



Android Developer Fundamentals Course

Learn to develop Android Applications

Practical Workbook

Developed by Google Developer Training Team

December 2016



Table of Contents

| | |
|----------------------------------------------------------------------------|---------|
| Pengantar | 1.1 |
| Unit 1. Mulai | 1.2 |
| Pelajaran 1: Membangun Aplikasi Pertama Anda | 1.2.1 |
| 1.1: Memasang Android Studio dan Menjalankan Hello World | 1.2.1.1 |
| Membuat UI Interaktif Pertama Anda | 1.2.1.2 |
| 1.2B: Menggunakan Layout | 1.2.1.3 |
| 1.3: Menggunakan Elemen TextView | 1.2.1.4 |
| 1.4: Mempelajari Tentang Sumber Daya yang Tersedia | 1.2.1.5 |
| Pelajaran 2: Aktivitas | 1.2.2 |
| 2.1: Membuat dan Memulai Aktivitas | 1.2.2.1 |
| 2.2: Daur Hidup dan Keadaan Aktivitas | 1.2.2.2 |
| 2.3: Aktivitas dan Maksud Implisit | 1.2.2.3 |
| Pelajaran 3: Menguji, Men-debug, dan Menggunakan Pustaka Dukungan | 1.2.3 |
| 3.1: Menggunakan Debugger | 1.2.3.1 |
| 3.2: Menguji Aplikasi Anda | 1.2.3.2 |
| 3.3: Menggunakan Pustaka Dukungan | 1.2.3.3 |
| Unit 2. Pengalaman Pengguna | 1.3 |
| Pelajaran 4: Interaksi Pengguna | 1.3.1 |
| 4.1. Menggunakan Keyboard, Kontrol Masukan, Peringatan, dan Picker | 1.3.1.1 |
| 4.2: Menggunakan Menu Opsi dan Tombol Radio | 1.3.1.2 |
| 4.3: Menggunakan Bilah Aplikasi dan Tab untuk Navigasi | 1.3.1.3 |
| 4.4: Membuat Recycler View | 1.3.1.4 |
| Pelajaran 5: Pengalaman Pengguna yang Menyenangkan | 1.3.2 |
| 5.1: Sumber Daya Dapat Digambar, Gaya, dan Tema | 1.3.2.1 |
| 5.2: Desain Material: Daftar, Kartu, dan Warna | 1.3.2.2 |
| 5.3: Mendukung Lanskap, Beberapa Ukuran Layar, dan Pelokalan | 1.3.2.3 |
| Pelajaran 6: Menguji UI Anda | 1.3.3 |
| 6.1: Menggunakan Espresso untuk Menguji UI Anda | 1.3.3.1 |
| Unit 3. Bekerja di Latar Belakang | 1.4 |
| Pelajaran 7: Tugas Latar Belakang | 1.4.1 |
| 7.1: Membuat AsyncTask | 1.4.1.1 |
| 7.2: Menghubungkan ke Internet dengan AsyncTask dan AsyncTaskLoader | 1.4.1.2 |
| 7.3: Penerima Siaran | 1.4.1.3 |
| Pelajaran 8: Memicu, Menjadwalkan, dan Mengoptimalkan Tugas Latar Belakang | 1.4.2 |
| 8.1: Notifikasi | 1.4.2.1 |
| 8.2: Alarm Manager | 1.4.2.2 |
| 8.3: Job Scheduler | 1.4.2.3 |
| Unit 4. Semua Tentang Data | 1.5 |
| Pelajaran 9: Preferensi dan Setelan | 1.5.1 |

| | |
|------------------------------------------------------|---------|
| 9.1: Preferensi Bersama | 1.5.1.1 |
| 9.2: Menambahkan Setelan ke Aplikasi | 1.5.1.2 |
| Pelajaran 10: Menyimpan Data Menggunakan SQLite | 1.5.2 |
| 10.1A: Database SQLite | 1.5.2.1 |
| 10.1B: Menelusuri Database SQLite | 1.5.2.2 |
| Pelajaran 11: Berbagi Data dengan Penyedia Materi | 1.5.3 |
| 11.1A: Mengimplementasikan Penyedia Materi Minimalis | 1.5.3.1 |
| 11.1B: Menambahkan Penyedia Materi ke Database Anda | 1.5.3.2 |
| 11.1C: Berbagi Materi dengan Aplikasi Lain | 1.5.3.3 |
| Pelajaran 12: Memuat Data Menggunakan Loader | 1.5.4 |
| 12.1: Memuat dan Menampilkan Data yang Diambil | 1.5.4.1 |
| Apendiks: Pekerjaan Rumah | 1.6 |
| Pekerjaan Rumah Pelajaran 1 | 1.6.1 |
| Pekerjaan Rumah Pelajaran 2 | 1.6.2 |
| Pekerjaan Rumah Pelajaran 3, 4 | 1.6.3 |
| Pekerjaan Rumah Pelajaran 5, 6 | 1.6.4 |
| Pekerjaan Rumah Pelajaran 7, 8 | 1.6.5 |
| Pekerjaan Rumah Pelajaran 9, 10, 11 | 1.6.6 |
| Apendiks: Utilitas | 1.7 |

Kursus Dasar-Dasar Developer Android — Praktik

Belajar mengembangkan aplikasi Android

Buku Kerja Praktik

Dikembangkan oleh Tim Pelatihan Developer Google

Terakhir diperbarui: Desember 2016



Karya ini berlisensi Creative Commons Attribution-NonCommercial 4.0 International License

1.1: Memasang Android Studio dan Menjalankan Hello World

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda BUTUHKAN](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1. Memasang Android Studio](#)
- [Tugas 2: Membuat aplikasi "Hello World"](#)
- [Tugas 3: Menjelajahi struktur proyek dan layout](#)
- [Tugas 4: Membuat perangkat virtual](#)
- [Tugas 5: Menjalankan aplikasi Anda di emulator](#)
- [Tugas 6. Menambahkan pernyataan log ke aplikasi Anda](#)
- [Tugas 7: Menjelajahi file AndroidManifest.xml](#)
- [Tugas 8. Menjelajahi file build.gradle](#)
- [Tugas 9. Menjalankan aplikasi Anda di perangkat](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Selamat datang di latihan praktis. Anda akan belajar:

- Memasang Android Studio, lingkungan development Android.
- Mempelajari proses development Android.
- Membuat dan menjalankan aplikasi Android Hello World pertama di emulator dan perangkat fisik.
- Menambahkan pencatatan log ke aplikasi Anda untuk men-debug.

Yang harus sudah Anda KETAHUI

Untuk praktik ini Anda harus bisa:

- Memahami proses development perangkat lunak umum untuk aplikasi berorientasi objek menggunakan IDE (Integrated Development Environment).
- Memiliki pengalaman pemrograman berorientasi objek paling sedikit 1-3 tahun, beberapa di antaranya fokus pada bahasa pemrograman Java. (Praktik ini tidak akan menjelaskan pemrograman berorientasi objek atau bahasa Java.)

Yang akan Anda BUTUHKAN

Untuk praktik ini, Anda akan membutuhkan:

- Komputer Mac, Windows, atau Linux. Lihat di bagian bawah [laman unduhan Android Studio](#) untuk [persyaratan sistem](#) terbaru.
- Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.

Yang akan Anda PELAJARI

Anda akan belajar:

- Memasang dan menggunakan Android IDE.
- Memahami proses development untuk membangun aplikasi Android.
- Membuat proyek Android dari template aplikasi dasar.

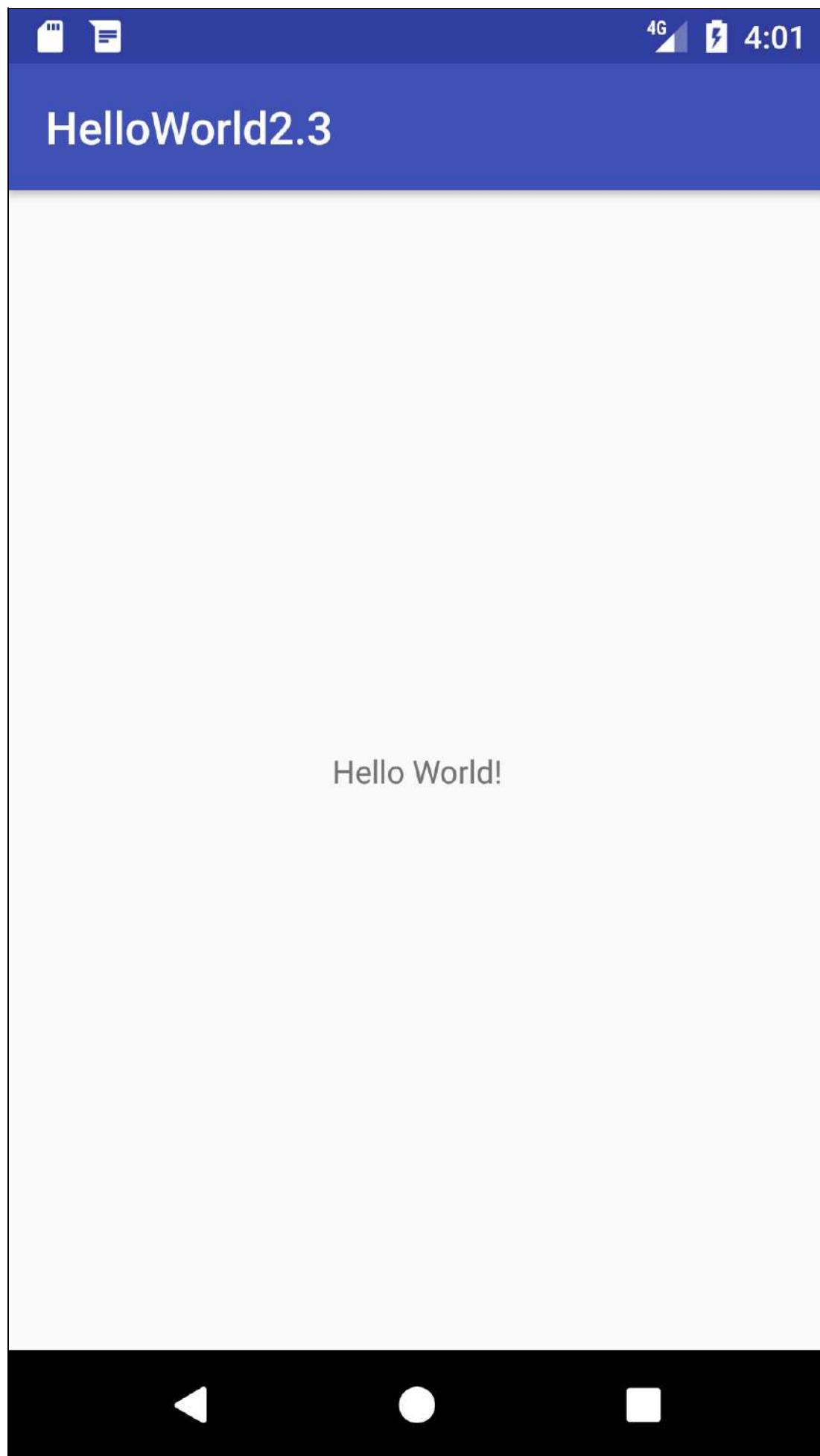
Yang akan Anda LAKUKAN

- Memasang lingkungan development Android Studio.
- Membuat emulator (perangkat virtual) untuk menjalankan aplikasi Anda di komputer.
- Membuat dan menjalankan aplikasi Hello World di perangkat virtual dan fisik.
- Menjelajahi layout proyek.
- Membuat dan menampilkan pernyataan log dari aplikasi Anda.
- Menjelajahi file AndroidManifest.xml.

Ringkasan Aplikasi

Setelah berhasil memasang Android Studio IDE, Anda akan membuat proyek Android baru untuk aplikasi 'Hello World' dari sebuah template. Aplikasi sederhana ini menampilkan string "Hello World" di layar perangkat Android virtual atau fisik.

Berikut tampilan aplikasi yang sudah selesai:



Tugas 1. Memasang Android Studio

Android Studio adalah IDE Google untuk aplikasi Android. Android Studio menyediakan editor kode tingkat lanjut dan serangkaian template aplikasi. Selain itu, Android Studio juga memiliki alat untuk development, men-debug, menguji, dan meningkatkan kinerja yang membuat pengembangan aplikasi lebih cepat dan mudah. Anda bisa menguji aplikasi dengan berbagai macam emulator yang telah dikonfigurasi sebelumnya atau di perangkat seluler Anda sendiri, dan membangun APK produksi untuk publikasi.

Catatan: Android Studio masih terus dikembangkan. Untuk informasi terbaru tentang persyaratan sistem dan petunjuk pemasangan, lihat dokumentasinya di developer.android.com.

Untuk menyiapkan dan mengoperasikan Android Studio:

- Anda mungkin perlu memasang Java Development Kit - Java 7 atau versi yang lebih baik.
- Memasang Android Studio

Android Studio dapat digunakan di komputer Windows, Mac, dan Linux. Pemasangannya sama untuk semua platform. Perbedaannya akan dijelaskan di bawah ini.

1.1. Memasang Java Development Kit

1. Buka jendela terminal di komputer.
2. Ketikkan `java -version`

Output termasuk baris:

```
Java(TM) SE Runtime Environment (build1.X.0_05-b13)
```

X adalah nomor versi yang perlu dilihat.

- Jika nomornya 7 atau lebih besar, Anda bisa lanjut memasang Android Studio.
- Jika versi Java SE di bawah 7 atau belum dipasang, Anda perlu memasang versi terbaru paket development Java SE sebelum memasang Android Studio.

Untuk mengunduh Java Standard Edition () Development Kit (JDK):

1. Buka [laman unduhan Oracle Java SE](https://www.oracle.com/java/technologies/javase/jdk8-downloads.html).
2. Klik ikon Java SE Downloads untuk membuka [laman Java SE Development Kit 8 Downloads](https://www.oracle.com/java/technologies/javase/jdk8-downloads.html).
3. Di kotak paket Java SE Development terbaru, Anda perlu menerima Perjanjian Lisensi agar dapat melanjutkan. Lalu unduh versi yang sesuai untuk komputer yang Anda gunakan untuk mengembangkan.
Penting: Jangan masuk ke demo dan contoh (menunya terlihat sangat mirip, jadi pastikan Anda membaca judul di atas).
4. Memasang paket development. Setelah pemasangan JDK selesai, yang seharusnya hanya memakan waktu beberapa menit, Anda bisa mengonfirmasi bahwa pemasangan sudah benar dengan memeriksa versi Java dari baris perintah.
5. Buka jendela terminal dan ketikkan `java -version` lagi untuk memverifikasi bahwa pemasangan telah berhasil.
6. Setel variabel lingkungan JAVA_HOME ke direktori pemasangan JDK.

Windows:

1. Setel JAVA_HOME ke lokasi pemasangan.
2. Start > Control Panel > System > Advanced System Settings > Environment Variables System Variables > New
 - o Nama variabel: JAVA_HOME
 - o Nilai variabel: C:\Program Files\Java\jdk1.7.0_80 (atau apa pun versi pemasangan Anda)
3. Jika variabel sudah ada, perbarui dengan versi JDK ini.
4. Verifikasi variabel JAVA_HOME dari terminal cmd.exe: `echo %JAVA_HOME%`

Lihat juga: https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/

Mac:

1. Buka Terminal.
2. Konfirmasi bahwa Anda memiliki JDK dengan mengetik "which java".
3. Periksa bahwa Anda memiliki versi Java yang diperlukan dengan mengetik "java -version".
4. Setel JAVA_HOME menggunakan perintah ini di Terminal: `ekspor JAVA_HOME=`which java``
5. masukkan `echo $JAVA_HOME` untuk mengonfirmasi jalurnya.

Linux:

Lihat: https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/

Penting: Jangan pasang Android Studio sampai setelah Java JDK dipasang. Tanpa salinan Java yang berfungsi dengan baik, proses tidak akan bekerja. Jika Anda tidak dapat mengunduh, telusuri pesan kesalahan dan cari solusinya secara online.

Pemecahan Masalah Dasar:

- Tidak ada ikon UI, Control Panel, atau Startup yang terkait dengan JDK.
- Verifikasi bahwa Anda telah memasang JDK dengan benar dengan membuka direktori tempat Anda memasangnya. Untuk mengidentifikasi tempat JDK, lihat variabel PATH dan/atau cari direktori "jdk" atau "java" atau "javac" yang dapat dijalankan di komputer Anda.

1.2. Memasang Android Studio

1. Buka [situs developer Android](#) dan ikuti petunjuk untuk mengunduh dan [memasang Android Studio](#).
 - Terima konfigurasi default untuk semua langkah.
 - Pastikan semua komponen dipilih untuk pemasangan.
2. Setelah selesai memasang. Wizard Persiapan akan mengunduh dan memasang beberapa komponen tambahan. Bersabarlah karena ini mungkin memakan waktu bergantung kecepatan internet Anda, dan beberapa langkah mungkin tampak berulang.
3. Setelah unduhan selesai, Android Studio akan dimulai, dan Anda siap membuat proyek pertama.

Pemecahan masalah: Jika menemui masalah dengan pemasangan, periksa dokumentasi terbaru, forum pemrograman, atau minta bantuan instruktur Anda.

Tugas 2: Membuat aplikasi "Hello World"

Dalam tugas ini, Anda akan mengimplementasikan aplikasi "Hello World" untuk memverifikasi bahwa Android Studio sudah dipasang dengan benar dan mempelajari dasar-dasar pengembangan dengan Android Studio.

2.1 Membuat aplikasi "Hello World"

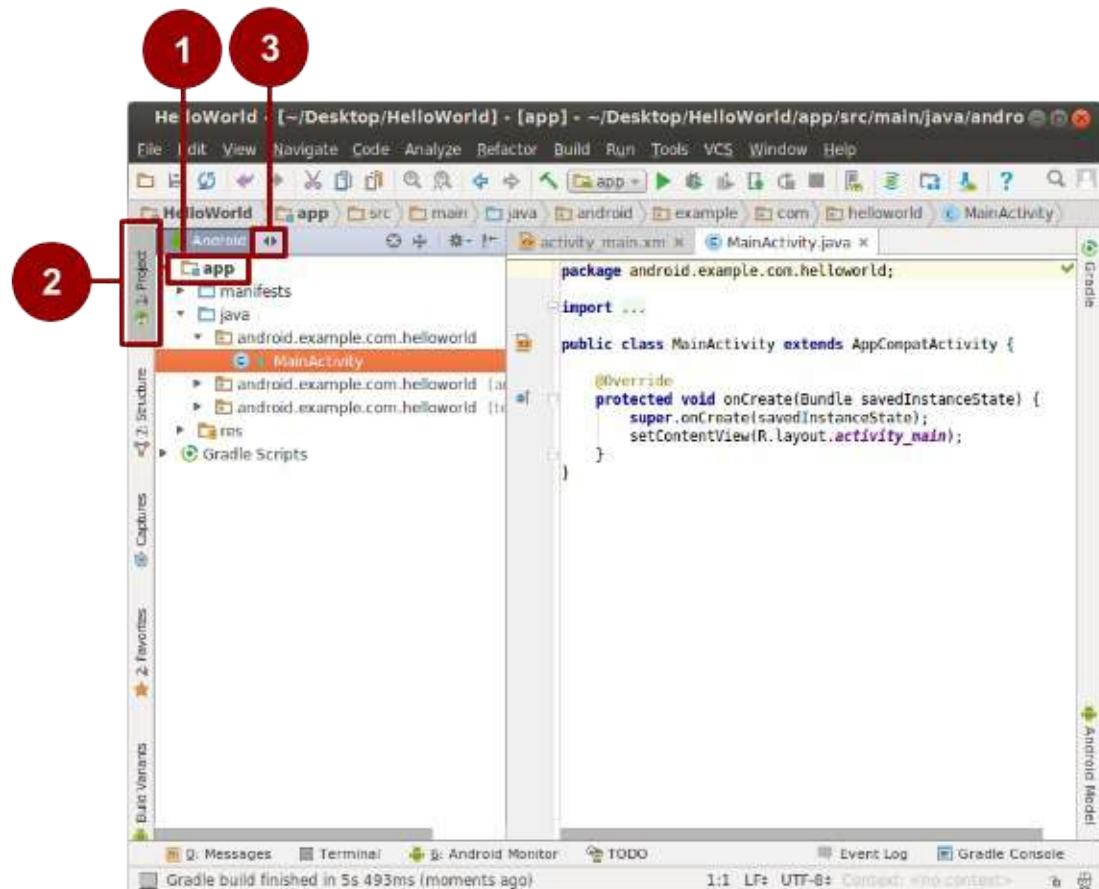
1. Buka Android Studio jika belum terbuka.
2. Di jendela utama **Welcome to Android Studio**, klik "Start a new Android Studio project".
3. Di jendela **New Project**, berikan aplikasi Anda **Application Name**, seperti "Hello World".
4. Verifikasi lokasi Proyek, atau pilih direktori yang berbeda untuk menyimpan proyek.
5. Pilih **Company Domain** yang unik.
 - Aplikasi yang dipublikasikan di Google Play Store harus memiliki nama paket yang unik. Karena domain unik, mengawali nama aplikasi dengan nama atau nama domain perusahaan Anda akan menghasilkan nama paket yang unik.
 - Jika tidak berencana mempublikasikan aplikasi, Anda bisa menerima domain contoh default. Perhatikan bahwa mengubah nama paket aplikasi di kemudian hari berarti melakukan pekerjaan ekstra.
6. Verifikasi bahwa **Project Location** default adalah tempat Anda menyimpan aplikasi Hello World dan proyek Android Studio lainnya, atau ubah lokasi ke direktori yang diinginkan. Klik Next.
7. Di layar **Target Android Devices**, "Phone and Tablet" harus dipilih. Dan Anda harus memastikan bahwa API 15: Android 4.0.3 IceCreamSandwich disetel sebagai **Minimum SDK**. (Perbaiki jika perlu.)

- Saat buku ini ditulis, memilih tingkat API ini menjadikan aplikasi "Hello World" kompatibel dengan 97% perangkat Android yang aktif di Google Play store.
 - Ini adalah setelan yang digunakan oleh contoh di buku ini.
8. Klik **Next**.
 9. Jika proyek Anda memerlukan komponen tambahan untuk SDK target terpilih, Android Studio akan memasangnya secara otomatis. Klik **Next**.
 10. Jendela **Customize the Activity**. Setiap aplikasi membutuhkan paling tidak satu aktivitas. Aktivitas mewakili satu layar dengan satu antarmuka pengguna dan Android Studio menyediakan template untuk membantu Anda memulai. Untuk proyek Hello World, pilih template termudah (saat ini ditulis, template proyek "Empty Activity" adalah template paling sederhana) yang tersedia.
 11. Adalah praktik yang biasa untuk menyebut aktivitas utama Anda **MainActivity**. Ini bukan persyaratan.
 12. Pastikan kotak **Generate Layout file** dicentang (jika terlihat).
 13. Pastikan kotak **Backwards Compatibility (App Compat)** dicentang.
 14. Biarkan **Layout Name** sebagai **activity_main**. Layout biasanya diberi nama sesuai dengan aktivitasnya. Terima defaultnya dan klik **Finish**.

Setelah langkah-langkah ini, Android Studio:

- Membuat folder untuk Proyek Android Studio Anda.
- Bangun proyek Anda dengan **Gradle** (ini mungkin memerlukan beberapa waktu). Android Studio menggunakan Gradle sebagai sistem build-nya. Lihat laman developer [Konfigurasi build](#) untuk informasi selengkapnya.
- Membuka editor kode dengan proyek Anda.
- Menampilkan tip hari ini.
 - Android Studio menawarkan banyak pintasan keyboard, dan membaca tip adalah cara yang bagus untuk mempelajarinya dari waktu ke waktu.

Jendela Android Studio harus terlihat mirip dengan diagram berikut:



Anda bisa melihat hierarki file untuk aplikasi Anda dengan beberapa cara.

1. Klik di folder Hello World folder untuk meluaskan hierarki file (1),
2. Klik di **Project (2)**.
3. Klik di menu **Android(3)**.
4. Jelajahi opsi tampilan yang berbeda untuk proyek Anda.

Catatan: Buku ini menggunakan tampilan Android dari file proyek, kecuali ditentukan lain.

Tugas 3: Jelajahi struktur proyek

Dalam praktik ini, Anda akan menjelajahi cara file proyek disusun di Android Studio.

Langkah-langkah ini mengasumsikan bahwa proyek Hello World Anda dimulai seperti yang ditampilkan pada diagram di atas.

3.1 Menjelajahi struktur proyek dan layout

Di tampilan Project > Android tugas sebelumnya, ada tiga folder level teratas di bawah folder **app** Anda: **manifests**, **java**, dan **res**.

1. Luaskan folder **manifests**.

Folder ini berisi **AndroidManifest.xml**. File ini menjelaskan semua komponen aplikasi Android Anda dan dibaca oleh sistem waktu proses Android saat program dijalankan.

2. Luaskan folder **java**. Semua file bahasa Java dikelola di folder ini. Folder **java** berisi tiga subfolder:
 - **com.example.hello.helloworld** (**atau nama domain yang telah Anda tetapkan**): Semua file untuk paket ada di folder yang bernama sama dengan paket tersebut. Untuk aplikasi Hello World, ada satu paket dan paket berisi `MainActivity.java` (ekstensi file dapat dihilangkan di tampilan Proyek).
 - **com.example.hello.helloworld(androidTest)**: Folder ini untuk pengujian berinstrumen Anda, dan memulai dengan file pengujian kerangka.
 - **com.example.hello.helloworld(test)**: Folder ini untuk pengujian unit dan memulai dengan file pengujian unit kerangka yang secara otomatis dibuat.
3. Luaskan folder **res**. Folder ini berisi semua sumber daya untuk aplikasi Anda, termasuk gambar, file layout, string, ikon, dan penataan gaya. Folder ini berisi subfolder berikut:
 - **drawable**. Simpan semua gambar aplikasi Anda di folder ini.
 - **layout**. Setiap aktivitas memiliki paling tidak satu file layout yang menjelaskan UI di XML. Untuk Hello World, folder ini berisi `activity_main.xml`.
 - **mipmap**. Simpan ikon peluncur Anda di folder ini. Ada subfolder untuk setiap kepadatan layar yang didukung. Android menggunakan kepadatan layar, yaitu, jumlah piksel per inci untuk menentukan resolusi gambar yang diperlukan. Android mengelompokkan kepadatan layar yang sebenarnya ke dalam kepadatan umum seperti medium (mdpi), tinggi (hdpi), atau ekstra-ekstra-tinggi (xxxhdpi). Folder `ic_launcher.png` berisi ikon peluncur default untuk semua kepadatan yang didukung oleh aplikasi Anda.
 - **values**. Sebagai ganti melakukan hardcode nilai-nilai seperti string, dimensi, dan warna di file XML dan Java, praktik terbaiknya adalah mendefinisikannya menurut file nilai. Ini membuatnya lebih mudah diubah dan konsisten di seluruh aplikasi Anda.
4. Luaskan subfolder **values** dalam folder res. Folder ini berisi subfolder berikut:
 - **colors.xml**. Tunjukkan warna default untuk tema pilihan Anda, dan Anda bisa menambahkan warna sendiri atau mengubahnya berdasarkan persyaratan aplikasi Anda.
 - **dimens.xml**. Simpan ukuran tampilan dan objek untuk resolusi yang berbeda.
 - **strings.xml**. Buat sumber daya untuk semua string Anda. Ini memudahkan penerjemahan string ke bahasa lain.
 - **styles.xml**. Semua gaya untuk aplikasi dan tema Anda ada di sini. Gaya membantu memberikan aplikasi Anda tampilan yang konsisten untuk semua elemen UI.

3.2 Sistem pembangunan Gradle

Android Studio menggunakan [Gradle](#) sebagai sistem pembangunan. Seiring dengan kemajuan Anda sepanjang praktik ini, Anda akan mempelajari lebih banyak tentang gradle dan apa yang dibutuhkan untuk membangun dan menjalankan aplikasi Anda.

1. Luaskan folder ***Gradle Scripts**. Folder ini berisi semua file yang dibutuhkan oleh sistem pembangunan.
2. Cari file **build.gradle(Module:app)**. Saat Anda menambahkan dependensi khusus aplikasi, seperti menggunakan pustaka tambahan, dependensi masuk ke dalam file ini.

Tugas 4: Membuat perangkat virtual (emulator)

Pada tugas ini, Anda akan menggunakan [Android Virtual Device \(AVD\) manager](#) untuk membuat perangkat virtual atau emulator yang menyimulasikan konfigurasi untuk jenis perangkat Android tertentu.

Menggunakan AVD Manager, Anda mendefinisikan karakteristik perangkat keras sebuah perangkat dan API level-nya, serta menyimpannya sebagai konfigurasi perangkat virtual.

Saat Anda mulai emulator Android, emulator membaca konfigurasi yang ditetapkan dan membuat perangkat emulasi yang bertindak persis seperti versi fisik perangkat tersebut, namun berada di komputer Anda.

Mengapa: Dengan perangkat virtual, Anda bisa menguji aplikasi di perangkat yang berbeda (tablet, ponsel) dengan API level berbeda untuk memastikan aplikasi terlihat bagus dan berfungsi untuk sebagian besar pengguna. Anda tidak perlu bergantung pada perangkat fisik yang tersedia untuk development aplikasi.

4.1 Membuat perangkat virtual

Untuk menjalankan emulator di komputer, Anda perlu membuat konfigurasi yang menjelaskan perangkat virtual.



1. Di Android Studio, pilih Tools > Android > AVD Manager atau klik ikon AVD Manager di bilah alat.
2. Klik **+Create Virtual Device....** (Jika Anda telah membuat perangkat virtual sebelumnya, jendela akan menunjukkan semua perangkat yang ada dan tombolnya ada di bagian bawah.)

Layar Select Hardware muncul menunjukkan daftar perangkat keras yang telah dikonfigurasi sebelumnya. Untuk setiap perangkat, tabel menunjukkan ukuran tampilan (Size), resolusi layar dalam piksel (Resolution), dan kepadatan piksel (Density).

Untuk perangkat Nexus 5, kepadatannya adalah xxhdpi, yang berarti aplikasi Anda menggunakan ikon peluncur di folder xxhdpi dari folder mipmap. Aplikasi Anda juga akan menggunakan layout dan drawable dari folder yang didefinisikan untuk kepadatannya juga.

3. Pilih perangkat keras Nexus 5 dan klik **Next**.
4. Pada layar **System Image**, dari tab **Recommended**, pilih versi sistem Android yang akan dijalankan pada perangkat virtual. Anda bisa memilih gambar sistem terbaru.

Ada lebih banyak versi yang tersedia dari yang ditunjukkan di tab **Recommended**. Lihat tab **x86 Images** dan **Other Images** untuk melihatnya.

5. Jika tautan **Download** terlihat di samping versi gambar sistem, berarti versi tersebut belum terpasang dan Anda perlu mengunduhnya. Jika perlu, klik tautan untuk mulai mengunduh dan klik **Finish** setelah selesai.
6. Pada layar **System Image**, pilih gambar sistem dan klik **Next**.
7. Verifikasi konfigurasi Anda dan klik **Finish**. (Jika jendela AVD Manager **Your Android Devices** tetap terbuka, Anda bisa melanjutkan dan menutupnya.)

Tugas 5: Menjalankan aplikasi Anda di emulator

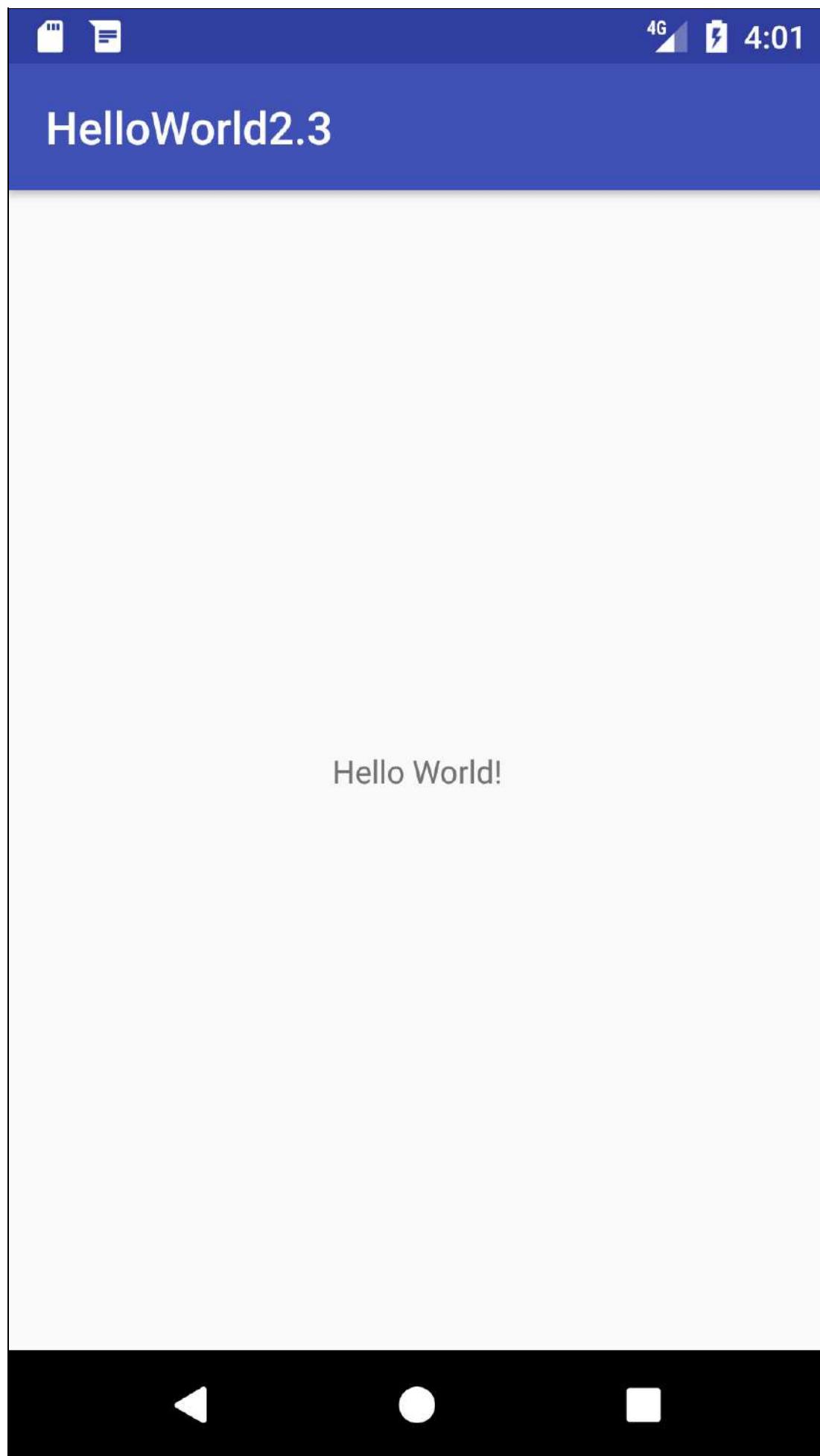
Pada tugas ini Anda akhirnya akan menjalankan aplikasi Hello World.

5.1 Jalankan aplikasi Anda di emulator

1. Di Android Studio, pilih **Run > Run app** atau klik **ikon Run**  di bilah alat.
2. Di jendela **Select Deployment Target**, di bawah **Available Emulators**, pilih **Nexus 5 API 23** dan klik **OK**.

Emulator dimulai dan di-boot, seperti perangkat fisik. Tergantung kecepatan komputer Anda, ini mungkin akan memerlukan beberapa saat. Aplikasi Anda dibangun, dan setelah emulator siap, Android Studio akan mengunggah aplikasi ke emulator dan menjalankannya.

Anda seharusnya melihat aplikasi Hello World seperti yang ditampilkan di tangkapan layar berikut.



Catatan:Saat menguji emulator, praktik terbaiknya adalah dengan memulainya sekali, di awal sesi Anda. Jangan tutup emulator sampai Anda selesai menguji aplikasi, agar aplikasi tidak perlu melalui proses booting lagi.

Tantangan penyusunan kode

Catatan:Semua tantangan penyusunan kode bersifat opsional dan tidak diwajibkan untuk praktik selanjutnya.

Tantangan: Anda bisa sepenuhnya menyesuaikan perangkat virtual.

- Pelajari [dokumentasi AVD Manager](#).
- Buat satu atau beberapa perangkat virtual khusus.

Anda mungkin melihat bahwa tidak semua kombinasi perangkat dan versi sistem berfungsi saat menjalankan aplikasi. Ini karena tidak semua gambar sistem dapat berjalan di semua perangkat keras.

Tugas 6. Menambahkan pernyataan log ke aplikasi Anda

Dalam praktik ini, Anda akan menambahkan pernyataan log ke aplikasi, yang ditampilkan di jendela pencatatan log Android Monitor.

Mengapa: Pesan log adalah alat debug bermanfaat yang bisa Anda gunakan untuk memeriksa nilai, jalur eksekusi, dan melaporkan pengecualian.

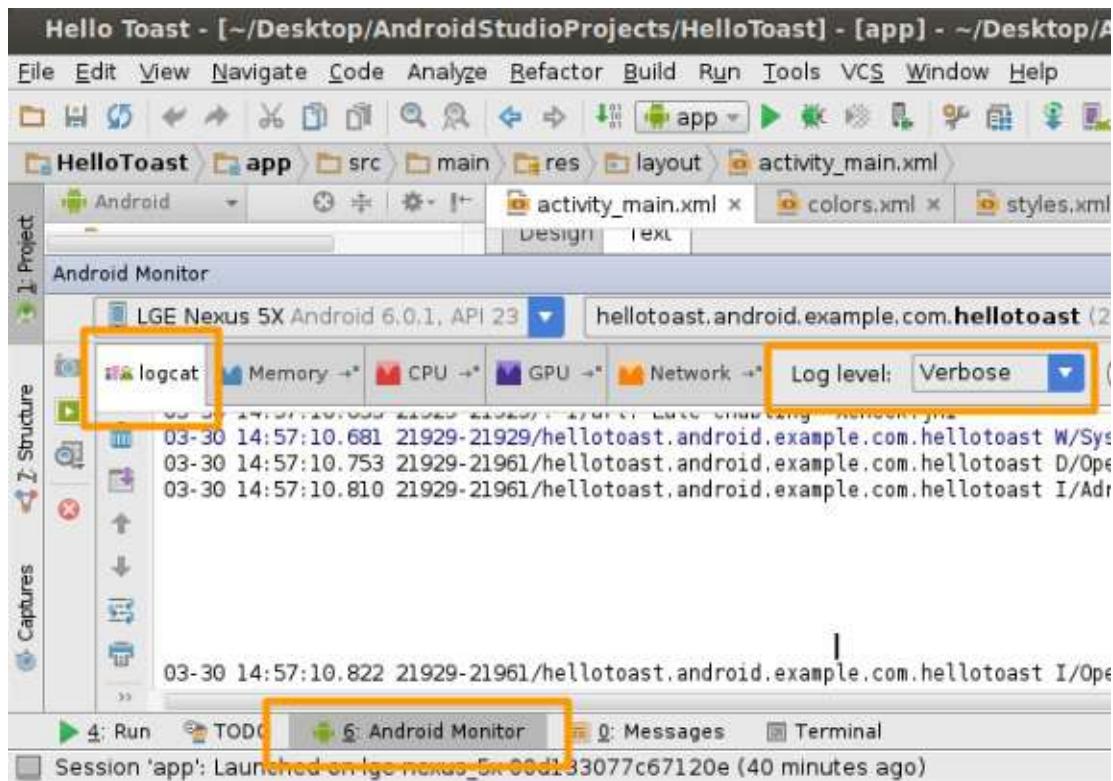
Android Monitor menampilkan informasi tentang aplikasi Anda.

1. Klik tombol **Android Monitor** di bawah Android Studio untuk membuka Android Monitor.

Secara default, ini membuka tab **logcat**, yang menampilkan informasi tentang aplikasi Anda saat sedang dijalankan. Jika Anda menambahkan pernyataan log ke aplikasi, pernyataan dicetak di sini juga.

Anda juga bisa memantau Memori, CPU, GPU, dan kinerja jaringan aplikasi dari tab lain Android Monitor. Ini dapat membantu debug dan kinerja menyempurnakan kode.

2. Level log default adalah **Verbose**. Di menu tarik-turun, ubah level log ke **Debug**.



Pernyataan log yang Anda tambahkan ke kode aplikasi mencetak pesan yang ditentukan oleh Anda di tab logcat Android Monitor. Misalnya:

```
Log.d("MainActivity", "Hello World");
```

Bagian dari pesan tersebut adalah:

- Log – [Kelas Log](#). API untuk mengirim pesan log.
- d – Tingkat Log. Digunakan untuk memfilter tampilan pesan di logcat. "d" untuk debug. Tingkat log lainnya adalah "e" untuk error, "w" untuk warning, dan "i" untuk info.
- "MainActivity" – Argumen pertama adalah tag yang dapat digunakan untuk memfilter pesan di logcat. Ini biasanya adalah nama aktivitas asal pesan. Tetapi, Anda dapat menjadikannya apa saja yang berguna untuk men-debug.

Sesuai ketentuan, tag log didefinisikan sebagai konstanta:

```
private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

- "Hello world" – Argumen kedua adalah pesan yang sebenarnya.

6.1 Menambahkan pernyataan log ke aplikasi Anda

1. Buka aplikasi Hello World di Android studio, dan buka file MainActivity.
2. **File > Settings > Editor > General >Auto Import** (Mac: **Android Studio > Preferences > Editor > General >Auto Import**). Pilih semua kotak centang dan setel **Insert imports on paste** ke **All**. Impor yang jelas sekarang ditambahkan secara otomatis ke file Anda. Catat bahwa opsi "add unambiguous imports on the fly" penting bagi beberapa fitur Android seperti NumberFormat. Jika tidak dicentang, NumberFormat menampilkan kesalahan. Klik 'Apply' lalu klik tombol 'OK'.
3. Dalam metode onCreate, tambahkan pernyataan log berikut:

```
Log.d("MainActivity", "Hello World");
```

4. Jika Android Monitor belum dibuka, klik tab Android Monitor di bawah Android Studio untuk membukanya. (Lihat

- tangkapan layar.)
5. Pastikan level Log di logcat Android Monitor disetel ke Debug atau Verbose (default).
 6. Jalankan aplikasi Anda.

Kode Solusi:

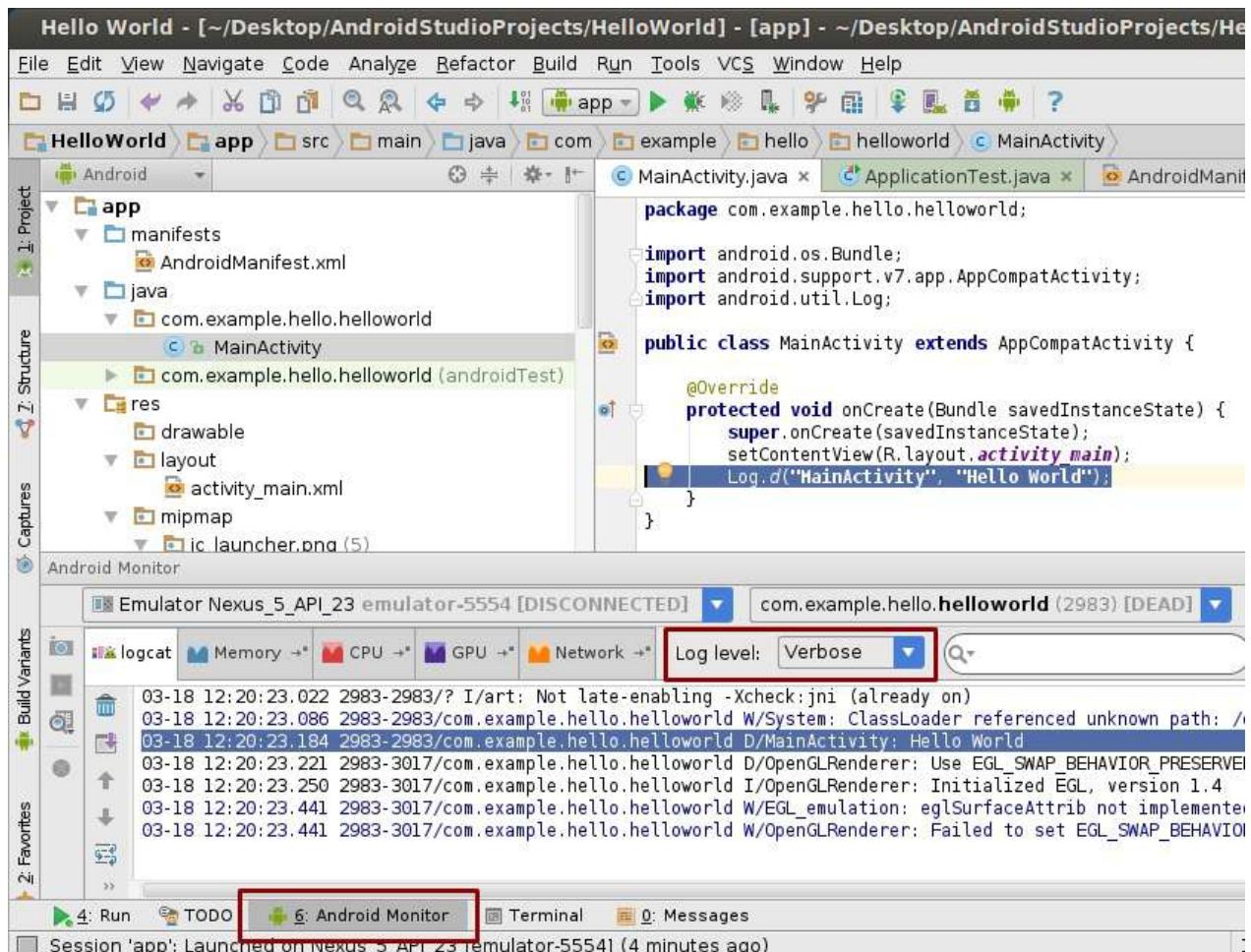
```
package com.example.hello.helloworld;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("MainActivity", "Hello World");
    }
}
```

Pesan Log Output

```
03-18 12:20:23.184 2983-2983/com.example.hello.helloworld D/MainActivity: Hello World
```



Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk bab berikutnya.

Tantangan: Penggunaan umum kelas Log adalah mencatat log pengecualian Java saat terjadi di program Anda. Ada beberapa metode berguna dalam kelas Log yang dapat digunakan untuk tujuan ini. Gunakan [dokumentasi kelas Log](#) untuk mencari tahu metode apa yang bisa Anda gunakan untuk menyertakan pengecualian dengan pesan log. Lalu, tulis kode di file MainActivity.java untuk memicu dan mencatat pengecualian.

Tugas 7: Menjelajahi file AndroidManifest.xml

Setiap aplikasi menyertakan file Manifes Android (`AndroidManifest.xml`). File manifes berisi informasi penting tentang aplikasi Anda dan menghadirkan informasi ini ke sistem waktu proses Android. Android harus memiliki informasi ini sebelum dapat menjalankan kode aplikasi apa pun.

Dalam praktik ini Anda akan menemukan dan membaca file `AndroidManifest.xml` untuk aplikasi Hello World.

Mengapa: Saat aplikasi menambahkan lebih banyak fungsionalitas dan pengalaman pengguna jadi lebih menarik dan interaktif, file `AndroidManifest.xml` berisi lebih banyak lagi informasi. Pada pelajaran berikutnya, Anda akan memodifikasi file ini untuk menambahkan fitur dan izin fitur.

7.1 Jelajahi file `AndroidManifest.xml`

1. Buka aplikasi Hello World di Android studio, dan di folder **manifests** folder, buka **AndroidManifest.xml**.
2. Baca file dan pertimbangkan apa yang ditunjukkan oleh setiap kode. Kode di bawah ini dianotasi untuk memberi Anda petunjuk.

Kode beranotasi:

```

<!-- XML version and character encoding -->
<?xml version="1.0" encoding="utf-8"?>
<!-- Required starting tag for the manifest -->
<manifest
    <!-- Defines the android namespace. Do not change. -->
    xmlns:android="http://schemas.android.com/apk/res/android"
    <!-- Unique package name of your app. Do not change once app is
        published. -->
    package="com.example.hello.helloworld">
    <!-- Required application tag -->
    <application
        <!-- Allow the application to be backed up and restored. -->
        android:allowBackup="true"
        <!-- Icon for the application as a whole,
            and default icon for application components. -->
        android:icon="@mipmap/ic_launcher"
        <!-- User-readable for the application as a whole,
            and default icon for application components. Notice that Android
            Studio first shows the actual label "Hello World".
            Click on it, and you will see that the code actually refers to a string
            resource. Ctrl-click @string/app_name to see where the resource is
            specified. This will be covered in a later practical . -->
        android:label="@string/app_name"
        <!-- Whether the app is willing to support right-to-left layouts.-->
        android:supportsRtl="true"
        <!-- Default theme for styling all activities. -->
        android:theme="@style/AppTheme">
        <!-- Declares an activity. One is required.
            All activities must be declared,
            otherwise the system cannot see and run them. -->
        <activity
            <!-- Name of the class that implements the activity;
                subclass of Activity. -->
            android:name=".MainActivity">
            <!-- Specifies the intents that this activity can respond to.-->
            <intent-filter>
                <!-- The action and category together determine what
                    happens when the activity is launched. -->
                <!-- Start activity as the main entry point.
                    Does not receive data. -->
                <action android:name="android.intent.action.MAIN" />
                <!-- Start this activity as a top-level activity in
                    the launcher . -->
                <category android:name="android.intent.category.LAUNCHER" />
            <!-- Closing tags -->
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode bersifat opsional.

Tantangan: Ada banyak elemen lain yang dapat disetel di Android Manifest. Jelajahi [dokumentasi Android Manifest](#) dan pelajari tentang elemen tambahan di Android Manifest.

Tugas 8. Menjelajahi file build.gradle

Android Studio menggunakan sistem pembangunan bernama Gradle. Gradle melakukan pembangunan bertahap, yang mengizinkan siklus edit-uji yang lebih singkat.

Untuk mengetahui selengkapnya tentang Gradle, buka:

- [Situs Gradle](#)

- Dokumentasi developer [Mengonfigurasi pembangunan](#)
- Telusuri "tutorial gradle" di internet.

Dalam tugas ini, Anda akan menjelajahi file `build.gradle`.

Mengapa: Saat menambahkan pustaka baru ke proyek Android, Anda mungkin juga harus memperbarui **file build.gradle**. Mengetahui di mana file tersebut dan strukturnya akan berguna bagi Anda.

8.1 Jelajahi file build.gradle(Module.app)

1. Di hierarki proyek, cari **Gradle Scripts** dan luaskan. Ada beberapa file build.gradle. Satu dengan arahan untuk seluruh proyek dan satu untuk setiap modul aplikasi. Modul untuk aplikasi Anda disebut "app". Pada tampilan Proyek, modul ini diwakili oleh folder **app** di tingkat atas tampilan Proyek.
2. Buka **build.gradle (Module.app)**.
3. Baca file dan pelajari apa yang ditunjukkan oleh setiap baris kode.

Solusi:

```
// Add Android-specific build tasks
apply plugin: 'com.android.application'
// Configure Android specific build options.
android {
    // Specify the target SDK version for the build.
    compileSdkVersion 23
    // The version of the build tools to use.
    buildToolsVersion "23.0.2"
    // Core settings and entries. Overrides manifest settings!
    defaultConfig {
        applicationId "com.example.hello.helloworld"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    // Controls how app is built and packaged.
    buildTypes {
        // Another common option is debug, which is not signed by default.
        release {
            // Code shrinker. Turn this on for production along with
            // shrinkResources.
            minifyEnabled false
            // Use ProGuard, a Java optimizer.
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
// This is the part you are most likely to change as you start using
// other libraries.
dependencies {
    // Local binary dependency. Include any JAR file inside app/libs.
    compile fileTree(dir: 'libs', include: ['*.jar'])
    // Configuration for unit tests.
    testCompile 'junit:junit:4.12'
    // Remote binary dependency. Specify Maven coordinates of the Support
    // Library needed. Use the SDK Manager to download and install such
    // packages.
    compile 'com.android.support:appcompat-v7:23.2.1'
}
```

- Untuk lebih memahami Gradle baca [Ringkasan Sistem Pembangunan](#) dan dokumentasi [Mengonfigurasi Build Gradle](#).
- Terdapat alat yang dapat membantu Anda [meringkas kode](#), menghapus pustaka/sumber daya tidak penting dan bahkan menyamarkan program untuk mencegah rekayasa balik yang tidak diinginkan.
- Android Studio sendiri menyediakan beberapa fitur yang berguna. Pelajari lebih lanjut tentang alat sumber terbuka yang bermanfaat yang bernama [ProGuard](#).

Tugas 9. [Opsiional] Menjalankan aplikasi Anda di perangkat

Pada tugas terakhir ini, Anda akan menjalankan aplikasi pada perangkat seluler fisik seperti ponsel atau tablet.

Mengapa: Pengguna akan menjalankan aplikasi di perangkat fisik. Anda harus selalu menguji aplikasi pada perangkat fisik dan virtual.

Yang Anda butuhkan:

- Perangkat Android seperti ponsel atau tablet.
- Kabel data untuk menghubungkan perangkat Android ke komputer melalui port USB.
- Jika menggunakan OS Linux atau Windows, Anda mungkin perlu melakukan langkah tambahan untuk menjalankannya di perangkat keras. Periksa dokumentasi [Menggunakan Perangkat Keras](#). Pada Windows, Anda mungkin perlu memasang driver USB yang sesuai untuk perangkat. Lihat [Driver US OEM](#).

Tugas 9.1. [Opsiional] Menjalankan aplikasi Anda di perangkat

Untuk mengizinkan Android Studio berkomunikasi dengan perangkat, Anda harus mengaktifkan USB Debugging di perangkat Android. Ini diaktifkan di setelan Developer options perangkat. Perlu diingat bahwa ini tidak sama dengan melakukan rooting perangkat.

Pada Android 4.2 dan yang lebih tinggi, layar Developer options disembunyikan secara default. Untuk menampilkan Developer options dan mengaktifkan USB Debugging:

1. Pada perangkat, buka **Settings > About phone** dan ketuk **Build number** tujuh kali.
2. Kembali ke layar sebelumnya (**Settings**). **Developer options** muncul di bawah daftar. Klik **Developer options**.
3. Pilih **USB Debugging**.

Sekarang Anda dapat menghubungkan perangkat dan menjalankan aplikasi dari Android Studio.

1. Hubungkan perangkat ke mesin development dengan kabel USB.
2. Di Android Studio, di bawah jendela, klik tab Android Monitor. Anda seharusnya melihat perangkat terdaftar di menu tarik-turun kiri atas.
3. Klik tombol **Run**  di bilah alat. Jendela **Select Deployment Target** terbuka dengan daftar emulator yang tersedia dan perangkat yang terhubung.
4. Pilih perangkat dan klik **OK**.

Android Studio seharusnya memasang dan menjalankan aplikasi di perangkat Anda.

Pemecahan Masalah

Jika Android Studio tidak mengenali perangkat, coba langkah berikut:

- Cabut dan colokkan kembali perangkat.
- Mulai ulang Android Studio.
- Jika komputer masih tidak mengenali perangkat dan mendeklarasikannya "unauthorized":
 1. Cabut perangkat.
 2. Pada perangkat, buka Settings->**Developer Options**.
 3. Ketuk **Revoke USB Debugging authorizations**.
 4. Hubungkan kembali perangkat ke komputer.
 5. Saat diminta, berikan otorisasi.
- Anda mungkin perlu memasang driver USB yang sesuai untuk perangkat. Baca [dokumentasi Menggunakan Perangkat Keras](#).
- Periksa dokumentasi terbaru, forum pemrograman, atau minta bantuan instruktur Anda.

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode bersifat opsional.

Tantangan: Sekarang karena Anda sudah siap dan mengenal alur kerja development, lakukan hal berikut:

1. Buat proyek baru di Android Studio.
2. Ganti sapaan dengan "Happy Birthday to " dan seseorang yang baru saja berulang tahun.
3. Ganti latar belakang aplikasi menggunakan gambar bertema ulang tahun.
4. Ambil tangkapan layar dari aplikasi yang telah selesai dan kirimkan lewat email ke seseorang yang ulang tahunnya Anda lupakan.

Rangkuman

Pada bab ini, Anda telah mempelajari cara:

- Memasang Android Studio
 - Mendapatkan pemahaman dasar tentang alur kerja development setelah membukanya di Android Studio.
 - Mendapatkan pemahaman dasar tentang struktur aplikasi Android di lingkungan pembangunannya.
 - Mendapatkan pemahaman dasar tentang Manifes Android dan kegunaannya.
 - Menambahkan pernyataan log ke kode yang memberi Anda alat dasar untuk proses debug.
- Menerapkan aplikasi Hello World pada emulator Android dan [opsional] perangkat seluler.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Membuat Aplikasi Android Pertama Anda](#)

Ketahui selengkapnya

- [Laman unduhan Android Studio](#)
- [Bagaimana cara memasang Java?](#)
- [Dokumentasi Android Studio](#)
- [Mendukung Beberapa Layar](#)
- [Laman Gradle di Wikipedia](#)
- [Membaca dan Menulis Log](#)

1.2A: Membuat UI Interaktif Pertama Anda

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1: Membuat proyek "Hello Toast"
- Tugas 2: Menambahkan tampilan ke "Hello Toast" dalam Layout Editor
- Tugas 3: Mengedit Layout "Hello Toast" dalam XML
- Tugas 4: Menambahkan handler on-click untuk tombol
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Antarmuka pengguna yang di pada layar perangkat seluler Android terdiri dari hierarki "tampilan". [Tampilan](#) adalah blok pembangun antarmuka pengguna dasar. Anda menentukan tampilan dalam file layout XML. Misalnya, tampilan dapat berupa komponen yang:

- menampilkan teks (kelas [TextView](#))
- mengizinkan Anda mengedit teks (kelas [EditText](#))
- mewakili tombol yang dapat diklik (kelas [Button](#)) dan komponen interaktif lainnya
- berisi teks yang dapat digulir ([ScrollView](#)) dan item yang dapat digulir ([RecyclerView](#))
- menampilkan gambar ([ImageView](#))
- menampung tampilan lain dan memposisikannya ([LinearLayout](#)).
- memunculkan [menu](#) dan komponen interaktif lainnya.

Anda bisa menjelajahi hierarki tampilan aplikasi dalam panel Component Tree Layout Editor.

Kode Java yang menampilkan dan menjalankan antarmuka pengguna berada dalam kelas yang memperluas [Aktivitas](#) dan berisi metode untuk mengembangkan tampilan, yaitu, mengambil layout XML tampilan dan menampilkannya di layar. Misalnya, MainActivity pada aplikasi Hello World mengembangkan tampilan teks dan mencetak Hello World. Pada aplikasi yang lebih kompleks, aktivitas dapat mengimplementasikan klik dan handler kejadian lainnya, meminta data dari database atau internet, atau menggambar konten grafis.

Android memudahkan proses pemisahan elemen UI dan data dari satu sama lain, dan menggunakan aktivitas tersebut untuk menyatukannya. Pemisahan ini adalah implementasi dari pola [MVP \(Model-View-Presenter\)](#).

Anda akan mengerjakan [Aktivitas](#) dan [Tampilan](#) di seluruh buku ini.

Yang harus sudah Anda KETAHUI

Untuk praktik ini Anda harus sudah mengetahui:

- Cara membuat aplikasi Hello World dengan Android Studio.

Yang akan Anda PELAJARI

Anda akan belajar:

- Cara membuat antarmuka pengguna interaktif dalam Layout Editor, dalam XML, dan secara terprogram.
- Banyak terminologi baru. Lihat [Kosakata](#) dan [konsep daftar istilah](#) untuk menemukan definisi yang mudah.

Yang akan Anda LAKUKAN

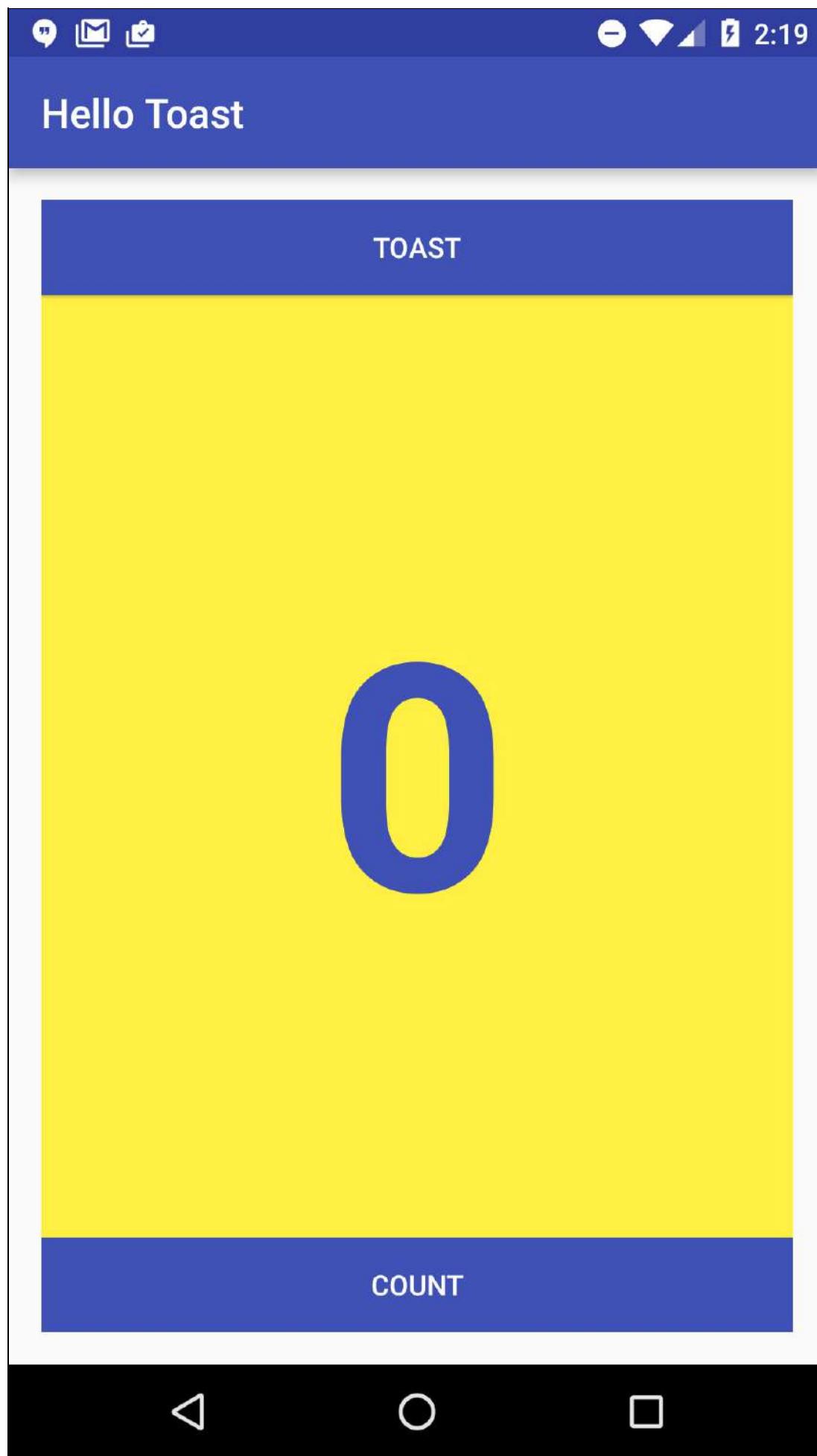
Dalam praktik ini Anda akan:

- Membuat aplikasi dan menggunakan elemen antarmuka pengguna seperti tombol dalam Layout Editor.
- Mengedit layout aplikasi dalam XML.
- Menambahkan tombol ke aplikasi. Menggunakan sumber daya string untuk label.
- Mengimplementasikan metode handler klik untuk tombol agar bisa menampilkan pesan pada layar saat pengguna mengekliknya.
- Mengubah metode handler klik untuk mengubah pesan yang ditampilkan pada layar.

Ringkasan Aplikasi

Aplikasi "Hello Toast" akan berisi dua tombol dan satu tampilan teks. Saat Anda mengeklik tombol pertama, pesan pendek atau toast akan ditampilkan pada layar. Mengeklik tombol kedua akan meningkatkan penghitung klik; jumlah total klik mouse akan ditampilkan pada tampilan teks.

Berikut gambaran aplikasi yang telah selesai:



Tugas 1: Membuat proyek "Hello Toast" yang baru

Dalam praktik ini Anda akan mendesain dan mengimplementasikan proyek untuk aplikasi "Hello Toast".

1.1 Membuat proyek "Hello Toast"

- Mulailah Android Studio dan buat proyek baru dengan parameter berikut:

| Atribut | Nilai |
|------------------------------|----------------------------------------|
| Application Name | Hello Toast |
| Company Name | com.example.android or your own domain |
| Phone and Tablet Minimum SDK | API15: Android 4.0.3 IceCreamSandwich |
| Template | Empty Activity |
| Generate Layout file box | Dicentang |
| Backwards Compatibility box | Dicentang |

- Pilih **Run > Run app** atau klik ikon Run  pada toolbar untuk membangun dan mengeksekusi aplikasi di emulator atau perangkat.

Tugas 2: Menambahkan tampilan ke "Hello Toast" dalam Layout Editor

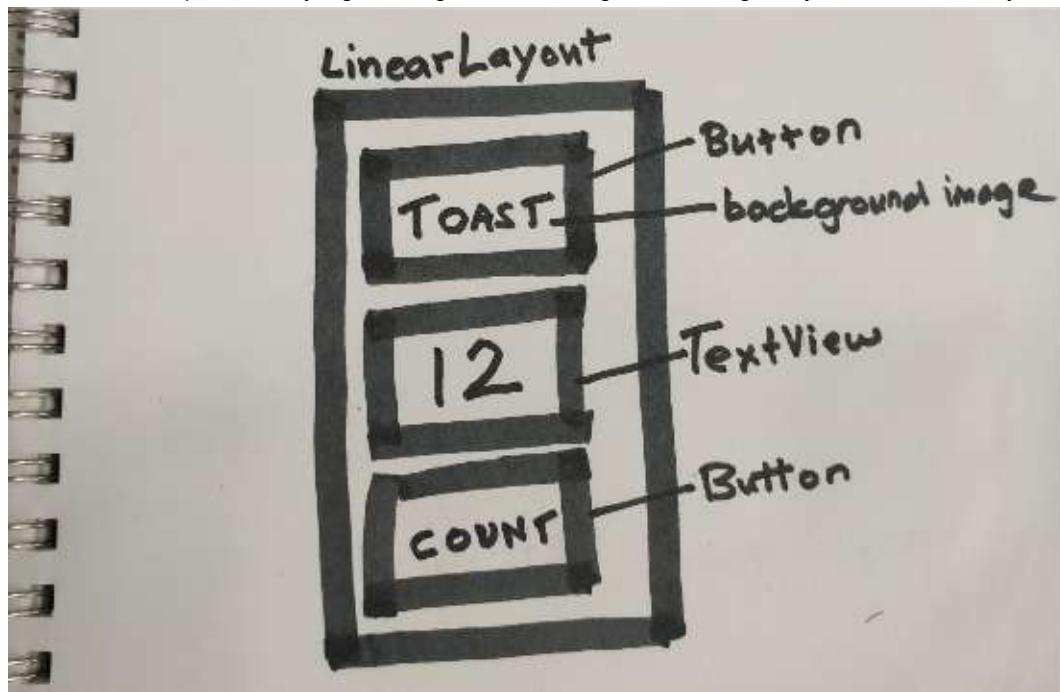
Dalam tugas ini, Anda akan membuat dan mengkonfigurasi antarmuka pengguna untuk aplikasi "Hello Toast" dengan mengatur komponen UI tampilan pada layar.

Mengapa: Setiap aplikasi harus dimulai dengan pengalaman pengguna, bahkan jika implementasi awalnya sangat dasar.

Tampilan yang digunakan untuk Hello Toast adalah:

- TextView** - Tampilan yang menampilkan teks.
- Tombol** - Tombol dengan label yang biasanya dikaitkan dengan handler klik.
- LinearLayout** - Tampilan yang berfungsi sebagai kontainer untuk mengatur tampilan lainnya. Jenis tampilan ini memperluas kelas **ViewGroup** dan juga disebut grup tampilan. LinearLayout adalah grup tampilan dasar yang mengatur kumpulan tampilan dalam baris horizontal atau vertikal.

Ini adalah sketsa kasar UI yang akan Anda bangun dalam latihan ini. Sketsa UI sederhana akan sangat berguna untuk memutuskan tampilan mana yang akan digunakan dan bagaimana mengurnya, terutama saat layout menjadi lebih rumit.



2.1 Menjelajahi layout editor

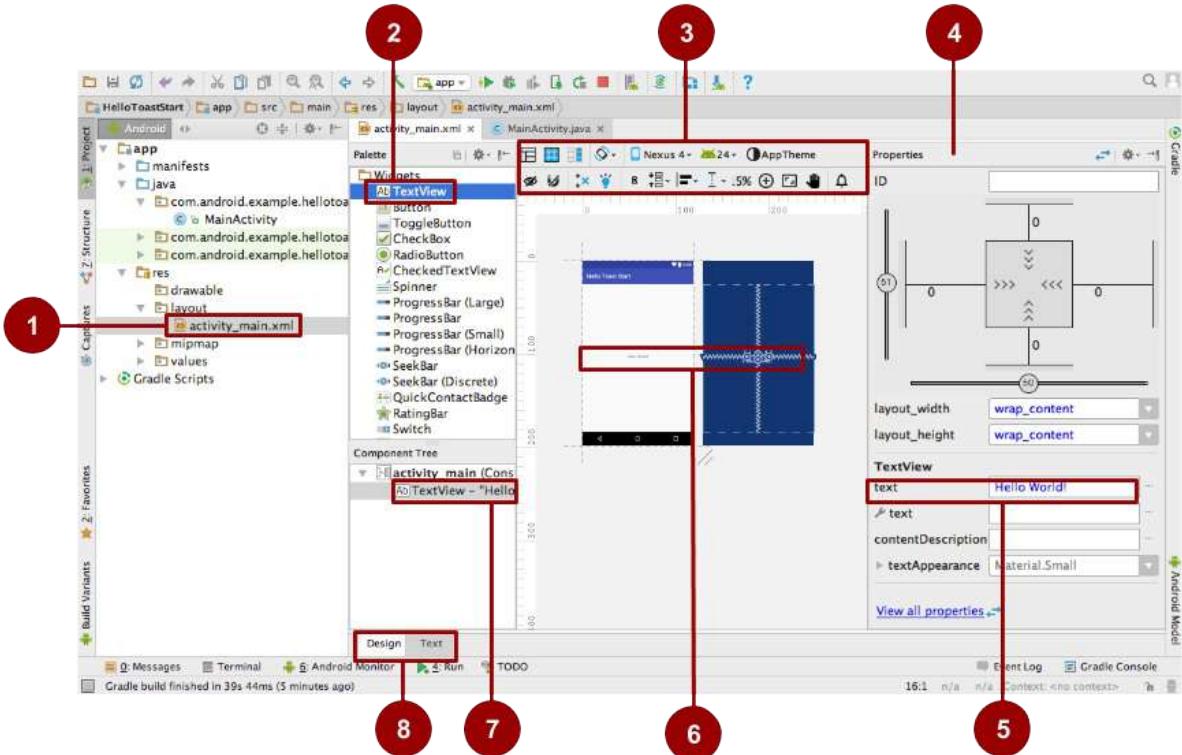
Gunakan Layout Editor untuk membuat layout elemen antarmuka pengguna dan mempratinjau aplikasi menggunakan perangkat yang berbeda serta tema, resolusi, dan orientasi aplikasi.

Lihat tangkapan layar bawah untuk mencocokkan

1. Dalam folder **app > res > layout**, buka file **activity_main.xml(1)**.

Layar Android Studio harus terlihat mirip dengan tangkapan layar di bawah ini. Jika Anda melihat kode XML untuk layout UI, klik tab **Design** di bawah Component Tree (8).

2. Menggunakan tangkapan layar beranotasi di bawah ini sebagai panduan, jelajahi Layout Editor.



3. Temukan berbagai cara yang berbeda untuk merepresentasikan elemen UI string "Hello World", atau TextView.
- Dalam Palette elemen UI (2) developer dapat membuat tampilan teks dengan menyeretnya ke dalam panel design.
 - Secara visual, panel Design (6).
 - Dalam **Component Tree** (7), sebagai komponen dalam hierarki tampilan UI yang disebut Hierarki Tampilan. Yakni, tampilan yang diatur ke dalam hierarki pohon induk dan anak, di mana anak mewarisi properti induknya.
 - Dalam panel **Properties** (4), sebagai daftar propertinya, di mana "Hello Toast" adalah nilai dari properti teks TextView (5).
4. Gunakan selector di atas perangkat virtual (3) untuk melakukan hal berikut ini:

- Mengubah tema aplikasi.
- Mengubah rotasi lanskap.
- Menggunakan versi berbeda dari SDK.
- Melakukan pratinjau pada gaya penulisan kanan ke kiri.
- Memilih item UI dan masuk ke kode sumbernya.

Menggunakan keterangan alat pada ikon untuk membantu Anda menemukan fungsinya.

5. Berpindah di antara tab **Design** dan **Text** (8). Beberapa perubahan UI hanya bisa dibuat dalam kode, dan beberapa lebih cepat dilakukan pada perangkat virtual.
6. Setelah selesai, batalkan perubahan (untuk perubahan UI, gunakan **Edit > Undo** atau pintasan keyboard untuk sistem operasi).

Lihat [Panduan Pengguna Android Studio](#) untuk dokumentasi Android Studio selengkapnya.

Catatan: Jika Anda mendapatkan kesalahan tentang Tema Aplikasi yang hilang, coba **File > Invalidate Caches / Restart** atau pilih tema yang tidak menghasilkan kesalahan. Bantuan tambahan bisa ditemukan di [pos stackoverflow ini](#).

2.2 Mengubah grup tampilan ke LinearLayout

Root hierarki tampilan adalah grup tampilan, seperti yang tersirat dari namanya, adalah tampilan yang berisi tampilan lain.

Secara default, Template Kosong menggunakan grup tampilan **RelativeLayout**. Layout ini menawarkan banyak fleksibilitas dalam menempatkan tampilan di grup tampilan.

Layout linear vertikal adalah salah satu layout yang paling umum. Layout ini sederhana, cepat, dan selalu merupakan titik permulaan yang baik. Ubah grup tampilan ke layout linear vertikal seperti berikut:

1. Dalam panel **Component Tree** (7 dalam tangkapan layar sebelumnya), temukan tampilan atas atau root di bawah Layar Perangkat, yang seharusnya RelativeLayout.
 2. Klik tab **Text** (8) untuk berpindah ke tampilan kode layout.
 3. Dalam baris kedua kode, cari **RelativeLayout** dan ganti ke **LinearLayout**.
 4. Pastikan tag penutup di bagian akhir kode sudah diubah menjadi </LinearLayout>. Jika belum berubah secara otomatis, ubah secara manual.
 5. `android:layout_height` ditetapkan sebagai bagian dari template. Orientasi layout default adalah baris horizontal. Untuk mengubah layout menjadi vertikal, tambahkan kode berikut di bawah `android:layout_height`.


```
        android:orientation="vertical"
```
 6. Dari bilah menu, pilih: **Code > Reformat Code...**
- Pesan "no lines changed: code is already properly formatted" mungkin muncul.
7. Jalankan kode untuk memastikan bahwa kode masih berfungsi.
 8. Kembali ke **Design**.
 9. Verifikasi dalam panel **Component Tree** bahwa elemen teratas sekarang adalah LinearLayout dan atribut orientasinya disetel ke "vertikal".

Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="hellotoast.android.example.com.hellotoast.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
</LinearLayout>
```

2.3: Menambahkan tampilan ke Linear Layout dalam Layout Editor

Pada tugas ini, Anda akan menghapus TextView saat ini (untuk latihan), dan menambahkan TextView baru dan dua tombol ke LinearLayout seperti yang ditampilkan dalam sketsa UI untuk tugas ini. Lihat diagram UI di atas, jika perlu.

Tambahkan Elemen UI

1. Klik tab **Design** (8) untuk menunjukkan layout perangkat virtual.
2. Klik TextView yang nilai teksnya "Hello World" dalam layout perangkat virtual atau panel **Component Tree** (7).
3. Tekan tombol DELETE untuk menghapus TextView.
4. Dari panel **Palette** (2), seret dan jatuhkan elemen Tombol, TextView, dan elemen Tombol lainnya, dalam urutan tersebut, satu di bawah yang lainnya ke dalam layout perangkat virtual.

Sesuaikan Elemen UI

1. Untuk mengidentifikasi setiap tampilan secara unik di dalam aktivitas, setiap tampilan memerlukan id unik. Dan agar ada gunanya, tombol memerlukan label dan tampilan teks perlu menunjukkan beberapa teks. Klik dua kali setiap elemen dalam Layout Manager untuk melihat properti dan mengubah string **text** dan **id** seperti berikut:

| Elemen | Teks | ID |
|---------------|-------|--------------|
| Top button | Toast | button_toast |
| Text view | 0 | show_count |
| Bottom button | Count | button_count |

1. Jalankan aplikasi Anda.

Layout Solusi:

Seharusnya ada tiga Tampilan pada layar Anda. Tampilan ini mungkin tidak cocok dengan ukuran gambar di bawah, tetapi selama Anda memiliki tiga Tampilan dalam layout vertikal, semuanya akan baik-baik saja.



Tantangan: Pikirkan tentang aplikasi yang mungkin Anda inginkan dan buat proyek dan layout untuk aplikasi tersebut menggunakan Layout Editor. Jelajahi selengkapnya fitur Layout Editor. Seperti yang disebutkan sebelumnya, Layout Editor memiliki banyak rangkaian fitur dan pintasan penyusunan kode. Lihat [dokumentasi Android Studio](#) untuk mencari tahu lebih jauh.

Tugas 3: Mengedit Layout "Hello Toast" dalam XML

Dalam praktik ini Anda akan mengedit kode XML untuk layout UI aplikasi Hello Toast. Anda juga akan mengedit properti tampilan yang sudah Anda buat. Anda bisa menemukan properti yang umum untuk semua tampilan dalam [Lihat dokumentasi kelas](#)

*Mengapa:** Meskipun Layout Editor adalah alat yang hebat, beberapa perubahan lebih mudah dibuat langsung di kode sumber XML. Menggunakan LayoutEditor grafis atau mengedit file XML secara langsung adalah pilihan personal.

1. Buka `res/layout/activity_main.xml` dalam mode Text.
2. Dalam bilah menu pilih **Code > Reformat Code**
3. Periksa kode yang dibuat oleh Layout Editor.

Perhatikan bahwa kode Anda mungkin bukan kecocokan eksak, tergantung pada perubahan yang Anda buat dalam Layout Editor. Gunakan solusi sederhana sebagai panduan.

3.1 Memeriksa properti LinearLayout

LinearLayout diperlukan untuk memiliki properti ini:

- layout_width
- layout_height
- orientation

layout_width dan layout_height dapat mengambil satu dari tiga nilai ini:

- Atribut **match_parent** meluaskan tampilan untuk mengisi induknya dengan lebar atau tinggi. Saat LinearLayout adalah tampilan rootnya, layout diperluas hingga ke ukuran tampilan induk.
- Atribut **wrap_content** menyusutkan dimensi tampilan, hanya cukup besar untuk membungkus kontennya. (Jika tidak ada konten, tampilan tidak akan terlihat.)
- Gunakan **angka tetap dp** ([piksel yang tidak tergantung perangkat](#)) untuk menentukan ukuran yang tetap, yang disesuaikan untuk ukuran layar perangkat. Misalnya, "16dp" berarti 16 piksel yang tidak tergantung perangkat.

Orientasinya bisa berupa:

- **horizontal**: tampilan diatur dari kiri ke kanan.
- **vertikal**: tampilan diatur dari atas ke bawah.

Ubah LinearLayout "Hello Toast" seperti berikut:

| Properti | Nilai |
|---------------|------------------------------------|
| layout_width | match_parent (untuk mengisi layar) |
| layout_height | match_parent (untuk mengisi layar) |
| orientasi | vertikal |

3.2 Membuat sumber daya string

Daripada melakukan hard-code string ke dalam kode XML, sebaiknya gunakan sumber daya string, yang mewakili string tersebut.

stringMengapa: String yang ada di file yang terpisah lebih mudah dikelola, terutama jika Anda menggunakan string ini lebih dari sekali. Selain itu, sumber daya string wajib ada untuk menerjemahkan dan melokalisasi aplikasi karena Anda akan membuat file sumber daya string untuk setiap bahasa.

1. Letakkan kursor pada kata "Toast".
2. Tekan **Alt-Enter (Option-Enter di Mac)**.
3. Pilih **Extract string resources**.
4. Setel **nama Sumber Daya** ke `button_label_toast` dan klik **OK**. (Jika Anda membuat kesalahan, batalkan perubahan dengan **Ctrl-Z**.)

Proses ini membuat sumber daya string dalam file `values/res/string.xml`, dan string dalam kode Anda digantikan dengan referensi ke sumber daya string tersebut.

```
@string/button_label_toast
```

5. Ekstrak dan beri nama string yang tersisa dari tampilan sebagai berikut:

| Tampilan | Nilai Sumber Daya / String | Nama sumber daya |
|----------|----------------------------|---------------------|
| Button | Hello Toast! | button_label_toast |
| TextView | 0 | count_initial_value |
| Button | Count | button_label_count |

6. Dalam tampilan Proyek, buka **values/strings.xml** untuk menemukan string Anda. Sekarang Anda bisa mengedit semua string di satu tempat.

3.3 Mengubah ukuran

Sama dengan string, sebaiknya ekstrak dimensi tampilan dari file XML layout utama ke dalam sumber daya dimensi yang terletak di dalam file.

Mengapa: Hal ini akan memudahkan pengelolaan dimensi, terutama jika Anda perlu menyesuaikan layout untuk resolusi perangkat yang berbeda. Ini juga akan memudahkan pengaturan ukuran yang konsisten, dan mengubah ukuran beberapa objek dengan mengubah satu properti.

Lakukan hal berikut:

1. Lihat file sumber daya dimens.xml. Harus ada nilai untuk margin layar default yang ditetapkan. Untuk dimensi tampilan, sebaiknya jangan gunakan nilai-nilai hard-code, karena nilai tersebut akan menghalangi tampilan untuk menyesuaikan ukuran dengan ukuran layar.
2. Kalau perlu, ubah layout_width semua elemen di dalam LinearLayout ke "match_parent".

Jika Anda ingin menggunakan Layout Editor grafis, klik tab **Design**, pilih setiap elemen dalam panel **Component Tree** dan ubah properti layout:width dalam panel **Properties**. Jika Anda ingin langsung mengedit file XML, klik tab **Text**, ubah android:layout_width untuk tombol pertama, TextView, dan Tombol terakhir.

3. Kalau perlu, ubah layout_height semua elemen di dalam LinearLayout ke "wrap_content".

3.4 Menyetel warna dan latar belakang

Gaya dan warna adalah properti tambahan yang dapat diekstrak ke dalam properti. Semua tampilan dapat memiliki latar belakang yang bisa berupa warna atau gambar.

Mengapa: Mengekstrak gaya dan warna memudahkan penggunaannya secara konsisten di seluruh aplikasi, dan mudah mengubah semua elemen UI.

Lakukan eksperimen dengan perubahan berikut:

1. Ubah ukuran teks TextView show_count. "sp" adalah singkatan dari piksel yang tidak tergantung pada skala, dan seperti dp, ini adalah unit yang skalanya menyesuaikan dengan kepadatan layar dan preferensi ukuran font pengguna. Sebaiknya Anda menggunakan unit ini saat menetapkan ukuran font, agar disesuaikan dengan kepadatan layar dan preferensi pengguna.

```
    android:textSize="160sp"
```

2. Ekstrak ukuran teks TextView sebagai sumber daya dimensi bernama `count_text_size`, sebagai berikut:
 - i. Klik tab **Text** untuk menunjukkan kode XML, jika belum.
 - ii. Letakkan kursor pada "`160sp`".
 - iii. Tekan **Alt-Enter (Option-Enter di Mac)**.
 - iv. Klik **Extract dimension resource**.
 - v. Setel **Resource name** ke `count_text_size` dan klik **OK**. (Jika Anda membuat kesalahan, batalkan perubahan dengan **Ctrl-Z**).
 - vi. Dalam tampilan Proyek, buka **values/dimens.xml** untuk menemukan dimensi Anda. File dimens.xml file berlaku untuk semua perangkat. File dimens.xml file untuk w820dp hanya berlaku untuk perangkat yang lebih lebar dari 820dp.
3. Ubah `textstyle` dari TextView `show_count` ke cetak tebal.


```
    android:textStyle="bold"
```
4. Ubah warna teks dalam tampilan teks `show_count` ke warna utama tema. Lihat file sumber daya colors.xml untuk melihat bagaimana file ditetapkan.

`colorPrimary` adalah salah satu dari warna dasar tema yang telah ditetapkan dan digunakan untuk bilah aplikasi. Misalnya, dalam aplikasi produksi Anda bisa menyesuaikan ini agar cocok dengan merek Anda. Menggunakan warna dasar untuk elemen UI membuat UI yang seragam. Lihat [Menggunakan Tema Material](#). Anda akan mengetahui selengkapnya tentang tema aplikasi dan desain material dalam praktik berikutnya.

```
    android:textColor="@color/colorPrimary"
```

5. Ubah warna kedua tombol ke warna utama tema.

```
    android:background="@color/colorPrimary"
```

6. Ubah warna teks dalam kedua tombol ke warna putih. Putih adalah salah satu warna yang disediakan sebagai Sumber Daya Platform Android. Lihat [Mengakses Sumber Daya](#).

```
    android:textColor="@android:color/white"
```

7. Tambahkan warna latar belakang ke TextView.

```
    android:background="#FFFF00"
```

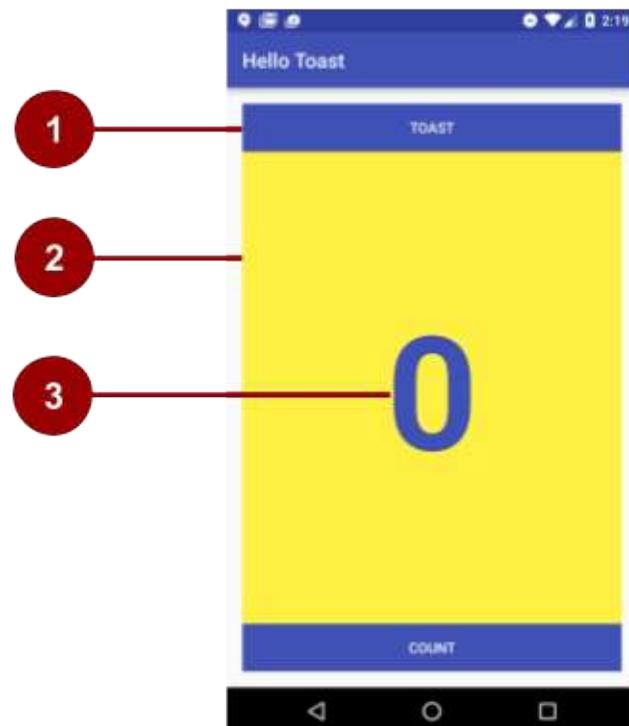
8. Dalam Layout Editor (tab Text), arahkan kursor ke warna ini dan tekan **Alt-Enter (Option-Enter di Mac)**.
9. Pilih **Choose color**, yang akan memunculkan pemilih warna, pilih warna yang Anda suka, atau pilih warna kuning yang sudah dipakai sekarang, lalu klik **Choose**.
10. Buka **values/colors.xml**. Perhatikan bahwa **colorPrimary** yang Anda gunakan sebelumnya ditetapkan di sini.
11. Menggunakan warna dalam **values/colors.xml** sebagai contoh, tambahkan sumber daya bernama

`myBackgroundColor` untuk warna latar belakang, lalu gunakan untuk menyetel latar belakang tampilan teks.

```
<color name="myBackgroundColor">#FFF043</color>
```

3.5 Gravitasi dan berat

Menentukan properti gravitasi dan berat memberi Anda kontrol tambahan untuk mengatur tampilan dan konten dalam



layout linear.

1. Android : atribut `layout_gravity` menentukan bagaimana tampilan diratakan dalam *Tampilan induk*. Karena lebar

- tampilan cocok dengan induknya, menyetelnya secara eksplisit tidak diperlukan. Anda dapat memusatkan tampilan sempit secara horizontal dalam induknya: `android:layout_gravity="center_horizontal"`
2. `android:atribut layout_weight` menunjukkan berapa banyak ruang ekstra dalam LinearLayout yang akan dialokasikan ke tampilan yang sudah menyetel parameter ini. Jika hanya satu tampilan yang memiliki atribut ini, tampilan itu akan mendapatkan semua ruang layar ekstra. Untuk beberapa tampilan, ruangan dipro-rata. Misalnya, jika kedua tombol memiliki berat 1 dan tampilan teks memiliki berat 2, sehingga totalnya 4, masing-masing tombol mendapatkan $\frac{1}{4}$ ruang, dan tampilan teks mendapatkan setengah.
 3. `Android:atribut gravity` menentukan perataan *konten Tampilan di dalam Tampilan itu sendiri*. Kounter dipusatkan dalam tampilannya dengan: `android:gravity="center"`

Lakukan hal berikut:

1. Pusatkan teks dalam TextView `show_count` secara horizontal dan vertikal: `android:gravity="center"`
2. Sesuaikan ukuran TextView `show_count` dengan ukuran layar:

```
    android:layout_weight="2"
```

Contoh Solusi: strings.xml

```
<resources>
    <string name="app_name">Hello Toast</string>
    <string name="button_label_count">Count</string>
    <string name="button_label_toast">Toast</string>
    <string name="count_initial_value">0</string>
</resources>
```

Contoh Solusi: dimens.xml

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="count_text_size">160sp</dimen>
</resources>
```

Contoh Solusi: colors.xml

```
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="myBackgroundColor">#FFF043</color>
</resources>
```

Contoh Solusi: activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="hellotoast.android.example.com.hellotoast.MainActivity">

    <Button
        android:id="@+id/button_toast"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/button_label_toast"
        android:background="@color/colorPrimary"
        android:textColor="@android:color/white" />

    <TextView
        android:id="@+id/show_count"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/count_initial_value"
        android:textSize="@dimen/count_text_size"
        android:textStyle="bold"
        android:textColor="@color/colorPrimary"
        android:background="@color/myBackgroundColor"
        android:layout_weight="2" />

    <Button
        android:id="@+id/button_count"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/button_label_count"
        android:background="@color/colorPrimary"
        android:textColor="@android:color/white" />
</LinearLayout>

```

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk bab berikutnya.

- Membuat proyek baru dengan 5 tampilan. Buat agar satu tampilan menggunakan setengah atas layar, dan tampilan 4 lainnya berbagi setengah bawah layar. Hanya gunakan LinearLayout, gravitasi, dan berat untuk melakukannya.
- Gunakan gambar sebagai latar belakang aplikasi Hello Toast. Tambahkan gambar ke folder drawable, lalu setel sebagai latar belakang tampilan root. Untuk keterangan lebih lanjut mengenai drawable, lihat [dokumentasi Sumber Daya Drawable](#).

Tugas 4: Menambahkan handler onclick untuk tombol

Pada tugas ini Anda akan menambahkan metode ke MainActivity yang mengeksekusi saat pengguna menekan tombol.

***Mengapa:** Aplikasi interaktif harus merespons ke masukan pengguna.

Untuk menghubungkan tindakan pengguna dalam tampilan ke kode aplikasi, Anda perlu melakukan dua hal:

- Tulis metode yang melakukan tindakan spesifik saat pengguna menekan tombol di layar.

- Kaitkan metode ini ke tampilan, agar metode ini dipanggil saat pengguna berinteraksi dengan tampilan.

4.1 Menambahkan properti onClick ke tombol

Handler klik adalah metode yang dipanggil saat pengguna menekuk elemen antarmuka pengguna. Dalam Android, Anda bisa menetapkan nama metode handler klik untuk setiap tampilan dalam file layout XML dengan properti `android:onClick`.

1. Buka `res/layout/activity_main.xml`.
2. Tambahkan properti berikut ini ke tombol `button_toast`.

```
    android:onClick="showToast"
```

3. Tambahkan atribut berikut ke tombol `button_count`:

```
    android:onClick="countUp"
```

4. Dalam `activity_main.xml`, letakkan kursor mouse Anda di atas setiap nama metode ini.

5. Tekan **Alt-Enter (Option-Enter di Mac)**, dan pilih **Create onClick event handler**.
6. Pilih **MainActivity** dan klik **OK**.

Tindakan ini akan membuat stub metode placeholder untuk metode `onClick` dalam `MainActivity.java`.

Catatan: Anda juga bisa menambahkan handler klik ke tampilan secara terprogram, yang akan Anda lakukan pada praktik berikutnya. Apakah Anda menambahkan handler klik di XML atau secara terprogram sepenuhnya adalah pilihan pribadi; meskipun, ada situasi saat Anda hanya bisa melakukannya secara terprogram.

MainActivity.java Solusi:

```
package hellotoast.android.example.com.hellotoast;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void countUp(View view) {
        // What happens when user clicks on the button_count Button goes here.
    }

    public void showToast(View view) {
        // What happens when user clicks on the button_toast Button goes here.
    }
}
```

4.2 Menampilkan toast saat tombol Toast diklik

Toast menyediakan feedback sederhana tentang operasi dalam popup kecil. Popup ini hanya akan memenuhi ruang yang diperlukan untuk pesan dan aktivitas saat ini tetap terlihat dan interaktif. **Toast** menyediakan cara lain untuk menguji interaktivitas aplikasi Anda.

Dalam `MainActivity.java`, tambahkan kode ke metode `showToast()` untuk menampilkan pesan toast.

Untuk membuat instance `Toast`, panggil metode pabrik `makeText()` pada kelas `Toast`, yang menyediakan konteks (lihat di bawah ini), pesan yang akan ditampilkan, dan durasi tampilan. Anda menampilkan toast yang memanggil `show()`. Ini adalah pola umum di mana Anda bisa menggunakan kembali kode yang akan ditulis.

- Dapatkan konteks aplikasi.

Menampilkan pesan `Toast` memerlukan konteks. `Konteks` aplikasi berisi informasi global tentang lingkungan aplikasi. Karena toast ditampilkan di atas UI yang terlihat, sistem memerlukan informasi tentang aktivitas saat ini.

```
Context context = getApplicationContext();
```

Saat sudah berada dalam konteks aktivitas yang konteksnya diperlukan, Anda juga bisa menggunakan `this` sebagai pintasan ke konteks.

- Panjang string toast bisa panjang atau pendek, dan Anda menetapkan yang mana dengan konstanta `Toast`.

- `Toast.LENGTH_LONG`
- `Toast.LENGTH_SHORT`

Panjang sebenarnya adalah 3,5d untuk toast panjang dan 2d untuk toast pendek. Nilai ditetapkan dalam kode sumber Android. Lihat detail [pos Stackoverflow ini](#).

- Buat instance kelas `Toast` dengan konteks, pesan, dan durasi.

- Konteksnya adalah konteks aplikasi yang kita dapatkan sebelumnya.
- Pesannya adalah string yang ingin Anda tampilkan.
- Durasinya adalah salah satu konstanta yang sudah ditetapkan sebelumnya `Toast.LENGTH_LONG` atau `Toast.LENGTH_SHORT`.

```
Toast toast = Toast.makeText(context, "Hello Toast", Toast.LENGTH_LONG);
```

- Ekstrak string "Hello Toast" ke dalam sumber daya string dan beri nama `toast_message`.

- Letakkan kursor pada string `"Hello Toast"`.
- Tekan **Alt-Enter (Option-Enter di Mac)**.
- Pilih **Extract string resources**.
- Setel **Resource name** ke `toast_message` dan klik **OK**.

Tindakan ini akan menyimpan "Hello World" sebagai nama sumber daya string `toast_message` dalam file sumber daya `string res/values/string.xml`. Parameter string dalam panggilan metode Anda akan digantikan dengan referensi ke sumber daya.

- `R` mengidentifikasi parameter sebagai sumber daya.
- `string` mereferensikan nama file XML tempat sumber daya ditetapkan.
- `toast_message` adalah nama sumber daya.

```
Toast toast = Toast.makeText(context, R.string.toast_message, duration);
```

- Tampilkan toast.

```
toast.show();
```

- Jalankan aplikasi dan verifikasi bahwa toast ditampilkan saat tombol **Toast** diketuk.

Solusi:

```

/*
 * When the TOAST button is clicked, show a toast.
 *
 * @param view The view that triggers this onClick handler.
 *             Since a toast always shows on the top, view is not used.
 */
public void showToast(View view) {
    // Create a toast show it.
    Toast toast = Toast.makeText(this, R.string.toast_message, Toast.LENGTH_LONG);
    toast.show();
}

```

4.3 Meningkatkan hitungan dalam tampilan teks saat tombol Count diklik

Untuk menampilkan hitungan saat ini dalam tampilan teks:

- Awasi hitungan saat berubah.
- Kirimkan hitungan terbaru ke tampilan teks untuk menampilkannya di antarmuka pengguna.

Implementasikan ini dengan cara berikut:

1. Dalam MainActivity.java, tambahkan variabel anggota privat `mCount` untuk melacak hitungan dan memulainya dari 0.
2. Dalam MainActivity.java, tambahkan variabel anggota privat `mShowCount` untuk mendapatkan referensi dari TextView `show_count`.
3. Dalam metode `countUp()`, tingkatkan nilai variabel hitungan setiap kali tombol diklik.
4. Dapatkan referensi ke tampilan teks menggunakan id yang Anda setel dalam file layout.

Tampilan, seperti string dan dimensi, adalah sumber daya yang dapat memiliki id. Panggilan `findViewById()` mengambil id tampilan sebagai parameternya dan mengembalikan tampilan. Karena metode ini mengembalikan Tampilan, Anda harus menampilkan hasilnya ke jenis tampilan yang diharapkan, dalam kasus ini (`TextView`) .

Untuk mendapatkan sumber daya ini sekali saja, gunakan variabel anggota dan setel di dalam `onCreate()` .

```
mShowCount = (TextView) findViewById(R.id.show_count);
```

5. Setel teks dalam tampilan teks ke nilai variabel count.

```

if (mShowCount != null)
    mShowCount.setText(Integer.toString(mCount));

```

6. Jalankan aplikasi untuk memverifikasi bahwa hitungan meningkat saat tombol **Count** ditekan.

Solusi:

Definisi kelas dan menginisialisasi variabel count:

```

public class MainActivity extends AppCompatActivity {
private int mCount = 0;
private TextView mShowCount;

```

dalam `onCreate()`:

```

mShowCount = (TextView) findViewById(R.id.show_count);

countUp Method:
public void countUp(View view) {
    mCount++;
    if (mShowCount != null)
        mShowCount.setText(Integer.toString(mCount));
}

```

Sumber daya:

- Ketahui selengkapnya tentang menangani [Kejadian Masukan Android](#).
- [Dokumentasi kelas konteks](#)

Kode solusi

Proyek Android Studio: [HelloToast](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk bab berikutnya.

Tantangan: Bahkan aplikasi sederhana seperti Hello Toast bisa menjadi dasar dari aplikasi pemberian skor atau pemesanan produk. Tulis satu aplikasi yang akan berguna bagi Anda, atau coba salah satu dari contoh berikut:

- Buat aplikasi pemesanan kopi. Tambahkan tombol untuk mengubah jumlah kopi yang dipesan. Hitung dan tampilkan harga total.
- Buat aplikasi pemberian skor untuk olahraga tim favorit Anda. Buat gambar latar belakang yang mewakili olahraga tersebut. Buat tombol untuk menghitung skor setiap tim.

Rangkuman

Dalam bab ini, Anda:

- Menambahkan elemen UI ke aplikasi dalam Layout Editor dan menggunakan XML.
- Membuat interaktif UI dengan tombol dan mengeklik listener.
- Menambahkan listener klik yang memperbarui tampilan teks untuk merespons masukan pengguna.
- Informasi yang ditampilkan ke pengguna menggunakan toast.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Layout, Tampilan, dan Sumber Daya](#)

Ketahui selengkapnya

- Semua tampilan adalah subkelas [kelas Tampilan](#) dan dengan demikian mewarisi banyak properti dari superkelas [Tampilan](#).
- Anda bisa menemukan informasi tentang semua properti Tombol di [dokumentasi kelas Tombol](#), dan semua properti [TextView](#) dalam [dokumentasi kelas TextView](#).
- Anda bisa menemukan informasi tentang semua properti [LinearLayout](#) dalam [dokumentasi kelas LinearLayout](#).
- [Dokumentasi sumber daya Android](#) akan menjelaskan jenis sumber daya lainnya.
- Konstanta warna Android: [sumber daya R.color standar Android](#)
- Informasi selengkapnya tentang dp dan sp dapat ditemukan di [Mendukung Kepadatan Berbeda](#)

Dokumentasi Developer

- [Dokumentasi Android Studio](#)
- [Kosakata dan konsep daftar istilah](#)
- [Layout developer.android.com](#)

- Dokumentasi kelas tampilan
- piksel yang tidak tergantung perangkat
- Dokumentasi kelas tombol
- Dokumentasi kelas TextView
- Dokumentasi sumber daya Android
- Selesaikan kode untuk aplikasi Hello Toast
- Pola Arsitektur (ringkasan)

1.2B: Menggunakan Layout

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Mengubah layout ke RelativeLayout](#)
- [Tugas 2: Mengubah layout ke ConstraintLayout](#)
- [Tugas 3: Membuat varian layout](#)
- [Tantangan Penyusunan Kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Saat Anda memulai proyek Android Studio, template yang dipilih menyediakan layout dasar dengan berbagai tampilan. Seperti yang telah dipelajari di praktik sebelumnya, Anda dapat menjajarkan tampilan dengan cepat dan mudah dalam sebuah layout menggunakan [LinearLayout](#), grup tampilan yang meluruskan tampilan anak di dalamnya secara horizontal maupun vertikal.

Praktik ini menjelajahi dua grup tampilan layout lainnya:

- [RelativeLayout](#): Grup tampilan anak tempat setiap tampilan diletakkan dan diluruskan relatif terhadap tampilan lainnya di dalam grup tampilan tersebut. Posisi tampilan anak dijelaskan dalam kaitannya dengan satu sama lain atau dengan grup tampilan induk.
- [ConstraintLayout](#): Layout yang serupa dengan RelativeLayout tetapi lebih fleksibel. Layout ini mengelompokkan tampilan anak menggunakan titik jangkar (koneksi ke tampilan lainnya), tepi, dan panduan untuk mengontrol bagaimana tampilan diletakkan relatif terhadap elemen lain di layout. ConstraintLayout didesain untuk membuat tampilan mudah diseret dan dilepaskan di editor layout Android Studio

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Membuat aplikasi Hello World dengan Android Studio.
- Menjalankan aplikasi di emulator atau perangkat.
- Mengimplementasikan TextView dalam layout untuk aplikasi.
- Membuat dan menggunakan sumber daya string.
- Mengonversi dimensi layout ke sumber daya.

Yang akan Anda PELAJARI

Anda akan belajar:

- Menggunakan editor layout di Android Studio
- Memposisikan tampilan di RelativeLayout
- Memposisikan tampilan di ConstraintLayout
- Membuat varian layout untuk orientasi lanskap dan tampilan yang lebih besar

Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

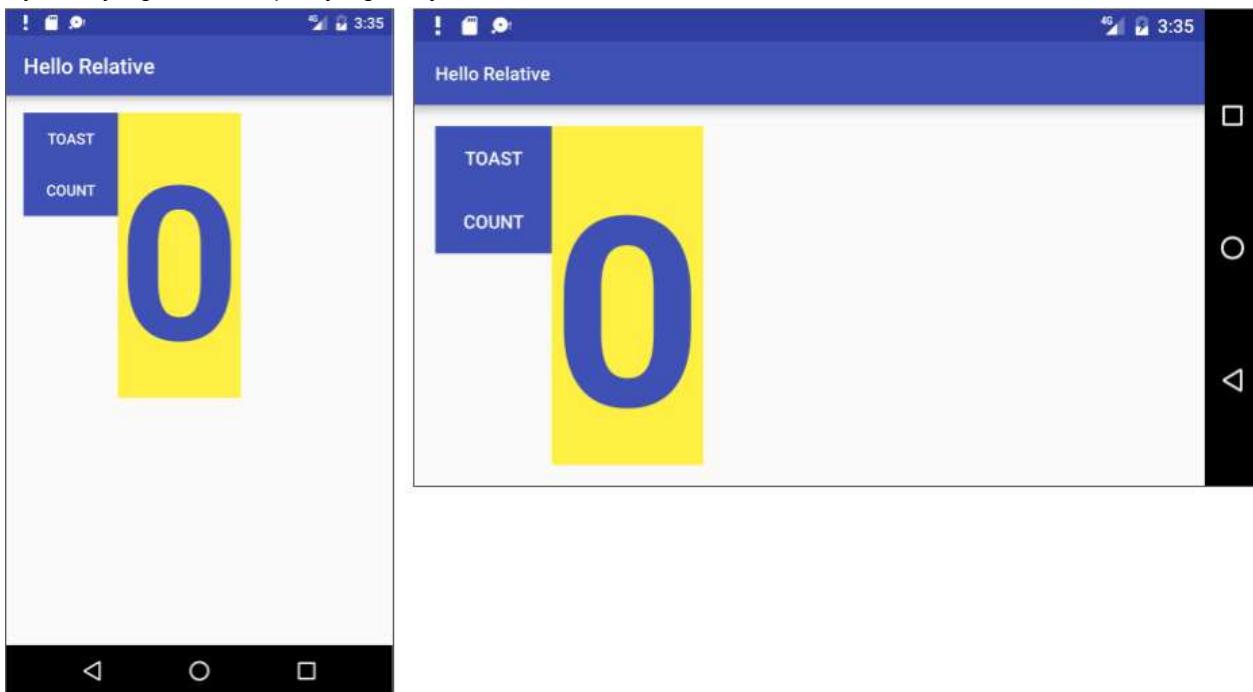
- Bereksperimen dengan menggunakan RelativeLayout dan ConstraintLayout.
- Menyalin dan melakukan refaktorisasi aplikasi Hello Toast untuk membuat aplikasi Hello Relative.
- Mengubah grup tampilan root di layout utama menjadi RelativeLayout.
- Mengatur ulang tampilan di layout utama menjadi relatif terhadap satu sama lain.
- Menyalin dan melakukan refaktorisasi aplikasi Hello Relative untuk membuat Hello Constraint.
- Mengubah grup tampilan root di layout utama menjadi ConstraintLayout.
- Memodifikasi layout untuk menambahkan pembatas ke tampilan.
- Memodifikasi tampilan untuk varian layout untuk orientasi lanskap dan tampilan yang lebih besar.

Ringkasan Aplikasi

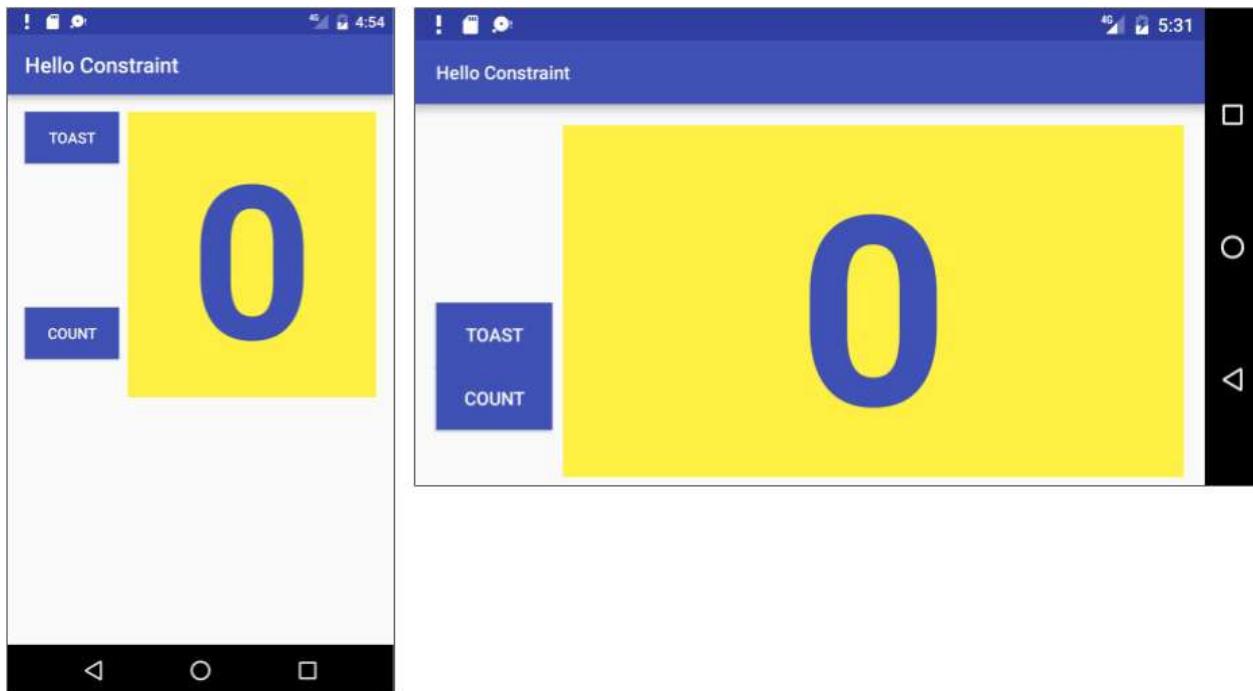
Aplikasi Hello Toast di praktik sebelumnya menggunakan LinearLayout untuk mengatur tampilan di layout aktivitas, seperti yang ditunjukkan pada gambar di bawah ini.



Untuk berlatih menggunakan editor layout, Anda akan menyalin aplikasi Hello Toast dan memanggil salinan baru Hello Relative, untuk bereksperimen dengan RelativeLayout. Anda akan menggunakan editor layout untuk mengatur tampilan di layout UI yang berbeda seperti yang ditunjukkan di bawah ini.



Akhirnya, Anda akan membuat salinan aplikasi dan menyebutnya Hello Constraint, dan mengganti LinearLayout dengan ConstraintLayout. ConstraintLayout menawarkan lebih banyak bantuan visual dan fitur pemosisian di editor layout. Anda akan membuat layout UI yang sangat berbeda, dan juga varian layout untuk orientasi lanskap dan tampilan yang lebih besar, seperti yang ditunjukkan di bawah ini.



Proyek Android Studio: [HelloToast](#)

Tugas 1: Mengubah layout ke RelativeLayout

RelativeLayout adalah pengelompokan tampilan tempat setiap tampilan diletakkan dan diluruskan relatif terhadap tampilan lainnya di dalam grup tersebut. Dalam tugas ini, Anda akan menyelidiki menggunakan RelativeLayout.

1.1 Menyalin dan melakukan refaktorisasi aplikasi Hello Toast

1. Salin folder proyek **HelloToast**, ganti namanya menjadi **HelloRelative**, dan lakukan refaktorisasi. (Lihat [Appendix](#) untuk membaca instruksi cara menyalin proyek.)
2. Setelah melakukan refaktorisasi, ubah nilai `<string name="app_name">` dalam file **strings.xml** (dalam **app > res > values**) menjadi **Hello Relative** (dengan spasi) sebagai nama aplikasi.

1.2 Mengubah LinearLayout menjadi RelativeLayout

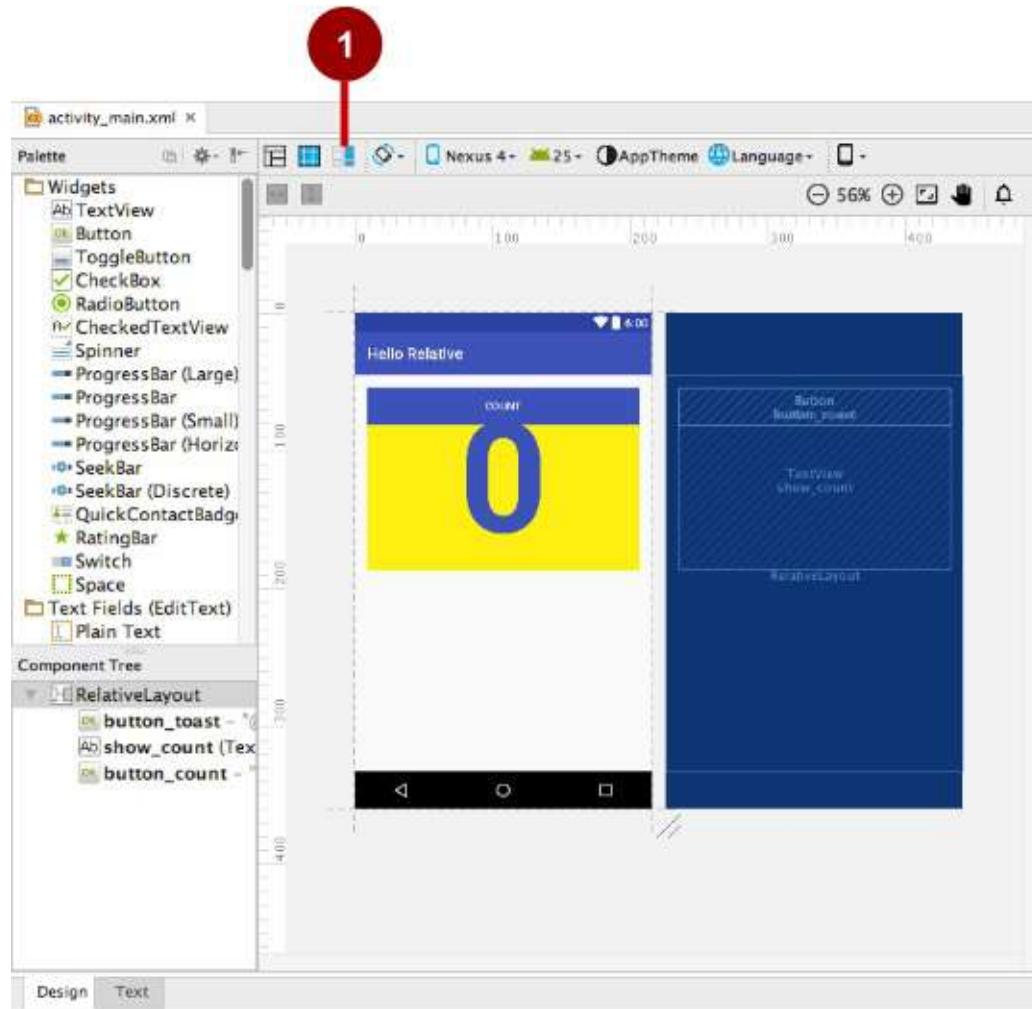
1. Buka file layout **activity_main.xml**, dan klik tab **Text** di bawah panel pengeditan untuk melihat kode XML.
2. Ubah `<LinearLayout` at the top to `<RelativeLayout` so that the statement looks like this:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

3. Gulir ke bawah untuk memastikan bahwa tag akhirannya `</LinearLayout>` has also changed to `</RelativeLayout>`; jika belum, ubah secara manual.

1.3 Atur ulang tampilan dengan tab Design

1. Klik tab **Design** di bagian bawah panel pengeditan.
2. Seharusnya panel pengeditan sekarang terlihat seperti gambar di bawah, dengan desain layout dan blueprint-nya. Jika hanya melihat desain layout, atau blueprint saja, klik tombol **Show Design + Blueprint** (#1 dalam gambar di bawah

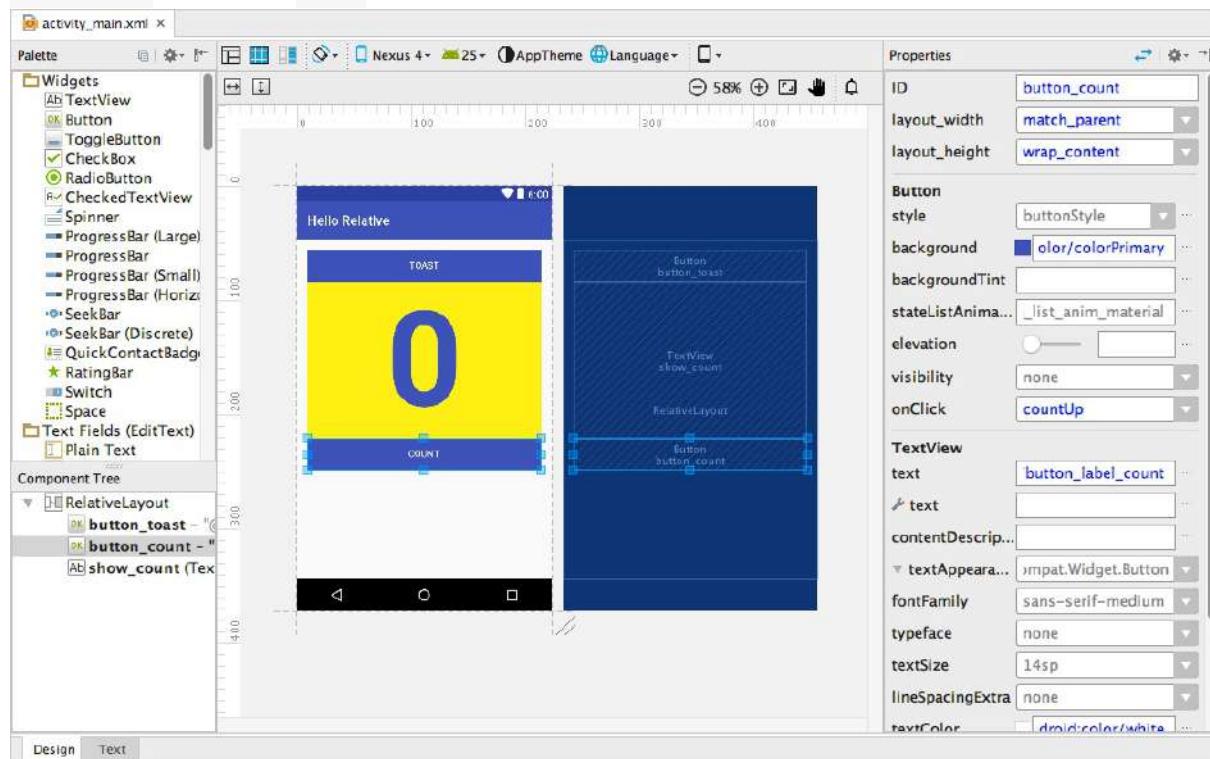


3. Dengan perubahan ke `RelativeLayout`, layout editor juga mengubah beberapa atribut tampilan. Contohnya:

- o Tampilan `button_count` untuk tombol `COUNT` button melapisi tampilan `button_toast` untuk tombol `TOAST`, karena itu Anda tidak dapat melihat tombol `TOAST`. Tetapi, di blueprint, Anda bisa melihat bahwa dua tombol

tersebut menempati ruang yang sama.

- Bagian atas tampilan `show_count` (yang menunjukkan 0) juga melapisi tombol `COUNT` dan `TOAST`.
4. Seret tampilan `button_count` (untuk tombol `COUNT`) ke area di bawah tampilan `show_count` (yang menunjukkan 0), lalu seret ke bagian bawah tampilan `show_count` sampai posisinya pas seperti yang terlihat di bawah ini. Juga seret tampilan `show_count` agar bagian atas tampilan dipaskan posisinya di bagian bawah tampilan `button_toast` untuk tombol `TOAST`.



Tips: Saat memilih tampilan di layout, propertinya muncul di panel Properties di sisi kanan panel pengeditan. Properti ini sesuai dengan atribut XML di kode XML untuk layout, yang akan Anda periksa di langkah berikutnya.

1.4 Memeriksa kode XML di tab Text

Ikuti langkah-langkah berikut untuk melihat tampilan aplikasi tersebut:

1. Jalankan aplikasi. Aplikasi bekerja dengan cara yang sama seperti sebelumnya. Satu-satunya perbedaan adalah layout menggunakan `RelativeLayout` untuk memposisikan elemen, alih-alih `LinearLayout`. Pada tugas berikutnya, Anda akan mengubah layout UI-nya.
2. Ubah orientasi perangkat atau emulator ke lanskap. Perhatikan bahwa tampilan `button_count` menghilang karena layout layar tidak mengakomodasi orientasi lanskap. Anda akan mengatasi masalah ini dalam tugas berikutnya pada praktik ini.
3. Klik tab **Text** di bawah panel pengeditan.
4. Periksa perubahan pada kode XML di panel pengeditan sebagai hasil mengubah `LinearLayout` ke `RelativeLayout`. Mulai dengan memeriksa Tombol kedua (`button_count`):

```
<Button
    android:id="@+id/button_count"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:onClick="countUp"
    android:text="@string/button_label_count"
    android:textColor="@android:color/white"
    android:layout_below="@+id/show_count"
    android:layout_centerHorizontal="true" />
```

Dua atribut XML baru secara otomatis ditambahkan ke layout editor setelah Anda memindahkan Tombol (`button_count`) dalam layout:

```
android:layout_below="@+id/show_count"
android:layout_centerHorizontal="true"
```

Atribut `android:layout_below` meletakkan tampilan `button_count` tepat di bawah tampilan `show_count`. Atribut ini adalah salah satu dari beberapa atribut untuk memposisikan tampilan di dalam `RelativeLayout` — Anda menempatkan tampilan dalam hubungannya dengan Kode XML untuk tampilan `show_count`, yang juga Anda pindahkan di layout editor, sekarang berada di posisi di bawah dua tombol pada tampilan Teks. Ini karena perubahan dari `LinearLayout` ke `RelativeLayout`. Tampilan `show_count` sekarang juga menyertakan atribut berikut, sebagai hasil memindahkan tampilan di editor layout:

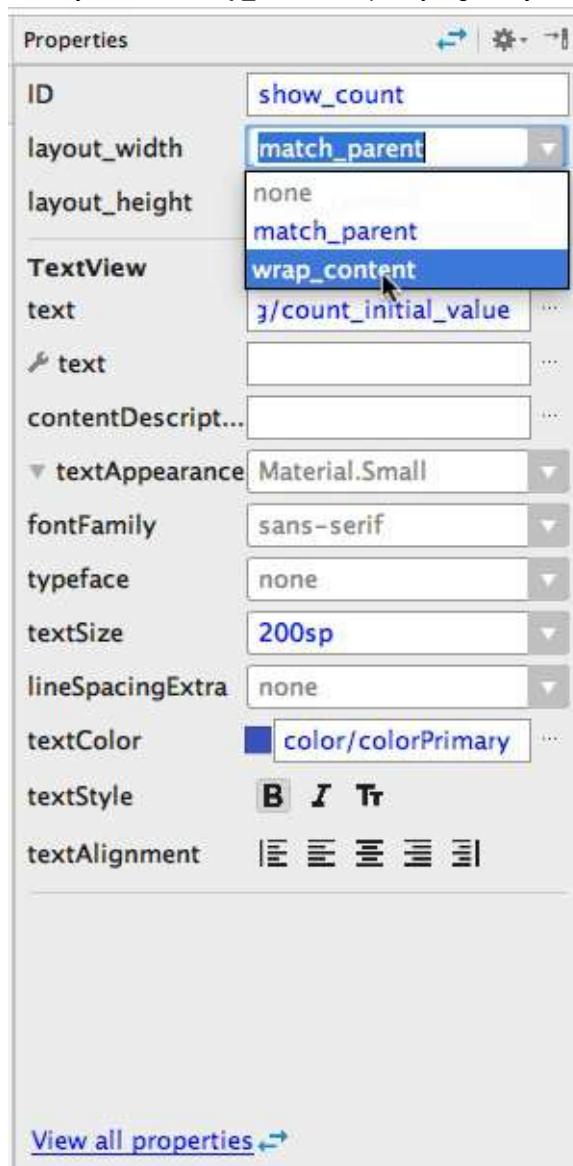
```
android:layout_below="@+id/button_toast"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
```

`android:layout_alignParentLeft` meratakan tampilan ke sisi kiri grup tampilan induk `RelativeLayout`. Walaupun atribut ini sendiri sudah cukup untuk meluruskan tampilan ke sisi kiri, Anda mungkin ingin tampilan diluruskan ke sisi kanan *jika* aplikasi berjalan di perangkat yang menggunakan bahasa kanan ke kiri. Oleh karena itu, atribut `android:layout_alignParentStart` membuat tepi "mulai" tampilan ini cocok dengan tepi mulai induknya. Tepi *mulai* adalah tepi kiri layar jika preferensinya kiri ke kanan, atau tepi kanan layar jika preferensinya kanan ke kiri.

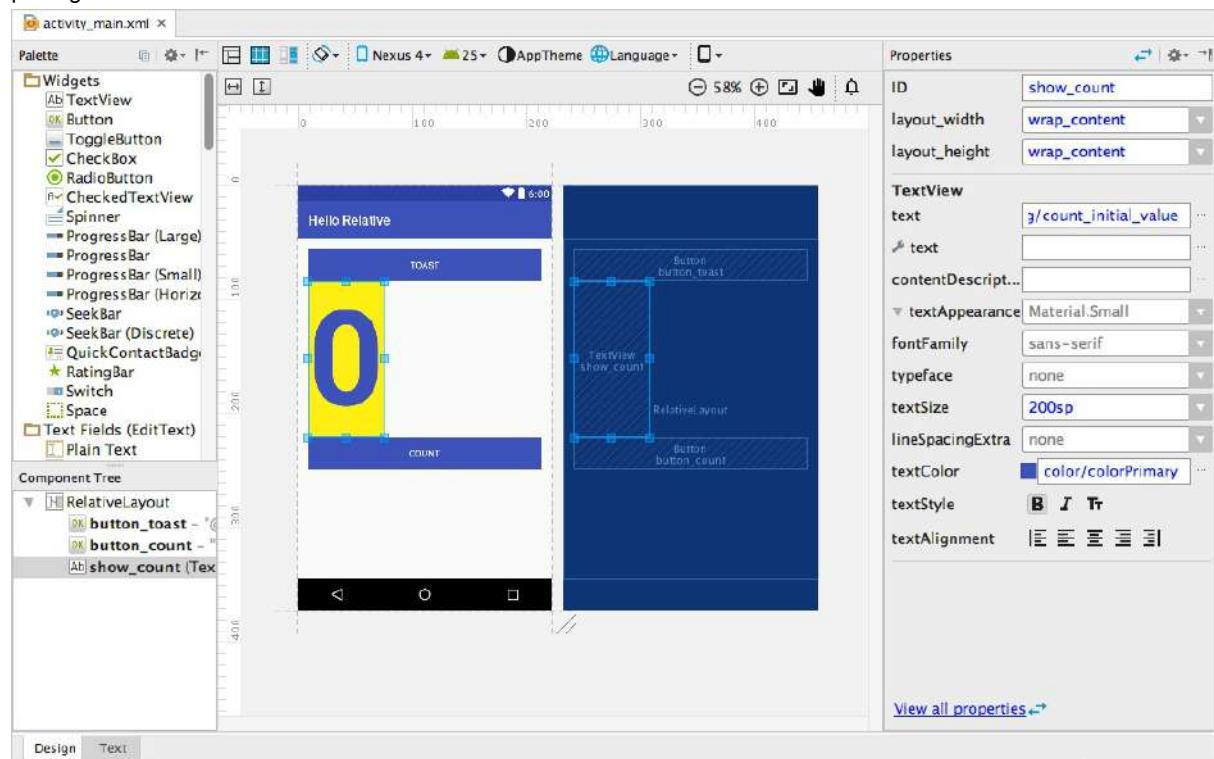
1.5 Mengatur ulang elemen dalam `RelativeLayout`

- Untuk bereksperimen lebih lanjut dengan `RelativeLayout`, pilih layout `activity_main.xml` lagi untuk mengedit (jika belum dipilih), dan klik tab **Design** di bawah panel pengeditan.

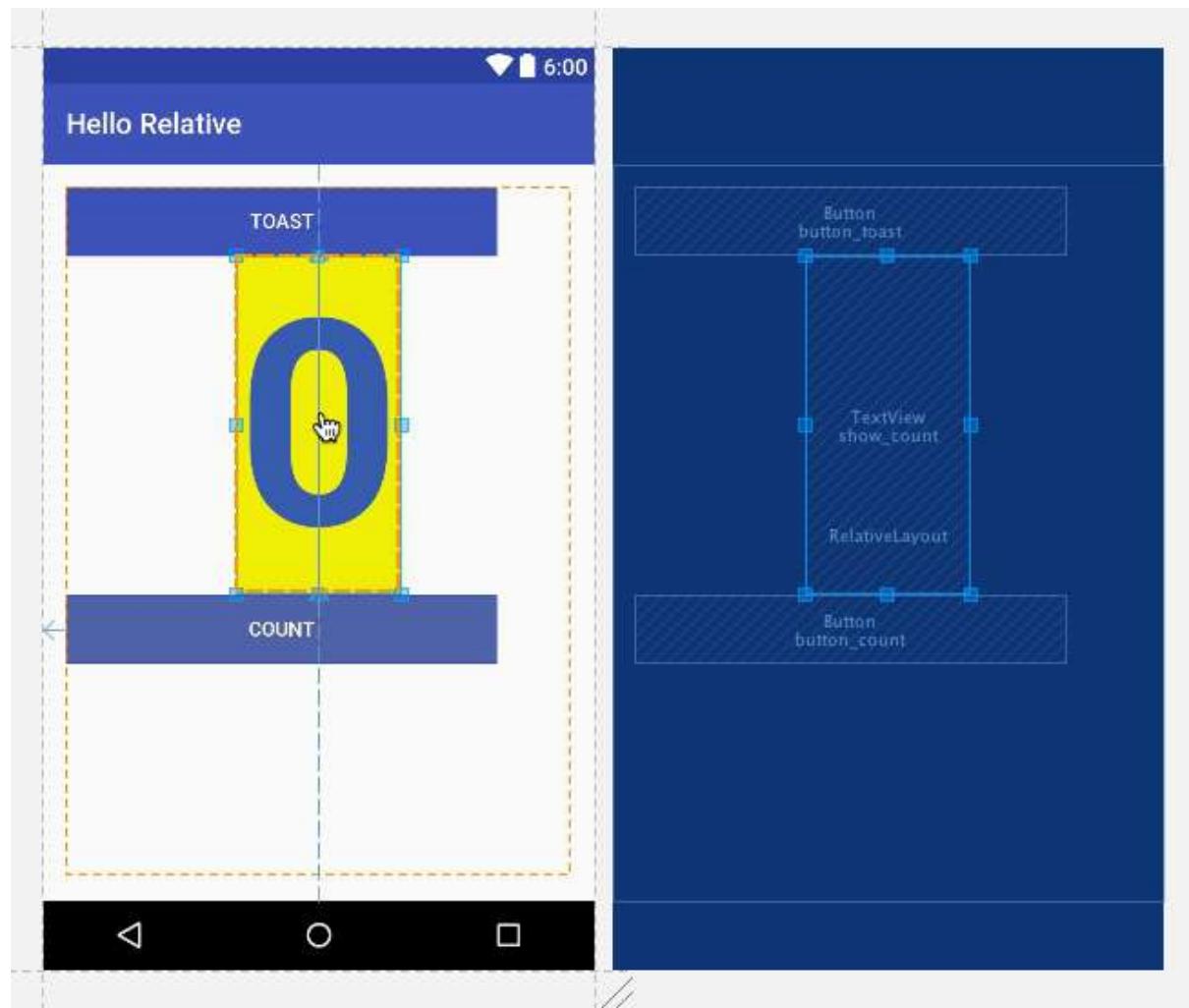
2. Pilih tampilan `show_count` dalam layout atau Component Tree, dan ubah `layout_width` dalam panel Properties di sisi kanan jendela ke **wrap_content** seperti yang ditunjukkan pada gambar di bawah ini.



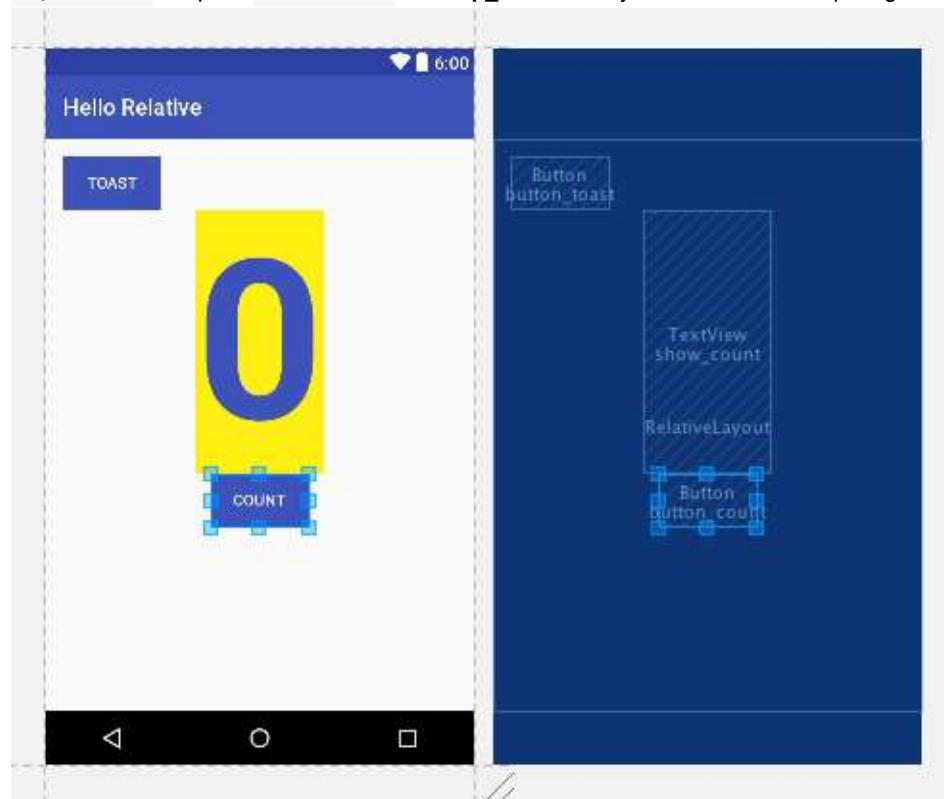
Layout editor menampilkan tampilan `show_count` yang diluruskan ke sisi kiri tampilan induk, seperti yang ditunjukkan pada gambar di bawah ini.



3. Seret tampilan `show_count` secara horizontal ke tengah layout. Saat Anda menyeret tampilan, panduan pusat muncul — pusat tampilan harus dipasangkan posisinya seperti panduan di bawah ini.

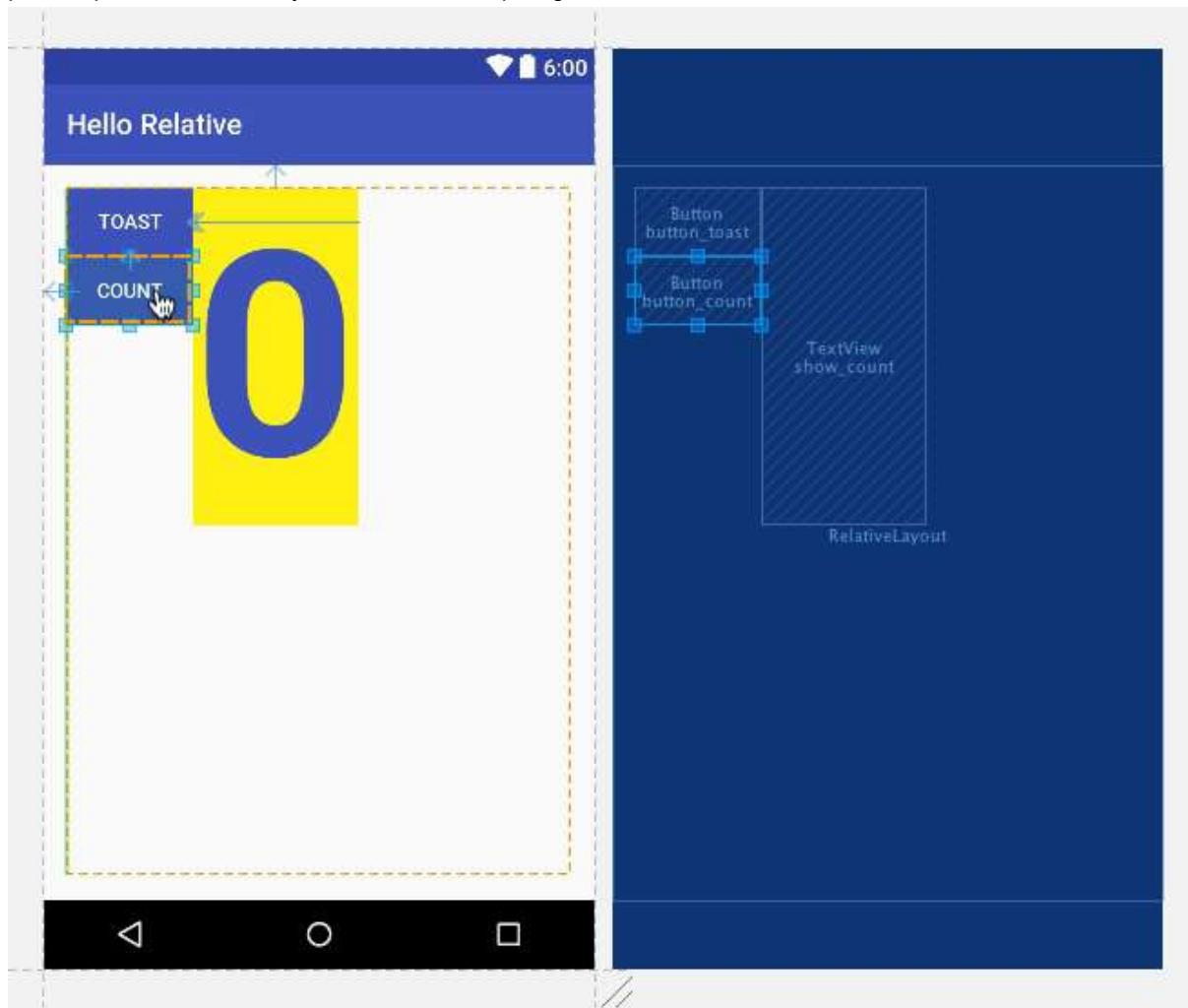


4. Pilih tampilan `button_toast` dan ubah `layout_width` ke **wrap_content** dalam panel Properties, lalu ubah `layout_width` tampilan `button_count` ke **wrap_content**. Layout harus terlihat seperti gambar di bawah ini.



5. Seret tampilan `button_count` tepat bawah tampilan `button_toast` agar posisinya pas di bagian bawah tampilan

`button_toast`, dan seret tampilan `show_count` ke atas di sebelah tepi kanan tampilan `button_toast` agar posisinya pas di tepi kanan tombol. Layout harus terlihat seperti gambar di bawah ini:



- Klik tab **Text** di bawah panel pengeditan dan periksa perubahan pada kode XML sebagai akibat dari pemindahan tampilan di layout:

- Tampilan `show_count` sekarang menggunakan atribut berikut untuk memposisikannya ke *kanan* dan *ujung* tampilan `button_toast` :

```

    android:layout_toRightOf="@+id/button_toast"
    android:layout_toEndOf="@+id/button_toast"

```

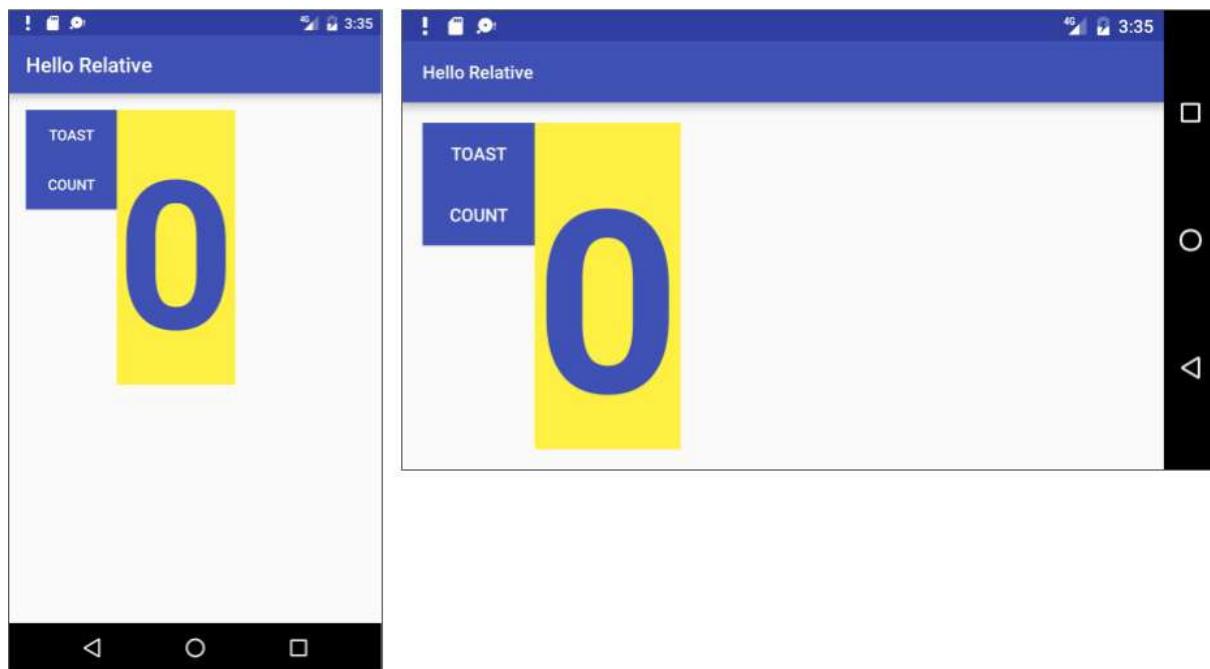
- Tampilan `button_count` sekarang menggunakan atribut berikut untuk memposisikannya ke *bawah* tampilan `button_toast` :

```

    android:layout_below="@+id/button_toast"

```

- Jalankan aplikasi. Aplikasi bekerja dengan cara yang sama seperti sebelumnya (karena kita tidak mengubah kode Java apa pun). Tetapi, layout-nya berbeda, seperti yang ditunjukkan pada gambar di bawah ini. Ubah orientasi perangkat atau emulator ke lanskap untuk melihat bahwa layout baru berfungsi untuk kedua orientasi.



Tips: Untuk mengetahui selengkapnya tentang cara memposisikan tampilan di RelativeLayout, lihat "[Memposisikan Tampilan](#)" di topik "Relative Layout" API Guide.

Kode Solusi: Android Studio project: [HelloRelative](#)

Tugas 2: Mengubah layout ke ConstraintLayout

ConstraintLayout adalah grup tampilan yang tersedia di pustaka Constraint Layout, yang telah disertakan bersama Android Studio 2.2 dan yang lebih tinggi. Layout berbasis pembatas memungkinkan developer membangun layout kompleks tanpa perlu melapis grup tampilan, yang dapat meningkatkan kinerja aplikasi. Layout dibangun ke dalam layout editor, sehingga alat penghambat dapat diakses dari tab **Design** tanpa perlu mengedit XML secara manual.

Pada tugas ini Anda akan menyalin dan melakukan refaktorisasi aplikasi Hello Toast untuk membuat aplikasi Hello Constraint. Lalu Anda akan mengubah grup tampilan `LinearLayout` root di layout utama menjadi `ConstraintLayout`. Setelah mengubah grup tampilan root, Anda akan mengatur ulang tampilan di layout utama untuk menghadirkan pembatas yang mengatur tampilannya.

2.1 Menyalin dan melakukan refaktorisasi aplikasi Hello Toast

1. Salin folder proyek **HelloToast**, ganti namanya menjadi **HelloConstraint**, dan lakukan refaktorisasi. (Lihat [Appendix](#) untuk membaca instruksi cara menyalin proyek.)
2. Setelah melakukan refaktorisasi, ubah nilai `string name="app_name"` dalam file `strings.xml` (dalam `app > res > values`) menjadi **Hello Constraint** (dengan spasi) sebagai nama aplikasi.

2.2 Menambahkan ConstraintLayout ke proyek Anda

Periksa untuk memastikan ConstraintLayout tersedia di proyek Anda:

1. Di Android Studio, pilih **Tools > Android > SDK Manager**.
2. Di panel kiri, klik **Android SDK**.
3. Di panel kanan, klik tab **SDK Tools** di bagian atas panel.
4. Luaskan **Support Repository** dan lihat apakah ConstraintLayout untuk Android dan Solver untuk ConstraintLayout sudah diperiksa.
 - Jika kata "Installed" muncul di kolom status, Anda sudah siap. Klik **Cancel**.
 - Jika "Not Installed" atau "Update" muncul:

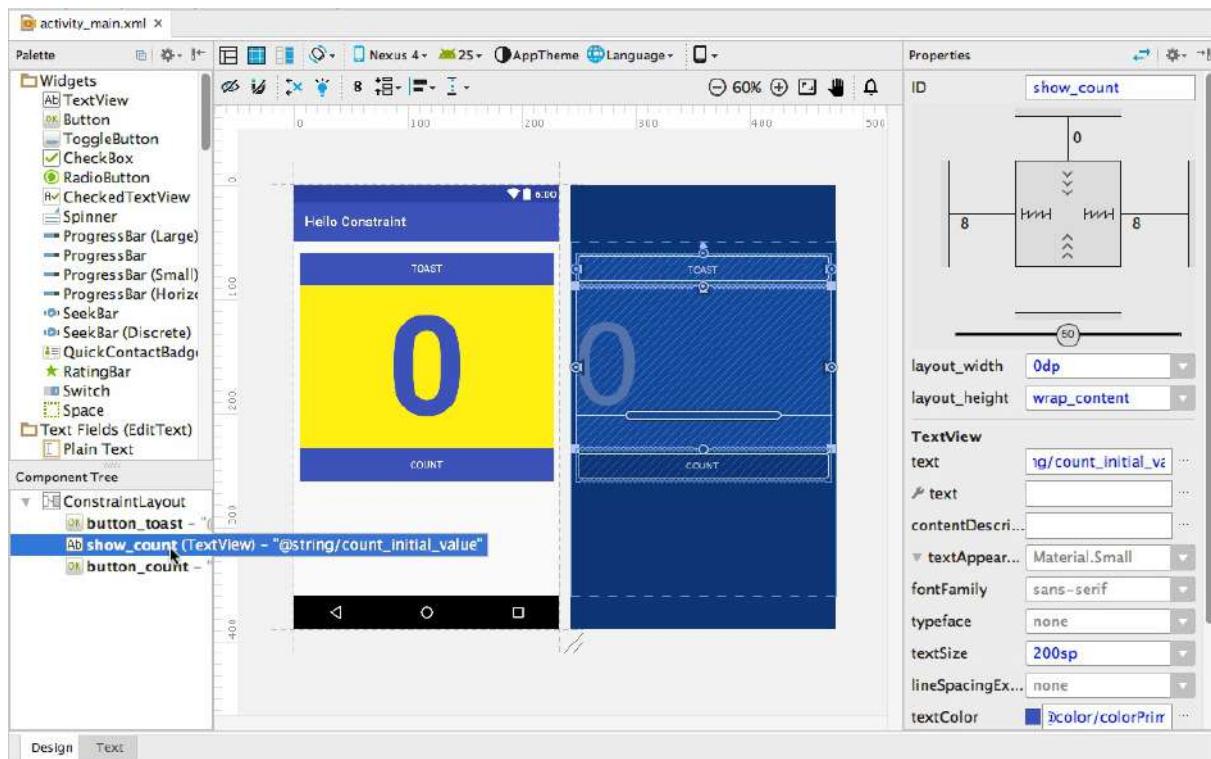
- i. Klik kotak centang di samping ConstraintLayout untuk Android dan Solver untuk ConstraintLayout. Ikon unduh seharusnya muncul di samping setiap kotak centang.
- ii. Klik salah satu dari yang berikut:
 - **Apply** untuk mulai memasang komponen dan tetap di SDK Manager untuk membuat perubahan lainnya.
 - **OK** untuk memasang komponen.
- iii. Setelah memasang komponen (dan membuat perubahan lain jika perlu), klik **Finish** saat selesai menggunakan SDK Manager.

2.3 Mengonversikan layout ke ConstraintLayout

Android Studio memiliki konverter bawaan untuk membantu Anda mengonversi layout ke `constraintLayout`. Ikuti langkah-langkah berikut:

1. Buka file layout (**activity_main.xml**) di Android Studio dan klik tab **Design** di bagian bawah jendela editor.
2. Di jendela Component Tree, klik kanan **LinearLayout** lalu pilih **Convert layout to ConstraintLayout** dari menu konteks.
3. Konverter menampilkan peringatan dengan dua kotak centang yang sudah dicentang. Jangan menghapus centang, pastikan kedua opsi tetap *dicentang*:
 - i. Ratakan Hierarki Layout: Opsi ini menghapus semua layout berlapis pada hierarki. Hasilnya adalah layout tunggal dan rata, yang mungkin lebih efisien untuk tujuan ini.
 - ii. Jangan meratakan layout yang direferensikan dari file lain: Jika layout tertentu mendefinisikan sebuah `android:id` attribute yang direferensikan di kode Java, Anda mungkin tidak ingin meratakannya karena kode mungkin tidak berfungsi lagi. Tetapi, pada HelloConstraint, Anda tidak memiliki `android:id` attribute untuk layout, hanya untuk tampilan.
4. Pada peringatan Add Project Dependency, klik **OK** untuk menambahkan pustaka layout pembatas. Android Studio secara otomatis menyalin dependensi yang sesuai ke file **build.gradle (Module: app)** proyek dan menyinkronkan perubahannya untuk Anda. Layout editor muncul kembali dengan `ConstraintLayout` sebagai grup tampilan root.

Catatan: Jika layout editor bermasalah dengan perubahan, Anda akan melihat peringatan Masalah Perenderan. Klik **build** pada pesan `Tip: Coba membangun proyek.` Ini akan menyinkronkan ulang file **build.gradle (Module: app)** proyek dengan dependensi baru.



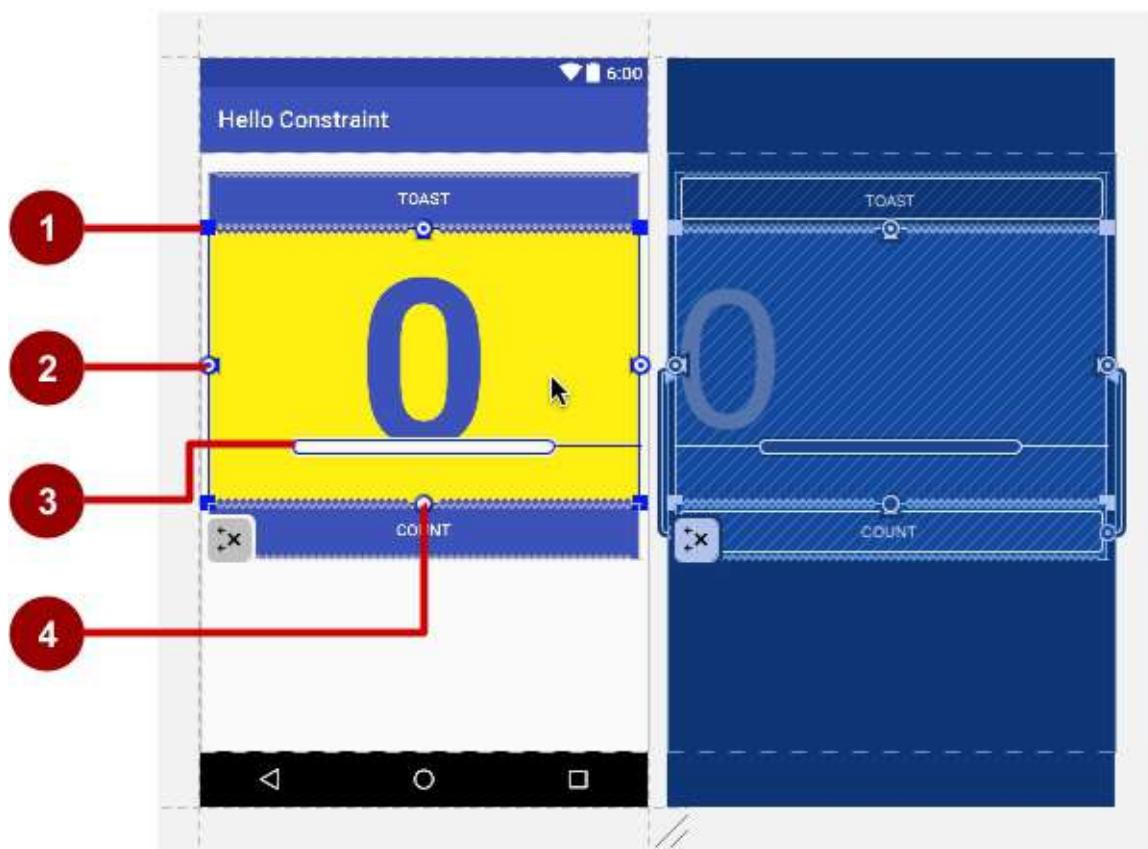
2.4 Menjelajahi layout editor

Layout editor menawarkan lebih banyak fitur di tab **Design** saat Anda menggunakan `ConstraintLayout`, termasuk lebih banyak alat layout visual dan baris kedua ikon untuk alat lainnya.

Layout visual dan blueprint menawarkan handle untuk mendefinisikan pembatas. *Pembatas* adalah koneksi atau perataan ke tampilan lain, layout induk, atau panduan yang tidak terlihat. Ikuti langkah-langkah berikut untuk menjelajah pembatas yang dibuat Android Studio saat Anda mengonversi `LinearLayout` ke `ConstraintLayout`:

1. Klik tampilan `show_count` di panel Component Tree.
2. Arahkan kursor ke atas tampilan `show_count` di layout, seperti yang ditunjukkan pada gambar di bawah ini.

Setiap pembatas muncul sebagai garis yang meluas dari handle lingkaran. Setiap tampilan memiliki handle pembatas lingkaran di tengah setiap sisi. Setelah memilih tampilan di panel Component Tree atau mengeklirknya di layout, tampilannya juga akan menunjukkan handle yang mengubah ukuran di setiap sudut.



Pada gambar di atas:

1. **Handle Mengubah Ukuran.**
2. **Garis dan handle pembatas.** Pada gambar, pembatas meratakan sisi kiri tampilan `show_count` ke sisi kiri tombol `button_toast`.
3. **Handle Patokan.** Handle patokan meratakan patokan teks tampilan ke patokan teks tampilan lainnya.
4. **Handle pembatas** tanpa baris pembatas. Layout editor juga menawarkan baris tombol yang memungkinkan Anda mengonfigurasi tampilan layout:



Pada gambar di atas:

1. **Design, Blueprint, dan Both:** Klik ikon **Design** (ikon pertama) untuk menampilkan pratinjau warna layout. Klik ikon **Blueprint** (ikon tengah) untuk menunjukkan hanya outline setiap tampilan. Anda bisa melihat tampilan *keduanya* berdampingan dengan mengeklik ikon ketiga.
2. **Screen orientation:** Klik untuk memutar perangkat antara lanskap dan potret.
3. **Device type and size:** Pilih tipe perangkat (ponsel/tablet, Android TV, atau Android Wear) dan konfigurasi layar (ukuran dan kepadatan).
4. **API version:** Pilih versi Android tempat pratinjau layout.
5. **App theme:** Pilih tema UI yang akan diterapkan pada pratinjau.
6. **Language:** Pilih bahasa untuk menampilkan string UI Anda. Daftar ini hanya menampilkan bahasa yang tersedia dalam sumber daya string.
7. **Layout Variants:** Beralih ke salah satu layout alternatif Anda untuk file ini, atau buat yang baru.

Tips: Untuk mengetahui selengkapnya tentang layout editor, lihat [Membangun UI dengan Layout Editor](#). Untuk mengetahui selengkapnya tentang cara membangun layout dengan `ConstraintLayout`, lihat [Membangun UI Responsif dengan ConstraintLayout](#).

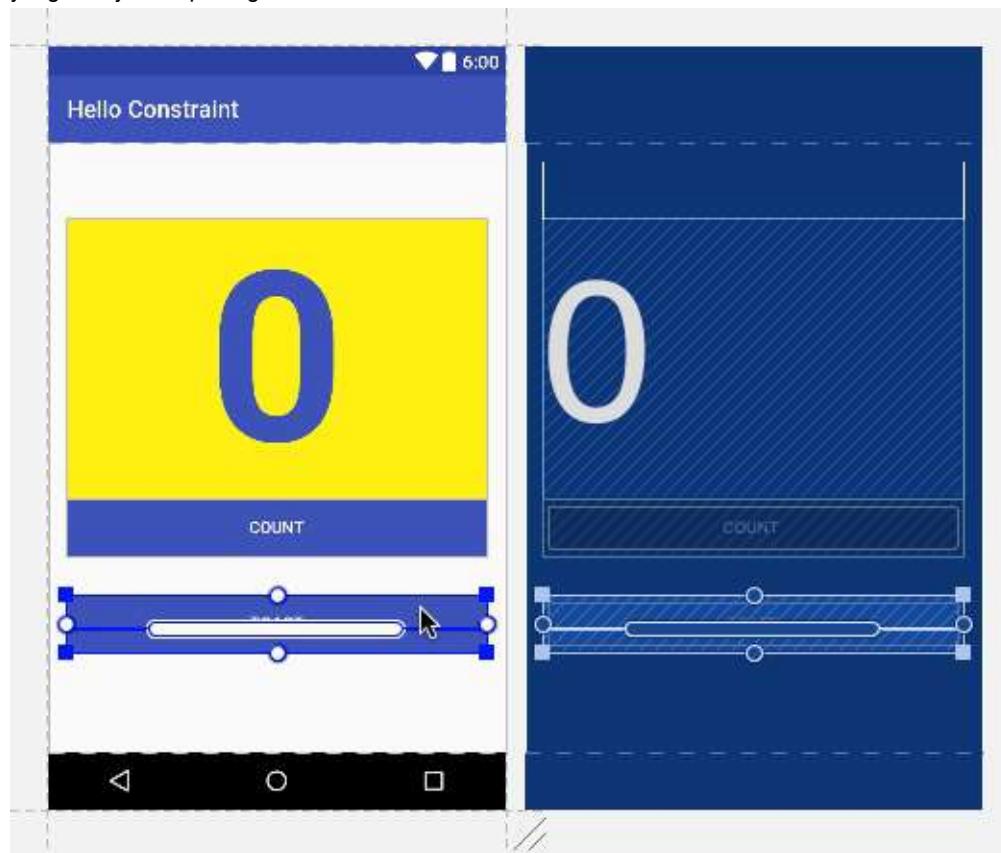
2.5 Menghapus Pembatas

Android Studio secara otomatis menebak pembatas untuk elemen layout saat Anda mengonversi layout untuk menggunakan `constraintLayout`. Tetapi, tebakannya mungkin tidak seperti yang Anda harapkan. Ikuti langkah-langkah berikut untuk menghapus pembatas untuk dengan bebas memposisikan elemen di layout:

1. Klik kanan (or Control-click) **ConstraintLayout** di panel Component Tree dan pilih **Clear All Constraints**.

Tips: Anda juga bisa menghapus satu baris pembatas dengan mengarahkan kursor ke handle pembatas sampai lingkaran merah muncul, lalu klik handlenya. Perintah **Clear All Constraints** lebih cepat dalam menghapus semua pembatas.

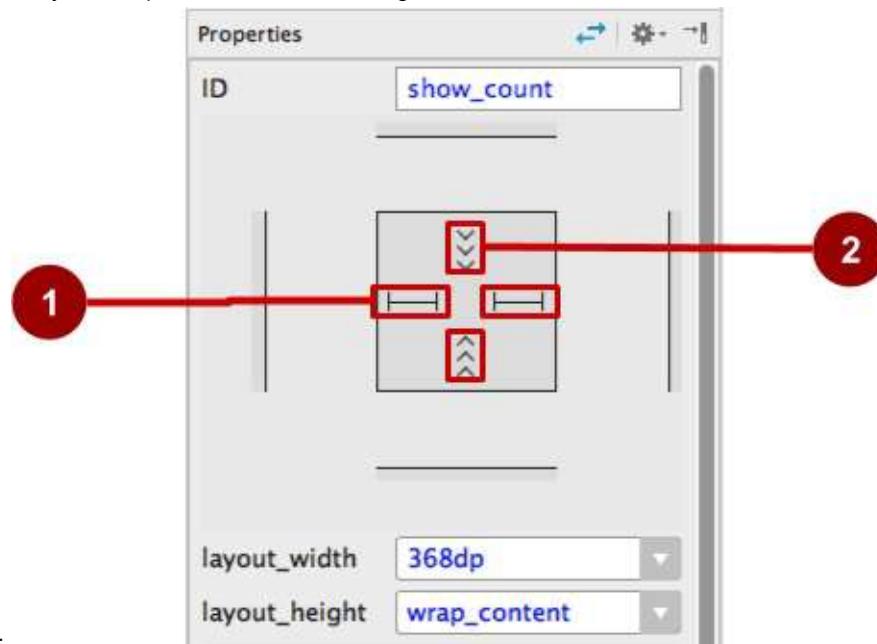
2. Setelah pembatas dihapus, Anda bisa memindahkan tampilan di layout dengan bebas. Seret tampilan `button_toast` ke posisi mana saja di bawah tampilan `button_count`, agar tampilan `show_count` kuning ada di bagian atas, seperti yang ditunjukkan pada gambar di bawah ini.



2.6 Mengubah ukuran tampilan

Layout editor menawarkan handle mengubah ukuran di empat sudut tampilan untuk mengubah ukuran tampilan dengan cepat. Anda bisa menyeret handle di setiap sudut tampilan untuk mengubah ukurannya. Namun, tindakan ini akan melakukan 'hard-code' pada lebar dan tinggi dimensi, yang sebaiknya dihindari untuk kebanyakan tampilan karena dimensi tampilan yang di-hardcode tidak dapat beradaptasi ke konten dan ukuran layar yang berbeda-beda. Sebagai gantinya,

gunakan panel Properties di sisi kanan layout editor untuk memilih mode ukuran yang tidak menggunakan dimensi yang di-hardcode. Panel Properties menyertakan panel ukuran kotak di bagian atas. Simbol di dalam kotak mewakili setelan tinggi



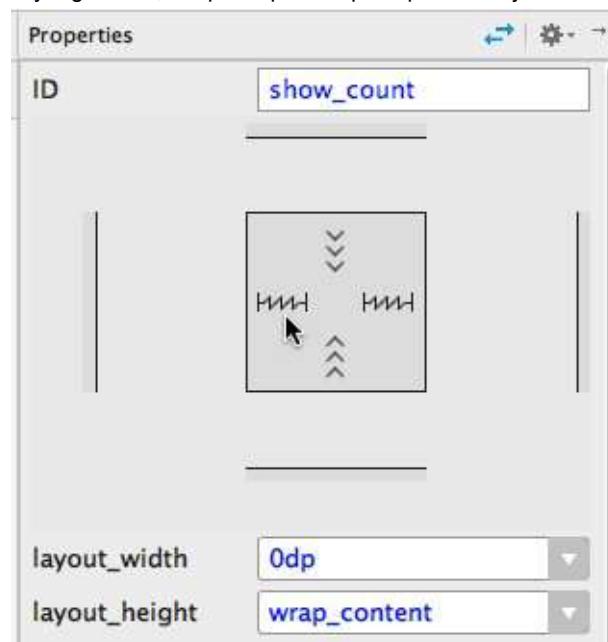
dan lebar seperti berikut ini:

Pada gambar di atas:

- Kontrol ukuran tampilan horizontal.** Kontrol ukuran horizontal, yang muncul di dua segmen di kiri dan kanan kotak, yang menetapkan `layout_width`. Garis lurus menandakan bahwa dimensinya telah ditetapkan dan disetel dalam properti `layout_width` bawah kotak.
- Kontrol ukuran tampilan vertikal.** Kontrol ukuran horizontal, yang muncul di dua segmen di sisi atas dan bawah kotak, menetapkan properti `layout_height`. Sudut menandakan bahwa kontrol ukuran ini disetel ke `wrap_content`, yang berarti tampilan akan diperluas persis seperti yang dibutuhkan agar pas dengan kontennya.

Ikuti langkah-langkah berikut untuk mengubah ukuran tampilan `show_count`:

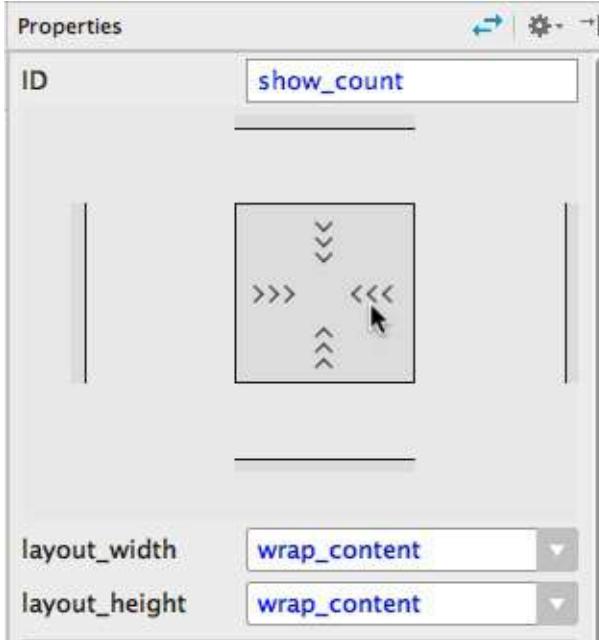
1. Klik tampilan `show_count` di panel Component Tree.
2. Klik kontrol ukuran tampilan horizontal di panel Properties. Garis lurus berubah menjadi lilitan per, seperti yang ditunjukkan pada gambar di bawah ini, yang mewakili "semua ukuran". Properti `layout_width` disetel ke nol karena tidak ada dimensi yang disetel, tetapi tampilan dapat diperluas sejauh mungkin untuk memenuhi pembatas dan



setelan margin.

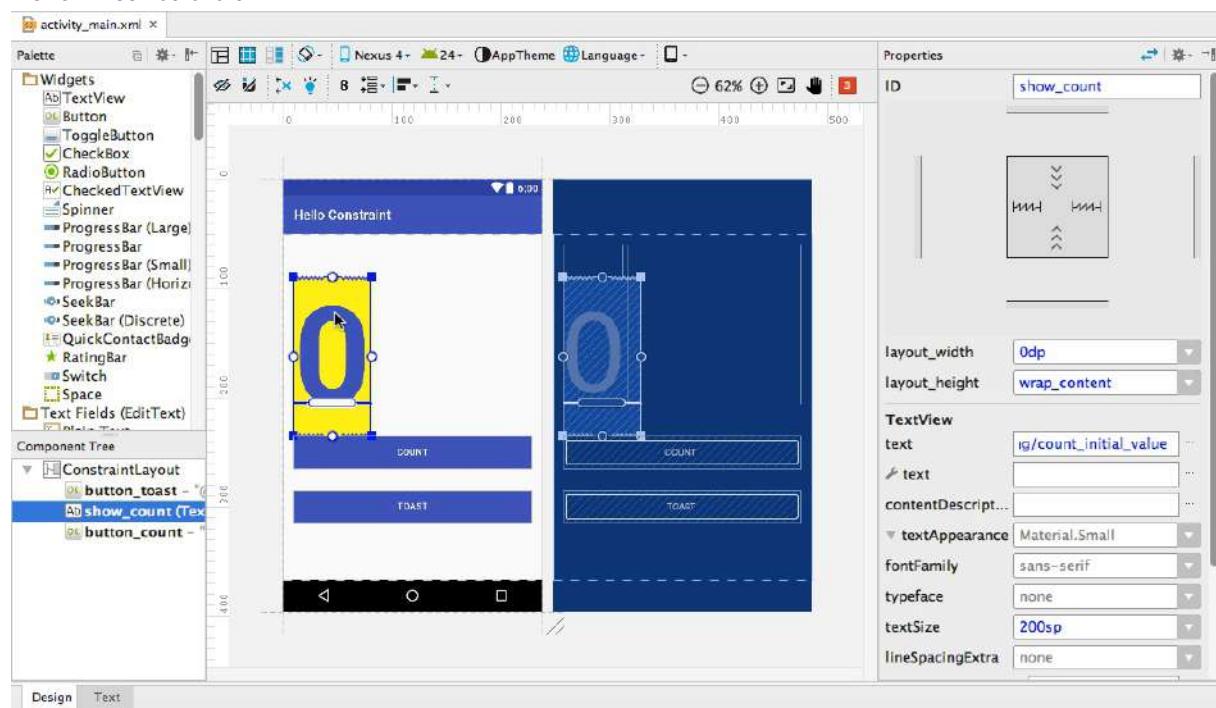
Anda akan menggunakan setelan ini untuk menambatkan ukuran tampilan ke pembatas. Tetapi, lanjutkan bereksperimen dengan setelan dahulu.

- Klik kontrol ukuran tampilan horizontal lagi (sisi kiri atau kanan) untuk melihat pilihan lain apa yang Anda punya. Lilitan per berubah menjadi sudut, seperti yang ditunjukkan pada gambar di bawah ini, yang menandakan bahwa



`layout_width` disetel ke `wrap_content`.

- Klik kontrol ukuran tampilan horizontal lagi, dan itu akan berubah kembali ke garis lurus, menandakan dimensi yang tetap. Klik lagi agar garis lurus berubah menjadi lilitan per, seperti yang ditunjukkan pada gambar di bawah ini, yang mewakili "semua ukuran".

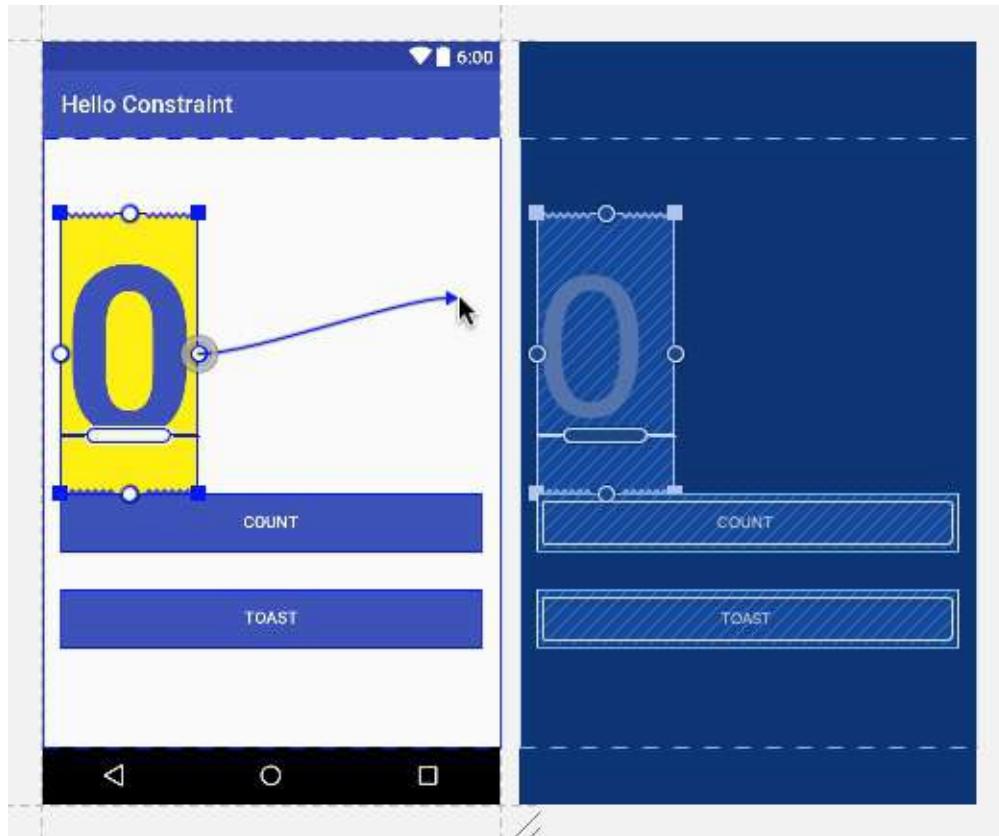


2.7 Menambahkan pembatas ke tampilan

Anda akan menambahkan pembatas ke tampilan `show_count` agar tampilan melebar ke tepi kanan layout, dan pembatas lain agar tampilan diposisikan di bawah tepi atas layout. Karena tampilan disetel ke "semua ukuran" di langkah sebelumnya, tampilan akan melebar sesuai kebutuhan agar cocok dengan pembatas.

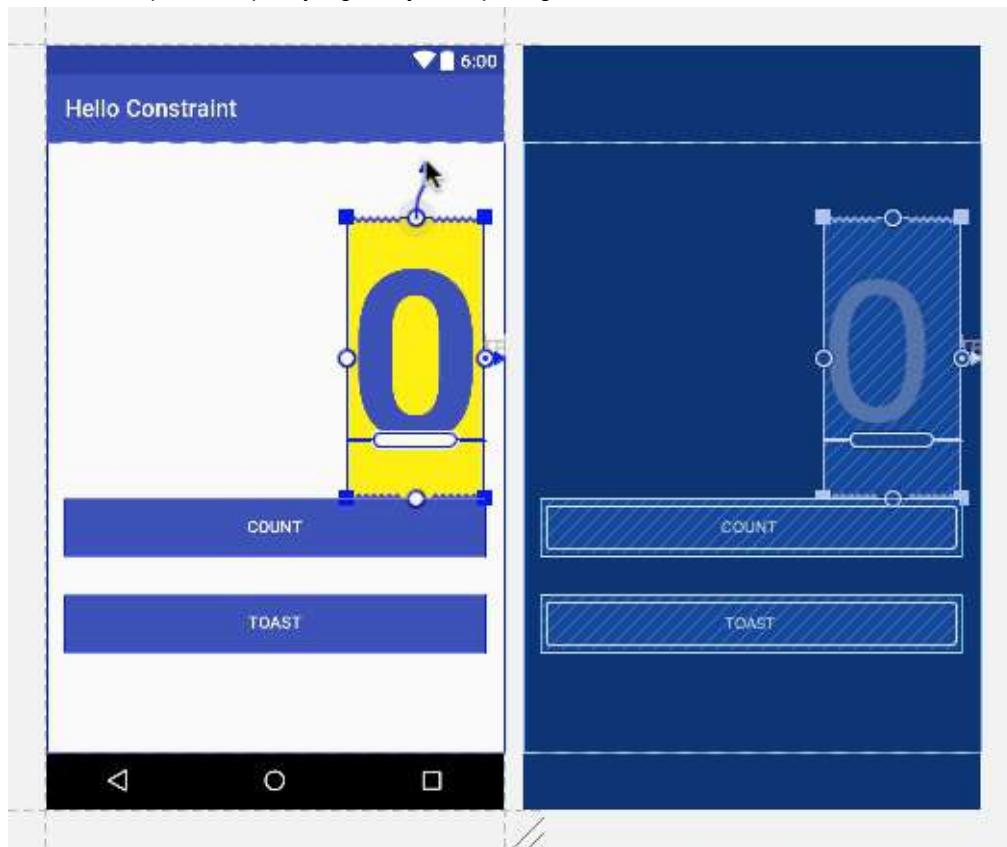
Anda juga akan memindahkan dua tombol ke posisi di sisi kiri tampilan `show_count`, membatasi tombol `button_toast` ke tepi atas dan kiri layout, dan membatasi tombol layout `button_count` agar patokan teks-nya cocok dengan patokan teks tampilan `show_count`.

- Untuk membuat pembatas sisi kanan untuk tampilan `show_count`, klik tampilan di layout, lalu arahkan cursor ke tampilan untuk melihat handle pembatasnya. Klik dan tahan handle pembatas di sisi kanan tampilan, dan seret garis pembatas yang muncul di tepi kanan layout, seperti yang ditunjukkan pada gambar di bawah ini.



Saat Anda melepaskan klik dan tahan, pembatas dibuat, dan tampilan `show_count` melompat ke tepi kiri layout.

2. Klik dan tahan handle pembatas di sisi atas tampilan, dan seret garis pembatas yang muncul di tepi atas layout di bawah bilah aplikasi, seperti yang ditunjukkan pada gambar di bawah ini.

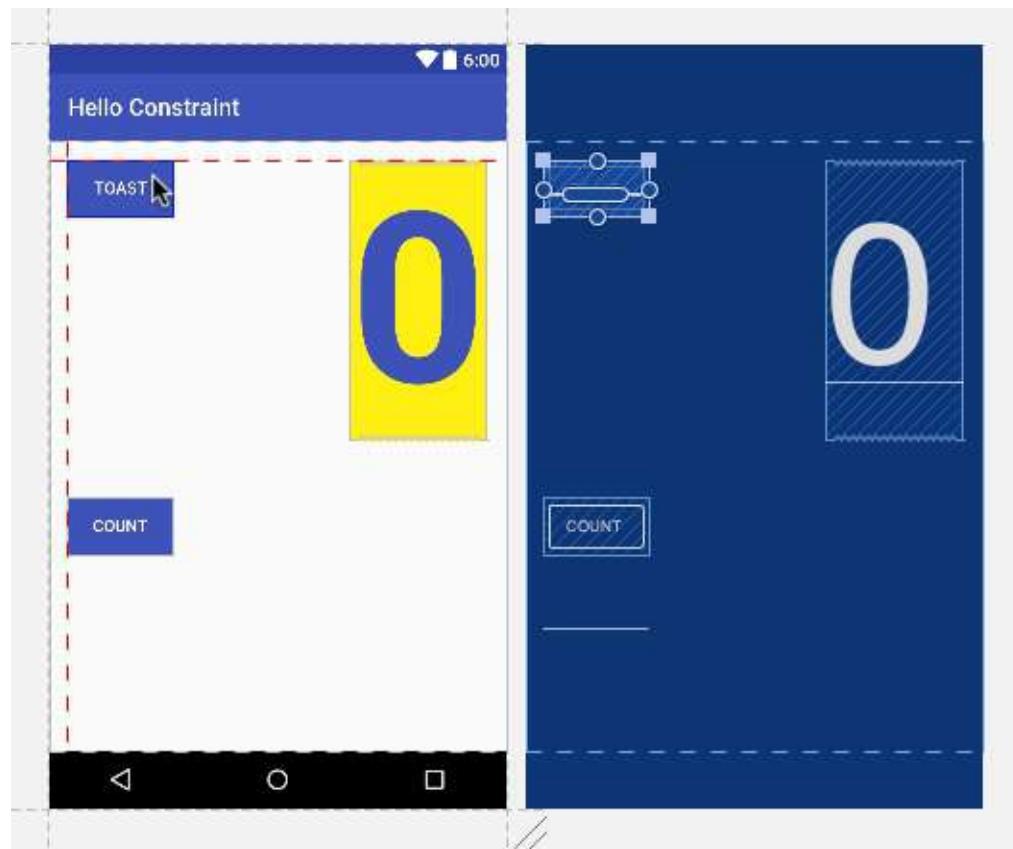


Ini membatasi tampilan ke tepi atas Setelah menyeret pembatas, tampilan `show_count` melompat ke tepi kanan atas layout, karena ditambatkan ke tepi atas dan kanan.

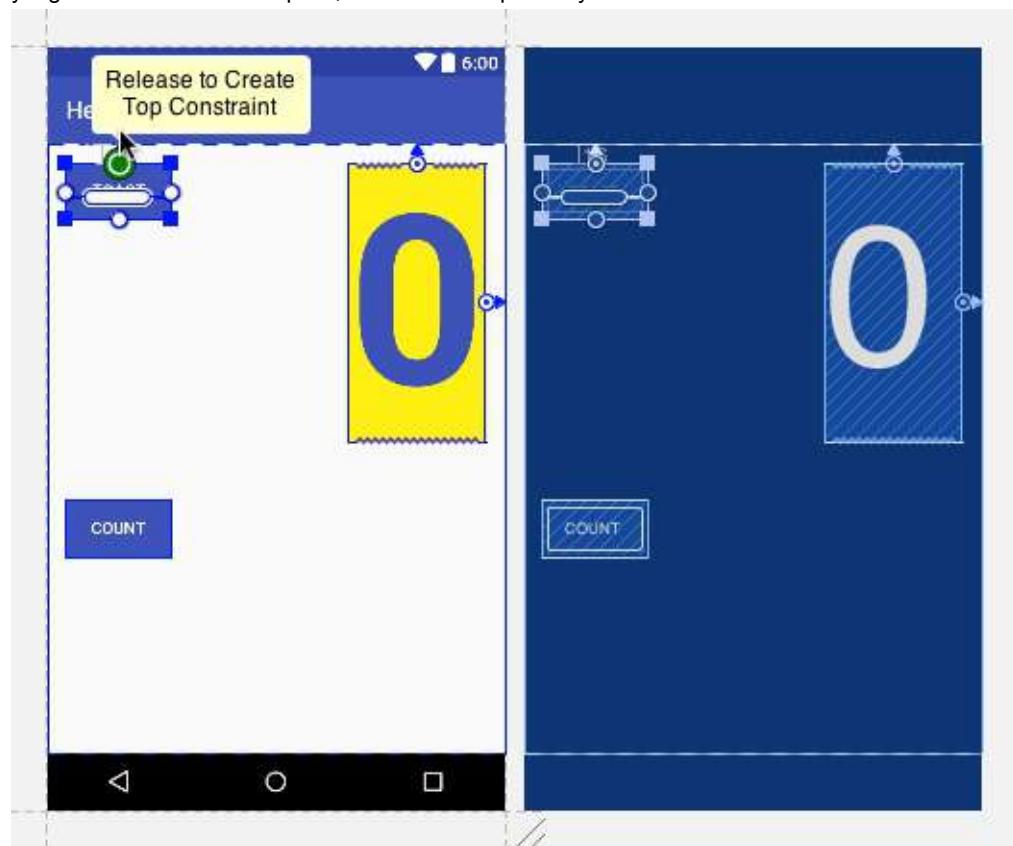
3. Klik tampilan `button_toast`, dan gunakan panel Properties seperti yang ditunjukkan sebelumnya untuk mengubah ukuran tampilan ke `wrap_content` untuk `layout_width` dan `layout_height`. Juga ubah ukuran tampilan `button_count` ke `wrap_content` untuk `layout_width` dan `layout_height`.

Anda menggunakan `wrap_content` untuk tombolnya agar jika teks tombol diterjemahkan ke bahasa lain, tombolnya akan muncul lebih lebar atau tipis untuk mengakomodasi kata dalam bahasa lain tersebut.

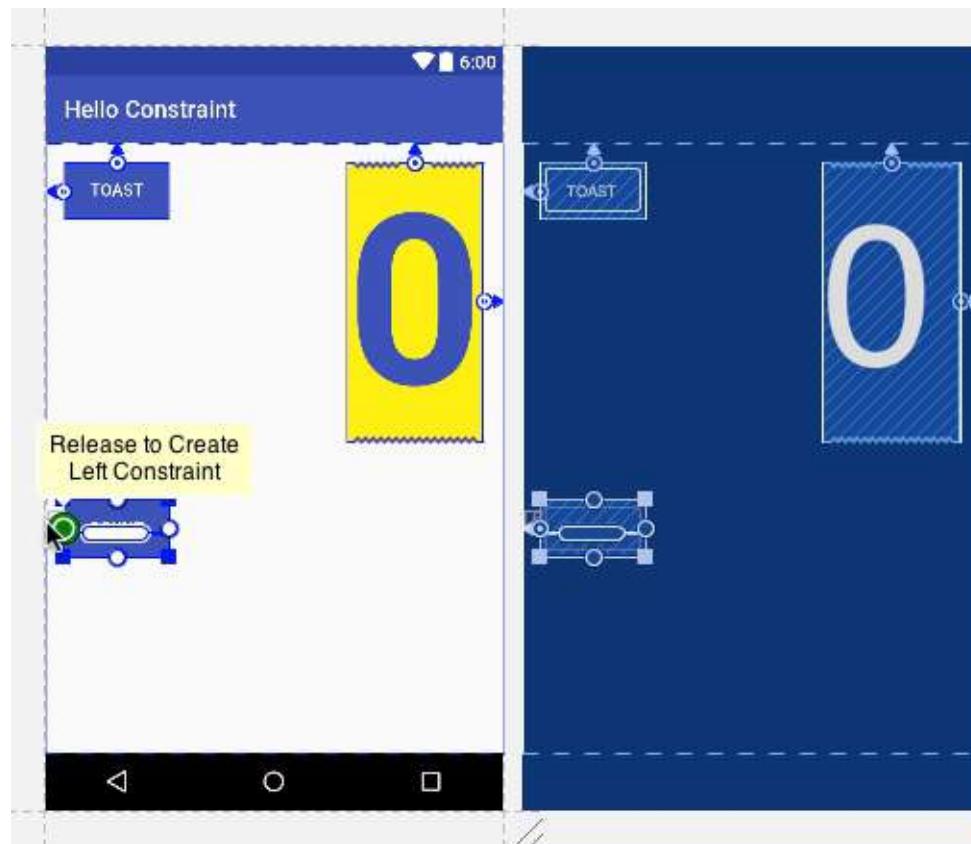
4. Seret tampilan `button_toast` ke posisi di sisi kiri tampilan `show_count` seperti yang ditunjukkan pada gambar di bawah ini. Panduan muncul agar Anda bisa mengepaskan tampilan ke posisi yang tepat terhadap margin atas dan kiri.



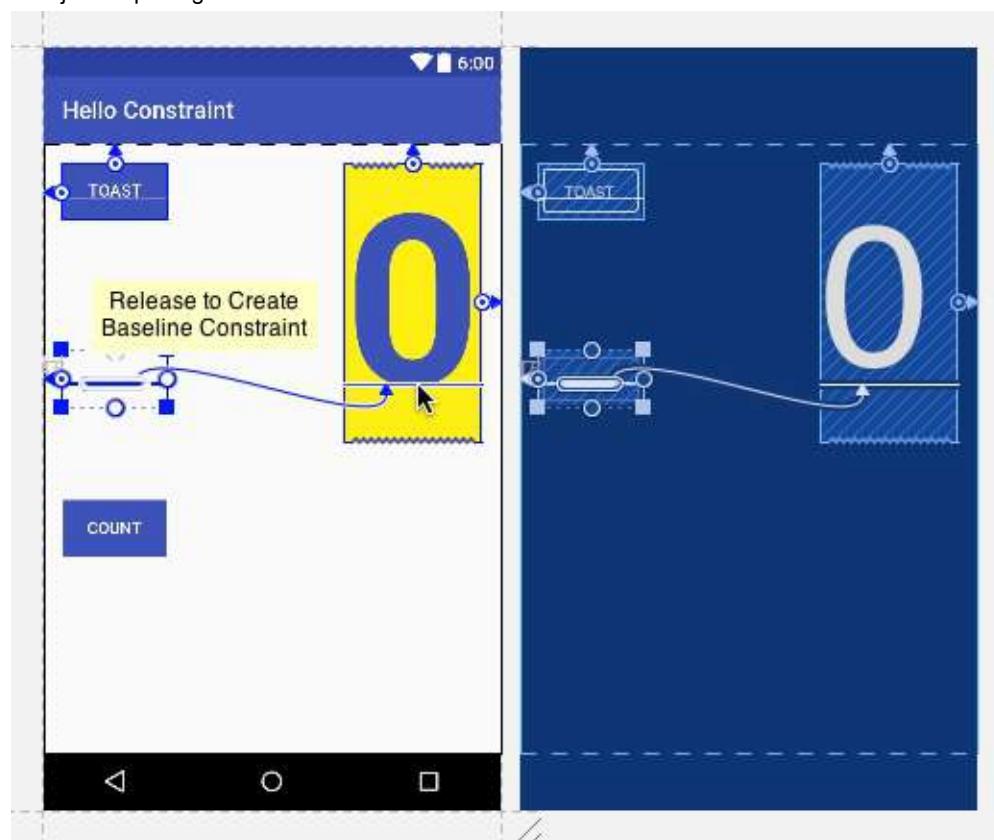
5. Pilih tampilan `button_toast` dalam layout, klik handle pembatas yang muncul di bagian atas tampilan, dan seret ke tepi atas layout di bawah bilah aplikasi seperti yang ditunjukkan pada gambar di bawah ini. Lalu klik handle pembatas yang muncul di sisi kiri tampilan, dan seret ke tepi kiri layout.



6. Klik tampilan `button_count`, klik handle pembatas yang muncul di sisi kiri tampilan, dan seret ke tepi kiri layout



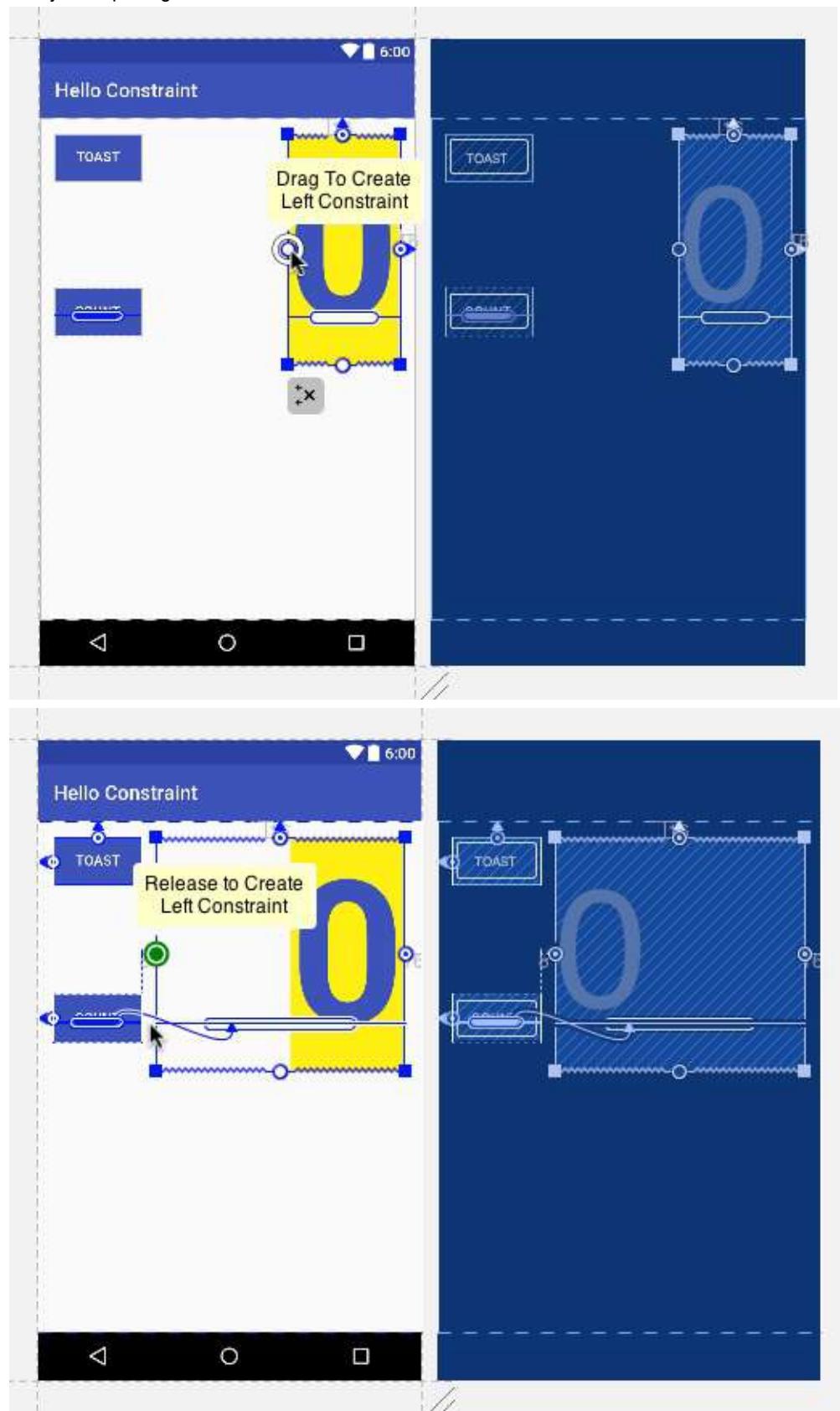
7. Untuk membuat pembatas patokan antara patokan tampilan `button_count` dan patokan teks tampilan `show_count`, pilih tampilan `button_count`, lalu arahkan cursor ke handle patokan tampilan selama dua detik sampai handle berkedip putih. Lalu klik dan seret garis pembatas yang muncul di patokan tampilan `show_count`, seperti yang ditunjukkan pada gambar di bawah ini.



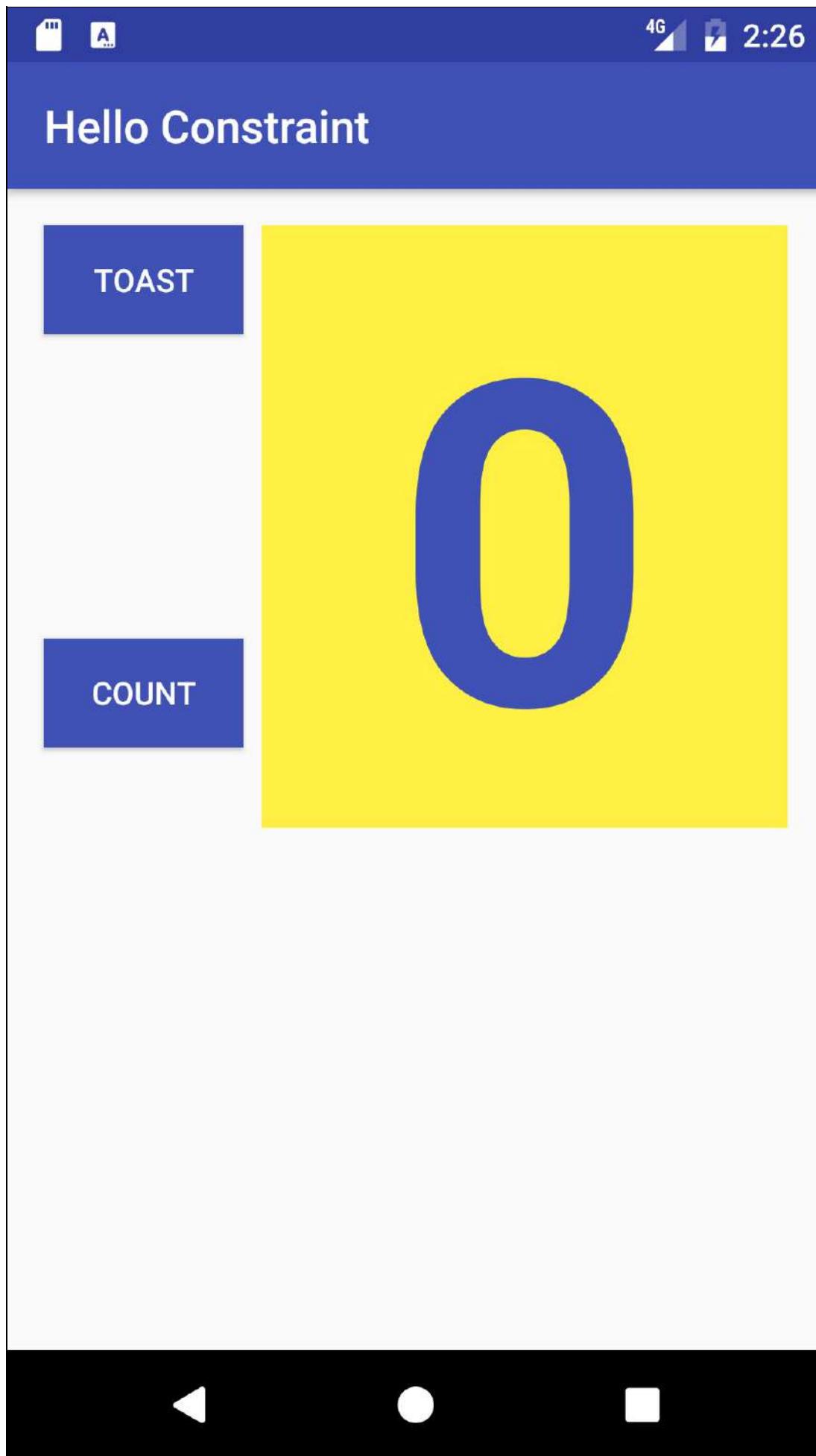
Anda sekarang memiliki layout di mana setiap tampilan disetel ke dimensi non-spesifik dan dibatasi ke layout. Satu teks tombol disejajarkan ke patokan TextView, jadi jika Anda memindahkan TextView, tombolnya juga akan pindah.

Tip: Jika tampilan tidak memiliki setidaknya *dua* pembatas, tampilan akan muncul di bagian atas layout.

8. Walaupun tampilan `show_count` sudah memiliki dua pembatas, Anda bisa menambahkan satu pembatas lagi. Seret garis pembatas dari handle pembatas di sisi kiri tampilan ke sisi kanan tampilan `button_count`, seperti yang ditunjukkan pada gambar di bawah ini.



9. Jalankan aplikasi. Layout mematuhi pembatasnya.

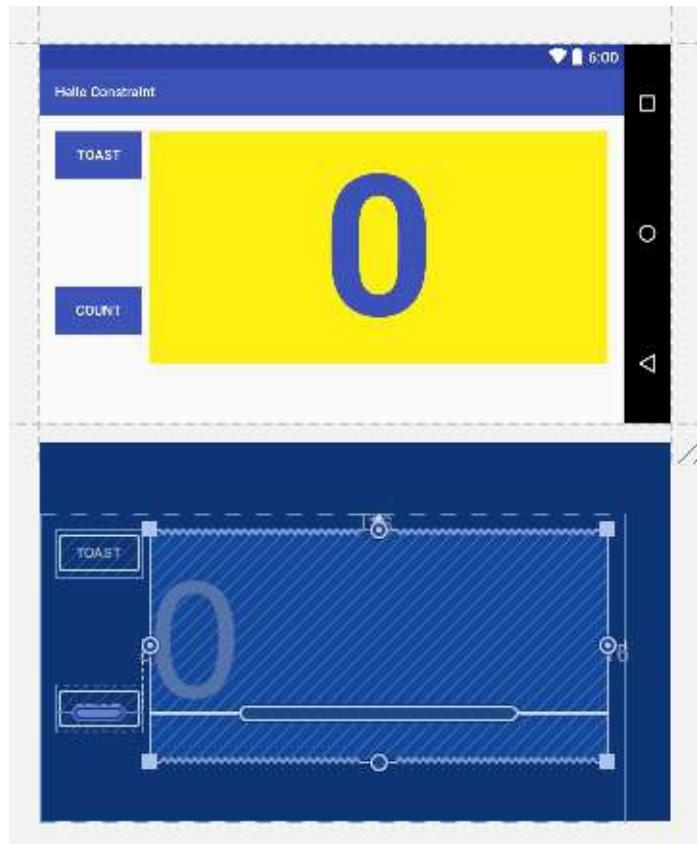


Kode Solusi: Proyek Android Studio: [HelloConstraint](#)

Tugas 3: Membuat varian layout

Anda bisa membuat varian layout untuk orientasi lanskap dan tampilan yang lebih besar. Anda akan membuat versi alternatif layout tugas sebelumnya untuk mengoptimalkan orientasi lanskap:

1. Buka file layout untuk aplikasi HelloConstraint, dan pastikan Anda menampilkan editor Design (Klik tab **Design** di bagian bawah jendela).
2. Klik ikon **Layout Variants** di baris kedua ikon (lihat gambar pada [Tugas 2 Langkah 4](#)) dan pilih **Create Landscape Variation**. Tab "land/activity_main.xml" muncul menampilkan layout untuk orientasi lanskap (horizontal), seperti yang



ditunjukkan pada gambar di bawah ini.

Anda bisa mengubah layout untuk versi lanskap (horizontal) tanpa mengubah orientasi potret aslinya (vertikal), sehingga memanfaatkan layar yang lebih lebar.

3. Pada Proyek: Pada tampilan Android di panel paling kiri Android Studio, lihat di dalam direktori **res > layout**, dan Anda akan melihat bahwa Android Studio secara otomatis membuat varian untuk Anda, yang disebut **activity_main.xml (land)**.
4. Tab "land/activity_main.xml" tab harus tetap terbuka di layout editor; jika tidak, klik dua kali file **activity_main.xml (land)** di direktori **layout**.
5. Klik tab **Text** untuk melihat layout di XML. Temukan pembatas untuk tampilan `button_toast` yang meratakan tepi atasnya dengan tampilan induk:

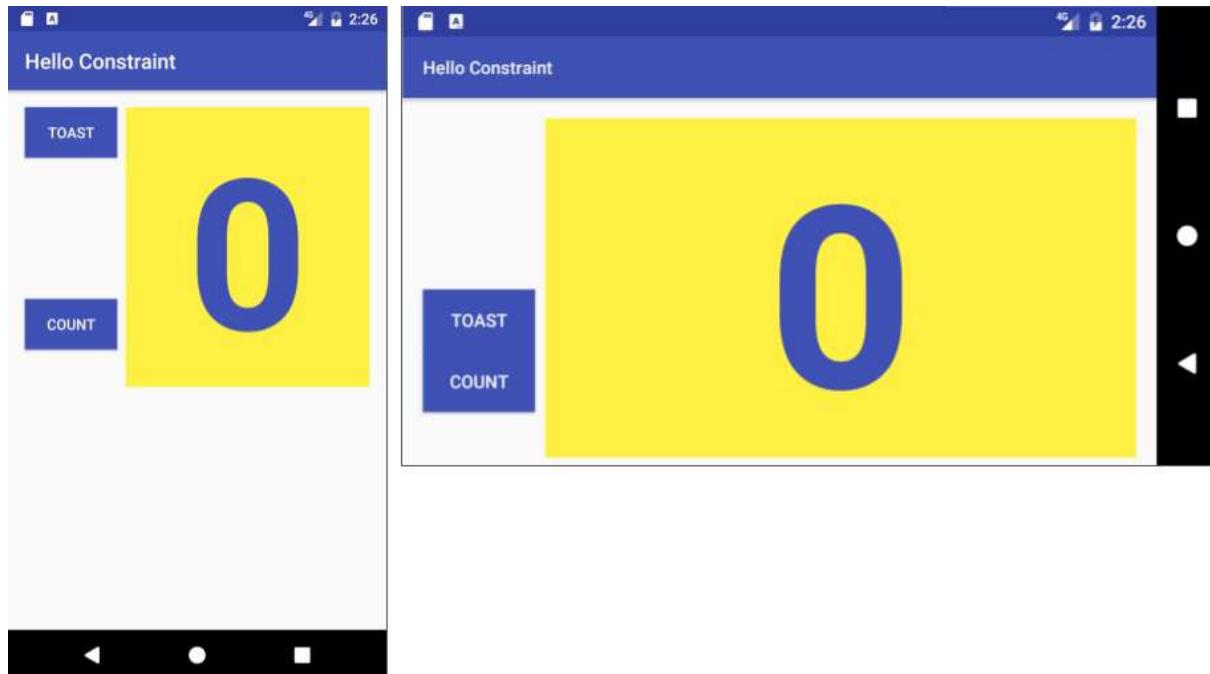
```
<Button
    android:id="@+id/button_toast"
    ...
    app:layout_constraintTop_toTopOf="parent"
    ...
```

6. Ubah pembatas ini agar tampilan `button_toast` diratakan dengan tepi atas tampilan `button_count`.

Petunjuk: Jika pembatas untuk meratakan bagian atas tampilan ke tampilan lain adalah `app:layout_constraintTop_toTopOf`, pembatas apa yang digunakan untuk meratakan bagian bawah tampilan ke bagian atas tampilan lainnya? Jawaban:

```
app:layout_constraintBottom_toTopOf="@+id/button_count"
```

7. Jalankan aplikasi, dan ubah ke mode lanskap untuk melihat layout yang berbeda. Layout seharusnya terlihat seperti di bawah ini.



Kode solusi

Proyek Android Studio: [HelloToast](#)

Android Studio project: [HelloRelative](#)

Proyek Android Studio: [HelloConstraint](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk bab berikutnya.

Tantangan: Tambahkan varian layout lain untuk tampilan yang besar. Varian layout seharusnya memanfaatkan ukuran layar yang lebih besar untuk menunjukkan elemen yang lebih besar.

Petunjuk: Klik ikon **Layout Variants** di toolbar dan pilih **Create layout-xlarge Variation**. Ubah ukuran dan posisi elemen di layout.

Rangkuman

Pada latihan ini Anda belajar:

- Mengatur ulang tampilan pada RelativeLayout menggunakan tab **Design** layout editor.
- Menampilkan desain layout dan blueprint-nya.
- Mengubah properti tampilan (atribut XML) di layout editor.

- Meratakan tampilan dengan RelativeLayout induk menggunakan:
 - `android:layout_alignParentTop` untuk meratakan tampilan ke bagian atas induk.
 - `android:layout_alignParentLeft` untuk meluruskan tampilan ke sisi kiri induk.
 - `android:layout_alignParentStart` untuk membuat tepi mulai tampilan cocok dengan tepi mulai induknya. Atribut ini berguna jika aplikasi Anda dijalankan pada perangkat yang bahasa atau preferensi lokalnya berbeda. Tepi *mulai* adalah tepi kiri layar jika preferensinya kiri ke kanan, atau tepi kanan layar jika preferensinya kanan ke kiri.
- Gunakan `android:layout_below` untuk memosisikan tampilan di bawah tampilan lain pada RelativeLayout.
- Menambahkan ConstraintLayout ke proyek Anda.
- Mengonversikan layout ke ConstraintLayout dengan mengeklik kanan grup tampilan root di panel Component Tree, lalu mengeklik **Convert layout to ConstraintLayout**.
- Menghapus semua pembatas pada ConstraintLayout dengan mengeklik kanan (atau Control-klik) **ConstraintLayout** di panel Component Tree, dan memilih **Clear All Constraints**.
- Menambahkan pembatas ke tampilan di layout ConstraintLayout, dan mengubah ukuran tampilan.
- Ubah tampilan properti, seperti `textAppearance` dan `textSize`.
- Membuat varian layout untuk orientasi lanskap dan ukuran layar yang lebih besar.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Layout, Tampilan, dan Sumber Daya](#)

Ketahui selengkapnya

Dokumentasi Developer

- [Tampilan](#)
- [Layout Relatif](#)
- [Membangun UI dengan Layout Editor](#)
- [Membangun UI Responsif dengan ConstraintLayout](#)

Lainnya:

- Codelabs: [Menggunakan ConstraintLayout untuk mendesain tampilan](#)

1.3: Menggunakan Elemen TextView

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Menambahkan beberapa tampilan teks](#)
- [Tugas 2: Menambahkan tautan web aktif dan ScrollView](#)
- [Tugas 3: Menggulir beberapa elemen](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Kelas [TextView](#) adalah subkelas dari kelas [View](#) yang menampilkan teks di layar. Anda bisa mengontrol bagaimana teks muncul dengan atribut [TextView](#) pada file layout XML. Praktik ini menunjukkan cara menggunakan beberapa elemen [TextView](#), termasuk yang kontennya dapat digulir secara vertikal oleh pengguna.

Jika Anda memiliki lebih banyak informasi dari yang dapat ditampilkan di layar perangkat, Anda dapat membuat *tampilan bergulir* agar pengguna dapat menggulir secara vertikal dengan mengusap ke atas atau ke bawah, atau secara horizontal dengan mengusap ke kanan atau ke kiri.

Anda biasanya akan menggunakan tampilan bergulir untuk berita, artikel, atau teks panjang apa pun yang tidak bisa benar-benar pas dengan layar perangkat. Anda juga bisa menggunakan tampilan bergulir untuk memungkinkan pengguna memasukkan beberapa baris teks, atau menggabungkan elemen UI (seperti bidang teks dan tombol) di dalam tampilan bergulir.

Kelas [ScrollView](#) menyediakan layout untuk tampilan bergulir. [ScrollView](#) adalah subkelas dari [FrameLayout](#), dan developer hanya boleh meletakkan *satu* tampilan sebagai tampilan anak di dalamnya, dan tampilan anak tersebut berisi seluruh konten yang dapat digulir. Tampilan anak ini dapat berupa grup tampilan (misalnya layout manager seperti [LinearLayout](#)) dengan hierarki objek yang kompleks. Perhatikan bahwa layout yang kompleks dapat memiliki masalah kinerja dengan tampilan anak seperti gambar. Pilihan yang bagus untuk tampilan di dalam [ScrollView](#) adalah [LinearLayout](#) yang diatur dalam orientasi vertikal, menampilkan item level atas yang dapat digulir pengguna.

Dengan [ScrollView](#), semua tampilan ada dalam memori dan hierarki tampilan bahkan jika tidak ditampilkan di layar. Ini menjadikan [ScrollView](#) ideal untuk menggulir halaman teks bentuk bebas dengan mudah, karena teks sudah ada dalam memori. Namun, [ScrollView](#) dapat menggunakan banyak memori, yang dapat memengaruhi kinerja aplikasi Anda. Untuk menampilkan daftar panjang item yang dapat ditambahkan ke, dihapus dari, atau diedit pengguna, pertimbangkan untuk menggunakan [RecyclerView](#), yang dijelaskan dalam praktik yang terpisah.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Membuat aplikasi Hello World dengan Android Studio.
- Menjalankan aplikasi di emulator atau perangkat.
- Mengimplementasikan [TextView](#) dalam layout untuk aplikasi.
- Membuat dan menggunakan sumber daya string.
- Mengonversi dimensi layout ke sumber daya.

Yang akan Anda PELAJARI

Anda akan belajar:

- Menggunakan kode XML untuk menambahkan beberapa elemen TextView.
- Menggunakan kode XML untuk menentukan tampilan bergulir.
- Menampilkan teks bentuk bebas dengan beberapa tag pemformatan HTML.
- Menata gaya warna latar belakang dan warna teks TextView.
- Menyertakan tautan web di teks.

Yang akan Anda LAKUKAN

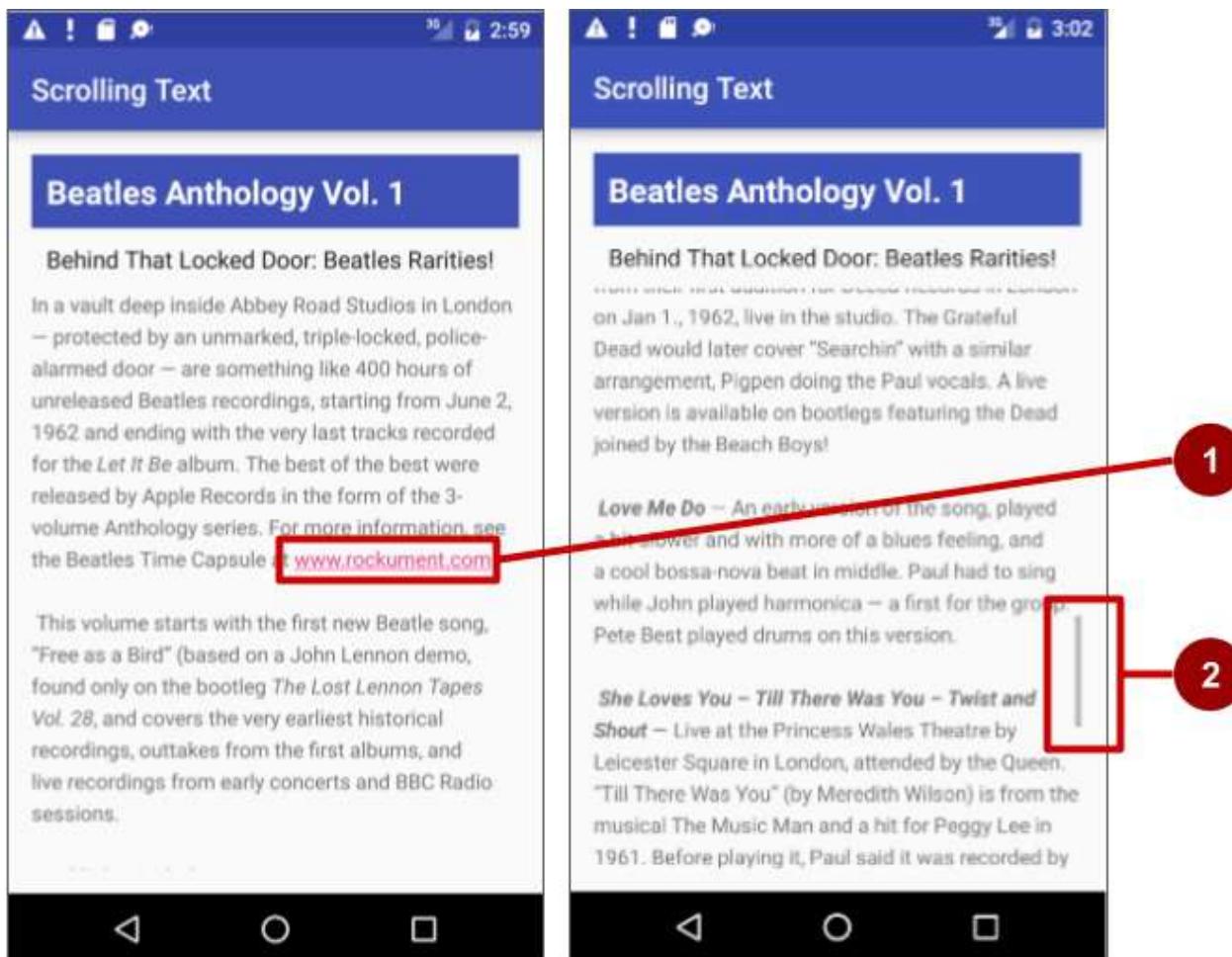
Dalam praktik ini Anda akan:

- Membuat aplikasi Scrolling Text.
- Menambahkan dua elemen TextView untuk heading dan subheading artikel.
- Menggunakan gaya dan warna TextAppearance untuk heading dan subheading artikel.
- Menggunakan tag HTML dalam string teks untuk mengontrol pemformatan.
- Menggunakan atribut lineSpacingExtra untuk menambahkan spasi baris agar meningkatkan keterbacaan.
- Menambahkan ScrollView ke layout untuk mengaktifkan pengguliran elemen TextView.
- Menambahkan atribut autoLink untuk mengaktifkan URL di teks agar aktif dan bisa diklik.

Ringkasan Aplikasi

Aplikasi Scrolling Text memperagakan komponen UI ScrollView. ScrollView adalah ViewGroup yang dalam contoh ini berisi TextView. ScrollView menunjukkan halaman teks panjang, dalam hal ini ulasan album musik, yang dapat digulir pengguna secara vertikal untuk dibaca dengan mengusap layar ke atas dan ke bawah. Bilah gulir muncul di margin kanan. Aplikasi ini menunjukkan cara menggunakan teks berformat dengan tag HTML minimal untuk mengubah teks menjadi cetak tebal atau

miring, dan dengan karakter baris baru untuk memisahkan paragraf. Anda juga bisa menyertakan tautan web aktif dalam teks.

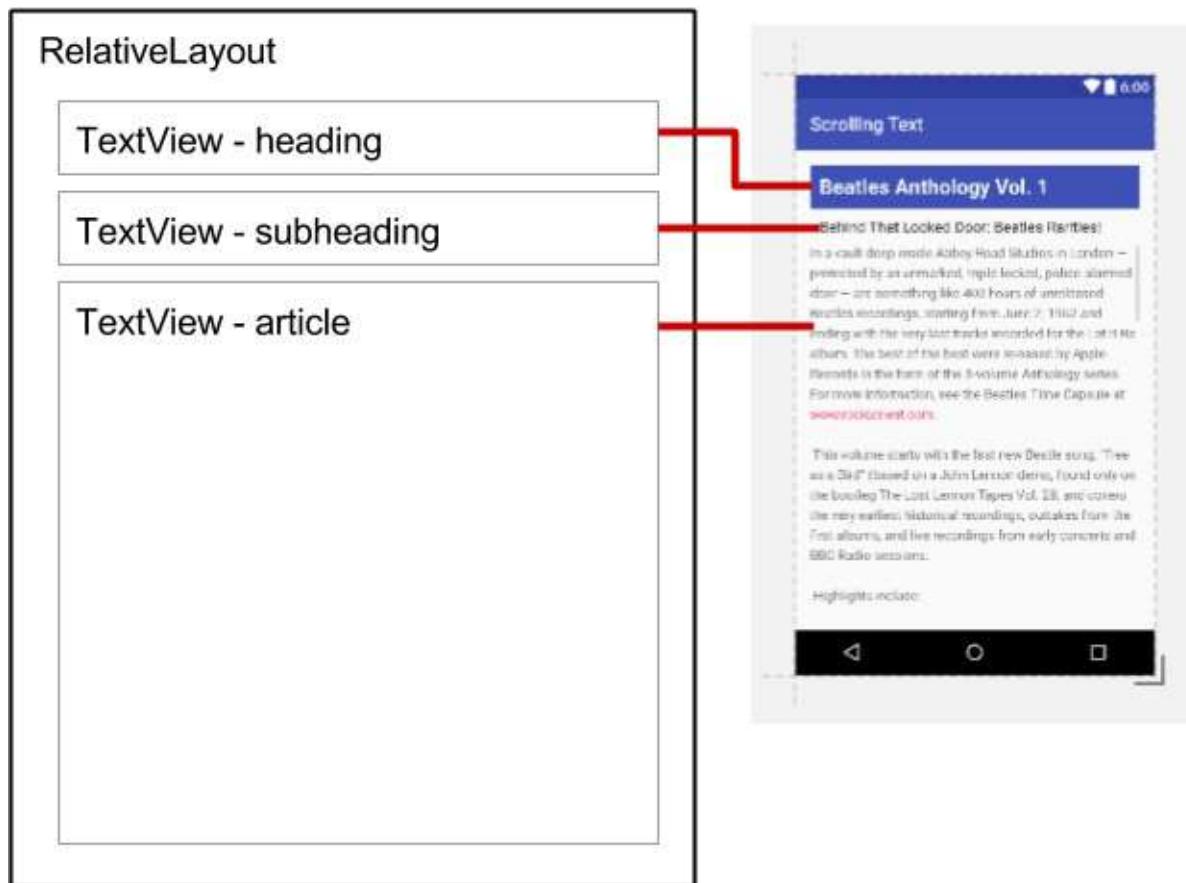


Pada gambar di atas, hal berikut muncul:

1. Tautan web aktif dalam teks bentuk bebas
2. Bilah gulir yang muncul saat menggulir teks.

Tugas 1: Menambahkan beberapa tampilan teks

Dalam praktik ini Anda akan membuat proyek Android untuk aplikasi Scrolling Text, menambahkan TextView ke layout untuk judul dan subjudul artikel, dan mengubah elemen TextView "Hello World" yang sudah ada untuk menunjukkan artikel yang panjang. Gambar di bawah ini adalah diagram layout.



Anda akan membuat semua perubahan ini dalam kode XML dan di file strings.xml. Anda akan mengedit kode XML untuk layout di panel Text, yang ditunjukkan dengan mengeklik tab **Text** tab, bukan dengan mengeklik tab **Design** untuk panel Design. Beberapa perubahan pada elemen dan atribut UI lebih mudah dibuat langsung pada panel Text menggunakan kode sumber XML.

1.1 Membuat proyek dan elemen TextView

1. Di Android Studio, buat proyek baru dengan parameter berikut:

| Atribut | Nilai |
|-------------------------------|------------------------------------------------|
| Application Name | Scrolling Text |
| Company Name | android.example.com (atau domain Anda sendiri) |
| Phone and Tablet Minimum SDK | API15: Android 4.0.3 IceCreamSandwich |
| Template | Empty Activity |
| Generate Layout File checkbox | Dicentang |

2. Pada folder app > res > layout, buka file **activity_main.xml**, dan klik tab **Text** untuk melihat kode XML jika belum dipilih.

Di bagian atas, atau *root*, hierarki tampilan adalah ViewGroup yang disebut **RelativeLayout**. Seperti ViewGroup lainnya, RelativeLayout adalah tampilan yang berisi tampilan lainnya. Selain itu, tampilan ini juga memungkinkan Anda untuk memposisikan Tampilan anak relatif terhadap satu sama lain atau relatif terhadap RelativeLayout induk itu sendiri. Elemen TextView "Hello World" default yang dibuat oleh template Empty Layout adalah Tampilan anak dalam grup tampilan RelativeLayout. Untuk informasi selengkapnya tentang cara menggunakan RelativeLayout, lihat [Panduan API RelativeLayout](#).

3. Tambahkan elemen `TextView` di atas `TextView` "Hello World". Saat Anda memasukkan `<TextView` to start a `TextView`, Android Studio automatically adds the ending `>`, which is shorthand for `</TextView>`. Tambahkan atribut-atribut berikut ke `TextView`:

| Atribut #1 TextView | Nilai |
|-------------------------------------|----------------------------------------------------|
| <code>android:id</code> | <code>"@+id/article_heading"</code> |
| <code>layout_width</code> | <code>"match_parent"</code> |
| <code>layout_height</code> | <code>"wrap_content"</code> |
| <code>android:background</code> | <code>"@color/colorPrimary"</code> |
| <code>android:textColor</code> | <code>"@android:color/white"</code> |
| <code>android:padding</code> | <code>"10dp"</code> |
| <code>android:textAppearance</code> | <code>"@android:style/TextAppearance.Large"</code> |
| <code>android:textStyle</code> | <code>"bold"</code> |
| <code>android:text</code> | <code>"Article Title"</code> |

Tip: Atribut untuk menata gaya teks dan latar belakang dirangkum dalam [Dokumentasi kelas TextView](#).

4. Ekstrak sumber daya string untuk string yang di-hardcode `"Article Title"` atribut `android:text` dalam `TextView` untuk membuat entri untuknya di `strings.xml`

Letakkan kursor di string yang di-hardcode, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract string resource**. Lalu edit nama sumber daya untuk nilai string ke `article_title`.

Tip: Sumber daya string dijelaskan secara detail di [dokumentasi Sumber Daya String](#).

5. Ekstrak sumber daya dimensi untuk string `"10dp"` hard-code atribut `android:padding` dalam `TextView` untuk membuat entri di `dimens.xml`.

Letakkan kursor di string hard-code, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract dimension resource**. Lalu edit name Sumber Daya ke `padding_regular`.

6. Tambahkan elemen `TextView` `TextView` di atas "Hello World" `TextView` bawah `TextView` yang Anda buat pada langkah sebelumnya. Tambahkan atribut-atribut berikut ke `TextView`:

| Atribut #2 TextView | Nilai |
|-------------------------------------|----------------------------------------------|
| <code>android:id</code> | <code>"@+id/article_subheading"</code> |
| <code>layout_width</code> | <code>"match_parent"</code> |
| <code>layout_height</code> | <code>"wrap_content"</code> |
| <code>android:layout_below</code> | <code>"@+id/article_heading"</code> |
| <code>android:padding</code> | <code>"@dimen/padding_regular"</code> |
| <code>android:textAppearance</code> | <code>"@android:style/TextAppearance"</code> |
| <code>android:text</code> | <code>"Article Subtitle"</code> |

Perhatikan karena Anda mengekstrak sumber daya dimensi untuk string `"10dp"` ke `padding_regular` pada `TextView` yang sebelumnya dibuat, Anda bisa menggunakan `"@dimen/padding_regular"` untuk atribut `android:padding` dalam `TextView` ini.

7. Ekstrak sumber daya string untuk string `"Article Subtitle"` yang di-hardcode atribut `android:text` dalam `TextView` ke `article_subtitle`.
8. Tambahkan atribut-atribut `TextView` berikut ke elemen `TextView` "Hello World", dan ubah atribut `android:text` :

| Atribut TextView | Nilai |
|--------------------------|---------------------------|
| android:id | "@+id/article" |
| android:lineSpacingExtra | "5sp" |
| android:layout_below | "@+id/article_subheading" |
| android:text | Ubah ke "Article text" |

9. Ekstrak sumber daya string untuk "Article text" ke `article_text`, dan ekstrak sumber dimensi "5sp" ke `line_spacing`.
10. Format ulang dan ratakan kode dengan memilih **Code > Reformat Code**. Sebaiknya format ulang dan ratakan kode agar memudahkan Anda dan orang lain memahaminya.

1.2 Menambahkan teks artikel

Pada aplikasi sebenarnya yang mengakses majalah atau artikel koran, artikel yang muncul mungkin berasal dari sumber online melalui penyedia konten, atau mungkin disimpan terlebih dulu dalam database di perangkat.

Untuk praktik ini, Anda akan membuat artikel dengan format satu string panjang pada sumber daya strings.xml.

1. Dalam folder `app > res > values`, buka `strings.xml`.
2. Masukkan nilai string `article_title` dan `article_subtitle` dengan judul karangan dan subjudul untuk artikel yang Anda tambahkan. Nilai masing-masing string harus berupa teks satu baris tanpa tag HTML atau beberapa baris.
3. Masukkan atau salin dan tempel teks untuk string `article_text`.

Gunakan teks yang disediakan untuk string `article_text` dalam file `strings.xml` dari [aplikasi ScrollingText] yang telah diselesaikan(<https://github.com/google-developer-training/android-fundamentals/tree/master/ScrollingText>), atau gunakan teks generik Anda sendiri. Anda bisa menempel dan menyalin kalimat yang sama berulang kali, selama hasilnya adalah bagian panjang teks yang tidak akan pas di layar. Perhatikan yang berikut ini (lihat gambar di bawah sebagai contoh):

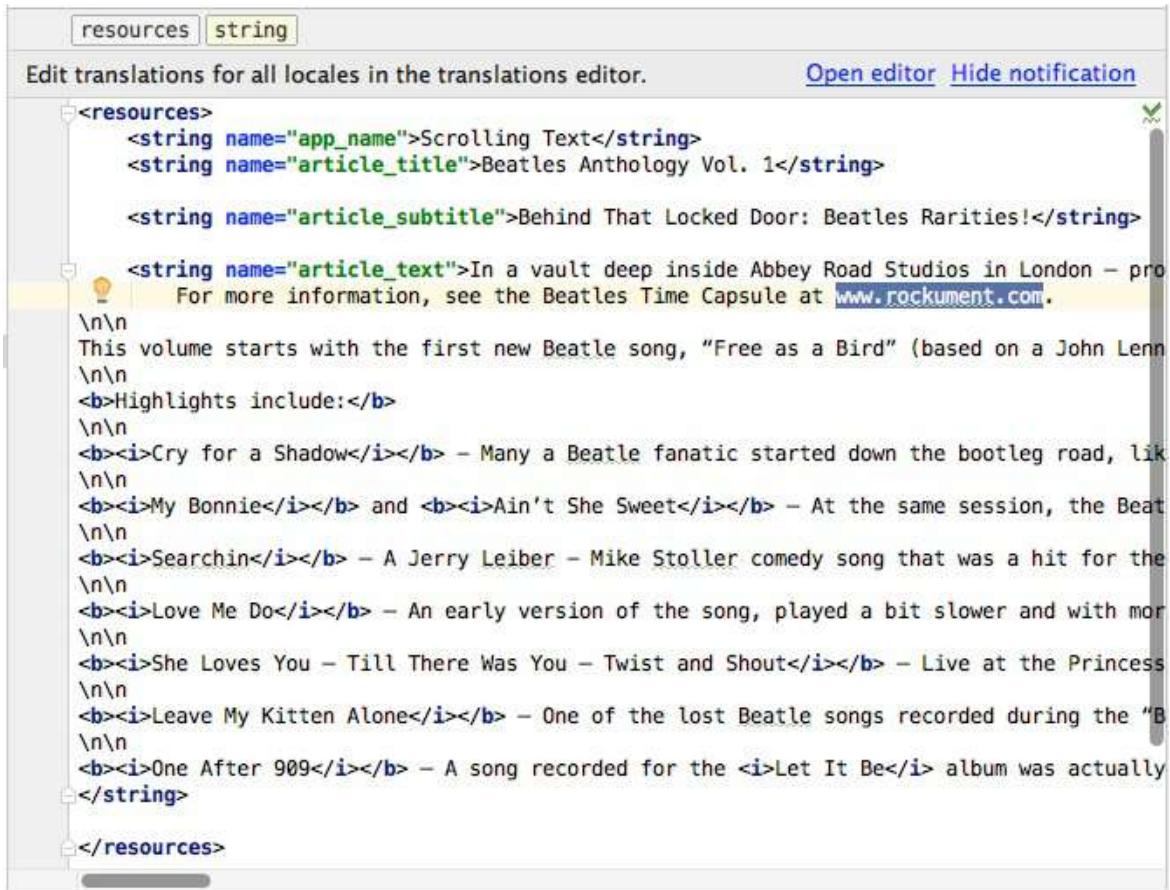
- i. Saat Anda memasukkan atau menempel teks di file `strings.xml`, baris teks tidak membungkus ke baris berikutnya, melainkan meluas melebihi margin kanan. Ini adalah perilaku yang benar, setiap baris teks baru yang dimulai di margin kiri mewakili seluruh paragraf.
- ii. Masukkan `\n` untuk mewakili akhir sebuah baris, dan `\n\n` untuk mewakili baris kosong.

*Mengapa?** Anda perlu menambahkan karakter akhir-baris untuk menjaga paragraf agar tidak bertabrakan satu dan lainnya.

Tips: Jika Anda ingin melihat teks dibungkus dalam strings.xml, Anda bisa menekan Return untuk memasukkan akhiran baris keras, atau memformat teksnya dulu di editor teks dengan akhiran baris keras.

- iii. Jika ada apostrof ('') dalam teks, Anda harus melakukan escape dengan mengawalinya dengan backslash (\'). Jika Anda memiliki tanda kutip ganda dalam teks, Anda harus melakukan escape (\""). Anda juga harus melakukan escape pada karakter non-ASCII lainnya. Lihat bagian "[Pemformatan dan Penataan gaya]" (<https://developer.android.com/guide/topics/resources/string-resource.html#FormattingAndStyling>) dari Sumber Daya String untuk detail selengkapnya.
- iv. Masukkan tag HTML dan `` di sekeliling kata yang harus dicetak tebal.
- v. Masukkan HTML ` dan ` di sekeliling kata yang harus dicetak miring. Namun perhatikan bahwa jika Anda menggunakan apostrof dengan frasa yang dimiringkan, Anda harus menggantinya dengan apostrof lurus.
- vi. Jika Anda bisa menggabungkan cetak tebal dan miring dengan menggabungkan tag-nya, seperti dalam ... kata-kata...`<i>`. Tag HTML lain diabaikan.
- vii. Kurung seluruh teks di dalam `<string name="article_text"> </string>` pada file strings.xml.

- viii. Sertakan tautan web untuk mengujinya, misalnya www.google.com (contoh bawah menggunakan www.rockument.com). *Jangan* gunakan tag HTML kecuali tag cetak tebal dan miring, karena akan diabaikan dan ditampilkan sebagai teks. Anda tentu tidak menginginkan hal ini.



The screenshot shows the 'string' tab in the 'resources' section of the Translations Editor. The code displays a string resource named 'article_text' containing a block of text about Beatles rarities. A blue link 'www.rockument.com' is present in the text, which is highlighted with a yellow background, indicating it is a live URL.

```

<resources>
    <string name="app_name">Scrolling Text</string>
    <string name="article_title">Beatles Anthology Vol. 1</string>

    <string name="article_subtitle">Behind That Locked Door: Beatles Rarities!</string>

    <string name="article_text">In a vault deep inside Abbey Road Studios in London – pro
        For more information, see the Beatles Time Capsule at www.rockument.com.
    \n\n
    This volume starts with the first new Beatle song, "Free as a Bird" (based on a John Lenn
    \n\n
    <b>Highlights include:</b>
    \n\n
    <b><i>Cry for a Shadow</i></b> – Many a Beatle fanatic started down the bootleg road, lik
    \n\n
    <b><i>My Bonnie</i></b> and <b><i>Ain't She Sweet</i></b> – At the same session, the Beat
    \n\n
    <b><i>Searchin</i></b> – A Jerry Leiber – Mike Stoller comedy song that was a hit for the
    \n\n
    <b><i>Love Me Do</i></b> – An early version of the song, played a bit slower and with mor
    \n\n
    <b><i>She Loves You – Till There Was You – Twist and Shout</i></b> – Live at the Princess
    \n\n
    <b><i>Leave My Kitten Alone</i></b> – One of the lost Beatle songs recorded during the "B
    \n\n
    <b><i>One After 909</i></b> – A song recorded for the <i>Let It Be</i> album was actually
</string>

</resources>

```

4. Jalankan aplikasi.

Artikel muncul, dan Anda bahkan bisa meng gulirnya, namun peng gulirannya tidak lancar dan tidak ada bilah gulir karena Anda belum memasukkan ScrollView (yang akan Anda lakukan di tugas berikutnya). Harap diingat bahwa mengetuk tautan web baru saat ini belum memberikan efek apa pun. Anda juga akan memperbaiknya di tugas berikutnya.

Kode Solusi

File layout activity_main.xml sekarang harus terlihat seperti ini:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.scrollingtext.MainActivity">

    <TextView
        android:id="@+id/article_heading"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:textColor="@android:color/holo_orange_light"
        android:textColorHighlight="@color/colorAccent"
        android:padding="10dp"
        android:textAppearance="@android:style/TextAppearance.Large"
        android:textStyle="bold"
        android:text="@string/article_title"/>

    <TextView
        android:id="@+id/article_subheading"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/article_heading"
        android:padding="10dp"
        android:textAppearance="@android:style/TextAppearance"
        android:text="@string/article_subtitle"/>

    <TextView
        android:id="@+id/article"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/article_subheading"
        android:lineSpacingExtra="5sp"
        android:text="@string/article_text"/>

</RelativeLayout>

```

Tugas 2: Menambahkan tautan Web aktif dan ScrollView

Pada tugas sebelumnya Anda membuat aplikasi Scrolling Text dengan TextViews untuk judul artikel, subjudul, dan teks artikel panjang. Anda juga menyertakan tautan web, namun tautannya belum aktif. Anda akan menambahkan kode untuk mengaktifkannya.

Selain itu, TextView sendiri tidak memungkinkan pengguna untuk menggulir teks artikel untuk melihat semuanya. Anda akan menambahkan grup tampilan baru `ScrollView` ke layout XML yang akan membuat TextView dapat digulir.

2.1 Menambahkan atribut autoLink untuk tautan web aktif

Tambahkan atribut `android:autoLink="web"` ke TextView `article`. Kode XML untuk TextView ini sekarang harus terlihat seperti berikut:

```

<TextView
    android:id="@+id/article"
    ...
    android:autoLink="web"
    ... />

```

2.2 Menambahkan ScrollView ke layout

Untuk membuat tampilan (seperti TextView) dapat digulir, sematkan tampilan tersebut *di dalam* ScrollView.

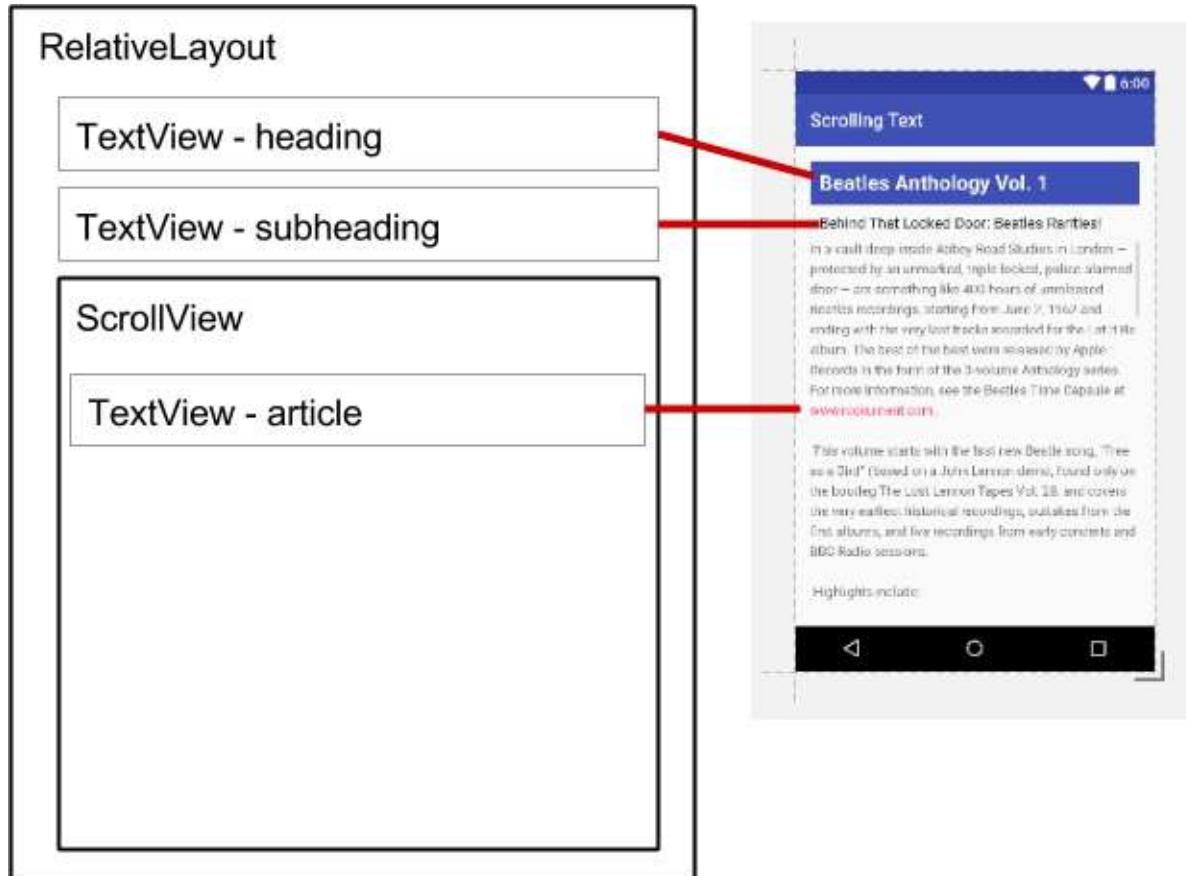
1. Tambahkan ScrollView di antara TextView article_subheading dan TextView artikel. Saat Anda memasukkan atribut <ScrollView, Android Studio automatically adds </ScrollView> at the end, and presents the android:layout_width and android:layout_height dengan saran. Pilih wrap_content dari saran untuk kedua atribut. Kode sekarang harus terlihat seperti ini:

```
<TextView  
    android:id="@+id/article_subheading"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/article_heading"  
    android:padding="10dp"  
    android:textAppearance="@android:style/TextAppearance"  
    android:text="@string/article_subtitle"/>  
  
<ScrollView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/article_subheading"></ScrollView>  
<TextView  
    android:id="@+id/article"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/article_subheading"  
    android:lineSpacingExtra="5sp"  
    android:autoLink="web"  
    android:text="@string/article_text"/>
```

2. Pindahkan </ScrollView> code after the article TextView so that the article TextView attributes are inside the ScrollView XML element.
3. Pindahkan atribut berikut dari TextView article, karena ScrollView sendiri akan ditempatkan bawah elemen article_subheading, dan atribut untuk TextView ini akan berkonflik dengan ScrollView:

```
    android:layout_below="@id/article_subheading"
```

Layout sekarang harus terlihat seperti ini:

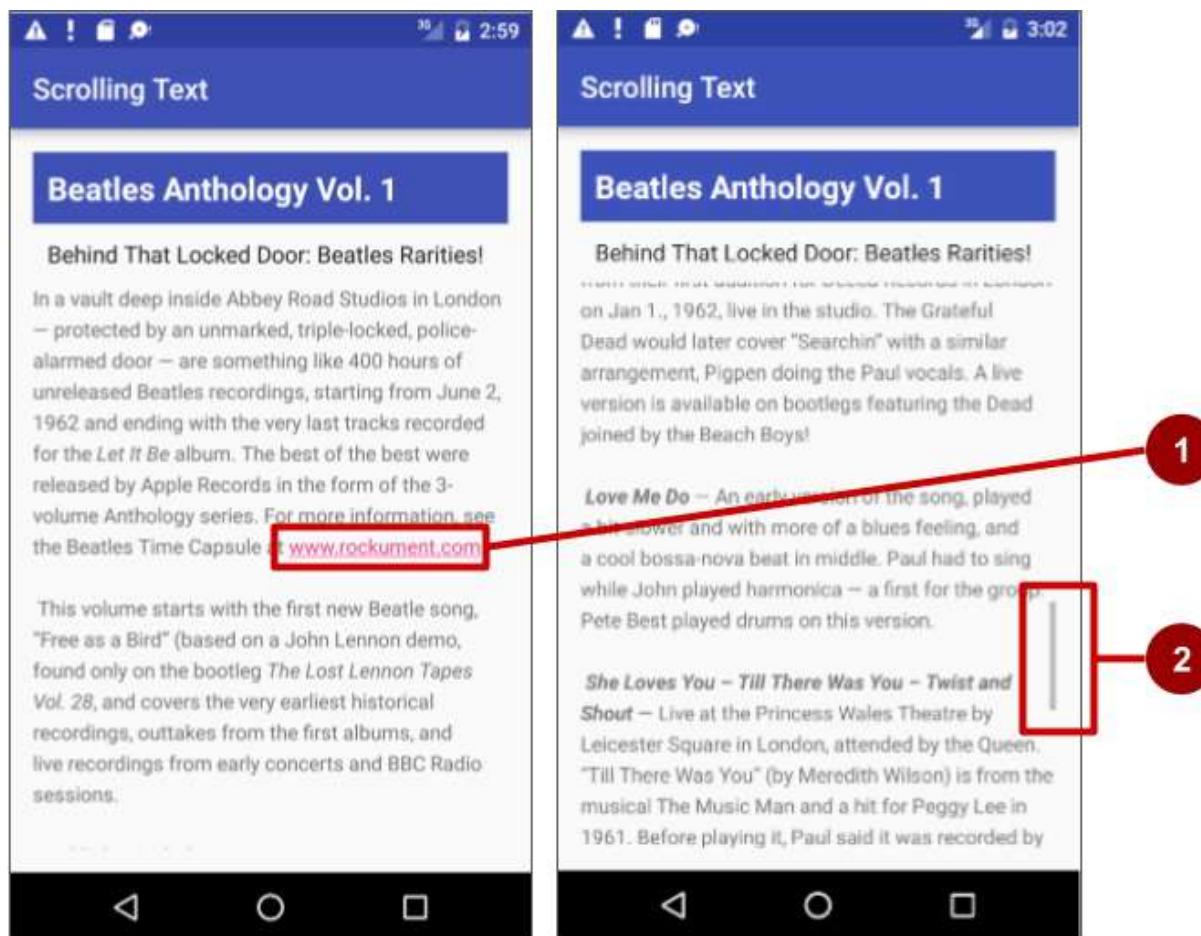


4. Pilih **Code > Reformat Code** untuk memformat ulang kode XML agar TextView `article` sekarang muncul terindentasi di dalam `<ScrollView>` code.
5. Jalankan aplikasi.

Gesek ke atas dan ke bawah untuk menggulir artikel. Bilah gulir muncul di margin kanan saat Anda menggulir.

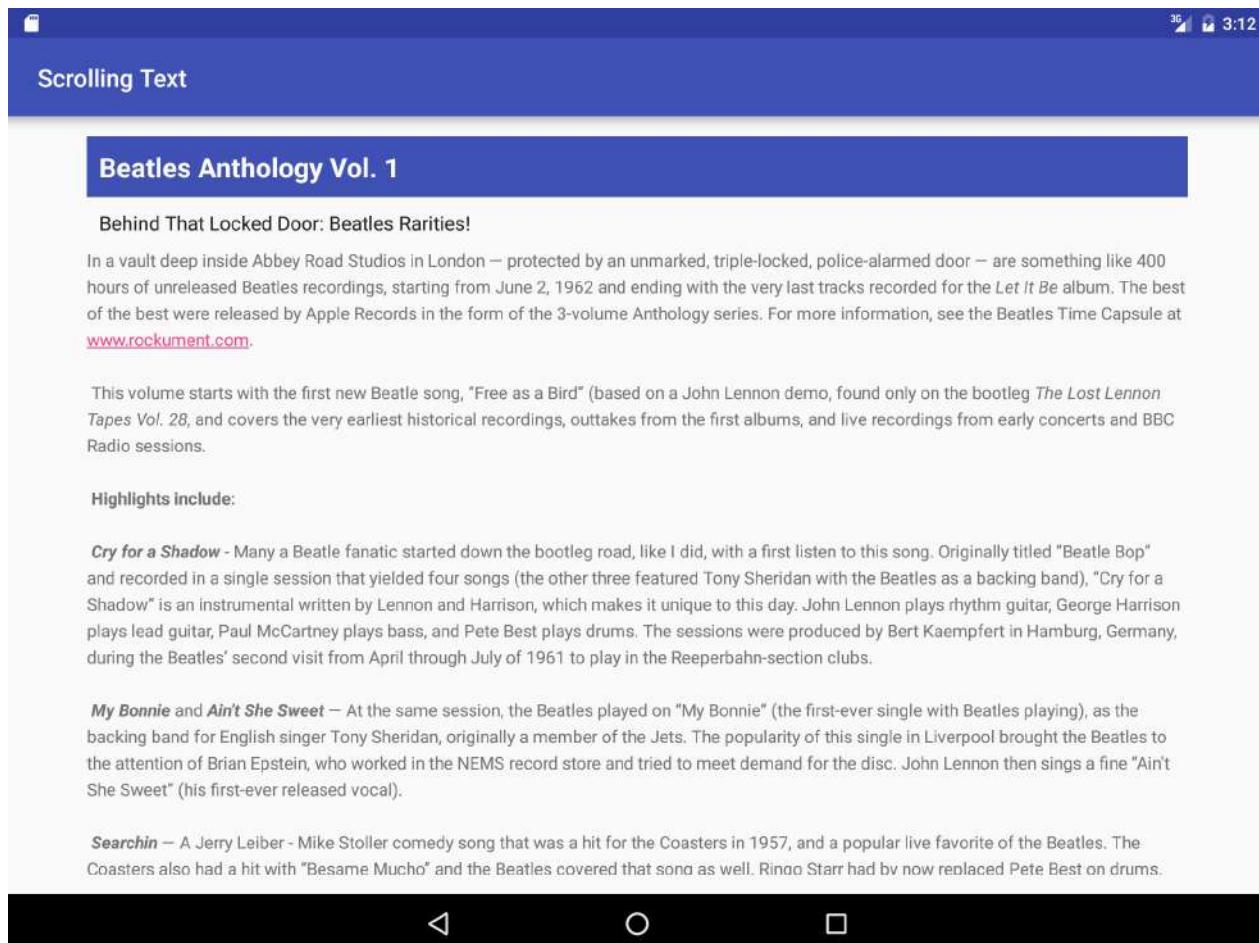
Ketuk tautan web untuk membuka laman web. Atribut `android:autoLink` menjadikan semua URL yang dapat dikenali di TextView (seperti www.rockument.com) tautan web.

6. Putar perangkat atau emulator saat menjalankan aplikasi. Perhatikan bagaimana tampilan bergulir melebar untuk menggunakan tampilan penuh dan tetap bergulir dengan benar.
7. Jalankan aplikasi di tablet atau emulator tablet. Perhatikan bagaimana tampilan bergulir melebar untuk menggunakan tampilan penuh dan tetap bergulir dengan benar.



Pada gambar di atas, hal berikut muncul:

1. Tautan web aktif dalam teks bentuk bebas
2. Bilah gulir yang muncul saat menggulir teks.



File layout **activity_main.xml** sekarang harus terlihat seperti ini:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.scrollingtext.MainActivity">

    <TextView
        android:id="@+id/article_heading"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:textColor="@android:color/white"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:textAppearance="@android:style/TextAppearance.Large"
        android:textStyle="bold"
        android:text="@string/article_title"/>

    <TextView
        android:id="@+id/article_subheading"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/article_heading"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:textAppearance="@android:style/TextAppearance"
        android:text="@string/article_subtitle"/>

    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/article_subheading">

        <TextView
            android:id="@+id/article"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:lineSpacingExtra="5sp"
            android:autoLink="web"
            android:text="@string/article_text"/>

    </ScrollView>
</RelativeLayout>

```

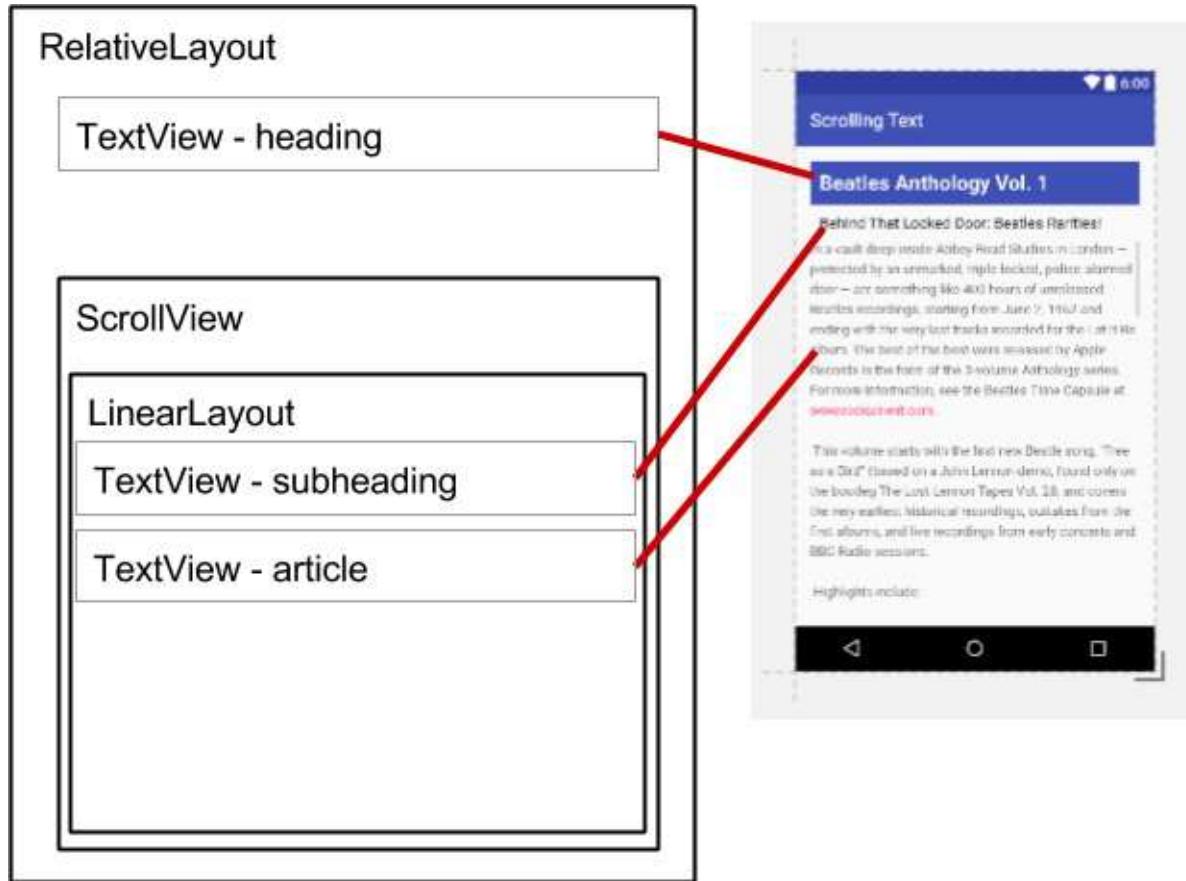
Kode solusi

Proyek Android Studio: [ScrollingText](#)

Tugas 3: Menggulir beberapa elemen

Seperti yang kita lihat sebelumnya, grup tampilan ScrollView bisa hanya berisi satu tampilan anak (seperti TextView `article` yang Anda buat); namun, Tampilan tersebut bisa saja grup tampilan lain yang berisi Tampilan, seperti `LinearLayout`. Anda bisa *melapiskan* grup tampilan seperti `LinearLayout` di dalam grup tampilan `ScrollView`, sehingga menggulir semua yang berada di dalam `LinearLayout`.

Misalnya, jika Anda ingin subheading artikel bergulir bersama artikelnya, tambahkan LinearLayout di dalam ScrollView, dan pindahkan subheading, bersama artikelnya, ke dalam LinearLayout. Grup tampilan LinearLayout menjadi Tampilan anak tunggal dalam ScrollView seperti yang ditunjukkan pada gambar di bawah ini, dan pengguna dapat menggulir seluruh grup tampilan: subheading dan artikelnya.



3.1 Menambahkan LinearLayout ke ScrollView

1. Di komputer Anda, buat salinan folder proyek Android Studio untuk ScrollingText, dan ganti nama proyek menjadi **ScrollingText2**. Untuk menyalin dan mengganti nama proyek, ikuti instruksi "Menyalin dan mengganti nama proyek" di [Appendiks](#).
2. Buka **ScrollingText2** di Android Studio, dan buka file **activity_main.xml** untuk mengubah kode layout XML.
3. Tambahkan LinearLayout di atas TextView `article` dalam ScrollView. Saat Anda memasukkan `<LinearLayout`, Android Studio automatically adds `</LinearLayout>` to the end, and presents the `android:layout_width` and `android:layout_height` attributes with suggestions. Pilih **match_parent** dan **wrap_content** dari saran untuk lebar dan tingginya, secara berturut-turut. Kode sekarang harus terlihat seperti ini:

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"></LinearLayout>
```

Anda menggunakan `match_parent` untuk mencocokkan lebar grup tampilan induk, dan `wrap_content` membuat grup tampilan cukup besar untuk mengurung konten dan padding-nya.

4. Pindahkan kode `</LinearLayout>` code after the `article` TextView but before the closing `</scrollview>` so that the LinearLayout includes the `article` TextView and seluruhnya di dalam ScrollView.
5. Tambahkan atribut `android:orientation="vertical"` ke LinearLayout untuk menyetel orientasi LinearLayout ke vertikal. LinearLayout di dalam ScrollView seharusnya sekarang terlihat seperti ini (pilih **Code > Reformat Code** untuk mengidentifikasi grup tampilan dengan benar):

```

<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/article_subheading">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:id="@+id/article"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:autoLink="web"
            android:lineSpacingExtra="5sp"
            android:text="@string/article_text" />

    </LinearLayout>
</ScrollView>

```

6. Pindahkan `TextView article_subheading` ke posisi di dalam `LinearLayout` di atas `TextView article`.
7. Pindahkan atribut `android:layout_below="@+id/article_heading"` dari `TextView article_subheading`. Karena `TextView` ini sekarang ada di dalam `LinearLayout`, atribut ini akan berkonflik dengan atribut `LinearLayout`.
8. Ubah atribut layout `ScrollView` dari `android:layout_below="@+id/article_subheading"` menjadi `android:layout_below="@+id/article_heading"`. Karena sekarang `subheading` sudah menjadi bagian dari `LinearLayout`, `ScrollView` harus diletakkan di bawah `heading`, bukan `subheading`.
9. Jalankan aplikasi.

Gesek ke atas dan ke bawah untuk menggulir artikel, dan perhatikan bahwa `subheading` sekarang bergulir dengan artikelnya sementara `heading` tetap di tempat.

Kode solusi

Proyek Android Studio: [ScrollingText2](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Tambahkan elemen UI lainnya—[Tombol](#)—ke grup tampilan `LinearLayout` yang berada di dalam `ScrollView`. Munculkan Tombol di bawah artikel. Pengguna harus menggulir ke bagian akhir artikel untuk melihat tombolnya. Gunakan teks "Add Comment" untuk Tombol, agar pengguna mengekliknya untuk menambahkan komentar ke artikel. Untuk tantangan ini, tidak perlu membuat metode penanganan tombol untuk bisa menambahkan komentar; cukup letakkan elemen Tombol di tempat yang tepat dalam layout.

▲ ! ⌂ . 3G 10:36

Scrolling Text

Beatles Anthology Vol. 1

was first recorded in 1962 by the Isley Brothers). A film of the performance shows the Queen smiling at John's remark.

Leave My Kitten Alone — One of the lost Beatle songs recorded during the "Beatles For Sale" sessions but never released. This song, written by Little Willie John, Titus Turner, and James McDougal, was a 1959 R&B hit for Little Willie John and covered by Johnny Preston before the Beatles tried it and shelved it. A reference to a "big fat bulldog" may have influenced John's "Hey Bulldog" (Yellow Submarine album), which is a similar rocker.

One After 909 — A song recorded for the *Let It Be* album was actually worked on way back in the beginning, six years earlier. This take shows how they did it much more slowly, with an R&B feel to it.

ADD COMMENT

◀ ○ □

Kode Solusi Tantangan

Proyek Android Studio: [ScrollingText3](#)

Rangkuman

Dalam praktik ini, Anda belajar tentang elemen tampilan Android Studio dan cara menggulir serta melapiskan kode. Anda bekerja untuk:

- Menambahkan beberapa elemen `TextView` ke layout XML.
 - Menampilkan teks bentuk bebas dalam `TextView` dengan tag pemformatan HTML untuk cetak tebal dan miring.
 - Menggunakan `\n` sebagai karakter akhir baris dalam teks bentuk bebas untuk menjaga agar satu paragraf tidak menabrak paragraf berikutnya.
 - Gunakan atribut `android:autoLink="web"` untuk membuat tautan web dalam teks dapat diklik.
- Menambahkan grup tampilan `ScrollView` ke layout untuk mendefinisikan tampilan bergulir dengan salah satu elemen `TextView`.
- Menambahkan grup tampilan `LinearLayout` di dalam `ScrollView` untuk menggulir beberapa elemen `TextView` bersama-sama.
- Mengekstrak nilai string ke dalam nama string dalam file `strings.xml` untuk penerjemahan sumber daya string yang lebih mudah.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Teks and Tampilan Bergulir](#)

Ketahui selengkapnya

Dokumentasi Developer

- [TextView](#)
- [ScrollView](#)
- [Sumber Daya String](#)
- [Tampilan](#)
- [Layout Relatif](#)

Lainnya:

- Blog Developer Android: [Linkify Teks!](#)
- Codepath: [Menggunakan TextView](#)

1.4: Mempelajari Tentang Sumber Daya yang Tersedia

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Menjelajahi dokumentasi Android resmi](#)
- [Tugas 2: Menggunakan template proyek](#)
- [Tugas 3: Belajar dari kode contoh](#)
- [Tugas 4: Sumber daya selengkapnya](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Dalam praktik ini Anda akan:

- Menjelajahi beberapa sumber daya yang tersedia untuk developer Android dari semua level.
- Menambahkan ikon layar beranda ke aplikasi World List, mengetuk ikon akan membuka aplikasi.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Memahami alur kerja dasar Android Studio.

Yang akan Anda PELAJARI

Lokasi sumber daya developer:

- Di dalam Android Studio.
- Dalam [dokumentasi developer Android](#) di web.
- Di tempat lain di internet.

Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Menjelajahi dan menggunakan sumber daya developer Android.
- Menggunakan sumber daya developer untuk mencari cara menambahkan ikon layar beranda ke perangkat Anda.
Saat ikon ini diklik, aplikasi akan terbuka.

Ringkasan Aplikasi

Anda akan menggunakan aplikasi HelloToast yang sudah ada dan menambahkan ikon peluncur ke aplikasi.

Tugas 1. Menjelajahi dokumentasi developer Android resmi

Anda bisa menemukan dokumentasi developer Android resmi di:

<http://developer.android.com/index.html>

Dokumentasi ini berisi beragam informasi yang terus diperbarui oleh Google.

1.1. Menjelajahi dokumentasi Android resmi

1. Buka <http://developer.android.com/index.html>.
2. Di bagian atas laman, cari tautan **Design**, **Develop**, dan ***Distribute**. Ikuti setiap tautan dan pahami struktur navigasinya.
 - o **Design** adalah tentang Desain Material, filosofi desain konseptual yang menguraikan bagaimana aplikasi harus terlihat dan bekerja pada perangkat seluler. Gulir ke bagian bawah laman landas untuk mencari tautan ke sumber daya seperti lembar stiker dan [palet warna](#).
 - o **Develop** adalah tempat Anda bisa menemukan informasi API, dokumentasi referensi, tutorial, panduan alat, dan contoh kode. Anda bisa menggunakan navigasi situs atau mencari apa yang Anda perlukan.
 - o **Distribute** adalah segala sesuatu yang terjadi setelah Anda menulis aplikasi: menyimpannya di Play Store, meningkatkan basis pengguna, dan [menghasilkan uang](#).
3. Gunakan penelusuran atau buka dokumentasi untuk menyelesaikan tugas berikut:
 - o Menambahkan ikon peluncur ke aplikasi World List. Lihat Panduan API untuk [Ikon Peluncur](#) untuk mempelajari lebih lanjut tentang cara mendesain ikon peluncur yang efektif.
 - o Mempelajari cara memantau penggunaan sumber daya aplikasi di Android Studio.

Tugas 2: Menggunakan template proyek

Android Studio menyediakan template untuk aplikasi umum dan yang disarankan serta desain aktivitas. Menggunakan template bawaan akan menghemat waktu dan membantu Anda mengikuti praktik terbaik.

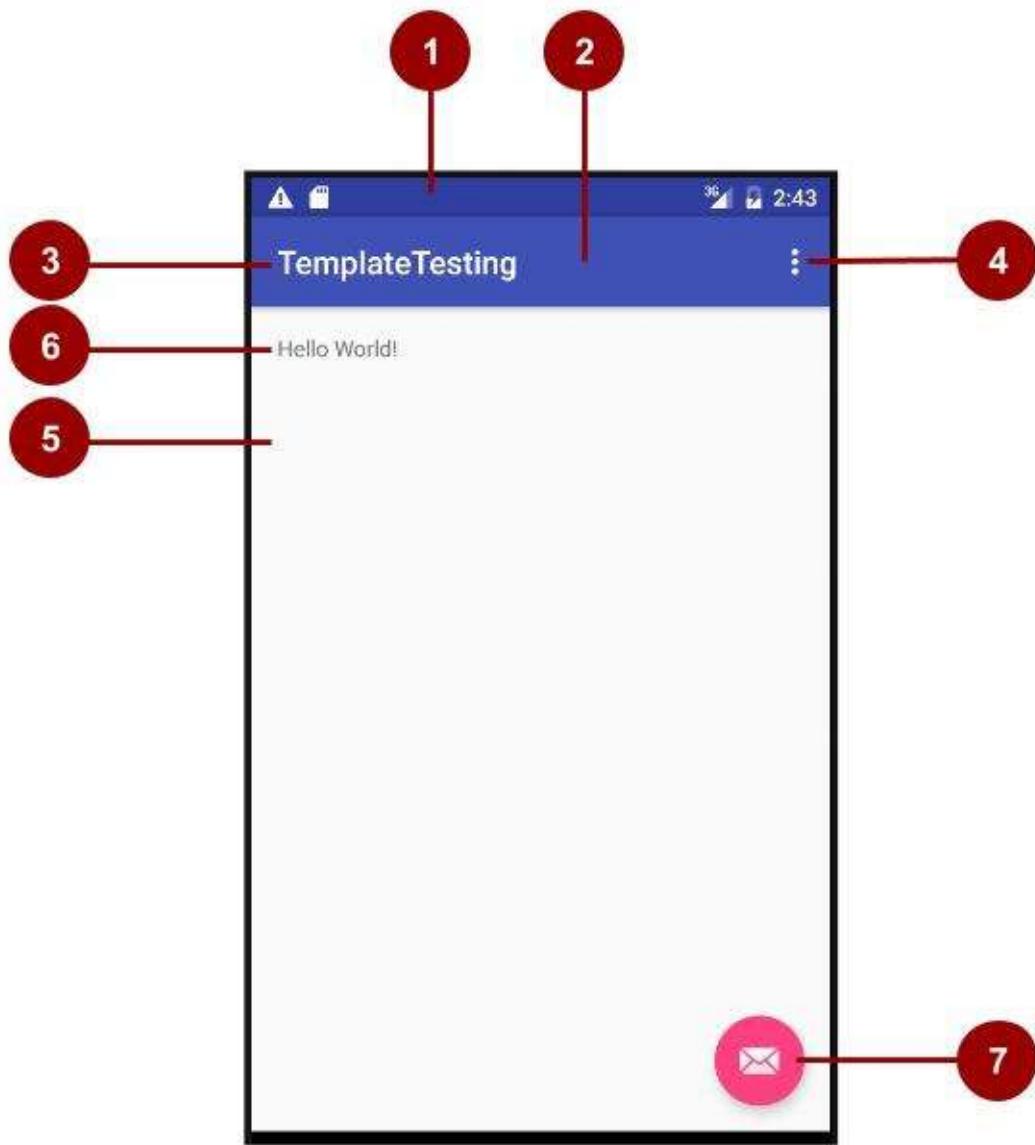
Setiap template menggabungkan aktivitas kerangkadan antarmuka pengguna. Anda sudah menggunakan template Empty Activity. Template Basic Activity memiliki lebih banyak fitur dan menggabungkan fitur-fitur aplikasi yang disarankan, seperti menu opsi.

2.1. Menjelajahi arsitektur Basic Activity

Template Basic Activity adalah template serbaguna yang disediakan oleh Android Studio untuk membantu Anda memulai development aplikasi.

1. Di Android Studio, buatlah proyek baru dengan template Basic Activity.
2. Bangun dan jalankan aplikasi.

3. Identifikasi bagian yang dilabeli pada tangkapan layar dan tabel di bawah ini. Temukan padanannya pada perangkat atau layar emulator.



Arsitektur

template Basic Activity

| # | Keterangan UI | Referensi kode |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Bilah status Bilah ini disediakan dan dikontrol oleh sistem Android. | Tidak terlihat di kode template. Dapat diakses dari aktivitas Anda. Misalnya, Anda bisa menyembunyikan bilah status , jika perlu. |
| 2 | AppBarLayout > Toolbar Bilah aplikasi (juga disebut bilah Tindakan) menyediakan struktur visual, elemen visual standar, dan navigasi. Untuk kompatibilitas mundur, AppBarLayout dalam template menyematkan Toolbar widget dengan fungsionalitas yang sama. Kelas ActionBar Tantangan: Tutorial Bilah Aplikasi | activity_main.xml Cari <code>android.support.v7.widget.Toolbar</code> di dalam <code>android.support.design.widget.AppBarLayout</code> . Ubah toolbar untuk mengubah tampilan induknya, bilah aplikasi. |
| 3 | Nama aplikasi Ini berasal dari nama paket, namun bisa juga apa pun yang Anda pilih. | AndroidManifest.xml <code>android:label="@string/app_name"</code> |
| 4 | Tombol luapan menu opsi Item menu untuk aktivitas, dan opsi global, seperti "Search" dan "Settings" untuk menu setelan. Item menu aplikasi masuk ke dalam menu ini. | MainActivity.java <code>onOptionsItemSelected()</code> mengimplementasikan apa yang terjadi jika item menu dipilih. res > menu > menu_main.xml Sumber daya yang menentukan item menu untuk menu opsi. |
| 5 | CoordinatorLayout CoordinatorLayout adalah layout yang berisi banyak fitur yang menyediakan mekanisme untuk tampilan agar dapat berinteraksi. Antarmuka pengguna aplikasi ada di dalam grup tampilan ini. | activity_main.xml Perhatikan bahwa tidak ada tampilan yang ditetapkan dalam layout ini, namun menyertakan layout lain dengan tempat tampilan ditetapkan. Hal ini memisahkan tampilan sistem dari tampilan unik bagi aplikasi Anda. |
| 6 | TextView Pada contoh berikut, digunakan untuk menampilkan "Hello World". Ganti ini dengan tampilan untuk aplikasi Anda. | content_main.xml Semua tampilan aplikasi Anda didefinisikan dalam file ini. |
| 7 | Floating Action button (FAB) | activity_main.xml MainActivity.java > onCreate memiliki stub yang menyertakan listener onClick di FAB. |

4. Periksa juga kode Java yang sesuai dan file konfigurasi XML.

Memahami kode sumber Java dan file XML akan membantu Anda memperluas dan menyesuaikan template sesuai kebutuhan Anda sendiri.

Lihat [Mengakses Sumber Daya](#) untuk detail tentang sintaks XML untuk mengakses sumber daya.

5. Setelah memahami kode template, coba yang berikut ini:

- Mengubah warna bilah aplikasi (toolbar).
- Melihat gaya yang berkaitan dengan bilah aplikasi (toolbar).
- Mengubah nama aplikasi Anda yang ditampilkan di bilah aplikasi (toolbar).

2.2. Menjelajahi cara menambahkan aktivitas menggunakan template

Untuk praktik sejauh ini, Anda telah menggunakan template Empty Activity dan Basic Activity. Dalam pelajaran berikutnya, template yang akan digunakan bervariasi, tergantung tugasnya.

Template aktivitas ini juga tersedia dari dalam proyek, sehingga Anda bisa menambahkan banyak aktivitas ke aplikasi Anda setelah penyiapan proyek awal. (Anda akan mempelajari lebih banyak tentang ini dalam bab berikutnya.)

1. Buat proyek baru atau pilih proyek yang sudah ada.
2. Dalam direktori proyek Anda, pada tampilan Android, **klik kanan** folder dengan file **java** Anda.
3. Pilih **New > Activity > Gallery**.
4. Tambahkan salah satu aktivitas tersebut, misalnya Navigation Drawer Activity. Temukan file layout untuk Navigation Drawer Activity dan tampilkan di **Design**.

Tugas 3: Belajar dari kode contoh

Android Studio dan dokumentasi Android menyediakan banyak contoh kode yang bisa Anda pelajari, salin, dan gabungkan dengan proyek Anda.

3.1. Contoh Kode Android

Anda dapat menjelajahi ratusan contoh kode langsung dari Android Studio.

1. Pada Android Studio, pilih **File > New > Import Sample**.
2. Jelajahi contoh.
3. Lihat tab Description dan Preview untuk mengetahui selengkapnya tentang setiap contoh.
4. Pilih contoh dan klik **Next**.
5. Terima defaultnya dan klik **Finish**.

Catatan: Contoh yang ada di sini dimaksudkan sebagai titik awal untuk development lebih jauh lagi. Kami mendorong Anda untuk merancang dan membangun ide Anda sendiri ke dalamnya.

3.2. Menggunakan SDK Manager untuk memasang dokumentasi offline

Memasang Android Studio juga memasang dasar-dasar Android SDK (Software Development Kit). Namun, pustaka dan dokumentasi tambahan tersedia, dan Anda bisa memasangnya menggunakan SDK Manager.

1. Pilih **Tools > Android > SDK Manager**. Ini akan membuka setelan Default Preferences.
2. Pada navigasi tangan kiri, temukan dan buka setelan untuk **Android SDK**.
3. Klik **SDK Platforms** di jendela setelan. Anda bisa memasang versi tambahan sistem Android dari sini.
4. Klik di **SDK Update Sites**. Android Studio memeriksa situs yang didaftarkan dan diperiksa secara teratur untuk mendapatkan pembaruan.
5. Klik tab **SDK Tools**. Di sini Anda bisa memasang SDK Tools tambahan yang tidak dipasang secara default, dan versi offline dokumentasi developer Android. Ini memberi Anda akses ke dokumentasi walaupun Anda tidak terhubung ke internet.
6. Periksa "Dokumentasi untuk Android SDK", klik **Apply**, dan ikuti perintahnya.
7. Buka direktori **Android/sdk** dan buka folder **docs**.
8. Temukan **index.html** dan buka.

Tugas 4: Sumber daya selengkapnya

- Saluran [YouTube Developer Android](#) adalah sumber daya yang bagus untuk tutorial dan tips.
- Tim Android mengeposkan berita dan tips di [Blog Android resmi](#).
- [Stack Overflow](#) adalah komunitas jutaan programmer yang saling membantu. Jika Anda menemukan masalah, kemungkinan seseorang telah mengeposkan jawaban di forum ini. Di [Stack Overflow](#), Anda bahkan bisa menanyakan, "Bagaimana cara menyiapkan dan menggunakan ADB dengan WiFi?", atau "Kebocoran memori apa yang paling

- umum di development Android?"
- Terakhir, ketik pertanyaan Anda di penelusuran Google, dan mesin penelusuran Google akan mengumpulkan hasil yang relevan dari semua sumber daya ini. Misalnya, "Versi OS Android mana yang paling populer di India?"

4.1. Telusuri di Stack Overflow menggunakan tag

1. Buka [Stack Overflow](#)
2. Di kotak pencarian, ketik [android].
Tanda kurung [] menandakan bahwa Anda ingin mencari pos yang telah ditag sebagai tag tentang Android.
3. Anda bisa menggabungkan tag dan istilah pencarian untuk membuat penelusuran Anda lebih spesifik. Telusuri
 - [android] dan [layout]
 - [android] "hello world"
4. [Baca selengkapnya](#) tentang banyak cara untuk mencari di Stackoverflow.

Rangkuman

- Dokumentasi Developer Android Resmi - <http://developer.android.com>
- Desain Material adalah filosofi desain konseptual yang menguraikan bagaimana aplikasi harus terlihat dan bekerja pada perangkat seluler.
- Google Play store adalah sistem distribusi digital Google untuk aplikasi yang dikembangkan dengan Android SDK.
- Android Studio menyediakan template untuk aplikasi umum dan yang disarankan serta desain aktivitas. Template ini menawarkan kode bekerja untuk kasus penggunaan umum.
- Saat Anda membuat proyek, Anda bisa memilih template untuk aktivitas pertama.
- Saat Anda mengembangkan aplikasi lebih lanjut, aktivitas dan komponen aplikasi lainnya dapat dibuat dari template bawaan.
- Android Studio berisi banyak contoh kode yang bisa Anda pelajari, salin, dan gabungkan dengan proyek Anda.

[Konsep terkait]

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Sumber Daya untuk Membantu Anda Belajar](#)

Ketahui selengkapnya

Dokumentasi Developer

- [Dokumentasi Android resmi](#)
- [Image Asset Studio](#)
- [Laman Android Monitor](#)
- [Blog Android Resmi](#)
- [Blog Pengembang Android](#)
- [Google I/O Codelabs](#)
- [Stack Overflow](#)
- [Kosakata Android](#)
- [Situs web Pelatihan Developer Google](#)

Kode

- [Kode sumber untuk latihan pada GitHub](#)
- [Contoh kode Android untuk developer](#)

Video

- [Saluran YouTube Pengembang Android](#)
- [Kursus online Udacity](#)

2.1: Membuat dan Memulai Aktivitas

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1. Membuat proyek TwoActivities
- Tugas 2. Membuat dan meluncurkan aktivitas kedua
- Tugas 3. Mengirim data dari aktivitas utama ke aktivitas kedua
- Tugas 4. Mengembalikan data ke aktivitas utama
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Sebuah aktivitas mewakili satu layar di aplikasi, yang digunakan pengguna untuk melakukan satu tugas terfokus seperti menelepon, mengambil foto, mengirim email, atau melihat peta. Aktivitas biasanya ditampilkan ke pengguna dalam jendela layar penuh.

Sebuah aplikasi biasanya terdiri atas beberapa aktivitas yang terikat secara longgar satu sama lain. Biasanya, satu aktivitas dalam aplikasi ditetapkan sebagai aktivitas "utama", yang ditampilkan kepada pengguna saat membuka aplikasi. Tiap aktivitas kemudian bisa memulai aktivitas lain untuk melakukan berbagai tindakan.

Tiap kali aktivitas baru dimulai, aktivitas sebelumnya akan dihentikan, namun sistem mempertahankan aktivitas dalam sebuah tumpukan ("back stack"). Saat sebuah aktivitas baru dimulai, aktivitas baru itu akan didorong ke atas back-stack dan mengambil fokus pengguna. Back-stack mematuhi mekanisme dasar tumpukan "masuk terakhir, keluar pertama", jadi jika pengguna selesai dengan aktivitas saat ini dan menekan tombol Back, aktivitas saat ini akan dikeluarkan dari tumpukan (dan dimusnahkan) dan aktivitas sebelumnya akan dilanjutkan.

Aktivitas Android dimulai atau diaktifkan dengan *intent*. Intent adalah pesan asinkron yang bisa Anda gunakan dalam aktivitas untuk meminta tindakan dari aktivitas lain (atau komponen aplikasi lainnya). Anda menggunakan intent untuk memulai satu aktivitas dari aktivitas lainnya dan meneruskan data di antara aktivitas.

Ada dua jenis intent: *eksplisit* dan *implisit*. Intent eksplisit adalah intent yang targetnya Anda ketahui. Anda sudah mengetahui nama kelas yang sepenuhnya memenuhi syarat dari aktivitas spesifik tersebut. Intent implisit adalah intent yang nama komponen targetnya tidak Anda ketahui, namun memiliki tindakan umum untuk dikerjakan. Dalam praktik ini Anda akan belajar tentang intent eksplisit. Anda akan mempelajari tentang intent implisit pada praktik berikutnya.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Membuat dan menjalankan aplikasi dalam Android Studio.
- Membuat dan mengedit elemen UI dengan Layout Editor grafis atau langsung di file layout XML.
- Menambahkan fungsionalitas onClick ke tombol.

Yang akan Anda PELAJARI

Anda akan belajar:

- Membuat aktivitas baru di Android Studio.
- Mendefinisikan aktivitas induk dan anak untuk navigasi "Up".

- Memulai aktivitas dengan intent eksplisit.
- Meneruskan data di antara aktivitas dengan ekstra intent.

Yang akan Anda LAKUKAN

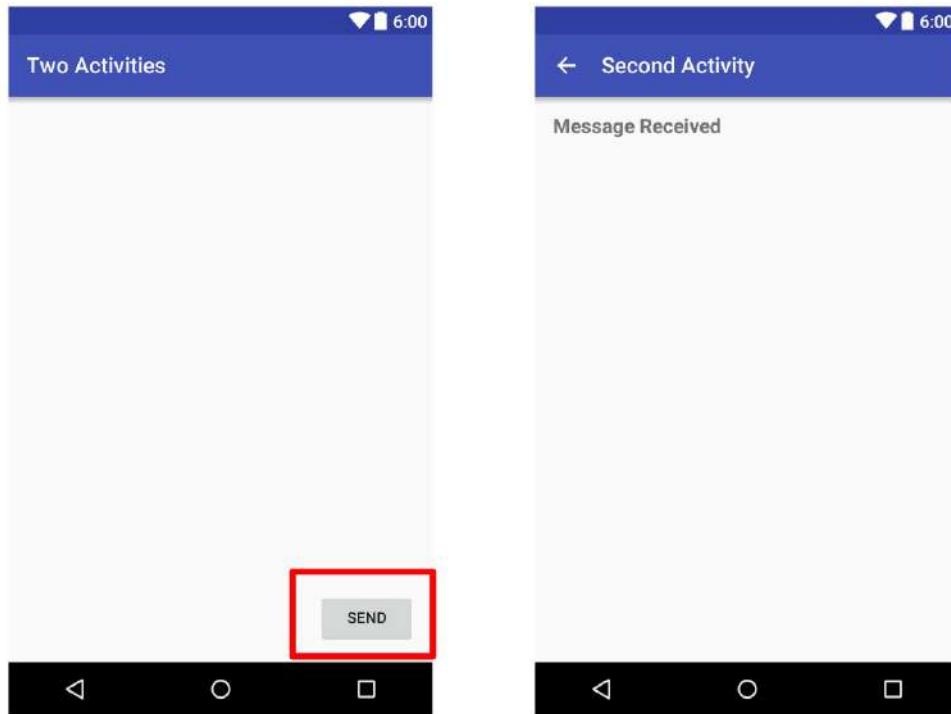
Dalam praktik ini Anda akan:

- Membuat aplikasi Android baru dengan dua aktivitas.
- Meneruskan beberapa data (string) dari aktivitas utama ke yang kedua menggunakan intent, dan menampilkan data tersebut dalam aktivitas kedua.
- Mengirimkan bit data kedua yang berbeda kembali ke aktivitas utama, menggunakan intent juga.

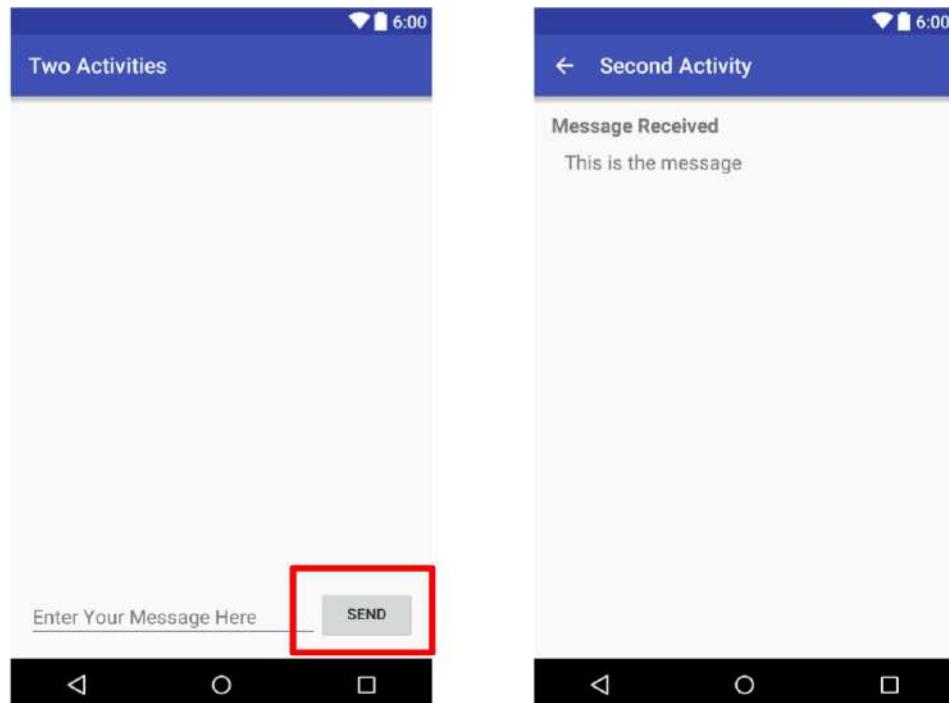
Ringkasan Aplikasi

Dalam bab ini Anda akan membuat dan membangun aplikasi bernama TwoActivities yang, ternyata, berisi dua aktivitas. Aplikasi ini akan dibangun dalam tiga tahap.

Pada tahap pertama, buat aplikasi yang aktivitas utamanya hanya berisi satu tombol (Send). Saat pengguna mengeklik tombol ini, aktivitas utama menggunakan intent untuk memulai aktivitas kedua.

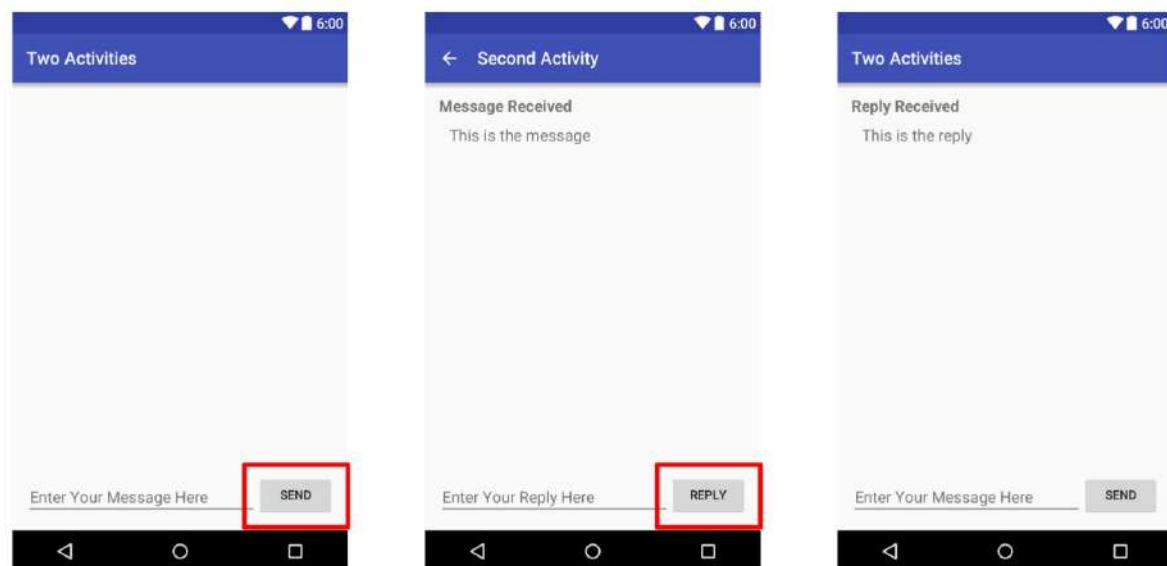


Dalam tahap kedua, Anda akan menambahkan tampilan EditText ke aktivitas utama. Pengguna memasukkan pesan dan mengeklik Send. Aktivitas utama menggunakan intent untuk memulai aktivitas kedua dan untuk mengirimkan pesan pengguna ke aktivitas tersebut. Aktivitas kedua menampilkan pesan yang diterimanya.



Main activity → Second activity

Pada tahap terakhir aplikasi TwoActivities, tambahkan tampilan EditText dan tombol Reply ke aktivitas kedua. Sekarang pengguna bisa mengetik pesan balasan dan mengeklik Reply, dan balasannya ditampilkan di aktivitas utama. Pada titik ini, gunakan intent di sini untuk meneruskan pesan balasan kembali dari aktivitas kedua ke aktivitas utama.



Main activity → Second activity → Back to Main activity

Tugas 1. Membuat proyek TwoActivities

Dalam tugas ini Anda akan menyiapkan proyek awal dengan aktivitas utama, mendefinisikan layout, dan mendefinisikan metode kerangka untuk peristiwa tombol `onClick`.

1.1 Membuat proyek TwoActivities

1. Mulai Android Studio dan buat proyek Android Studio baru.

Panggil aplikasi "TwoActivities" dan ubah domain perusahaan ke "android.example.com." Pilih SDK Minimum yang Anda gunakan di proyek sebelumnya.

2. Pilih **Empty Activity** untuk template proyek. Klik **Next**.
3. Terima nama aktivitas default (MainActivity). Pastikan kotak **Generate Layout file** dicentang. Klik **Finish**.

1.2 Definisikan layout untuk aktivitas utama

1. Buka `res/layout/activity_main.xml`. Pada Layout Editor, klik tab Design di bagian bawah layar.
2. Hapus TextView yang berbunyi "Hello World."
3. Tambahkan Tombol ke layout dalam posisi apa pun.
4. Beralih ke XML Editor (klik tab Text) dan modifikasi atribut ini di dalam Tombol:

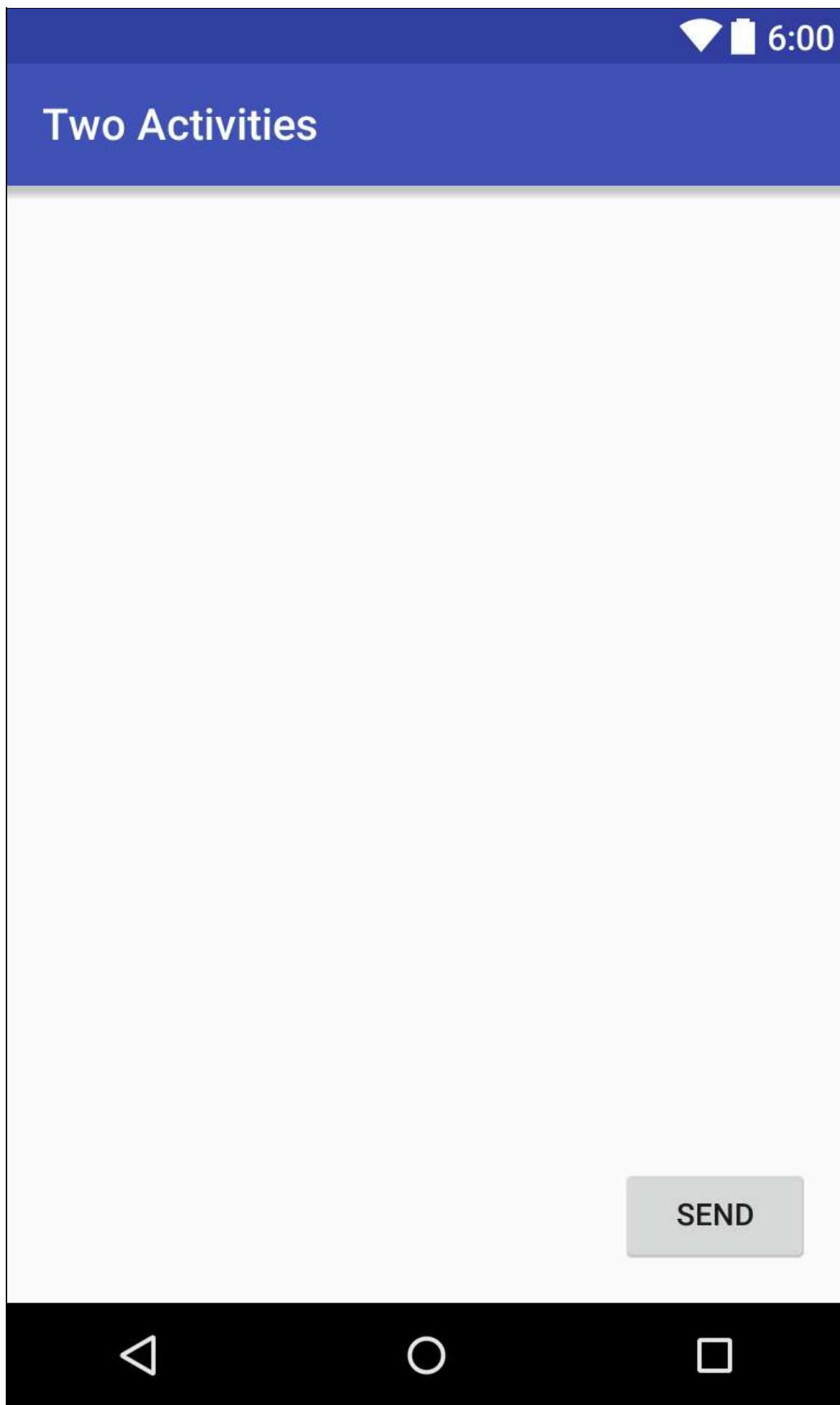
| Atribut | Nilai |
|-----------------------------------------------|-------------------------------------|
| <code>android:id</code> | <code>"@+id/button_main"</code> |
| <code>android:layout_width</code> | <code>wrap_content</code> |
| <code>android:layout_height</code> | <code>wrap_content</code> |
| <code>android:layout_alignParentRight</code> | <code>"true"</code> |
| <code>android:layout_alignParentBottom</code> | <code>"true"</code> |
| <code>android:layout_alignParentEnd</code> | <code>"true"</code> |
| <code>android:text</code> | <code>"Send"</code> |
| <code>android:onClick</code> | <code>"launchSecondActivity"</code> |

Ini mungkin membuat kesalahan "Method launchSecondActivity is missing in MainActivity." Abaikan kesalahan ini sekarang. Masalah ini akan diatasi pada tugas berikutnya.

5. Letakkan kursor pada kata `"Send"`.
6. Tekan **Alt-Enter (Option-Enter di Mac)**, dan pilih **Extract string resources**.
7. Setel **nama Sumber Daya** ke `button_main` dan klik **OK**.

Proses ini membuat sumber daya string dalam file `values/res/string.xml`, dan string dalam kode Anda digantikan dengan referensi ke sumber daya string tersebut.

8. Pilih **Code > Reformat Code** untuk memformat kode XML, jika perlu.
9. Pratinjau layout aktivitas utama menggunakan Layout Editor. Layout harus terlihat seperti ini:



Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.MainActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_main"
        android:id="@+id/button_main"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="launchSecondActivity"/>
</RelativeLayout>
```

1.3 Mendefinisikan tindakan tombol

Dalam tugas ini Anda akan mengimplementasikan metode onClick yang didefinisikan dalam layout.

1. Dalam Editor XML, letakkan kursor pada kata "launchSecondActivity" .
2. Tekan **Alt-Enter (Option-Enter di Mac)** dan pilih **Create 'launchSecondActivity(View)' dalam 'MainActivity'**.

File MainActivity.java terbuka, dan Android Studio membuat metode kerangka untuk handler onClick.

3. Dalam `launchSecondActivity` , tambahkan pernyataan log yang bernama "Button Clicked!"

```
Log.d(LOG_TAG, "Button clicked!");
```

`LOG_TAG` akan terlihat berwarna merah. Definisi untuk variabel tersebut akan ditambahkan dalam langkah selanjutnya.

4. Tempatkan kursor pada kata "Log" dan tekan **Alt-Enter (Option-Enter di Mac)**. Android Studio menambahkan pernyataan impor untuk `android.util.Log`.
5. Di bagian atas kelas, tambahkan konstanta untuk variabel `LOG_TAG`:

```
private static final String LOG_TAG =
    MainActivity.class.getSimpleName();
```

Konstanta ini menggunakan nama kelas itu sendiri sebagai tag-nya.

6. Jalankan aplikasi Anda. Saat mengeklik tombol "Send", Anda akan melihat pesan "Button Clicked!" di Android Monitor (logcat). Jika ada terlalu banyak keluaran di monitor, ketik `MainActivity` di dalam kotak pencarian dan log hanya akan menunjukkan baris yang cocok dengan tag tersebut.

Kode Solusi:

```

package com.example.android.twoactivities;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;

public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = MainActivity.class.getSimpleName();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void launchSecondActivity(View view) {
        Log.d(LOG_TAG, "Button clicked!");
    }
}

```

Tugas 2. Membuat dan meluncurkan aktivitas kedua

Setiap aktivitas baru yang Anda tambahkan ke proyek memiliki layout dan file Java-nya sendiri, terpisah dari milik aktivitas utama. Aktivitas tersebut juga memiliki elemen `<activity>` sendiri dalam manifes Android. Seperti aktivitas utama, aktivitas baru yang Anda buat di Android Studio juga meluas dari kelas `AppCompatActivity`.

Semua aktivitas di aplikasi terhubung secara longgar dengan satu sama lain. Namun, Anda dapat mendefinisikan sebuah aktivitas sebagai induk dari akti lain dalam file `AndroidManifest.xml`. Hubungan induk-anak ini memungkinkan Android untuk menambahkan petunjuk navigasi seperti panah yang menghadap ke kiri pada bilah judul untuk setiap aktivitas.

Aktivitas berkomunikasi dengan satu sama lain (di dalam aplikasi yang sama dan di berbagai aplikasi yang berbeda) dengan *intent*. Ada dua tipe intent, eksplisit dan implisit. Intent eksplisit adalah intent yang targetnya Anda ketahui. Anda sudah mengetahui nama kelas yang sepenuhnya memenuhi syarat dari aktivitas spesifik tersebut. Intent implisit adalah intent yang nama komponen targetnya tidak Anda ketahui, namun memiliki tindakan umum untuk dikerjakan. Anda akan mempelajari tentang intent implisit pada praktik berikutnya.

Pada tugas ini Anda akan menambahkan aktivitas kedua pada aplikasi, dengan layout-nya sendiri. Anda akan memodifikasi manifes Android untuk mendefinisikan aktivitas utama sebagai induk aktivitas kedua. Lalu Anda akan memodifikasi metode kejadian `onClick` dalam aktivitas utama untuk menyertakan intent yang meluncurkan aktivitas kedua saat tombol diklik.

2.1 Membuat aktivitas kedua

1. Klik folder `app` untuk proyek Anda dan pilih **File > New > Activity > Empty Activity**.
2. Namakan aktivitas baru "SecondActivity." Pastikan **Generate Layout File** dicentang, dan nama layout akan diisi dengan `activity_second`.
3. Klik **Finish**. Android Studio menambahkan layout aktivitas baru (`activity_second`) dan file Java baru (`SecondActivity`) ke proyek Anda untuk aktivitas baru tersebut. Ini juga akan memperbarui manifes Android untuk menyertakan aktivitas baru.

2.2 Memodifikasi manifes Android

1. Buka `manifests/AndroidManifest.xml`.
2. Temukan elemen `<activity>` yang dibuat Android Studio untuk aktivitas kedua.

```
<activity android:name=".SecondActivity"></activity>
```

3. Tambahkan atribut ini ke elemen `<activity>` :

| Atribut | Nilai |
|----------------------------|-------------------|
| android:label | "Second Activity" |
| android:parentActivityName | ".MainActivity" |

Atribut `label` menambahkan judul aktivitas ke bilah tindakan.

Atribut `parentActivityName` menandakan bahwa aktivitas utama adalah induk aktivitas kedua. Hubungan aktivitas induk ini digunakan untuk navigasi "ke atas" di dalam aplikasi Anda. Dengan mendefinisikan atribut ini, bilah tindakan untuk aktivitas kedua akan muncul dengan panah yang menghadap ke kiri untuk memungkinkan pengguna untuk bergerak "ke atas" ke aktivitas utama.

4. Tempatkan kursor di "Second Activity" dan tekan **Alt-Enter (Option-Enter di Mac)**.
5. Pilih **Extract string resource**, namakan sumber daya "activity2_name", dan klik **OK**. Android Studio menambahkan sumber daya string untuk label aktivitas.
6. Tambahkan elemen `<meta-data>` element inside the `<activity>` di dalam elemen untuk aktivitas kedua. Gunakan atribut ini:

| Atribut | Nilai |
|---------------|--------------------------------------------------|
| android:name | "android.support.PARENT_ACTIVITY" |
| android:value | "com.example.android.twoactivities.MainActivity" |

Elemen `<meta-data>` menyediakan informasi arbitrer tambahan tentang aktivitas sebagai pasangan kunci-nilai. Dalam hal ini, atribut ini mencapai hal yang sama dengan atribut `android:parentActivityName`, yakni mendefinisikan hubungan antara dua aktivitas untuk tujuan navigasi ke atas. Atribut ini diperlukan untuk versi Android lebih lama. `android:parentActivityName` hanya tersedia untuk API level 16 dan yang lebih tinggi.

Kode Solusi:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.twoactivities">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SecondActivity">
            android:label="@string/activity2_name"
            android:parentActivityName=".MainActivity">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="com.example.android.twoactivities.MainActivity" />
        </activity>
    </application>
</manifest>

```

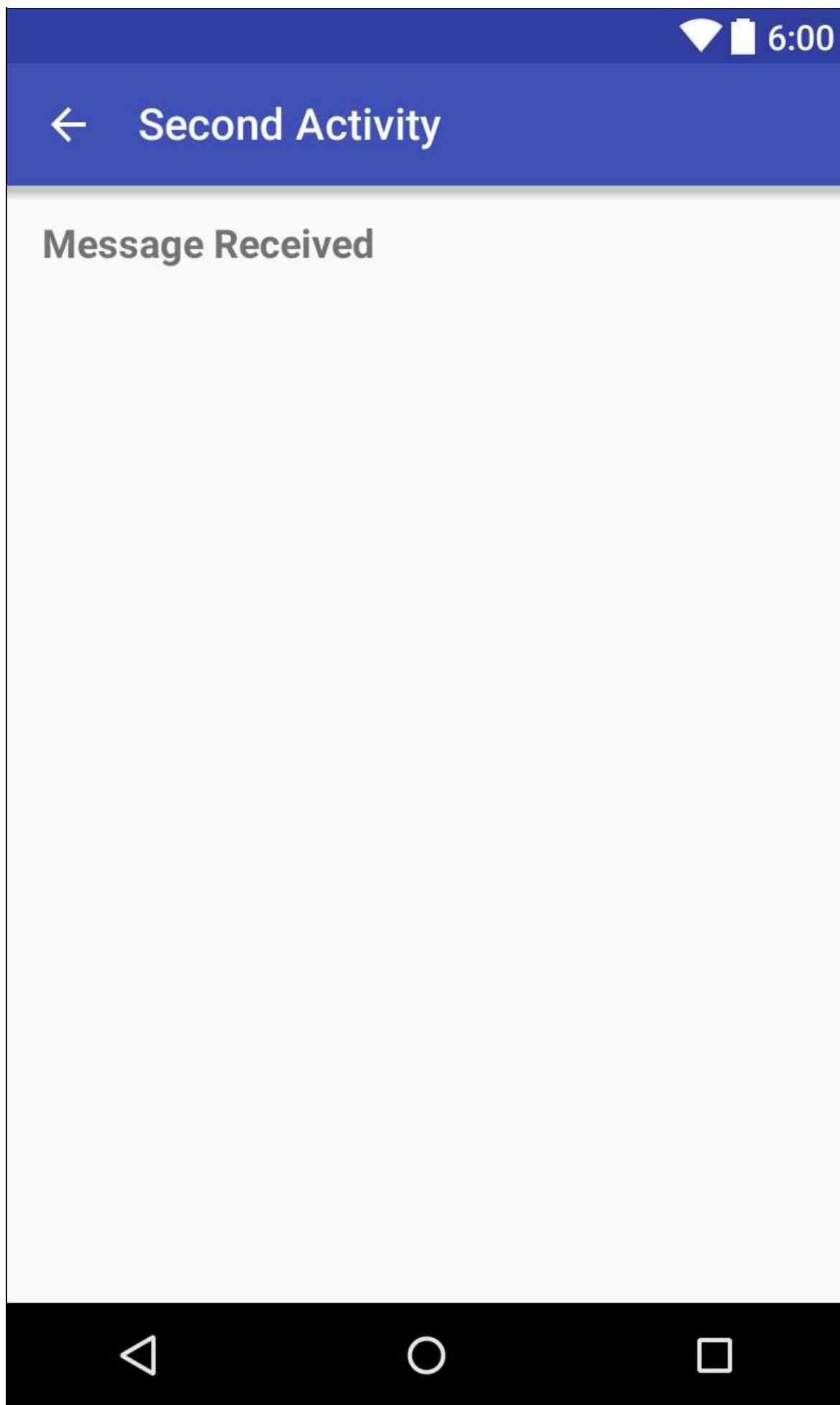
2.3 Mendefinisikan layout untuk aktivitas utama

1. Buka `res/layout/activity_second.xml`.
2. Tambahkan `TextView` ("Plain Textview" dalam Layout Editor). Berikan atribut berikut pada `TextView`:

| Atribut | Nilai |
|-----------------------------|--------------------------------------|
| android:id | "@+id/text_header" |
| android:layout_width | wrap_content |
| android:layout_height | wrap_content |
| android:layout_marginBottom | "@dimen/activity_vertical_margin" |
| android:text | "Message Received" |
| android:textAppearance | "?android:attr/textAppearanceMedium" |
| android:textStyle | "bold" |

Nilai textAppearance adalah atribut bertema Android yang mendefinisikan gaya font dasar font kecil, medium, dan besar. Anda akan mempelajari selengkapnya tentang tema pada pelajaran berikutnya.

3. Ekstrak string "Message Received" ke sumber daya bernama text_header.
4. Pratinjau layout dalam Layout Editor. Layout harus terlihat seperti ini:



Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".SecondActivity">

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:text="@string/text_header"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold" />
</RelativeLayout>
```

2.4 Menambahkan intent ke aktivitas utama

Dalam aktivitas ini Anda akan menambahkan intent eksplisit ke aktivitas utama. Intent ini digunakan untuk mengaktifkan aktivitas kedua saat tombol Send diklik.

1. Buka file Java untuk `MainActivity` (`java/com.example.android.twoactivities/MainActivity`).
2. Buat intent baru dalam metode `launchSecondActivity()`.

Konstruktor intent memerlukan dua argumen untuk intent eksplisit: konteks `Aplikasi` dan komponen spesifik yang akan menerima intent tersebut. Di sini Anda harus menggunakan `this` sebagai konteksmnya, dan `SecondActivity.class` sebagai kelas spesifiknya.

```
Intent intent = new Intent(this, SecondActivity.class);
```

3. Tempatkan kursor di intent dan tekan **Alt-Enter (Option-Enter di Mac)** untuk menambahkan impor ke kelas intent.
4. Panggil metode `startActivity()` dengan intent baru sebagai argumennya.

```
startActivity(intent);
```

5. Jalankan aplikasi.

Saat mengeklik tombol Send, aktivitas utama mengirim intent dan sistem Android membuka aktivitas kedua. Aktivitas kedua muncul di layar. Untuk kembali ke aktivitas utama, klik tombol Kembali Android di kiri bawah layar, Anda bisa menggunakan panah kiri di atas aktivitas kedua untuk kembali ke aktivitas utama.

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode bersifat opsional.

Tantangan: Apa yang terjadi jika Anda menghapus `android:parentActivityName` dan elemen `<meta-data>` dari manifest? Buat perubahan ini dan jalankan aplikasi.

Tugas 3. Mengirim data dari aktivitas utama ke aktivitas kedua

Dalam tugas terakhir, Anda akan menambahkan intent eksplisit ke aktivitas utama yang mengaktifkan aktivitas kedua. Anda juga bisa menggunakan intent untuk mengambil data dari aktivitas satu ke aktivitas lainnya.

Dalam tugas ini, Anda akan memodifikasi intent eksplisit dalam aktivitas utama untuk menyertakan data tambahan (dalam kasus ini, string yang dimasukkan pengguna) dalam ekstra intent. Lalu Anda akan memodifikasi aktivitas kedua untuk mendapatkan data kembali dari ekstra intent dan menampilkannya di layar.

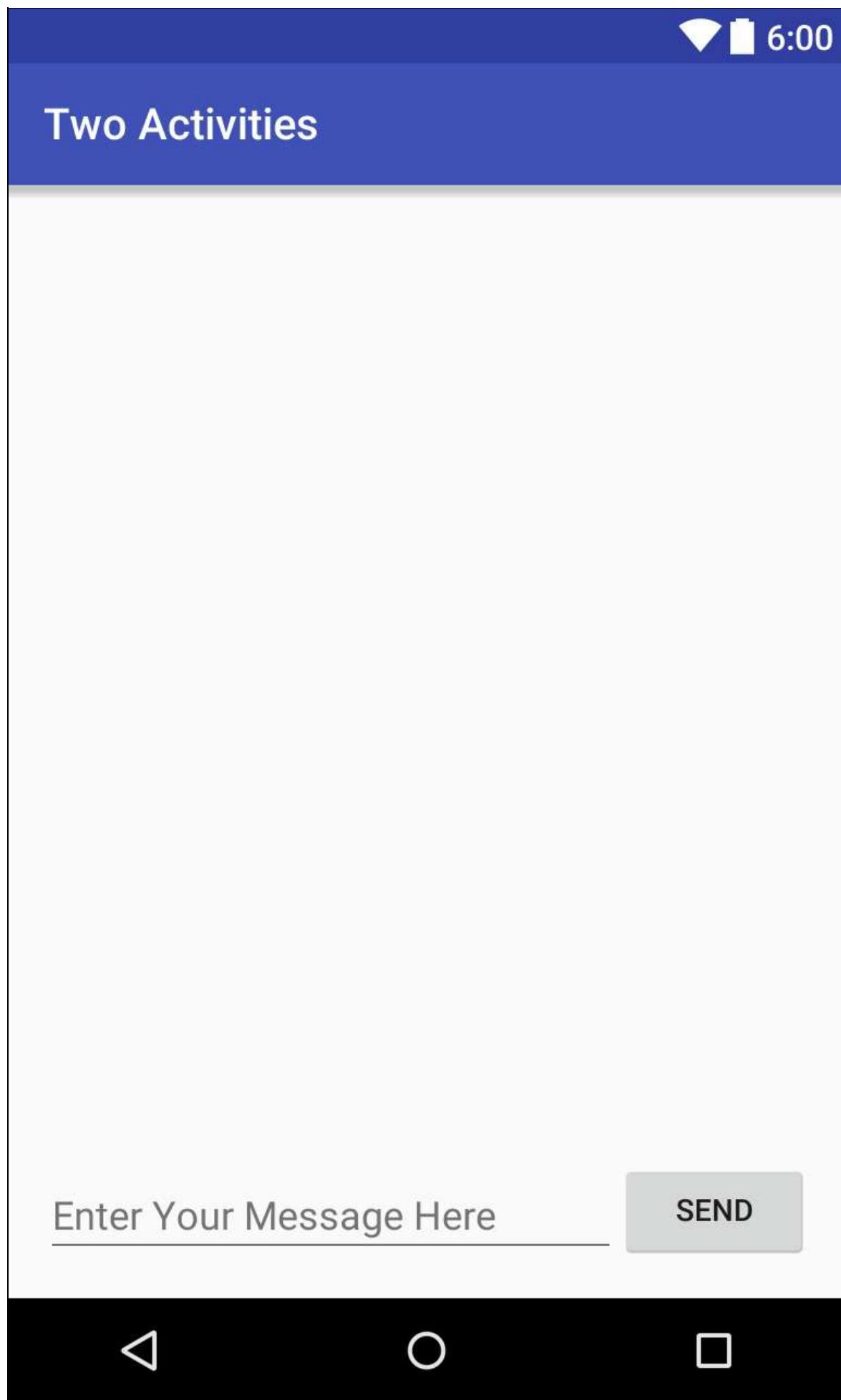
3.1 Menambahkan EditText ke layout aktivitas utama

1. Buka res/layout/activity_main.xml.
2. Tambahkan tampilan EditText (Plain Text dalam Layout Editor.) Berikan EditText atribut berikut:

| Atribut | Nilai |
|----------------------------------|---------------------------|
| android:id | "@+id/editText_main" |
| android:layout_width | match_parent |
| android:layout_height | wrap_content |
| android:layout_toLeftOf | "@+id/button_main" |
| android:layout_toStartOf | "@+id/button_main" |
| android:layout_alignParentBottom | "true" |
| android:hint | "Enter Your Message Here" |

3. Hapus android:text attribute.
4. Ekstrak string "Enter Your Message Here" ke sumber daya bernama editText_main.

Layout baru untuk aktivitas utama harus terlihat seperti ini:



Kode Solusi:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_main"
        android:id="@+id/button_main"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="launchSecondActivity"/>

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button_main"
        android:layout_toStartOf="@+id/button_main"
        android:hint="@string/editText_main" />
</RelativeLayout>

```

3.2 Menambahkan string ke ekstra intent aktivitas utama

Objek intent dapat meneruskan data ke aktivitas target dalam dua cara: dalam bidang data, atau dalam *ekstra* intent. Data intent adalah URI yang menandakan data spesifik untuk dikerjakan. Jika informasi yang ingin diteruskan ke aktivitas melalui intent bukan URI, atau jika Anda memiliki lebih dari satu informasi yang ingin dikirimkan, Anda bisa meletakkan informasi tambahan tersebut ke dalam *ekstra* intent.

Ekstra intent adalah pasangan kunci/nilai dalam *Bundel*. *Bundel* adalah kumpulan data, yang disimpan sebagai pasangan kunci/nilai. Untuk meneruskan informasi dari satu aktivitas ke yang lainnya, letakkan kunci dan nilai ke dalam bundel ekstra intent dari aktivitas mengirim, dan keluarkan lagi ke aktivitas menerima.

1. Buka `java/com.example.android.twoactivities/MainActivity`.
2. Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk ekstra intent:

```

public static final String EXTRA_MESSAGE =
    "com.example.android.twoactivities.extra.MESSAGE";

```

3. Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText. Mengimpor kelas EditText.

```

private EditText mMessageEditText;

```

4. Dalam metode `onCreate()`, gunakan `findViewById` untuk mendapatkan referensi tentang contoh EditText dan menetapkannya ke variabel privat tersebut:

```

mMessageEditText = (EditText) findViewById(R.id.editText_main);

```

5. Dalam metode `launchSecondActivity()`, di bawah intent baru, dapatkan teks dari EditText sebagai string:

```

String message = mMessageEditText.getText().toString();

```

6. Tambahkan string tersebut ke intent sebagai sebuah ekstra dengan konstanta EXTRA_MESSAGE sebagai kunci dan string-nya sebagai nilai:

```
intent.putExtra(EXTRA_MESSAGE, message);
```

Kode Solusi:

```
package com.example.android.twoactivities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = MainActivity.class.getSimpleName();

    public static final String EXTRA_MESSAGE =
"com.example.android.twoactivities.extra.MESSAGE";

    private EditText mMessageEditText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mMessageEditText = (EditText) findViewById(R.id.editText_main);
    }

    public void launchSecondActivity(View view) {
        Log.d(LOG_TAG, "Button clicked!");

        Intent intent = new Intent(this, SecondActivity.class);
        String message = mMessageEditText.getText().toString();

        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
```

3.3 Menambahkan TextView ke aktivitas kedua untuk pesan

1. Buka `res/layout/activity_second.xml`.
2. Tambahkan TextView kedua. Berikan atribut berikut pada TextView:

| Atribut | Nilai |
|----------------------------|-------------------------------------|
| android:id | "@+id/text_message" |
| android:layout_width | wrap_content |
| android:layout_height | wrap_content |
| android:layout_below | "@+id/text_header" |
| android:layout_marginLeft | "@dimen/activity_horizontal_margin" |
| android:layout_marginStart | "@dimen/activity_horizontal_margin" |
| android:textSize | |

3. Hapus atribut android:text (jika ada).

Layout baru untuk aktivitas kedua terlihat sama dengan layout pada tugas sebelumnya karena TextView baru tidak berisi (belum) teks apa pun, sehingga tidak muncul di layar.

Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.SecondActivity">

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_header"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/text_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_header"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginStart="@dimen/activity_horizontal_margin"
        android:textAppearance="?android:attr/textAppearanceMedium" />

</RelativeLayout>
```

3.4 Memodifikasi aktivitas kedua untuk mendapatkan ekstra dan menampilkan pesan

1. Buka `java/com.example.android.twoactivities/SecondActivity`.
2. Dalam metode `onCreate()`, dapatkan intent yang mengaktifkan aktivitas ini:

```
Intent intent = getIntent();
```

3. Dapatkan string berisi pesan dari ekstra intent menggunakan variabel statis `MainActivity.EXTRA_MESSAGE` sebagai kuncinya:

```
String message =
    intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
```

4. Gunakan `findViewById` untuk mendapatkan referensi ke `TextView` untuk pesan dari layout (Anda mungkin perlu mengimpor kelas `TextView`):

```
TextView textView = (TextView) findViewById(R.id.text_message);
```

5. Setel teks `TextView` tersebut ke string dari ekstra intent:

```
textView.setText(message);
```

6. Jalankan aplikasi. Saat Anda mengetik pesan dalam aktivitas utama dan mengeklik Send, aktivitas kedua terbuka dan menampilkan pesan tersebut.

Kode Solusi:

```

package com.example.android.twoactivities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        Intent intent = getIntent();
        String message =
            intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
        TextView textView = (TextView) findViewById(R.id.text_message);
        textView.setText(message);
    }
}

```

Tugas 4. Mengembalikan data ke aktivitas utama

Sekarang karena Anda telah memiliki aplikasi yang meluncurkan aktivitas baru dan mengirimkan data ke aplikasi, langkah terakhir adalah mengembalikan data ke aktivitas utama. Anda juga akan menggunakan intent dan *ekstra* intent untuk tugas ini.

4.1 Menambahkan EditText dan Tombol ke layout aktivitas kedua

1. Salin EditText dan Tombol dari file layout aktivitas utama dan tempel ke layout kedua.
2. Dalam file activity_second.xml, modifikasi nilai atribut untuk Tombol dan tampilan EditText. Gunakan nilai ini:

| Atribut Lama (Tombol) | Atribut Baru (Tombol) |
|----------------------------------------------|------------------------------------------------|
| android:id="@+id/button_main" | android:id="@+id/button_second" |
| android:onClick= "launchSecondActivity" | android:onClick="returnReply" |
| android:text= "@string/button_main" | android:text= "@string/button_second" |
| Atribut Lama (EditText) | Atribut Baru (EditText) |
| android:id="@+id/editText_main" | android:id="@+id/editText_second" |
| android:layout_toLeftOf= "@+id/button_main" | android:layout_toLeftOf= "@+id/button_second" |
| android:layout_toStartOf= "@+id/button_main" | android:layout_toStartOf= "@+id/button_second" |
| android:hint= "@string/editText_main" | android:hint= "@string/editText_second" |

3. Buka `res/values/strings.xml` dan tambahkan sumber daya String untuk teks tombol dan petunjuknya di EditText:

```

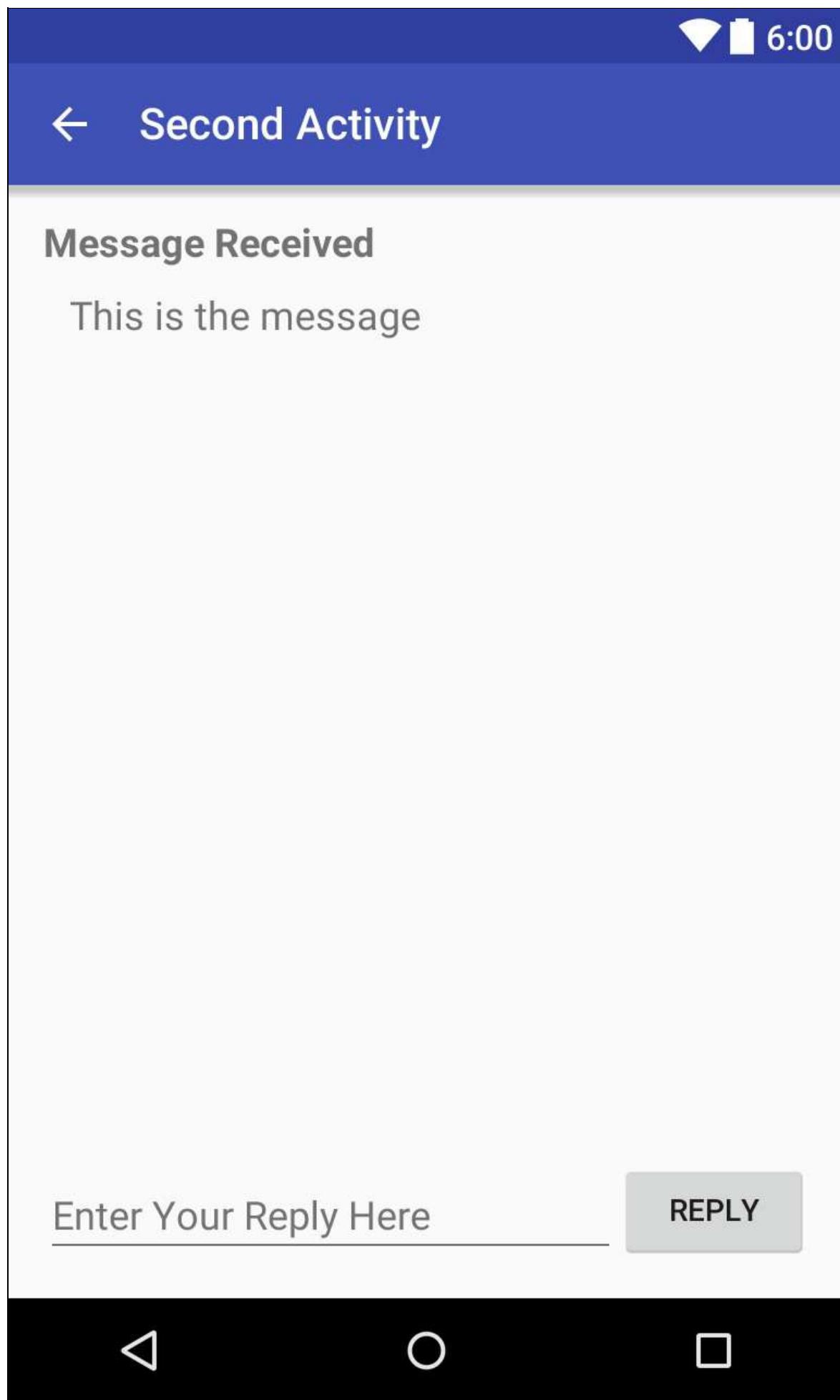
<string name="button_second">Reply</string>
<string name="editText_second">Enter Your Reply Here</string>

```

4. Dalam editor layout XML, letakkan kursor pada `"returnReply"`, tekan **Alt-Enter (Option-Enter di Mac)** dan pilih **Create 'launchSecondActivity(View)' dalam 'SecondActivity'**.

File SecondActivity.java terbuka, dan Android Studio membuat metode kerangka untuk handler onClick. Anda akan mengimplementasikan metode ini dalam tugas berikutnya.

Layout baru untuk aktivitas kedua terlihat seperti ini:



Kode Solusi:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.SecondActivity">

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_header"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/text_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_header"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginStart="@dimen/activity_horizontal_margin"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_second"
        android:id="@+id/button_second"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="returnReply"/>

    <EditText
        android:id="@+id/editText_second"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button_second"
        android:layout_toStartOf="@+id/button_second"
        android:hint="@string/editText_second" />

</RelativeLayout>

```

4.2 Membuat intent respons dalam aktivitas kedua

1. Buka `java/com.example.android.twoactivities/SecondActivity` .
2. Di atas kelas, tambahkan konstanta publik untuk mendefinisikan kunci untuk ekstra intent:

```

public static final String EXTRA_REPLY =
    "com.example.android.twoactivities.extra.REPLY";

```

3. Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText.

```

private EditText mReply;

```

4. Dalam metode `onCreate()`, gunakan `findViewById()` untuk mendapatkan referensi tentang contoh `EditText` dan menetapkannya ke variabel privat tersebut:

```
mReply = (EditText) findViewById(R.id.editText_second);
```

5. Dalam metode `returnReply()`, dapatkan teks EditText sebagai string:

```
String reply = mReply.getText().toString();
```

6. Membuat intent baru untuk respons tersebut.

Catatan: Jangan gunakan ulang objek intent yang diterima dari permintaan asalnya. Buat intent baru untuk respons tersebut.

```
Intent replyIntent = new Intent();
```

7. Tambahkan string balasan dari EditText ke intent baru sebagai *ekstra* intent. Karena *ekstra* adalah pasangan kunci-nilai, di sini kuncinya adalah EXTRA_REPLY dan nilainya adalah balasan:

```
replyIntent.putExtra(EXTRA_REPLY, reply);
```

8. Setel hasilnya ke `RESULT_OK` untuk menunjukkan bahwa responsnya berhasil. Kode hasil (termasuk `RESULT_OK` dan `RESULT_CANCELLED`) didefinisikan oleh kelas [Aktivitas](#).

```
setResult(RESULT_OK, replyIntent);
```

9. Panggil `finish()` untuk menutup aktivitas dan kembali ke aktivitas utama.

```
finish();
```

Kode Solusi:

```

package com.example.android.twoactivities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {
    public static final String EXTRA_REPLY =
        "com.example.android.twoactivities.extra.REPLY";

    private EditText mReply;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        mReply = (EditText) findViewById(R.id.editText_second);

        Intent intent = getIntent();
        String message =
            intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
        TextView textView = (TextView) findViewById(R.id.text_message);
        textView.setText(message);
    }

    public void returnReply(View view) {
        String reply = mReply.getText().toString();

        Intent replyIntent = new Intent();
        replyIntent.putExtra(EXTRA_REPLY, reply);
        setResult(RESULT_OK, replyIntent);
        finish();
    }
}

```

4.3 Menambahkan TextViews ke layout aktivitas utama untuk menampilkan balasan

Aktivitas utama memerlukan cara untuk menampilkan balasan yang dikirim kembali ke aktivitas kedua. Dalam tugas ini Anda akan menambahkan TextViews ke layout aktivitas utama untuk menampilkan balasan tersebut. Untuk memudahkannya, Anda dapat menyalin TextViews yang digunakan dalam aktivitas kedua.

1. Salin dua TextViews untuk tampilan pesan dari file layout aktivitas kedua dan tempel ke layout utama di atas EditText yang ada dan tampilan Tombol.
2. Modifikasi nilai atribut untuk kedua TextViews baru ini. Gunakan nilai ini:

| Atribut Lama (header TextView) | Atribut Baru (header TextView) |
|------------------------------------------|------------------------------------------------|
| android:id="@+id/text_header" | android:id="@+id/text_header_reply" |
| android:text="@string/text_header" | android:text= "@string/text_header_reply" |
| Atribut Lama (TextView pesan) | Atribut Baru (TextView pesan) |
| android:id="@+id/text_message" | android:id="@+id/text_message_reply" |
| android:layout_below= "@+id/text_header" | android:layout_below= "@+id/text_header_reply" |

3. Tambahkan atribut `android:visibility` ke setiap TextView agar tidak terlihat di awal. (Membuatnya terlihat di layar tanpa konten apa pun dapat membuat pengguna bingung.) Anda akan menjadikan TextView terlihat setelah data respons diteruskan kembali dari aktivitas kedua.

```
    android:visibility="invisible"
```

4. Buka res/values/strings.xml dan tambahkan sumber daya string untuk header balasan:

```
<string name="text_header_reply">Reply Received</string>
```

Layout untuk aktivitas utama terlihat sama dengan tugas sebelumnya, walaupun Anda sudah menambahkan dua TextView baru ke layout. Namun, karena Anda menyetel TextView menjadi tidak terlihat, TextView tidak muncul di layar.

Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.twoactivities.MainActivity">
    <TextView
        android:id="@+id/text_header_reply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_header_reply"
        android:visibility="invisible"
        android:layout_marginBottom="@dimen/activity_vertical_margin"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/text_message_reply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_header_reply"
        android:visibility="invisible"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginStart="@dimen/activity_horizontal_margin"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_main"
        android:id="@+id/button_main"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="launchSecondActivity"/>

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button_main"
        android:layout_toStartOf="@+id/button_main"
        android:hint="@string/editText_main" />
</RelativeLayout>
```

4.4 Mendapatkan balasan dari ekstra intent dan menampilkannya

Saat Anda menggunakan intent eksplisit untuk memulai aktivitas, Anda mungkin tidak berharap untuk mendapatkan data kembali, Anda hanya mengaktifkan aktivitas tersebut. Dalam kasus seperti ini, gunakan StartActivity() untuk memulai aktivitas baru, seperti yang Anda lakukan sebelumnya pada pelajaran ini. Tetapi jika ingin mendapatkan data kembali dari

aktivitas yang diaktifkan, Anda perlu memulainya dengan `startActivityForResult()`.

Dalam tugas ini Anda akan memodifikasi aplikasi untuk memulai aktivitas kedua dan mengharapkan hasil, untuk mengekstrak data pengembalian itu dari intent, dan untuk menampilkan data tersebut dalam `TextView` yang dibuat dalam tugas terakhir.

1. Buka `java/com.example.android.twoactivities/MainActivity` .
2. Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk tipe respons tertentu yang Anda inginkan:

```
public static final int TEXT_REQUEST = 1;
```

3. Tambahkan dua variabel privat untuk menampung header balasan dan `TextViews` balasan:

```
private TextView mReplyHeadTextView;  
private TextView mReplyTextView;
```

4. dalam metode `onCreate()`, gunakan `findViewById` untuk mendapatkan referensi dari layout ke header balasan dan `TextView` balasan. Tetapkan dua contoh tampilan ke variabel privat:

```
mReplyHeadTextView = (TextView) findViewById(R.id.text_header_reply);  
mReplyTextView = (TextView) findViewById(R.id.text_message_reply);
```

5. Dalam metode `launchSecondActivity()` , modifikasi panggilan ke `startActivity()` menjadi `startActivityForResult()` , dan sertakan kunci `TEXT_REQUEST` sebagai argumen:

```
startActivityForResult(intent, TEXT_REQUEST);
```

6. Buat metode callback `onActivityResult()` dengan tanda tangan ini:

```
public void onActivityResult(int requestCode, int resultCode,  
    Intent data) {}
```

7. Dalam `onActivityResult()`, panggil `super.onActivityResult()` :

```
super.onActivityResult(requestCode, resultCode, data);
```

8. Tambahkan kode untuk `TEXT_REQUEST` (untuk memproses hasil intent yang benar, jika ada beberapa) dan `RESULT_CODE` (untuk memastikan permintaannya sukses):

```
if (requestCode == TEXT_REQUEST) {  
    if (resultCode == RESULT_OK) {  
    }  
}
```

9. Di bagian dalam blok if, dapatkan ekstra intent dari intent respons (`data`). Di sini, kunci untuk ekstra adalah konstanta `EXTRA_REPLY` dari `SecondActivity`:

```
String reply = data.getStringExtra(SecondActivity.EXTRA_REPLY);
```

10. Setel visibilitas header balasan ke benar:

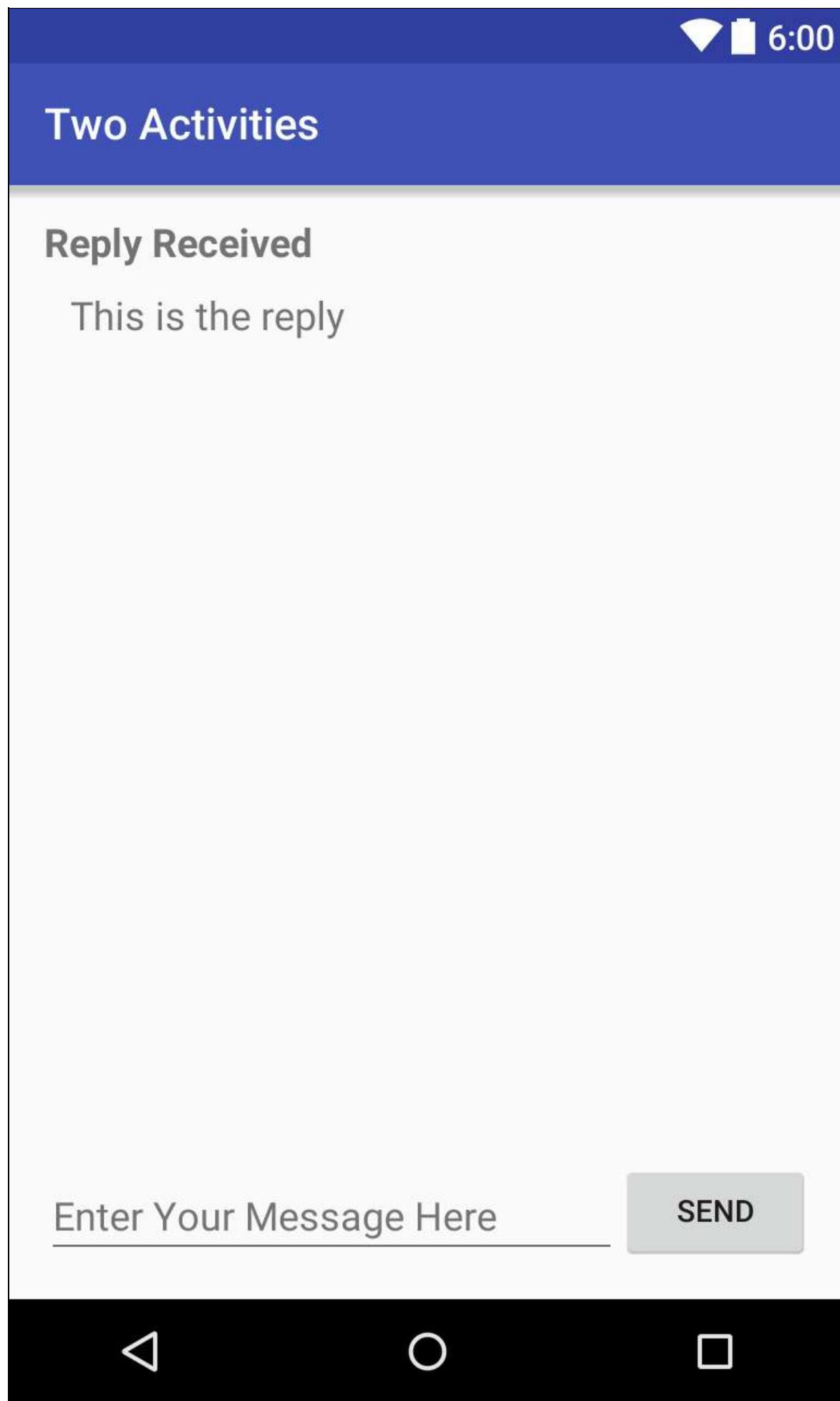
```
mReplyHeadTextView.setVisibility(View.VISIBLE);
```

11. Setel teks `textView` balasan ke balasan, dan setel visibilitasnya ke benar:

```
mReplyTextView.setText(reply);  
mReplyTextView.setVisibility(View.VISIBLE);
```

12. Jalankan aplikasi.

Sekarang, saat Anda mengirimkan pesan ke aktivitas kedua dan mendapatkan balasan kembali, aktivitas utama diperbarui untuk menampilkan balasan.



Kode Solusi:

```

package com.example.android.twoactivities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = MainActivity.class.getSimpleName();

    public static final String EXTRA_MESSAGE =
        "com.example.android.twoactivities.extra.MESSAGE";

    public static final int TEXT_REQUEST = 1;

    private EditText mMessageEditText;
    private TextView mReplyHeadTextView;
    private TextView mReplyTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mMessageEditText = (EditText) findViewById(R.id.editText_main);
        mReplyHeadTextView = (TextView) findViewById(R.id.text_header_reply);
        mReplyTextView = (TextView) findViewById(R.id.text_message_reply);
    }

    public void launchSecondActivity(View view) {
        Log.d(LOG_TAG, "Button clicked!");

        Intent intent = new Intent(this, SecondActivity.class);
        String message = mMessageEditText.getText().toString();

        intent.putExtra(EXTRA_MESSAGE, message);
        startActivityForResult(intent, TEXT_REQUEST);
    }

    public void onActivityResult(int requestCode, int resultCode,
                                 Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == TEXT_REQUEST) {
            if (resultCode == RESULT_OK) {
                String reply =
                    data.getStringExtra(SecondActivity.EXTRA_REPLY);

                mReplyHeadTextView.setVisibility(View.VISIBLE);
                mReplyTextView.setText(reply);
                mReplyTextView.setVisibility(View.VISIBLE);
            }
        }
    }
}

```

Kode Solusi

Proyek Android Studio: [TwoActivities](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Buat aplikasi dengan tiga tombol yang diberi label: Text One, Text Two, dan Text Three. Saat tombol mana saja diklik, aktivitas kedua terbuka. Aktivitas kedua itu harus berisi ScrollView yang menampilkan salah satu dari tiga bagian teks (Anda bisa menyertakan pilihan bagian). Gunakan intent untuk membuka aktivitas kedua dan ekstra intent untuk menunjukkan mana dari tiga bagian tersebut yang akan ditampilkan.

Rangkuman

Dalam praktik ini Anda telah belajar:

- Sebuah Aktivitas adalah komponen aplikasi yang menyediakan layar tunggal yang fokus pada tugas seorang pengguna.
 - Setiap aktivitas memiliki file layout antarmuka pengguna masing-masing.
 - Anda bisa menetapkan hubungan induk/anak pada aktivitas untuk mengaktifkan navigasi "ke atas" di dalam aplikasi.
 - Untuk mengimplementasikan aktivitas di dalam aplikasi, lakukan hal berikut:
- Buat kelas Java aktivitas.
- Implementasikan antarmuka pengguna untuk aktivitas itu.
- Deklarasikan aktivitas baru itu dalam manifes aplikasi.
- Saat membuat proyek baru untuk aplikasi, atau menambahkan aktivitas baru ke aplikasi, dalam Android Studio (dengan File > new > Activity), kode template untuk setiap tugas ini disediakan untuk Anda.
- Intent mengizinkan Anda meminta tindakan dari komponen lain dalam aplikasi, misalnya, untuk memulai satu aktivitas dari lainnya. Intent dapat menjadi eksplisit atau implisit.
 - Dengan intent eksplisit Anda mengindikasikan komponen target spesifik untuk menerima data.
 - Dengan intent implisit Anda menetapkan fungsionalitas yang Anda inginkan tetapi tidak dengan komponen target.
 - Intent dapat menyertakan data tempat melakukan tindakan (sebagai URI) atau informasi tambahan sebagai *ekstra* intent.
 - *Ekstra* intent adalah pasangan kunci-nilai dalam bundel yang dikirimkan bersama intent-nya.
- Tampilan dapat dijadikan terlihat atau tidak terlihat dengan atribut `android:visibility`

[Konsep terkait]

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Aktivitas dan Intent](#)

Ketahui selengkapnya

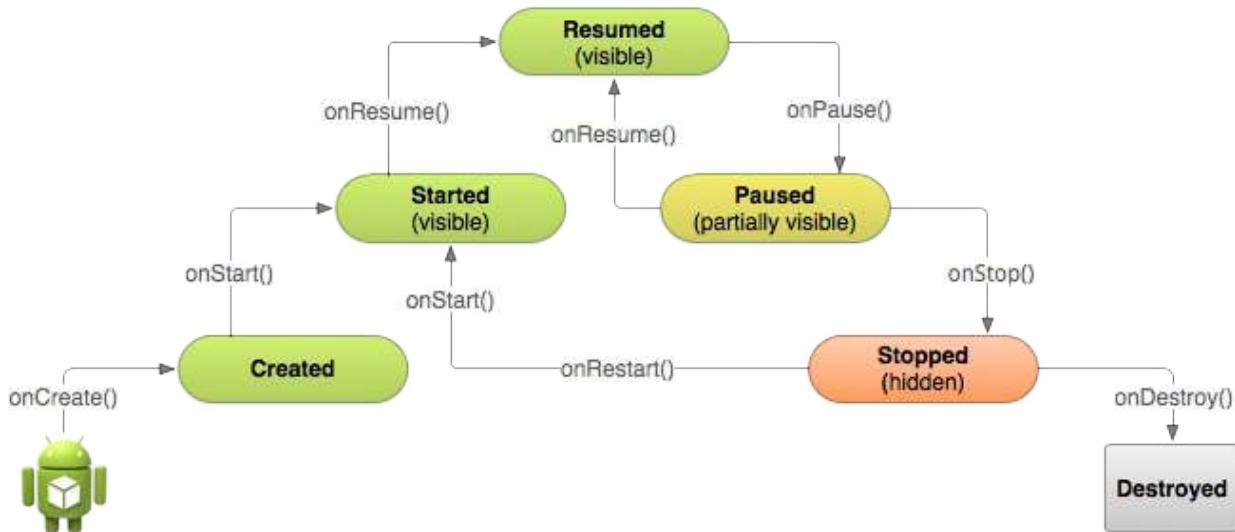
- [Dasar-Dasar Aplikasi Android](#)
- [Memulai Aktivitas Lain](#)
- [Aktivitas \(Panduan API\)](#)
- [Aktivitas \(Referensi API\)](#)
- [Intent Filter Intent API Guide \(Panduan API\)](#)
- [Intent \(Referensi API\)](#)

2.2: Siklus Hidup Aktivitas dan Status Instance

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1. Menambahkan callback siklus hidup ke TwoActivities
- Tugas 2. Menyimpan dan memulihkan aktivitas status
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Dalam praktik ini Anda akan belajar lebih banyak tentang *siklus hidup aktivitas*. Siklus hidup aktivitas adalah serangkaian status tempat aktivitas mungkin berada sepanjang daur hidupnya, mulai dari saat aktivitas pertama dibuat hingga dihancurkan dan sistem mengklaim kembali sumber daya aktivitas tersebut. Saat pengguna menelusuri di antara aktivitas dalam aplikasi (di dalam dan di luar aplikasi juga), masing-masing aktivitas tersebut bertransisi di antara status yang berbeda dalam siklus hidup aktivitas.



Setiap tahap dalam siklus hidup aktivitas memiliki metode callback yang cocok (`onCreate()`, `onStart()`, `onPause()`, dan lainnya). Saat sebuah aktivitas mengubah status, metode callback terkait akan dipanggil. Anda sudah melihat salah satu metode ini: `onCreate()`. Dengan mengganti metode callback siklus hidup dalam kelas aktivitas, Anda bisa mengubah perilaku default bagaimana aktivitas berperilaku untuk merespons pengguna atau tindakan sistem yang berbeda.

Perubahan pada status aktivitas juga dapat terjadi sebagai respons terhadap perubahan konfigurasi perangkat seperti memutar perangkat dari orientasi potret ke lanskap. Perubahan konfigurasi ini menyebabkan aktivitas dihancurkan dan sepenuhnya dibuat ulang dalam status default-nya, yang dapat menyebabkan hilangnya informasi yang telah dimasukkan pengguna ke dalam aktivitas tersebut. Penting untuk mengembangkan aplikasi untuk menghindarinya, agar pengguna tidak bingung. Nanti dalam praktik ini kita akan bereksperimen dengan perubahan konfigurasi dan belajar cara mempertahankan status aktivitas sebagai respons terhadap perubahan konfigurasi atau peristiwa siklus hidup Aktivitas lainnya.

Dalam praktik ini Anda akan menambahkan pernyataan pencatatan log ke aplikasi TwoActivities dan mengamati perubahan siklus hidup saat menggunakan aplikasi dengan berbagai cara. Lalu Anda akan mulai bekerja dengan perubahan tersebut dan mencari tahu cara menangani input pengguna dalam kondisi ini..

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Membuat dan menjalankan proyek aplikasi dalam Android Studio.
- Menambahkan pernyataan log ke aplikasi dan menampilkan log tersebut dalam Android Monitor (logcat).
- Memahami dan bekerja dengan aktivitas dan intent, dan nyaman berinteraksi dengan keduanya.

Yang akan Anda PELAJARI

Anda akan belajar:

- Memahami siklus hidup aktivitas dan kapan aktivitas dibuat, dijeda, dihentikan, dan dimusnahkan.
- Memahami metode callback siklus hidup yang terkait dengan perubahan aktivitas.
- Memahami efek tindakan seperti perubahan konfigurasi yang dapat menghasilkan peristiwa siklus hidup aktivitas.
- Mempertahankan status aktivitas di seluruh peristiwa siklus hidup.

Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Memperluas aplikasi TwoActivities dari praktik sebelumnya untuk mengimplementasikan berbagai callback siklus hidup aktivitas untuk menyertakan pernyataan pencatatan log.
- Mengamati perubahan status saat aplikasi berjalan dan Anda berinteraksi dengan aktivitas di dalam aplikasi.
- Memodifikasi aplikasi untuk mempertahankan status instance aktivitas yang tanpa terduga dibuat ulang sebagai respons terhadap perilaku pengguna atau perubahan konfigurasi pada perangkat.

Ringkasan Aplikasi

Untuk praktik ini Anda akan menambahkan ke aplikasi TwoActivities. Aplikasi ini terlihat dan berperilaku kurang lebih sama dengan yang ada pada bab terakhir: dengan dua aktivitas dan dua pesan yang bisa Anda kirimkan di antara keduanya. Perubahan yang Anda buat pada aplikasi dalam praktik ini tidak akan memengaruhi perilaku pengguna yang terlihat.

Tugas 1. Menambahkan Callback Siklus Hidup ke TwoActivities

Dalam tugas ini Anda akan mengimplementasikan semua metode callback siklus hidup aktivitas untuk mencetak pesan ke logcat saat metode itu dipanggil. Pesan log ini akan mengizinkan Anda melihat kapan status perubahan siklus hidup aktivitas dan bagaimana perubahan status siklus hidup memengaruhi aplikasi Anda saat aplikasi berjalan.

1.1 (Opsiional) Menyalin Proyek TwoActivities

Untuk tugas dalam praktik ini, Anda akan memodifikasi proyek [TwoActivities] yang sudah ada (<https://github.com/google-developer-training/android-fundamentals/tree/master/TwoActivities>) yang dibuat pada praktik sebelumnya. Jika memilih untuk menjaga proyek TwoActivities sebelumnya tetap utuh, ikuti langkah-langkah pada [Appendix](#) untuk membuat salinan proyek.

1.2 Mengimplementasikan callback ke MainActivity

1. Buka `java/com.example.android.twoactivities/MainActivity`.
2. Dalam metode `onCreate()`, tambahkan pernyataan log berikut:

```
Log.d(LOG_TAG, "-----");
Log.d(LOG_TAG, "onCreate");
```

3. Tambahkan metode baru untuk callback onStart(), dengan pernyataan ke log untuk peristiwa tersebut:

```
@Override
public void onStart(){
    super.onStart();
    Log.d(LOG_TAG, "onStart");
}
```

TIPS: Pilih **Code > Override Methods** dalam Android Studio. Dialog muncul dengan semua metode yang mungkin dan bisa Anda gantikan dalam kelas. Memilih satu atau beberapa metode callback dari daftar menyisipkan template lengkap untuk metode tersebut, termasuk panggilan wajib ke superkelas.

4. Gunakan metode onStart() sebagai template untuk mengimplementasikan callback siklus hidup lainnya:

- onPause()
- onRestart()
- onResume()
- onStop()
- onDestroy()

Semua metode callback memiliki tanda tangan yang sama (kecuali untuk nama). Jika Anda menyalin dan menempel onStart() untuk membuat metode callback lain, jangan lupa untuk memperbarui kontennya agar memanggil metode yang sama dalam superkelas dan mencatat log metode yang benar.

5. Bangun dan jalankan aplikasi Anda.

Kode Solusi (bukan seluruh kelas):

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Log.d(LOG_TAG, "-----");
    Log.d(LOG_TAG, "onCreate");

    mMessageEditText = (EditText) findViewById(R.id.editText_main);
    mReplyHeadTextView = (TextView) findViewById(R.id.text_header_reply);
    mReplyTextView = (TextView) findViewById(R.id.text_message_reply);
}

@Override
public void onStart(){
    super.onStart();
    Log.d(LOG_TAG, "onStart");
}

@Override
public void onRestart() {
    super.onRestart();
    Log.d(LOG_TAG, "onRestart");
}

@Override
public void onResume() {
    super.onResume();
    Log.d(LOG_TAG, "onResume");
}

@Override
public void onPause() {
    super.onPause();
    Log.d(LOG_TAG, "onPause");
}

@Override
public void onStop() {
    super.onStop();
    Log.d(LOG_TAG, "onStop");
}

@Override
public void onDestroy() {
    super.onDestroy();
    Log.d(LOG_TAG, "onDestroy");
}

```

1.3 Mengimplementasikan callback siklus hidup dalam SecondActivity

Karena sekarang Anda sudah mengimplementasikan metode callback siklus hidup untuk MainActivity, lakukan hal yang sama untuk SecondActivity.

1. Buka `java/com.example.android.twoactivities/SecondActivity`.
2. Di bagian atas kelas, tambahkan konstanta untuk variabel LOG_TAG:

```

private static final String LOG_TAG =
    SecondActivity.class.getSimpleName();

```

3. Tambahkan callback siklus hidup dan pernyataan log ke aktivitas kedua. (Anda juga bisa menyalin dan menempel metode callback dari MainActivity)
4. Tambahkan pernyataan log ke metode `returnReply()`, tepat sebelum metode `finish()`:

```

Log.d(LOG_TAG, "End SecondActivity");

```

Kode Solusi (bukan seluruh kelas):

```

private static final String LOG_TAG = SecondActivity.class.getSimpleName();

public void returnReply(View view) {
    String reply = mReply.getText().toString();

    Intent replyIntent = new Intent();
    replyIntent.putExtra(EXTRA_REPLY, reply);
    setResult(RESULT_OK, replyIntent);

    Log.d(LOG_TAG, "End SecondActivity");
    finish();
}

@Override
protected void onStart() {
    super.onStart();
    Log.d(LOG_TAG, "onStart");
}

@Override
public void onRestart() {
    super.onRestart();
    Log.d(LOG_TAG, "onRestart");
}

@Override
public void onResume() {
    super.onResume();
    Log.d(LOG_TAG, "onResume");
}

@Override
public void onPause() {
    super.onPause();
    Log.d(LOG_TAG, "onPause");
}

@Override
public void onStop() {
    super.onStop();
    Log.d(LOG_TAG, "onStop");
}

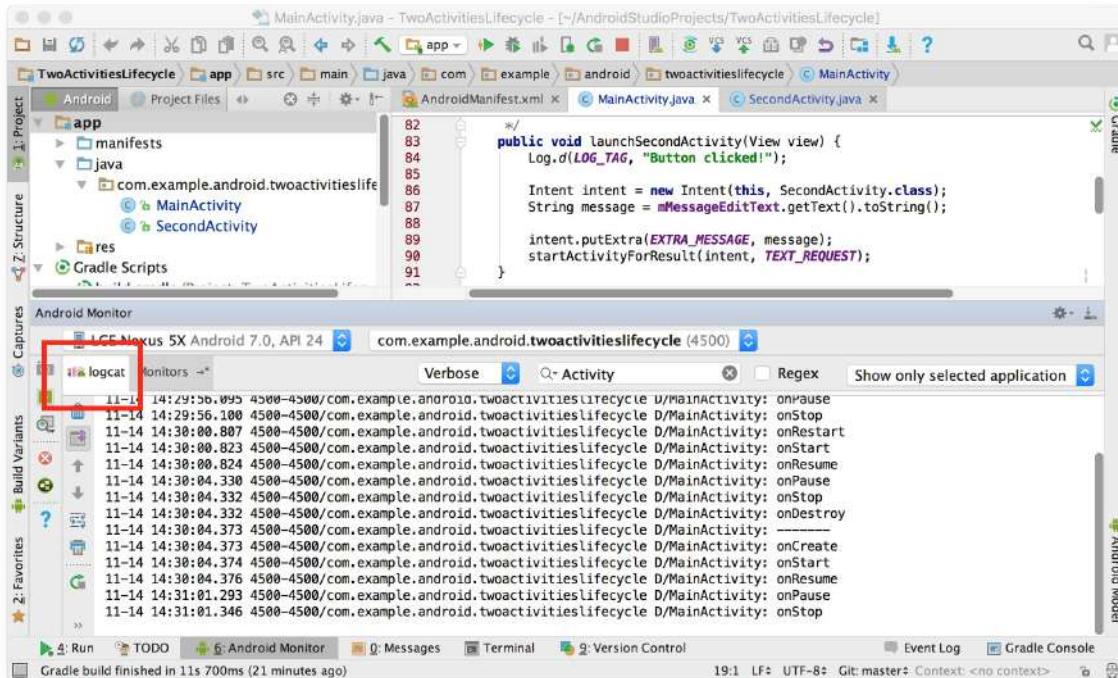
@Override
public void onDestroy() {
    super.onDestroy();
    Log.d(LOG_TAG, "onDestroy");
}

```

1.4 Amati log saat aplikasi berjalan

1. Jalankan aplikasi Anda.
2. Klik **Android Monitor** di bawah Android Studio untuk membuka Android Monitor.
3. Pilih tab logcat.
4. Ketik "Activity" dalam kotak pencarian Android Monitor.

Logcat Android bisa menjadi sangat panjang dan berantakan. Karena variabel LOG_TAG di setiap kelas berisi kata MainActivity atau SecondActivity, kata kunci ini memungkinkan Anda memfilter log hanya untuk hal-hal yang Anda minati.



5. Berekspresikanlah dengan aplikasi dan catat bahwa peristiwa siklus hidup terjadi untuk merespons tindakan yang berbeda. Khususnya, coba hal-hal berikut ini:

- Gunakan aplikasi dengan normal (mengirim pesan, membalas dengan pesan lain.)
- Gunakan tombol kembali untuk kembali dari aktivitas kedua ke aktivitas utama.
- Gunakan panah kiri pada bilah tindakan untuk kembali dari aktivitas kedua ke aktivitas utama.
- Putar perangkat pada aktivitas utama dan kedua pada waktu yang berlainan dalam aplikasi dan amati apa yang terjadi dalam log dan layar.**TIPS:**Jika menjalankan aplikasi dalam emulator, Anda bisa menyimulasikan putaran dengan Ctrl-F11 atau Ctrl-Fn-F11.
- Tekan tombol ringkasan (tombol kotak di sebelah kanan Beranda) dan tutup aplikasi (ketuk X).
- Kembali ke layar beranda dan mulai ulang aplikasi Anda.

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Amati onDestroy() pada khususnya. Mengapa onDestroy() kadang dipanggil (setelah mengeklik tombol kembali, atau saat perangkat diputar) dan bukan yang lainnya (menghentikan dan memulai ulang aplikasi secara manual)?

[Tugas 2. Menyimpan dan memulihkan status instance aktivitas.

Tergantung pada sumber daya sistem dan perilaku pengguna, aktivitas dalam aplikasi Anda dapat dimusnahkan dan dibuat ulang jauh lebih sering dari yang Anda pikirkan. Anda mungkin telah memperhatikan serangkaian aktivitas ini pada bagian sebelumnya saat memutar perangkat atau emulator. Memutar perangkat adalah salah satu contoh *perubahan konfigurasi*. Walaupun putaran adalah yang paling umum, semua hasil perubahan konfigurasi mengakibatkan aktivitas saat ini dimusnahkan dan dibuat ulang seperti baru. Jika Anda tidak mempertimbangkan perilaku ini dalam kode, saat perubahan konfigurasi terjadi layout aktivitas mungkin akan kembali ke tampilan default dan nilai awalnya, dan pengguna Anda bisa kehilangan tempat, data, atau status perkembangannya dalam aplikasi Anda.

Status setiap aktivitas disimpan sebagai serangkaian pasangan kunci/nilai dalam objek Bundel bernama *status instance aktivitas*. Sistem menyimpan informasi status default ke bundel status instance sebelum aktivitasnya dihentikan, dan meneruskan bundel itu ke instance aktivitas baru untuk disimpan.

Agar aktivitas tidak kehilangan data saat secara tidak terduga dimusnahkan dan dibuat ulang, Anda perlu mengimplementasikan metode `onSaveInstanceState()`. Sistem akan memanggil metode ini pada aktivitas Anda (di antara `onPause()` dan `onStop()`) saat ada kemungkinan aktivitas akan dimusnahkan dan dibuat ulang.

Data yang disimpan pada status instance spesifik untuk instance aktivitas spesifik ini saja selama sesi aplikasi saat ini. Saat Anda berhenti dan memulai ulang sesi aplikasi baru, status instance aktivitas hilang dan aktivitas Anda akan kembali ke tampilan default-nya. Jika Anda perlu menyimpan data pengguna antar sesi aplikasi, gunakan database atau preferensi bersama. Anda akan mempelajari tentang keduanya pada praktik berikutnya.

2.1 Simpan status instance aktivitas dengan `onSaveInstanceState()`

Anda mungkin telah memperhatikan bahwa memutar perangkat tidak memengaruhi status aktivitas kedua sama sekali. Ini karena layout dan status aktivitas kedua dihasilkan dari layout dan intent yang mengaktifkannya. Bahkan jika aktivitas tersebut dibuat ulang, intent-nya akan tetap ada di sana dan data di dalam intent tersebut masih digunakan setiap `onCreate()` aktivitas kedua dipanggil.

Selain itu, Anda mungkin memperhatikan bahwa dalam kedua aktivitas setiap teks yang Anda ketik ke dalam pesan atau balasan EditTexts dipertahankan bahkan ketika perangkat diputar. Ini karena informasi status beberapa tampilan dalam layout secara otomatis disimpan di semua perubahan konfigurasi, dan nilai saat ini EditText adalah salah satu kasus ini.

Satu-satunya status aktivitas yang harus diperhatikan adalah `TextView` untuk header balasan dan teks balasan dalam aktivitas utama. Kedua `TextView` secara default tidak terlihat, hanya muncul ketika Anda mengirimkan pesan kembali ke aktivitas utama dari aktivitas kedua.

Dalam tugas ini Anda akan menambahkan kode untuk mempertahankan status instance kedua `TextView` menggunakan `onSaveInstanceState()`.

1. Buka `java/com.example.android.twoactivities/MainActivity`.
2. Tambahkan implementasi skeleton `onSaveInstanceState()` ke aktivitas, atau gunakan **Code > Override Methods** untuk menyisipkan pengganti kerangka.

```
@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
}
```

3. Periksa untuk melihat apakah header saat ini terlihat, dan jika demikian letakkan status visibilitas ke dalam bundel status dengan metode `putBoolean()` dan kunci "reply_visible".

```
if (mReplyHeadTextView.getVisibility() == View.VISIBLE) {
    outState.putBoolean("reply_visible", true);
}
```

Ingin bahwa header dan teks balasan ditandai sebagai tidak terlihat sampai ada balasan dari aktivitas kedua. Jika header terlihat, maka ada data balasan yang perlu disimpan. Kita hanya berfokus pada status visibilitas -- teks header sebenarnya tidak perlu disimpan, karena teks tersebut tidak pernah berubah.

4. Dalam pemeriksaan yang sama, tambahkan teks balasan ke dalam bundel.

```
outState.putString("reply_text", mReplyTextView.getText().toString());
```

Jika header terlihat Anda bisa mengasumsikan bahwa pesan balasan juga terlihat. Anda tidak perlu menguji atau menyimpan status visibilitas pesan balasan saat ini. Hanya teks sebenarnya pesan yang masuk ke dalam bundel status dengan kunci "reply_text".

Kita hanya menyimpan tampilan yang dapat berubah setelah aktivitas dibuat.

Tampilan lain dalam aplikasi (EditText, Tombol) dapat dibuat ulang dari layout default kapan saja.

Catatan: Sistem akan menyimpan status beberapa tampilan, seperti konten EditText.

Kode Solusi (bukan seluruh kelas):

```
@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    // If the heading is visible, we have a message that needs to be saved.
    // Otherwise we're still using default layout.
    if (mReplyHeadTextView.getVisibility() == View.VISIBLE) {
        outState.putBoolean("reply_visible", true);
        outState.putString("reply_text", mReplyTextView.getText().toString());
    }
}
```

2.2. Memulihkan status instance aktivitas dalam onCreate()

Setelah menyimpan status instance aktivitas, Anda juga perlu memulihkannya saat aktivitas dibuat ulang. Anda bisa melakukannya dalam onCreate(), atau dengan mengimplementasikan callback onRestoreInstanceState(), yang dipanggil setelah onStart() setelah aktivitas dibuat.

Biasanya tempat terbaik untuk memulihkan status aktivitas adalah di onCreate(), untuk memastikan antarmuka pengguna termasuk statusnya tersedia secepat mungkin. Kadang lebih nyaman melakukannya dalam onRestoreInstanceState() setelah semua inisialisasi dilakukan, atau untuk mengizinkan subkelas memutuskan apakah akan menggunakan implementasi default Anda.

1. Dalam metode onCreate(), tambahkan tes untuk memastikan bundelnya tidak null.

```
if (savedInstanceState != null) {
```

Saat aktivitas dibuat, sistem meneruskan bundel status ke onCreate() karena itu hanya argumen. Pertama kali onCreate() dipanggil dan aplikasi Anda dimulai, bundelnya null, tidak ada status saat pertama kali aplikasi dimulai. Panggilan berikutnya ke onCreate() memiliki bundel yang diisi dengan data apa pun yang Anda simpan dalam onSaveInstanceState().

2. Dalam pemeriksaan tersebut, dapatkan visibilitas saat ini (benar atau salah) dari bundel dengan kunci "reply_visible"

```
if (savedInstanceState != null) {
    boolean isVisible =
        savedInstanceState.getBoolean("reply_visible");
}
```

3. Tambahkan tes di bawah baris sebelumnya untuk variabel isVisible.

```
if (isVisible) {
```

Jika ada kunci "reply_visible" dalam bundel status (maka isVisible benar), kita perlu memulihkan statusnya.

4. Dalam tes isVisible, buat header-nya terlihat.

```
mReplyHeadTextView.setVisibility(View.VISIBLE);
```

5. Dapatkan pesan balasan teks dari bundel dengan kunci "reply_text", dan setel TextView balasan untuk menunjukkan string tersebut.

```
mReplyTextView.setText(savedInstanceState.getString("reply_text"));
```

6. Jadikan TextView balasan terlihat juga:

```
mReplyTextView.setVisibility(View.VISIBLE);
```

7. Jalankan aplikasi. Coba putar perangkat atau emulator untuk memastikan bahwa pesan balasan (jika ada) tetap ada di layar setelah aktivitas dibuat ulang.

Kode Solusi (bukan seluruh kelas):

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Log.d(LOG_TAG, "-----");
    Log.d(LOG_TAG, "onCreate");

    // Initialize all the view variables.
    mMessageEditText = (EditText) findViewById(R.id.editText_main);
    mReplyHeadTextView = (TextView) findViewById(R.id.text_header_reply);
    mReplyTextView = (TextView) findViewById(R.id.text_message_reply);

    // Restore the saved state. See onSaveInstanceState() for what gets saved.
    if (savedInstanceState != null) {
        boolean isVisible = savedInstanceState.getBoolean("reply_visible");
        // Show both the header and the message views. If isVisible is
        // false or missing from the bundle, use the default layout.
        if (isVisible) {
            mReplyHeadTextView.setVisibility(View.VISIBLE);

            mReplyTextView.setText(savedInstanceState.getString("reply_text"));
            mReplyTextView.setVisibility(View.VISIBLE);
        }
    }
}
```

Kode Solusi

Proyek Android Studio: [Siklus Hidup TwoActivities](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Membuat aplikasi pembuat daftar belanja dengan dua aktivitas. Aktivitas utama berisi daftar itu sendiri, yang dibuat dari sepuluh tampilan teks (kosong). Tombol pada aktivitas utama dilabeli "Add Item" membuka aktivitas kedua yang berisi daftar barang belanjaan (Keju, Nasi, Apel, dan seterusnya). Gunakan Tombol untuk menampilkan item. Memilih item akan mengembalikan Anda ke aktivitas utama, dan memperbarui TextView kosong untuk menyertakan barang yang dipilih.

Gunakan intent untuk meneruskan informasi di antara dua aktivitas. Pastikan status daftar belanja saat ini disimpan saat Anda memutar perangkat.

Rangkuman

- Siklus hidup aktivitas adalah serangkaian status tempat aktivitas bermigrasi, yang dimulai saat aktivitas baru dibuat dan berakhir saat sistem Android mengklaim kembali sumber daya aktivitas tersebut.

- Saat pengguna beralih antara aktivitas, di dalam dan di luar aplikasi, setiap aktivitas bergerak di antara status dalam siklus hidup aktivitas.
- Setiap status dalam siklus hidup aktivitas memiliki metode callback yang sesuai dan dapat Anda ganti dalam kelas aktivitas. Metode siklus hidup tersebut adalah:
 - onCreate()
 - onStart()
 - onPause()
 - onRestart()
 - onResume()
 - onStop()
 - onDestroy()
- Mengganti metode callback siklus hidup mengizinkan Anda menambahkan perilaku yang terjadi saat aktivitas bertransisi ke status tersebut.
- Anda dapat menambahkan metode pengganti kerangka ke kelas dalam Android Studio dengan **Code > Override**.
- Perubahan konfigurasi perangkat seperti hasil putaran dalam aktivitas dimusnahkan dan dibuat ulang seperti baru.
- Porsi status aktivitas tersebut dipertahankan pada perubahan konfigurasi, termasuk nilai saat ini EditTexts. Untuk semua data lainnya, Anda harus secara eksplisit menyimpan data sendiri.
- Simpan status instance aktivitas dalam metode onSaveInstanceState().
- Data status instance disimpan sebagai pasangan kunci/nilai dalam Bundel. Gunakan metode Bundel untuk meletakkan data ke dalam dan mengeluarkan data dari bundel.
- Pulihkan status instance dalam onCreate(), cara yang sebaiknya dipilih, atau onRestoreInstanceState().

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Siklus Hidup Aktivitas dan Keadaan Menyimpan](#)

Ketahui selengkapnya

- [Aktivitas \(Panduan API\)](#)
- [Aktivitas \(Referensi API\)](#)
- [Mengelola Siklus Hidup Aktivitas](#)
- [Membuat Ulang Aktivitas](#)
- [Menangani Perubahan Waktu Proses](#)

2.3: Memulai Aktivitas dengan Intent Implisit

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1. Membuat proyek dan layout baru
- Tugas 2. Mengimplementasikan buka situs web
- Tugas 3. Mengimplementasikan buka lokasi
- Tugas 4. Mengimplementasikan bagikan teks ini
- Tugas 5. Menerima Intent Implisit
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Dalam bagian sebelumnya Anda belajar tentang intent *eksplisit* -- mengaktifkan aktivitas spesifik dalam aplikasi Anda atau aplikasi berbeda dengan mengirimkan intent dengan nama kelas mutlak dari aktivitas tersebut. Dalam bagian ini Anda akan mempelajari selengkapnya tentang intent *implisit* dan cara menggunakannya untuk mengaktifkan aktivitas juga.

Intent implisit memungkinkan Anda untuk mengaktifkan aktivitas jika tahu tindakannya, tetapi tidak tahu aplikasi spesifik atau aktivitas yang akan menangani tindakan tersebut. Contohnya, jika Anda ingin aplikasi mengambil foto, atau mengirim email, atau menampilkan lokasi pada peta, Anda biasanya tidak peduli aplikasi atau aktivitas spesifik mana yang melakukan tindakan ini.

Sebaliknya, aktivitas Anda dapat mendeklarasikan satu atau beberapa filter intent dalam manifes Android yang mengiklankan kemampuan aktivitas tersebut untuk menerima intent implisit dan mendefinisikan tipe intent tertentu yang akan diterimanya.

Untuk mencocokkan permintaan Anda dengan aplikasi spesifik yang dipasang pada perangkat, sistem Android mencocokkan intent implisit dengan aktivitas yang filter intent-nya menandakan bahwa intent itu dapat melakukan tindakan tersebut. Jika ada beberapa aplikasi terpasang yang cocok, pengguna akan diberikan pemilih aplikasi yang akan mengizinkan pengguna memilih aplikasi mana yang ingin digunakan untuk menangani intent tersebut.

Dalam praktik ini Anda akan membangun aplikasi yang mengirimkan tiga intent implisit untuk membuka URL pada browser web, untuk membuka lokasi pada peta dan membagikan sedikit teks. Berbagi -- mengirim informasi ke orang lain melalui email atau media sosial -- adalah fitur umum dan populer pada banyak aplikasi. Untuk tindakan berbagi kami menggunakan kelas ShareCompat.IntentBuilder, yang memudahkan pembangunan intent untuk membagikan data.

Akhirnya, kami membuat aplikasi penerima intent sederhana yang menerima intent implisit untuk tindakan yang spesifik.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Membuat dan menggunakan aktivitas.
- Membuat dan mengirim intent di antara aktivitas.

Yang akan Anda PELAJARI

Anda akan belajar:

- Membuat intent implisit dan menggunakan tindakan dan kategorinya.
- Menggunakan kelas pembantu ShareCompat.IntentBuilder untuk membuat intent implisit agar dapat membagikan data dengan mudah.
- Iklankan bahwa aplikasi Anda dapat menerima intent implisit dengan mendeklarasikan filter intent dalam manifest Android.

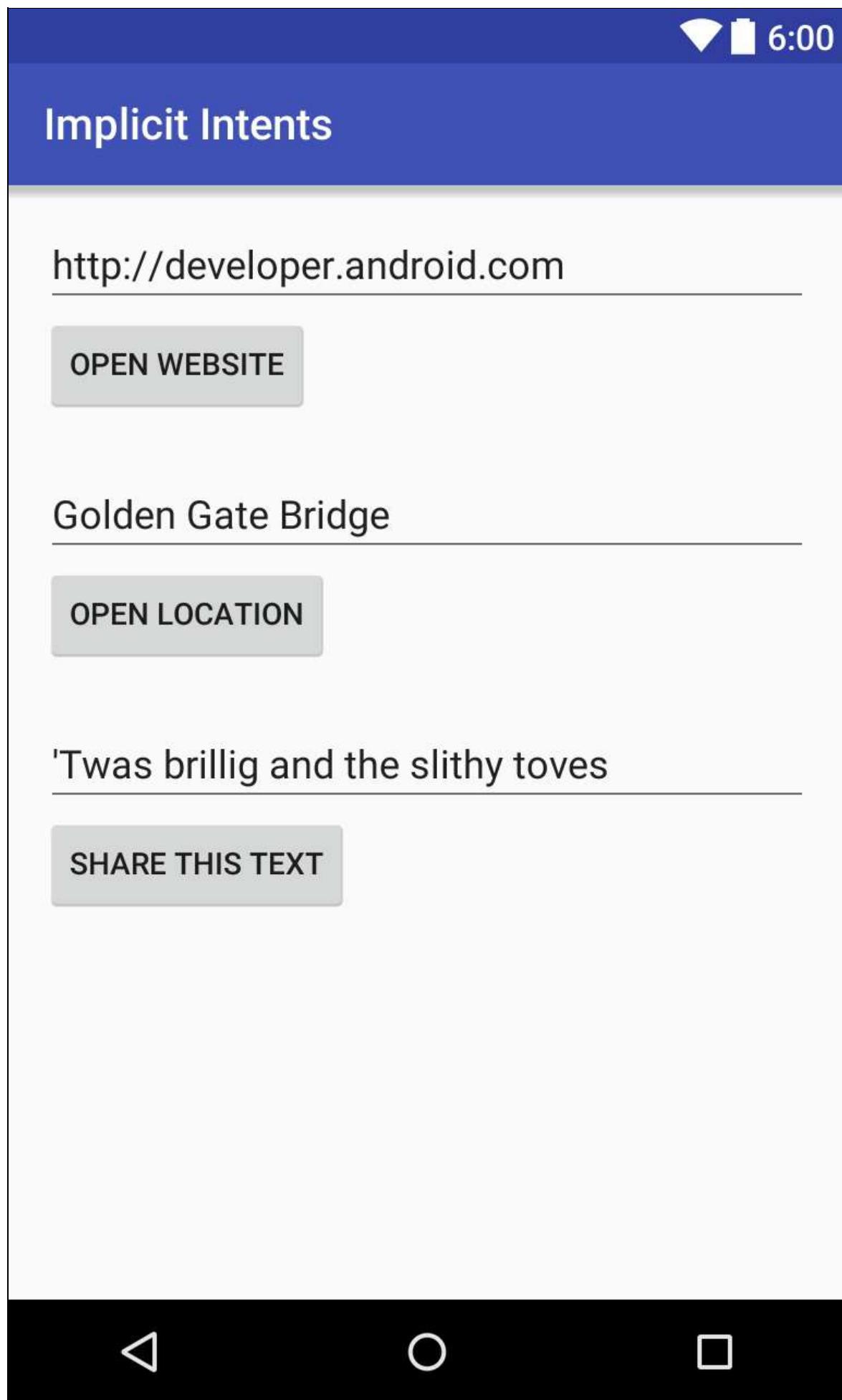
Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Membuat aplikasi baru untuk mengirimkan intent implisit.
- Mengimplementasikan dua intent implisit yang membuka halaman web dan membuka lokasi pada peta.
- Mengimplementasikan tindakan untuk membagikan cuplikan teks.
- Membuat aplikasi baru yang dapat menerima intent implisit untuk membuka laman web.

Ringkasan Aplikasi

Dalam bagian ini Anda akan membuat aplikasi baru dengan satu aktivitas dan tiga opsi untuk tindakan ini: membuka situs web, membuka lokasi pada peta, dan membagikan cuplikan teks. Semua bidang teks dapat diedit (EditText), tetapi berisi nilai default.



Tugas 1. Membuat proyek dan layout baru

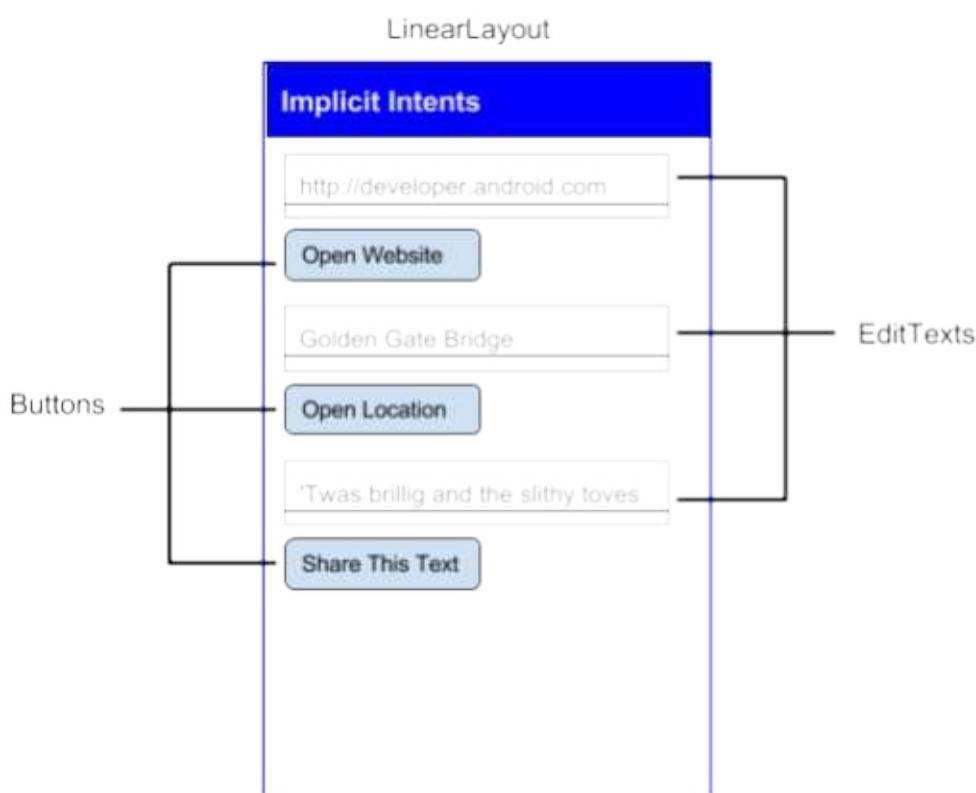
Untuk latihan ini, Anda akan membuat proyek baru dan aplikasi bernama Implicit Intents dengan layout baru.

1.1 Membuat proyek

1. Mulai Android Studio dan buat proyek Android Studio baru. Panggil "Implicit Intents" aplikasi.
2. Pilih **Empty Activity** untuk template proyek. Klik **Next**.
3. Terima nama aktivitas default (MainActivity). Pastikan kotak **Generate Layout file** dicentang. Klik **Finish**.

1.2 Membuat Layout

Dalam tugas ini, buat layout untuk aplikasi. Gunakan LinearLayout, tiga Tombol, dan tiga EditText, seperti ini:



1. Edit res/values/strings.xml untuk menyertakan sumber daya string ini:

```

<string name="edittext_uri">http://developer.android.com</string>
<string name="button_uri">Open Website</string>

<string name="edittext_loc">Golden Gate Bridge</string>
<string name="button_loc">Open Location</string>

<string name="edittext_share">'\Twas brillig and the slithy toves</string>
<string name="button_share">Share This Text</string>

```

2. Ubah RelativeLayout default ke LinearLayout. Tambahkan atribut android:orientation dan berikan nilai "vertical."

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.implicitintents.MainActivity"
    android:orientation="vertical">
```

3. Hapus TextView "Hello World".
4. Tambahkan EditText dan Tombol ke layout untuk fungsi Buka Situs web. Gunakan nilai atribut ini:

| Atribut (EditText) | Nilai (EditText) |
|-----------------------------|----------------------------|
| android:id | "@+id/website_edittext" |
| android:layout_width | "match_parent" |
| android:layout_height | "wrap_content" |
| android:text | "@string/edittext_uri" |
| Atribut (Tombol) | Nilai (Tombol) |
| android:id | "@+id/open_website_button" |
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:layout_marginBottom | "24dp" |
| android:text | "@string/button_uri" |
| android:onClick | "openWebsite" |

5. Tambahkan EditText kedua dan Tombol ke layout untuk fungsi Open Website.
6. Gunakan atribut yang sama dengan atribut pada langkah sebelumnya. Namun, modifikasi atribut seperti yang di bawah ini:

| Atribut (EditText) | Nilai (EditText) |
|--------------------|-----------------------------|
| android:id | "@+id/location_edittext" |
| android:text | "@string/edittext_loc" |
| Atribut (Tombol) | Nilai (Tombol) |
| android:id | "@+id/open_location_button" |
| android:text | "@string/button_loc" |
| android:onClick | "openLocation" |

7. Tambahkan EditText ketiga dan Tombol untuk fungsi Share This. Buatlah perubahan ini:

| Atribut (EditText) | Nilai (EditText) |
|--------------------|--------------------------|
| android:id | "@+id/share_edittext" |
| android:text | "@string/edittext_share" |
| Atribut (Tombol) | Nilai (Tombol) |
| android:id | "@+id/share_text_button" |
| android:text | "@string/button_share" |
| android:onClick | "shareText" |

Kode Solusi:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.implicitintents.MainActivity"
    android:orientation="vertical">

    <EditText
        android:id="@+id/website_edittext"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/edittext_uri" />

    <Button
        android:id="@+id/open_website_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:onClick="openWebsite"
        android:text="@string/button_uri" />

    <EditText
        android:id="@+id/location_edittext"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/edittext_loc" />

    <Button
        android:id="@+id/open_location_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:onClick="openLocation"
        android:text="@string/button_loc" />

    <EditText
        android:id="@+id/share_edittext"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/edittext_share" />

    <Button
        android:id="@+id/share_text_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:onClick="shareText"
        android:text="@string/button_share" />
</LinearLayout>
```

Tugas 2. Mengimplementasikan "buka situs web"

Dalam tugas ini Anda akan mengimplementasikan metode handler on-click untuk tombol pertama dalam layout ("Open Website.") Tindakan ini menggunakan intent implisit untuk mengirimkan URI yang diberikan ke aktivitas yang dapat menangani intent implisit tersebut (seperti browser web).

2.1 Mendefinisikan metode openWebsite

1. Buka MainActivity.java.
2. Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText untuk URI situs web.

```
private EditText mWebsiteEditText;
```

3. Dalam metode onCreate(), gunakan findViewById() untuk mendapatkan referensi tentang instance EditText dan menetapkannya ke variabel privat tersebut:

```
mWebsiteEditText = (EditText) findViewById(R.id.website_edittext);
```

4. Buat metode baru bernama openWebsite(), dengan tanda tangan ini:

```
public void openWebsite(View view) { }
```

5. Dapatkan nilai string EditText:

```
String url = mWebsiteEditText.getText().toString();
```

6. Enkode dan parse string ke dalam objek Uri:

```
Uri webpage = Uri.parse(url);
```

7. Buat intent baru dengan Intent.ACTION_VIEW sebagai tindakan dan URI sebagai data:

```
Intent intent = new Intent(Intent.ACTION_VIEW, webpage);
```

Konstruktor intent ini berbeda dari yang Anda gunakan untuk membuat intent eksplisit. Dalam konstruktor sebelumnya, Anda menetapkan konteks saat ini dan komponen spesifik (kelas aktivitas) untuk mengirim intent. Dalam konstruktor ini Anda menetapkan tindakan dan data untuk tindakan tersebut. Tindakan didefinisikan oleh kelas intent dan dapat menyertakan ACTION_VIEW (untuk melihat data yang diberikan), ACTION_EDIT (untuk mengedit data yang diberikan), atau ACTION_DIAL (untuk menghubungi nomor telepon). Dalam hal ini, tindakan adalah ACTION_VIEW karena kita ingin membuka dan melihat laman web yang ditetapkan oleh URI dalam variabel laman web.

8. Gunakan resolveActivity() dan pengelola paket Android untuk menemukan aktivitas yang dapat menangani intent implisit. Periksa untuk memastikan permintaan berhasil diatasi.

```
if (intent.resolveActivity(getApplicationContext()) != null) { }
```

Permintaan yang cocok dengan tindakan intent dan data dengan intent Anda ini memfilter aplikasi yang dipasang pada perangkat untuk memastikan paling tidak ada satu aktivitas yang bisa menangani permintaan Anda.

9. Dalam pernyataan if, panggil startActivity() untuk mengirimkan intent.

```
startActivity(intent);
```

10. Tambahkan blok else untuk mencetak pesan log jika intent tidak bisa diatasi.

```

} else {
    Log.d("ImplicitIntents", "Can't handle this!");
}

```

Kode Solusi (bukan seluruh kelas):

```

public void openWebsite(View view) {
    // Get the URL text.
    String url = mWebsiteEditText.getText().toString();

    // Parse the URI and create the intent.
    Uri webpage = Uri.parse(url);
    Intent intent = new Intent(Intent.ACTION_VIEW, webpage);

    // Find an activity to hand the intent and start that activity.
    if (intent.resolveActivity(getApplicationContext()) != null) {
        startActivity(intent);
    } else {
        Log.d("ImplicitIntents", "Can't handle this intent!");
    }
}

```

Tugas 3. Mengimplementasikan "buka lokasi"

Dalam tugas ini, Anda akan mengimplementasikan metode handler on-click untuk tombol kedua dalam UI("Open Location.") Metode ini hampir identik dengan metode openWebsite(). Perbedaannya adalah penggunaan URI geo untuk menunjukkan lokasi peta. Anda bisa menggunakan URI geo dengan lintang dan bujur, atau gunakan string kueri untuk lokasi umum. Dalam contoh ini kami telah menggunakanya untuk tujuan yang kedua.

3.1 Definisikan metode openLocation

1. Buka MainActivity.java (java/com.example.android.implicitintents/MainActivity).
2. Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText bagi URI lokasi.

```
private EditText mLocationEditText;
```

3. Dalam metode onCreate(), gunakan findViewById() untuk mendapatkan referensi tentang contoh EditText dan menetapkannya ke variabel privat tersebut:

```
mLocationEditText = (EditText) findViewById(R.id.location_edittext);
```

4. Membuat metode baru bernama openLocation untuk digunakan sebagai metode onClick untuk tombol Open Location. Gunakan tanda tangan metode yang sama sebagai openWebsite().
5. Dapatkan nilai string EditText mLocationEditText.

```
String loc = mLocationEditText.getText().toString();
```

6. Parse string itu ke dalam objek Uri dengan kueri penelusuran geo:

```
Uri addressUri = Uri.parse("geo:0,0?q=" + loc);
```

7. Buat intent baru dengan Intent.ACTION_VIEW sebagai tindakan dan loc sebagai datanya.

```
Intent intent = new Intent(Intent.ACTION_VIEW, addressUri);
```

8. Pecahkan intent dan periksa untuk memastikan intent berhasil diatasi. Jika demikian, startActivity(), jika tidak catat log pesan kesalahan.

```

if (intent.resolveActivity(getApplicationContext()) != null) {
    startActivity(intent);
} else {
    Log.d("ImplicitIntents", "Can't handle this intent!");
}

```

Kode Solusi (bukan seluruh kelas):

```

public void openLocation(View view) {
    // Get the string indicating a location. Input is not validated; it is
    // passed to the location handler intact.
    String loc = mLocationEditText.getText().toString();

    // Parse the location and create the intent.
    Uri addressUri = Uri.parse("geo:0,0?q=" + loc);
    Intent intent = new Intent(Intent.ACTION_VIEW, addressUri);

    // Find an activity to handle the intent, and start that activity.
    if (intent.resolveActivity(getApplicationContext()) != null) {
        startActivity(intent);
    } else {
        Log.d("ImplicitIntents", "Can't handle this intent!");
    }
}

```

Tugas 4. Mengimplementasikan bagikan teks ini

Tindakan berbagi adalah cara mudah bagi pengguna untuk berbagi item dalam aplikasi Anda dengan jaringan sosial dan aplikasi lain. Walaupun Anda bisa membangun tindakan berbagi dalam aplikasi Anda sendiri menggunakan intent implisit, Android menyediakan kelas pembantu [ShareCompat.IntentBuilder](#) untuk memudahkan implementasi berbagi. Anda bisa menggunakan ShareCompat.IntentBuilder untuk membangun intent dan meluncurkan pemilih untuk memungkinkan pengguna memilih aplikasi tujuan untuk berbagi.

Dalam tugas terakhir ini, kita akan mengimplementasikan sedikit teks dalam text edit dengan kelas ShareCompat.IntentBuilder.

4.1 Mengimplementasikan metode shareText

1. Buka MainActivity.java.
2. Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText untuk URI situs web.

```
private EditText mShareTextEdit;
```

3. Dalam metode onCreate(), gunakan findViewById() untuk mendapatkan referensi tentang instance EditText dan menetapkannya ke variabel privat tersebut:

```
mShareTextEdit = (EditText) findViewById(R.id.share_edittext);
```

4. Membuat metode baru bernama shareThis() untuk digunakan sebagai metode onClick untuk tombol This Text. Gunakan tanda tangan metode yang sama sebagai openWebsite().
5. Dapatkan nilai string EditText mShareTextEdit .

```
String txt = mShareTextEdit.getText().toString();
```

6. Definisikan tipe mime teks untuk dibagikan:

```
String mimeType = "text/plain";
```

7. Panggil ShareCompat.IntentBuilder dengan metode ini:

```
ShareCompat.IntentBuilder
    .from(this)
    .setType(mimeType)
    .setChooserTitle("Share this text with: ")
    .setText(txt)
    .startChooser();
```

Panggilan ke `hareCompat.IntentBuilder` ini menggunakan metode berikut:

</tr> </table> Format ini, dengan semua metode setter builder dirangkai dalam satu pernyataan, adalah cara shorthand mudah untuk membuat dan meluncurkan intent. Anda bisa menambahkan metode tambahan apa pun ke daftar ini.

| Metode | Keterangan |
|--------------------------------|--------------------------------------------------------|
| <code>from()</code> | Aktivitas yang meluncurkan intent berbagi (ini). |
| <code>setType()</code> | Tipe MIME item yang akan dibagikan. |
| <code>setChooserTitle()</code> | Judul yang muncul pada pemilih aplikasi sistem. |
| <code>setText()</code> | Teks sebenarnya yang akan dibagikan |
| <code>startChooser()</code> | Tunjukkan pemilih aplikasi sistem dan kirimkan intent. |

Kode Solusi (bukan seluruh kelas):

```
public void shareText(View view) {
    String txt = mShareTextEdit.getText().toString();
    String mimeType = "text/plain";

    ShareCompat.IntentBuilder
        .from(this)
        .setType(mimeType)
        .setChooserTitle("Share this text with: ")
        .setText(txt)
        .startChooser();
}
```

Tugas 5. Menerima Intent Implisit

Sejauh ini, Anda sudah membuat aplikasi yang menggunakan intent eksplisit dan implisit untuk membuka beberapa aktivitas aplikasi lainnya. Dalam tugas ini kita akan melihat masalah dari sudut pandang sebaliknya: mengizinkan aktivitas dalam aplikasi untuk merespons intent implisit yang dikirim dari aplikasi lain.

Aktivitas dalam aplikasi Anda selalu bisa diaktifkan dari dalam atau luar aplikasi dengan intent eksplisit. Untuk mengizinkan aktivitas menerima intent implisit, definisikan *filter intent* dalam manifes Anda untuk menandakan intent implisit mana yang akan ditangani oleh aktivitas.

Untuk mencocokkan permintaan Anda dengan aplikasi spesifik yang dipasang pada perangkat, sistem Android mencocokkan intent implisit dengan aktivitas yang filter intent-nya menandakan bahwa intent itu dapat melakukan tindakan tersebut. Jika ada beberapa aplikasi terpasang yang cocok, pengguna akan diberikan pemilih aplikasi yang akan mengizinkan pengguna memilih aplikasi mana yang ingin digunakan untuk menangani intent tersebut.

Saat aplikasi pada perangkat mengirimkan intent implisit, sistem Android mencocokkan tindakan dan data intent tersebut dengan aktivitas yang tersedia, termasuk filter intent yang benar. Jika filter intent aktivitas cocok dengan intent, aktivitas dapat menangani intent itu sendiri (jika itu satu-satunya aktivitas yang cocok), atau (jika ada beberapa yang cocok) sebuah pemilih aplikasi akan muncul dan mengizinkan pengguna memilih aplikasi mana yang mereka inginkan untuk melakukan tindakan itu.

Dalam tugas ini Anda akan membuat aplikasi sangat sederhana yang menerima intent implisit untuk membuka URI untuk laman web. Saat diaktifkan oleh intent implisit, aplikasi itu menampilkan URI yang diminta sebagai string dalam TextView.

5.1 Membuat Proyek & Layout

1. Mulai Android Studio dan buat proyek Android Studio baru.
2. Panggil "Penerima Intent Implisit" aplikasi Anda.
3. Pilih **Empty Activity** untuk template proyek.
4. Terima nama aktivitas default (MainActivity). Klik **Berikutnya**.
5. Pastikan kotak **Generate Layout file** dicentang. Klik **Finish**.
6. Buka `res/layout/activity_main.xml`.
7. Ubah TextView ("Hello World") yang sudah ada dengan atribut berikut:

| Atribut | Nilai |
|------------------------------------|--------------------------------------|
| <code>android:id</code> | <code>"@+id/text_uri_message"</code> |
| <code>android:layout_width</code> | <code>wrap_content</code> |
| <code>android:layout_height</code> | <code>wrap_content</code> |
| <code>android:textSize</code> | <code>"18sp"</code> |
| <code>android:textStyle</code> | <code>"bold"</code> |

8. Hapus atribut `android:text`. Tidak ada teks dalam TextView ini secara default, Anda akan menambahkan URI dari intent dalam `onCreate()`.

5.2 Memodifikasi Manifes Android untuk menambahkan filter intent

1. Buka `manifests/AndroidManifest.xml`.
2. Perhatikan bahwa aktivitas utama telah memiliki filter intent ini:

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Filter intent ini, yang merupakan bagian dari manifes proyek default, menunjukkan bahwa aktivitas ini adalah titik masuk utama untuk aplikasi Anda (filter ini memiliki tindakan intent "android.intent.action.MAIN"), dan bahwa aktivitas ini seharusnya muncul sebagai item level teratas dalam peluncur (kategorinya adalah `"android.intent.category.LAUNCHER"`)

3. Tambahkan `<intent-filter>` tag inside `<activity>`, dan sertakan elemen-elemen berikut ini:

```
<action android:name="android.intent.action.VIEW" />
<category android:name="android.intent.category.DEFAULT" />
<category android:name="android.intent.category.BROWSABLE" />
<data android:scheme="http" android:host="developer.android.com" />
```

Baris ini mendefinisikan filter intent untuk aktivitas tersebut, jenis intent yang dapat ditangani aktivitas. Filter intent mendeklarasikan elemen berikut:

| Jenis Filter | Nilai | Kecocokan |
|--------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| tindakan | "android.intent.action.VIEW" | Semua intent dengan tindakan tampilan. |
| kategori | "android.intent.category.DEFAULT" | Semua intent implisit. Kategori ini harus disertakan agar aktivitas Anda menerima intent implisit apa pun . |
| kategori | "android.intent.category.BROWSABLE" | Permintaan untuk tautan yang dapat dijelajahi dari laman web, email, atau sumber lainnya. |
| data | android:scheme="http" android:host="developer.android.com" | URI yang berisi skema http DAN hostname dari developer.android.com. |

Catat bahwa filter (<https://developer.android.com/guide/topics/manifest/data-element.html>) [data] memiliki batasan untuk jenis tautan yang akan diterima dan hostname untuk URI tersebut. Jika Anda memilih agar penerima dapat menerima tautan apa pun, Anda bisa meninggalkan `<data>` elemen semuanya.

Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.implicitintentsreceiver">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />
                <data android:scheme="http"
                    android:host="developer.android.com" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

5.3 Memproses Intent

Dalam metode `onCreate()` untuk aktivitas, Anda memproses intent yang masuk untuk data atau ekstra yang disertakan. Dalam hal ini, intent implisit yang masuk memiliki URI yang disimpan dalam data intent.

1. Buka `MainActivity.java`.
2. Dalam metode `onCreate()`, dapatkan intent yang datang dan digunakan untuk mengaktifkan aktivitas tersebut:

```
Intent intent = getIntent();
```

3. Dapatkan data intent. Data intent selalu objek URI:

```
Uri uri = intent.getData();
```

4. Periksa untuk memastikan variabel `uri` tidak null. Jika pemeriksaan lulus, buat string dari objek URI:

```
if (uri != null) {
    String uri_string = "URI: " + uri.toString();
}
```

5. Di dalam blok if yang sama, dapatkan tampilan teks untuk pesannya:

```
TextView textView = (TextView) findViewById(R.id.text_uri_message);
```

6. Juga di dalam blok if, setel teks TextView itu ke URI:

```
textView.setText(uri_string);
```

7. Jalankan aplikasi penerima.

Menjalankan aplikasi itu sendiri menunjukkan aktivitas kosong tanpa teks. Ini karena aktivitas diaktifkan dari peluncur sistem, dan tidak dengan intent dari aplikasi lain.

8. Jalankan aplikasi ImplicitIntents, dan klik Open Website dengan URI default.

Pemilih aplikasi muncul menanyakan apakah Anda ingin menggunakan browser default atau aplikasi ImplicitIntentsReceiver. Pilih "Just Once" untuk aplikasi penerima. Aplikasi ImplicitIntentsReceiver terbuka dan pesan menunjukkan URI dari permintaan yang asli.

9. Tap tombol kembali dan masukkan URI yang berbeda. Klik Open Website.

Aplikasi penerima memiliki filter intent yang hanya cocok dengan protokol URI yang sama persis (http) and host (developer.android.com). URI lainnya terbuka pada browser web default.

Kode Solusi

Proyek Android Studio: [ImplicitIntents](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Dalam tantangan bagian terakhir Anda membuat builder aplikasi daftar belanja dengan dua aktivitas: satu untuk menampilkan daftar dan yang satu lagi untuk memilih item. Tambahkan EditText dan Tombol ke aktivitas daftar belanja untuk menemukan toko tertentu di peta.

Rangkuman

- Intent implisit memungkinkan Anda untuk mengaktifkan aktivitas jika tahu tindakannya, tetapi tidak tahu aplikasi spesifik atau aktivitas yang akan menangani tindakan tersebut.
- Aktivitas yang dapat menerima intent implisit harus mendefinisikan filter intent dalam manifes Android yang cocok dengan satu atau beberapa tindakan intent dan kategori.
- Sistem Android cocok dengan konten intent implisit dan filter intent semua aktivitas yang tersedia untuk menentukan aktivitas mana yang akan diaktifkan. Jika ada lebih dari satu aktivitas yang tersedia, sistem menyediakan pemilih agar pengguna dapat memilih salah satunya.
- Kelas ShareCompat.IntentBuilder memudahkan pembuatan intent implisit untuk berbagi data ke media sosial atau email.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Aktivitas dan Intent Implisit](#)

Ketahui Selengkapnya

- [Aktivitas](#) (Panduan API)
- [Aktivitas](#) (Referensi API)
- [Intent Filter Intent API Guide](#) (Panduan API)
- [Intent](#) (Referensi API)
- [Uri](#)
- [Intent Google Maps](#)
- [ShareCompat.IntentBuilder](#) (Referensi API)

3.1 P: Menggunakan Debugger

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1. Membuat Proyek dan Aplikasi SimpleCalc](#)
- [Tugas 2. Menjalankan SimpleCalc di Debugger](#)
- [Tugas 3: Menjelajahi Fitur Debugger](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Pada praktik sebelumnya Anda menggunakan kelas Log untuk mencetak informasi ke log sistem (logcat) saat aplikasi dijalankan. Menambahkan pernyataan catatan log ke aplikasi adalah salah satu cara untuk menemukan kesalahan dan meningkatkan operasi aplikasi. Cara lainnya adalah dengan menggunakan debugger yang sudah disediakan oleh Android Studio.

Dalam praktik ini Anda akan mempelajari cara men-debug aplikasi dalam emulator dan pada perangkat, menyetel dan melihat breakpoint, menyusuri kode, dan memeriksa variabel.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Membuat proyek Android Studio, bekerja dengan tampilan EditText dan Tombol.
- Membuat dan menjalankan aplikasi di Android Studio, pada emulator dan perangkat.
- Membaca dan menganalisis pelacakan tumpukan, termasuk yang pertama menyala dan pertama mati.
- Menambahkan pernyataan log dan melihat log sistem (logcat) dalam Android Monitor.

Yang akan Anda PELAJARI

Anda akan belajar:

- Menjalankan aplikasi dalam mode debug di emulator atau perangkat.
- Menyusuri eksekusi aplikasi.
- Menyetel dan mengatur breakpoint.
- Memeriksa dan memodifikasi variabel dalam debugger.

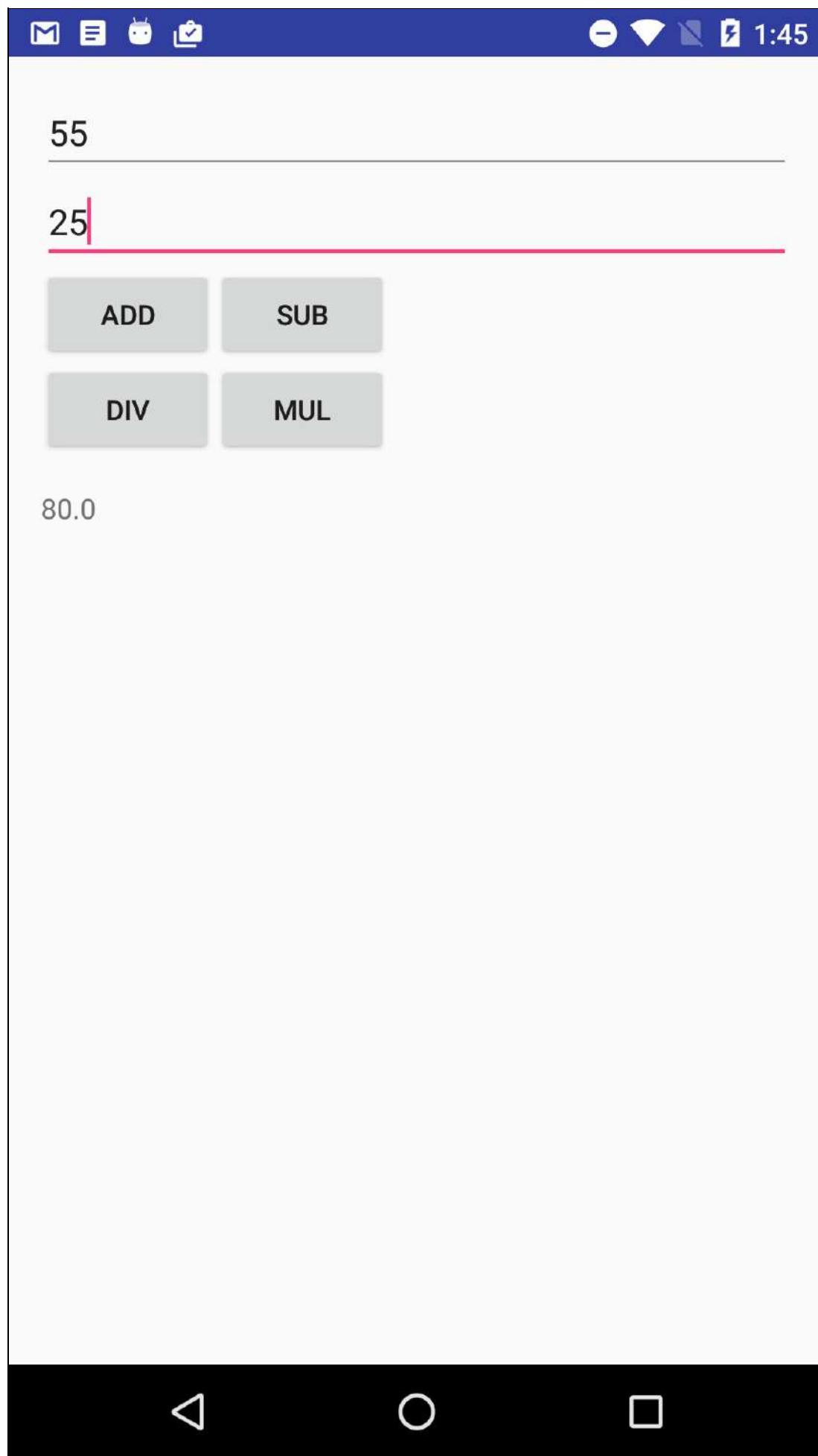
Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Membangun aplikasi SimpleCalc.
- Menyetel dan melihat breakpoints dalam kode untuk SimpleCalc.
- Menyusuri kode saat aplikasi berjalan.
- Memeriksa variabel dan mengevaluasi ekspresi.
- Mengidentifikasi dan memperbaiki masalah dalam aplikasi contoh.

Ringkasan Aplikasi

Aplikasi SimpleCalc memiliki dua teks edit dan empat tombol. Saat Anda memasukkan dua angka dan meneklik tombol, aplikasi melakukan penghitungan untuk tombol tersebut dan menampilkan hasilnya.



Tugas 1. Membuat Proyek dan Aplikasi SimpleCalc

Untuk praktik ini Anda tidak akan membangun aplikasi SimpleCalc sendiri. Proyek selengkapnya tersedia di [SimpleCalc](#). Dalam tugas ini Anda akan membuka proyek SimpleCalc ke dalam Android Studio dan menjelajahi beberapa fitur utama aplikasinya.

1.1 Unduh dan buka proyek SimpleCalc

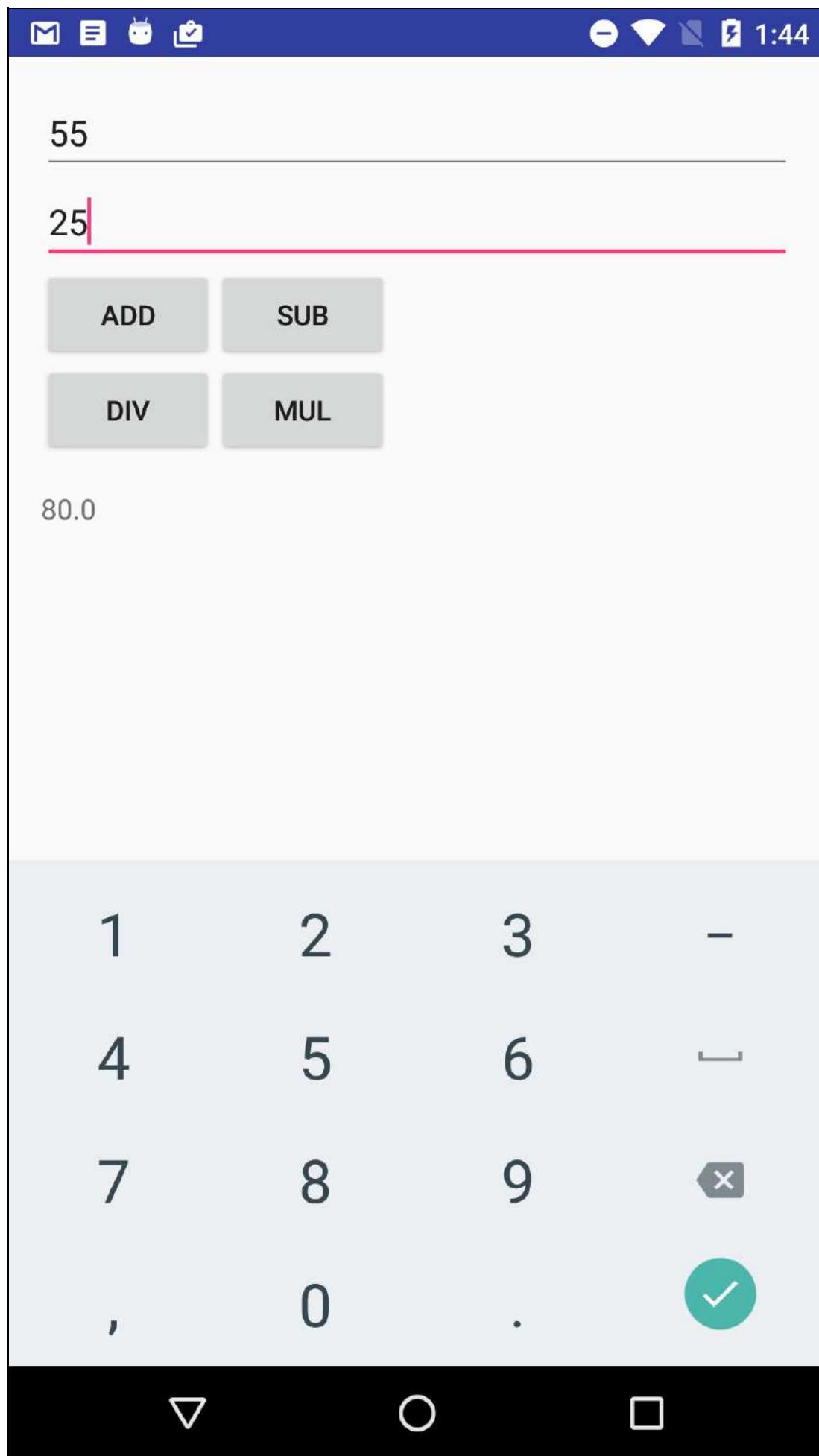
1. Unduh dan ekstrak folder proyek [SimpleCalc](#) project folder.
2. Mulai Android Studio dan pilih **File > Open**.
3. Buka folder untuk menemukan SimpleCalc, pilih file folder, dan klik **OK**.

Pembangunan Proyek SimpleCalc. Buka tampilan proyek jika belum terbuka.

Peringatan: Aplikasi ini berisi kesalahan yang akan Anda temukan dan perbaiki. Jika menjalankan aplikasi pada perangkat atau emulator Anda mungkin menemukan perilaku tidak terduga yang mungkin termasuk crash dalam aplikasi.

1.2 Menjelajahi Layout

1. Buka `res/layout/activity_main.xml`.
2. Pratinjau layout dalam Layout Editor.
3. Periksa kode layout, desain, dan catat hal-hal berikut ini:
 - Layout berisi dua EditTexts untuk input, empat tampilan Tombol untuk penghitungan, dan satu TextViews untuk menampilkan hasilnya.
 - Setiap tombol penghitungan memiliki handler onClick-nya sendiri (onAdd, OnSub, dan sebagainya.)
 - TextView untuk hasil tidak memiliki teks di dalamnya secara default.
 - Tampilan EditText memiliki properti `android:inputType` dan nilai `"numberDecimal"`. Properti ini menunjukkan bahwa EditText hanya menerima angka sebagai input. Keyboard yang muncul pada layar juga hanya akan berisi angka. Anda akan mengetahui selengkapnya tentang tipe input untuk EditText dalam praktik berikutnya.

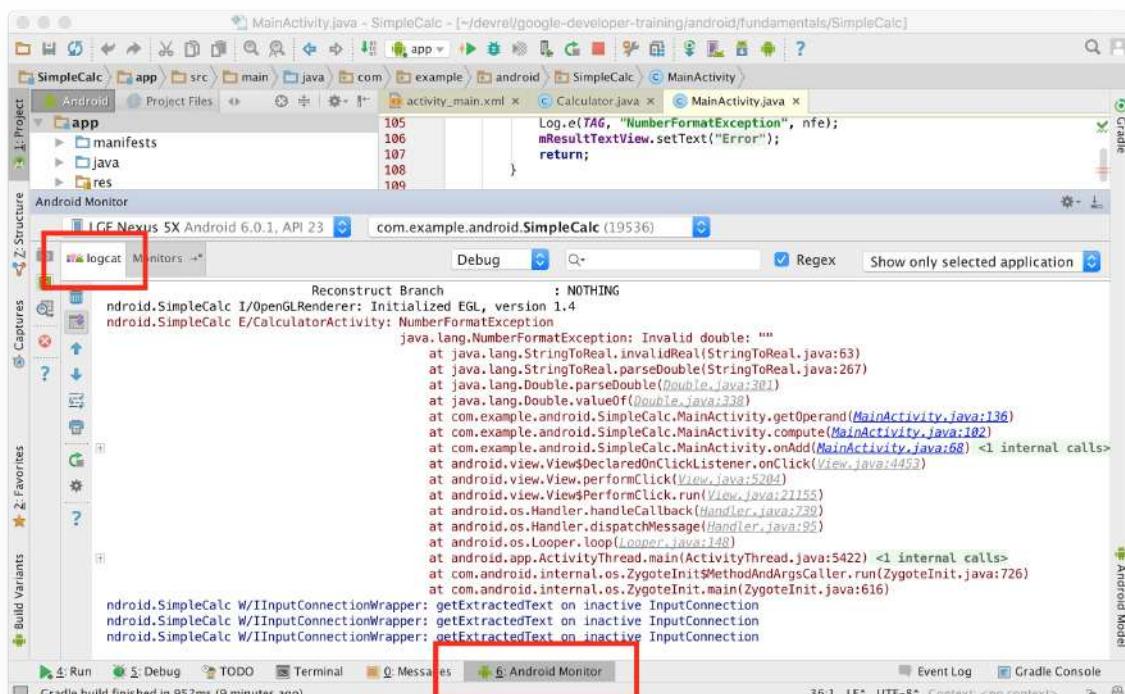


1.3 Jelajahi kode aplikasi

- Perluas folder aplikasi/java folder dalam tampilan proyek Android. Selain kelas MainActivity, proyek ini juga menyertakan kelas Kalkulator utilitas.
- Buka Kalkulator (java/com.example.android.simplecalc/Calculator.java). Periksa kodennya. Setelah pemeriksaan, Anda bisa melakukan observasi berikut:
 - Operasi yang dapat dilakukan ditentukan oleh enum Operator.
 - Semua metode operasi bersifat publik.
- Buka MainActivity (java/com.example.android.simplecalc/MainActivity). Periksa kodennya. Observasi apa yang bisa Anda lakukan tentang kode dan aktivitas? Pikirkan jawabannya dan konfirmasi hal-hal berikut:
 - Semua handler onClick memanggil metode compute() privat, dengan nama operasi sebagai salah satu nilai dari enumerasi Calculator.Operator.
 - Metode compute() memanggil metode privat getOperand() (yang bergantian memanggil getOperandText()) untuk mengambil nilai angka dari EditTexts.
 - Metode compute() menggunakan switch pada nama operand untuk memanggil metode yang tepat dalam kelas Calculator.
 - Metode penghitungan dalam kinerja kelas Calculator melakukan aritmetika yang sebenarnya dan mengembalikan nilai.
 - Bagian terakhir metode compute() memperbarui TextView dengan hasil penghitungan,
- Jalankan aplikasi. Coba hal-hal berikut ini:
 - Masukkan integer dan nilai titik-mengambang untuk penghitungan.
 - Masukkan nilai titik-mengambang dengan pecahan desimal besar (misalnya, 1.6753456)
 - Bagi angka dengan nol.
 - Biarkan salah satu atau kedua tampilan EditText kosong, dan coba penghitungan apa pun.
- Periksa pelacakan tumpukan dalam Android Studio saat aplikasi melaporkan kesalahan.

Jika pelacakan tumpukan tidak terlihat, klik tombol Android Monitor pada tombol Android Studio, lalu klik logcat.

Jika satu atau kedua tampilan EditText dalam SimpleCalc kosong, aplikasi melaporkan "Kesalahan" dan log sistem menampilkan status tumpukan eksekusi saat aplikasi menghasilkan kesalahan. Pelacakan tumpukan biasanya menyediakan informasi penting tentang mengapa kesalahan terjadi.



Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Periksa pelacakan tumpukan dan coba cari tahu apa yang menyebabkan kesalahan (tapi jangan perbaiki dulu.)

Tugas 2. Menjalankan SimpleCalc di Debugger

Dalam tugas ini Anda akan mendapatkan pengenalan ke debugger dalam Android Studio dan belajar cara menjalankan aplikasi dalam mode debug.

2.1 Memulai dan Menjalankan aplikasi dalam mode debug

1. Dalam Android Studio, pilih **Run > Debug app** atau klik ikon  di toolbar.

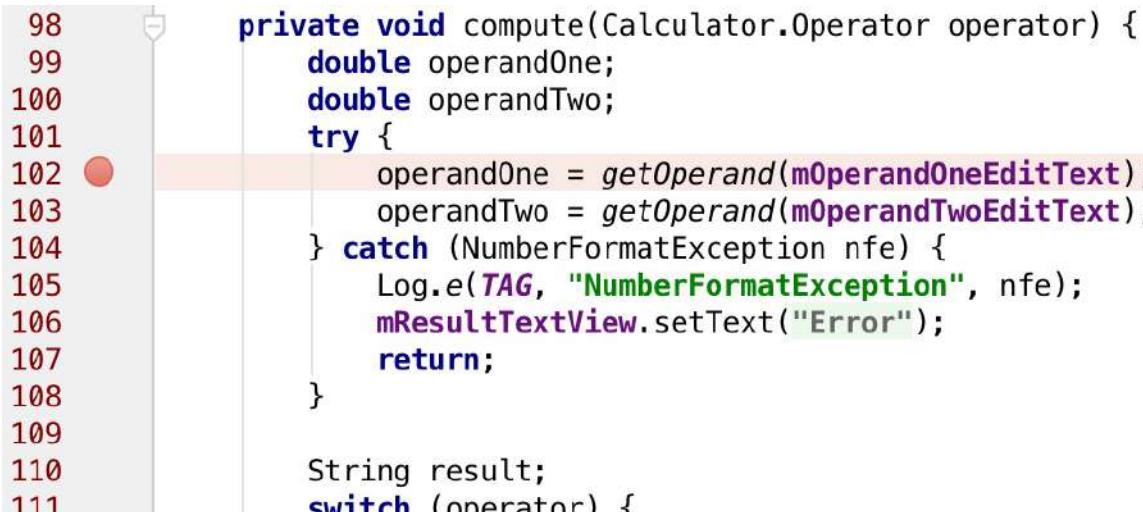
Jika aplikasi sudah berjalan, Anda akan ditanyakan apakah ingin memulai ulang aplikasi dalam mode debug. Klik **Restart app**.

Android Studio membangun dan menjalankan aplikasi pada emulator atau perangkat. Men-debug sama saja pada kedua kasus. Saat Android Studio memulai debugger, Anda mungkin melihat pesan yang berbunyi "Waiting for debugger" pada perangkat sebelum Anda bisa menggunakan aplikasi.

Jika tampilan Debug tidak secara otomatis muncul dalam Android Studio, klik tab **Debug** di bagian bawah layar, lalu tab **Debugger**.

2. Buka file **MainActivity.java** dan klik baris keempat metode **compute()** (baris setelah pernyataan **try**).
3. Klik di gutter kiri jendela editor baris tersebut, sebelah nomor baris. Titik merah muncul di baris itu, menunjukkan breakpoint.

Anda juga bisa menggunakan **Run > Toggle Line Breakpoint** atau Control-F8 (Command-F8 di OS X) untuk menyetel atau menghapus breakpoint pada baris.



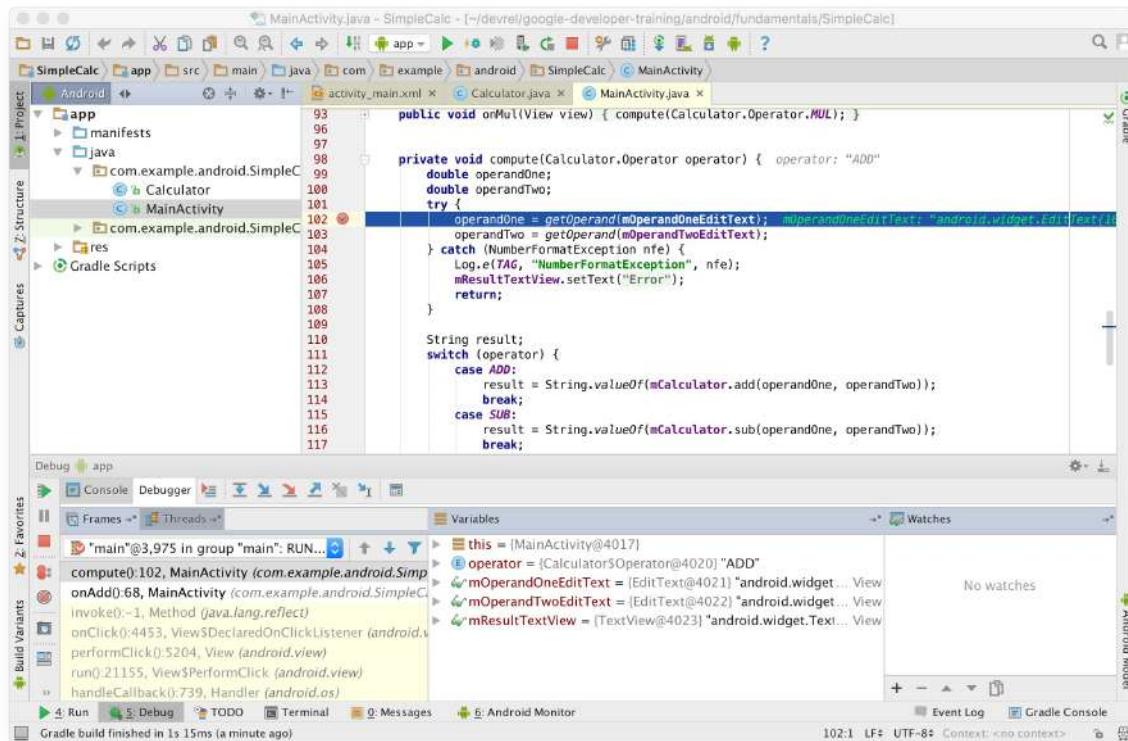
```

98     private void compute(Calculator.Operator operator) {
99         double operandOne;
100        double operandTwo;
101       try {
102           operandOne = getOperand(mOperandOneEditText);
103           operandTwo = getOperand(mOperandTwoEditText);
104       } catch (NumberFormatException nfe) {
105           Log.e(TAG, "NumberFormatException", nfe);
106           mResultTextView.setText("Error");
107           return;
108       }
109
110      String result;
111      switch (operator) {

```

4. Dalam aplikasi SimpleCalc, masukkan nomor pada tampilan **EditText** dan klik salah satu tombol hitung.

Eksekusi aplikasi berhenti saat mencapai breakpoint yang Anda setel, dan debugger menampilkan status saat ini aplikasi Anda pada breakpoint tersebut.



5. Periksa jendela Debug. Termasuk bagian ini:
6. **Panel Frames:** menunjukkan frame tumpukan eksekusi saat ini untuk thread yang diberikan. Tumpukan eksekusi menunjukkan setiap kelas dan metode yang telah dipanggil dalam aplikasi dan dalam waktu proses Android, dengan metode terbaru di bagian atas. Thread muncul pada menu tarik turun. Aplikasi saat ini berjalan dalam thread utama, dan aplikasi menjalankan metode compute() dalam MainActivity.
7. **Panel Variables:** menampilkan variabel dalam lingkup saat ini dan nilainya. Pada tahap eksekusi aplikasi ini, variabel yang tersedia adalah: this (untuk aktivitas), operator (nama operator dari Calculator.Operator, tempat metode dipanggil), dan variabel global untuk EditText dan TextView. Setiap variabel dalam panel ini memiliki segitiga pengungkapan yang memungkinkan Anda melihat properti objek yang berada dalam variabel tersebut. Coba luaskan variabel untuk menjelajahi propertiinya.
8. **Panel Watches:** menampilkan nilai-nilai untuk variabel watches yang sudah Anda setel. Watches mengizinkan Anda mengawasi variabel spesifik dalam program dan melihat variabel tersebut berubah saat program dijalankan.
9. Lanjutkan eksekusi aplikasi dengan **Run > Resume Program** atau klik **Resume** ikon di sisi kiri tampilan jendela debugger.

Aplikasi SimpleCalc terus berjalan, dan Anda bisa berinteraksi dengan aplikasi sampai eksekusi kode lain waktu mencapai breakpoint.

2.2 Men-debug aplikasi yang berjalan

Jika aplikasi sudah berjalan pada perangkat atau emulator, dan Anda memutuskan untuk men-debug perangkat itu, Anda dapat memindahkan aplikasi yang sudah berjalan ke mode debug.

1. Jalankan aplikasi SimpleCalc secara normal, dengan **ikon Run** ikon.
2. Pilih **Run > Attach debugger to Android process** atau klik **Attach** di bilah alat.



- Pilih proses aplikasi dari dialog yang muncul. Klik **OK**.

Jendela Debug muncul, dan Anda sekarang bisa men-debug aplikasi jika sudah memulainya dalam mode debug.

Catatan: Jika jendela Debug tidak secara otomatis muncul, klik tab Debug di bagian bawah layar, lalu tab Debugger.

Tugas 3: Menjelajahi Fitur Debugger

Dalam tugas ini kita akan menjelajahi beragam fitur dalam debugger Android Studio, termasuk menjalankan aplikasi baris per baris, bekerja dengan breakpoint, dan memeriksa variabel.

3.1 Menyusuri eksekusi aplikasi

Setelah breakpoint, Anda bisa menggunakan debugger untuk mengeksekusi setiap baris kode dalam aplikasi satu per satu, dan memeriksa status variabel saat aplikasi berjalan.

- Debug aplikasi dalam Android Studio, dengan breakpoint yang sudah disetel pada tugas terakhir.
- Dalam aplikasi, masukkan angka dalam tampilan EditText dan klik tombol Add.

Eksekusi aplikasi Anda berhenti pada breakpoint yang sudah disetel sebelumnya, dan debugger menunjukkan status aplikasi saat ini. Baris saat ini disorot dalam kode Anda.

- Klik tombol **Step Over**  di bagian atas jendela debugger.

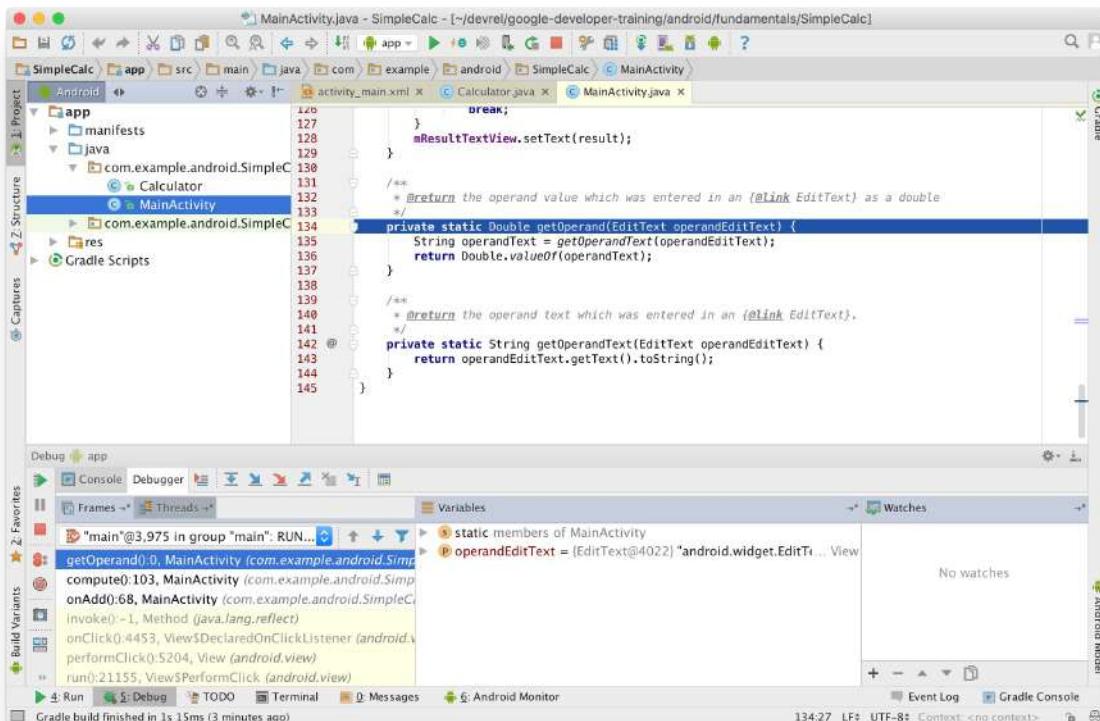
Debugger mengeksekusi baris saat ini dalam metode compute() (tempat breakpoint, penetapan untuk operandOne), dan sorotan pindah ke baris berikutnya dalam kode (penetapan untuk operandTwo). Panel Variables diperbarui untuk merefleksikan status eksekusi baru, dan nilai variabel saat ini juga muncul setelah setiap baris kode sumber yang dicetak miring.

Anda juga bisa menggunakan **Run > Step Over**, atau F8, untuk melangkah kode Anda.

4. Di baris berikutnya (penetapan untuk operandTwo), klik ikon **Step Into**  ikon.

Step Into melompat ke eksekusi panggilan metode dalam baris saat ini (bukan hanya mengeksekusi metode dan tetap berada pada baris yang sama). Dalam hal ini, karena penetapan termasuk panggilan ke getOperand(), debugger menggulir kode MainActivity ke definisi metode tersebut.

Saat melangkah ke dalam metode, panel Frames diperbarui untuk menunjukkan bingkai baru dalam tumpukan panggilan (di sini getOperand()), dan panel Variables menunjukkan variabel yang tersedia dalam lingkup metode baru. Anda bisa mengeklik baris mana saja pada panel Frames untuk melihat titik dalam bingkai tumpukan, tempat metode dipanggil.



Anda juga bisa menggunakan **Run > Step Into**, atau F7, untuk memulai metode.

5. Klik **Step Over**  untuk menjalankan setiap baris dalam getOperand(). Perhatikan bahwa saat metode selesai debugger mengembalikan Anda ke titik awal saat Anda mulai metode, dan semua panel diperbarui dengan informasi baru.
6. Gunakan **Step Over** dua kali untuk memindahkan titik eksekusi ke baris pertama di dalam pernyataan kasus untuk ADD.
7. Klik **Step Into** .

Debugger mengeksekusi metode yang sesuai dan didefinisikan dalam kelas Kalkulator, buka Calculator.java file, dan gulir ke titik eksekusi dalam kelas tersebut. Lagi, beragam pembaruan panel untuk merefleksikan status baru.

8. Gunakan ikon **Step Out**  untuk mengeksekusi sisa metode penghitungan itu dan mengembalikan ke metode compute() dalam MainActivity. Lalu Anda bisa meneruskan men-debug metode compute() yang tadi ditinggalkan.
- Anda juga bisa menggunakan **Run > Step Out** atau Shift-F8 untuk keluar dari eksekusi metode.

3.2 Bekerja dengan Breakpoint

Gunakan breakpoint untuk menunjukkan di mana dalam kode Anda ingin menyela eksekusi aplikasi untuk men-debug porsi itu dari aplikasi.

1. Temukan breakpoint yang disetel pada tugas terakhir saat memulai metode compute() dalam MainActivity.
2. Tambahkan breakpoint ke awal pernyataan switch.
3. Klik kanan di breakpoint baru itu dan masukkan tes berikut ke dalam bidang Condition:

```
(operandOne == 42) || (operandTwo == 42)
```

4. Klik **Done**.

Breakpoint kedua ini adalah breakpoint *bersyarat*. Eksekusi aplikasi hanya akan berhenti pada breakpoint ini jika tes dalam kondisi ini benar. Dalam kasus ini, ekspresi hanya benar jika satu atau operand lain yang Anda masukkan adalah 42. Anda dapat memasukkan ekspresi Java apa pun selama menghasilkan boolean.

5. Jalankan aplikasi dalam mode debug (**Run > Debug**), atau klik **Resume**  jika sudah berjalan. Dalam aplikasi, masukkan dua nomor selain 42 dan klik tombol Add. Eksekusi berhenti pada breakpoint pertama dalam metode compute().
6. Klik **Resume** untuk melanjutkan men-debug aplikasi. Amati bahwa eksekusi tidak berhenti pada breakpoint kedua, karena kondisi tidak terpenuhi.
7. Klik kanan breakpoint pertama dan hapus centang **Enabled**. Klik **Done**. Amati bahwa ikon breakpoint sekarang memiliki titik hijau dengan garis merah.

Menonaktifkan breakpoint memungkinkan Anda untuk "membisukan" breakpoint sementara waktu tanpa menghapusnya dari kode. Jika Anda menghapus breakpoint sekaligus Anda juga akan kehilangan kondisi yang Anda buat, jadi menonaktifkannya adalah pilihan yang lebih baik.

Anda juga bisa membisukan semua breakpoint sekaligus dalam aplikasi dengan ikon **Mute Breakpoints** .

8. Dalam aplikasi, masukkan 42 dalam EditText dan klik tombol apa saja. Amati bahwa breakpoint bersyarat pada pernyataan switch menghentikan eksekusi (kondisi terpenuhi.)

9. Klik View ikon Breakpoints  di tepi kiri jendela debugger. Jendela Breakpoints muncul.

Jendela Breakpoints memungkinkan Anda untuk melihat semua breakpoint dalam aplikasi, mengaktifkan atau menonaktifkan breakpoint individual dan menambahkan fitur breakpoint tambahan termasuk kondisi, dependensi pada breakpoints lain, dan mencatat log.

10. Klik **Done** untuk menutup jendela breakpoint.

3.3 Memeriksa dan memodifikasi variabel

Debugger Android Studio mengizinkan Anda memeriksa status variabel dalam aplikasi saat aplikasi itu berjalan.

1. Jalankan aplikasi SimpleCalc dalam mode debug jika belum dijalankan.
2. Dalam aplikasi, masukkan dua nomor, salah satunya 42, dan klik tombol Add.

Breakpoint pertama dalam compute() masih dibisukan. Eksekusi berhenti pada breakpoint kedua (pada pernyataan switch), dan debugger muncul.

3. perhatikan dalam panel Variables bagaimana variabel operandOne dan operandTwo memiliki nilai yang Anda masukkan ke dalam aplikasi.
4. Amati bahwa variabel ini adalah objek MainActivity. Klik panah pengungkapan untuk melihat variabel anggota objek itu.
5. Klik kanan variabel operandOne dalam panel Variables, dan pilih **Set Value**. Anda juga bisa menggunakan F2.
6. Ubah nilai operandOne ke 10 dan tekan Return.
7. Modifikasi nilai operandTwo ke 10 dengan cara yang sama dan tekan Return.
8. Klik ikon Resume untuk terus menjalankan aplikasi Anda. Amati bahwa hasil di aplikasi sekarang 20, berdasarkan nilai variabel yang Anda ubah dalam debugger.

9. Dalam aplikasi, klik tombol Add. Eksekusi menghentikan breakpoint.
10. Klik ikon Evaluate Expression , atau pilih **Run > Evaluate Expression**. Jendela Evaluate Code Fragment muncul. Anda juga bisa mengeklik kanan pada variabel apa pun dan memilih Evaluate Expression.

Gunakan Evaluate Expression untuk menjelajahi status variabel dan objek dalam aplikasi Anda, termasuk metode memanggil pada objek tersebut. Anda bisa memasukkan kode apa pun ke dalam jendela ini.
11. Ketikkan `mOperandOneEditText.getHint()` ke dalam jendela Expression dan klik Evaluate.
12. Jendela Evaluate Expression diperbarui dengan hasil ekspresi itu. Petunjuk untuk EditText ini adalah string "Type Operand 1", seperti yang awalnya ditetapkan dalam XML untuk EditText yang dimaksud.

Hasil yang Anda dapat dari mengevaluasi ekspresi didasarkan pada status aplikasi saat ini. Tergantung nilai variabel dalam aplikasi saat Anda mengevaluasi ekspresi, Anda mungkin akan mendapatkan hasil yang berbeda.

Perhatikan bahwa jika Anda menggunakan Evaluate Expression untuk mengubah nilai variabel atau properti objek, Anda akan mengubah status berjalan aplikasi.
13. Klik **Close** untuk menyembunyikan jendela Evaluate Expression.

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Dalam Tugas 1.3, Anda mencoba menjalankan aplikasi SimpleCalc tanpa nilai dalam tampilan EditText, yang menyebabkan pesan kesalahan. Gunakan debugger untuk menyusuri eksekusi kode dan menentukan mengapa tepatnya kesalahan ini terjadi. Perbaiki bug yang menyebabkan kesalahan ini.

Rangkuman

- Lihat informasi pencatatan log dalam Android Studio dengan tab logcat di panel Android Monitor.
- Jalankan aplikasi dalam mode debug dengan mengeklik ikon debug atau memilih **Run > Debug app**.
- Breakpoint adalah tempat dalam kode di mana Anda akan menjeda eksekusi normal aplikasi untuk melakukan tindakan yang lain. Setel atau hapus breakpoint debug dengan mengeklik di gutter kiri jendela editor tepat di samping baris target.
- Jendela Debug dalam Android Studio menunjukkan (stack) Frames, Variabel dalam bingkai tersebut dan Watches (pelacakan aktif variabel saat program berjalan).

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Men-debug Aplikasi Anda](#)

Ketahui selengkapnya

- [Men-debug Aplikasi Anda](#) (Panduan Pengguna Android Studio)
- [Men-debug dan Menguji di Android Studio](#) (video)

3.2: Menguji Aplikasi dengan Pengujian Unit

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1. Menjelajahi dan Menjalankan SimpleCalc di Android Studio](#)
- [Tugas 2. Menambahkan lebih banyak pengujian unit ke CalculatorTest](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Menguji kode dapat membantu menemukan bug secepat mungkin selama development, saat yang paling murah untuk menangani bug — dan meningkatkan ketangguhan kode saat aplikasi semakin besar dan kompleks. Dengan pengujian terhadap kode, Anda bisa melatih sebagian kecil aplikasi dalam isolasi, dan dengan cara yang otomatis dan dapat diulang.

Android Studio dan Android Testing Support Library mendukung beberapa jenis pengujian dan framework pengujian. Dalam praktik ini Anda akan menjelajahi fungsionalitas bawaan Android Studio untuk pengujian, dan belajar menulis serta menjalankan pengujian unit lokal.

Pengujian unit lokal adalah pengujian yang dikompilasi dan dijalankan sepenuhnya pada komputer lokal Anda dengan Java Virtual Machine (JVM). Gunakan pengujian unit lokal untuk menguji bagian aplikasi Anda (seperti logika internal) yang tidak membutuhkan akses ke framework Android atau perangkat atau emulator Android, atau bagian yang membuat objek palsu ("tiruan") yang berpura-pura berperilaku seperti framework yang setara. Pengujian unit ditulis dengan JUnit, framework pengujian unit yang umum untuk Java.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah memahami:

- Cara membuat proyek di Android Studio.
- Komponen utama Android Studio (manifest, sumber daya, file Java, file gradle).
- Cara membuat dan menjalankan aplikasi.

Yang akan Anda PELAJARI

- Cara mengelola dan menjalankan pengujian di Android Studio.
- Apa itu pengujian unit dan cara menulis pengujian unit untuk kode.
- Cara membuat dan menjalankan pengujian unit lokal di Android Studio.

Yang akan Anda LAKUKAN

- Menjalankan pengujian awal di aplikasi SimpleCalc.
- Menambahkan pengujian lainnya ke aplikasi SimpleCalc.
- Menjalankan pengujian unit untuk melihat hasilnya.

Ringkasan Aplikasi

Praktik ini menggunakan aplikasi [SimpleCalc](#) yang sama dengan aplikasi dari praktik sebelumnya. Anda bisa memodifikasi aplikasi ini dalam aplikasi langsung, atau [salin proyek Anda](#) ke aplikasi baru.

Tugas 1. Menjelajahi dan menjalankan SimpleCalc di Android Studio

Anda menulis dan menjalankan pengujian (pengujian unit dan pengujian berinstrumen) di dalam Android Studio, di samping kode untuk aplikasi. Setiap proyek Android baru menyertakan kelas contoh dasar untuk pengujian yang bisa Anda perpanjang atau ganti dengan penggunaan sendiri.

Dalam tugas ini kita akan kembali ke aplikasi SimpleCalc, yang menyertakan kelas pengujian unit dasar.

1.1 Menjelajahi rangkaian sumber dan SimpleCalc

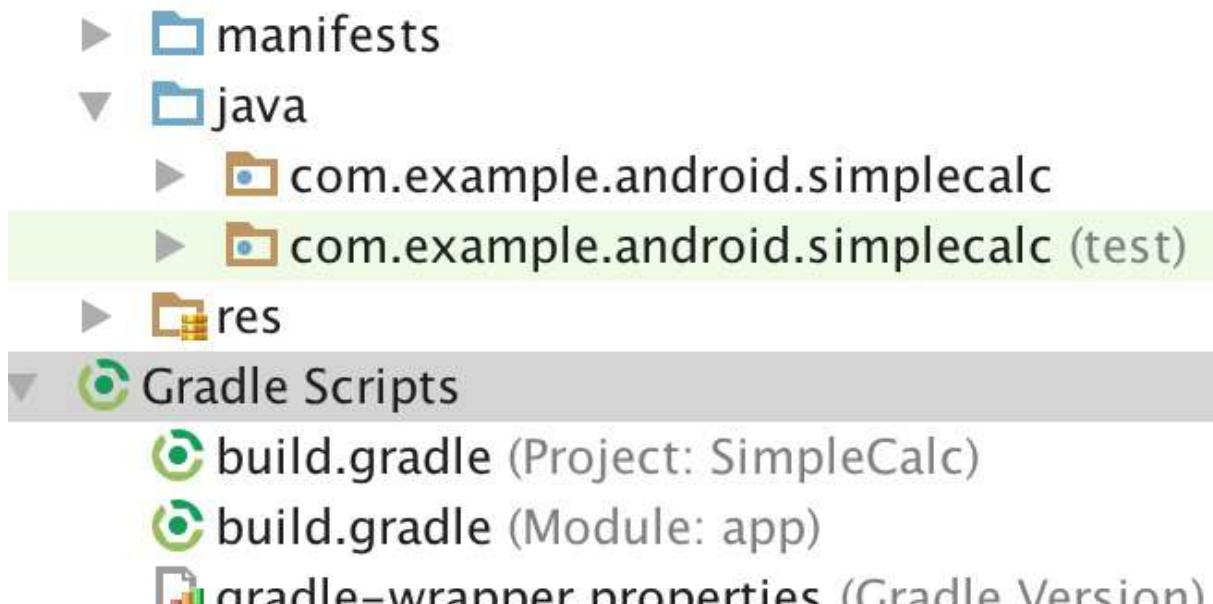
Rangkaian sumber adalah kumpulan kode terkait dalam proyek untuk target membangun yang berbeda atau "variasi" lainnya dari aplikasi Anda. Saat Android Studio membuat proyek, Android Studio membuat tiga rangkaian sumber:

- Rangkaian sumber utama, untuk kode dan sumber daya aplikasi Anda.
- Rangkaian sumber **pengujian**, untuk pengujian unit lokal aplikasi.
- Rangkaian sumber **androidTest**, untuk pengujian berinstrumen Android.

Dalam tugas ini Anda akan mempelajari bagaimana rangkaian sumber ditampilkan dalam Android Studio, memeriksa konfigurasi gradle untuk pengujian, dan menjalankan pengujian untuk aplikasi SimpleCalc. Anda akan menggunakan rangkaian sumber androidTest dengan lebih detail pada praktik berikutnya.

1. Membuka proyek SimpleCalc dalam Android Studio jika belum. Jika belum memiliki SimpleCalc Anda bisa menemukannya di [tautan unduhan] ini(<https://github.com/google-developer-training/android-fundamentals/tree/master/SimpleCalc>).
2. Buka tampilan Project, dan perluas folder **aplikasi** dan **java**.

Folder java dalam tampilan Android mencantumkan semua rangkaian sumber dalam aplikasi menurut nama paket (com.android.example.simplecalc), dengan **pengujian** dan **androidTest** ditampilkan dalam tanda kurung setelah nama paket. Dalam aplikasi SimpleCalc, hanya rangkaian sumber utama dan pengujian yang digunakan.



3. Luaskan folder **com.android.example.simplecalc (test)**.

Ini adalah folder tempat Anda meletakkan pengujian unit lokal aplikasi. Android Studio membuat kelas pengujian contoh untuk Anda dalam folder ini untuk proyek baru, tetapi untuk SimpleCalc, kelas pengujinya disebut CalculatorTest.

4. Buka **CalculatorTest.java**.

5. Periksa kode dan catat hal-hal berikut ini:

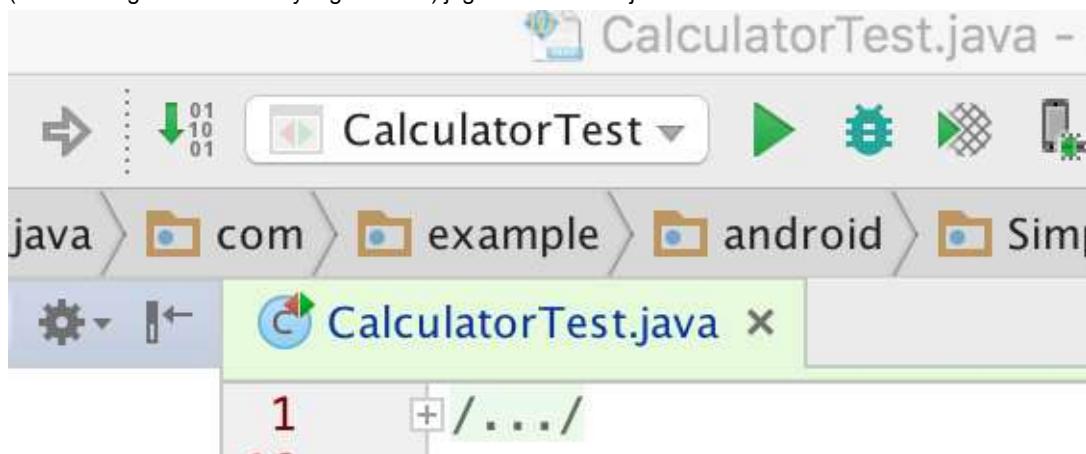
- Satu-satunya impor adalah dari paket org.junit, org.hamcrest, dan android.test. Tidak ada dependensi pada kelas framework Android di sini.
- Anotasi `@RunWith(JUnit4.class)` menunjukkan runner yang akan digunakan untuk menjalankan pengujian dalam kelas ini. Runner pengujian adalah pustaka atau serangkaian alat yang memungkinkan pengujian terjadi dan hasilnya dicetak ke log. Untuk pengujian dengan persiapan atau persyaratan infrastruktur yang lebih rumit (seperti Espresso), Anda akan menggunakan runner pengujian yang berbeda. Untuk contoh berikut kita menggunakan runner pengujian JUnit4 dasar.
- Anotasi `@SmallTest` menunjukkan bahwa semua pengujian dalam kelas ini adalah pengujian unit yang tidak memiliki dependensi dan berjalan dalam milidetik. Anotasi `@SmallTest`, `@MediumTest`, dan `@LargeTest` adalah konvensi yang memudahkan membundel grup pengujian ke dalam kelompok fungsionalitas yang serupa.
- Metode `setUp()` digunakan untuk menyiapkan lingkungan sebelum pengujian, dan menyertakan anotasi `@Before`. Dalam hal ini, persiapan membuat instance baru kelas Calculator dan menetapkannya ke variabel anggota `mCalculator`.
- Metode `addTwoNumbers()` adalah pengujian yang sebenarnya, dan dianotasi dengan `@Test`. Hanya metode dalam kelas tes yang memiliki anotasi `@Test` yang dianggap pengujian bagi runner pengujian. Perhatikan bahwa metode pengujian konvensi tidak termasuk kata "test".
- Baris pertama `addTwoNumbers()` memanggil metode `add()` dari kelas Calculator. Anda hanya bisa menguji metode yang bersifat publik atau dilindungi paket. Dalam hal ini Calculator adalah kelas publik dengan metode publik, jadi semuanya berjalan baik.
- Baris kedua adalah pernyataan untuk pengujian. Pernyataan adalah ekspresi yang harus mengevaluasi dan menghasilkan benar untuk lulus pengujian. Dalam hal ini, pernyataannya adalah bahwa hasil yang Anda dapatkan dari metode penambahan $(1 + 1)$ cocok dengan nomor 2 yang diberikan. Anda akan mempelajari selengkapnya tentang cara membuat pernyataan nanti dalam praktik ini.

1.2 Menjalankan pengujian dalam Android Studio

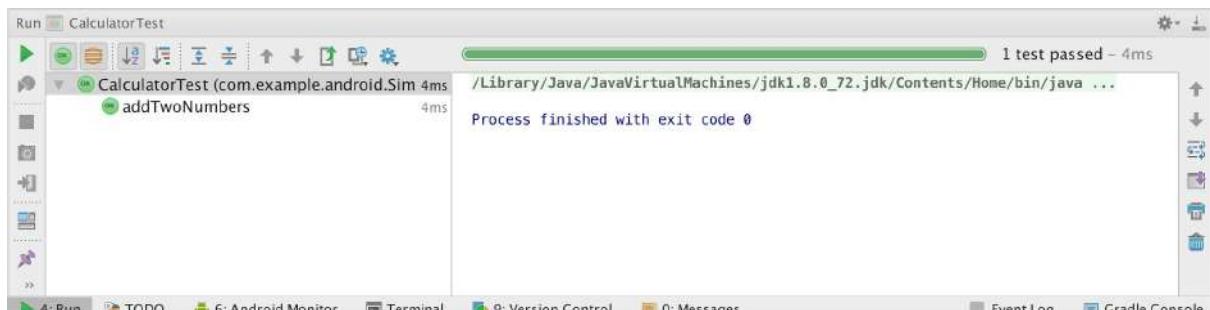
Dalam tugas ini Anda akan menjalankan pengujian unit dalam folder pengujian dan melihat keluaran untuk pengujian yang berhasil dan gagal.

1. Dalam tampilan proyek, klik kanan kelas **CalculatorTest** dan pilih **Run 'CalculatorTest'**.

Proyek membangun, jika perlu, dan tampilan pengujian muncul di bagian bawah layar. Di bagian atas layar, tarik turun (untuk konfigurasi eksekusi yang tersedia) juga berubah menjadi **CalculatorTest**.



Semua pengujian dalam kelas **CalculatorTest** berjalan, dan jika pengujian tersebut berhasil, bilah kemajuan di bagian atas tampilan akan berubah menjadi warna hijau. (Dalam hal ini, hanya ada satu pengujian saat ini.) Pesan status dalam footer juga melaporkan "Tests Passed."



2. Dalam kelas CalculatorTest, ubah pernyataan dalam addTwoNumbers() ke:

```
assertThat(resultAdd, is(equalTo(3d)));
```

3. Dalam menu tarik turun run configurations di bagian atas layar, pilih **CalculatorTest** (jika belum dipilih) dan klik **Run**



Pengujian berjalan lagi seperti sebelumnya, namun kali ini pernyataannya gagal (3 tidak sama dengan $1 + 1$.) Bilah kemajuan dalam tampilan berjalan berubah menjadi merah, dan log pengujian menunjukkan di mana (pernyataan) gagal dan alasannya.

4. Ubah pernyataan dalam addTwoNumbers() kembali ke pengujian yang benar dan jalankan pengujian lagi agar berhasil.
5. Dalam menu tarik turun run configurations, pilih **app** untuk menjalankan aplikasi Anda dengan normal.

Tugas 2. Menambahkan lebih banyak pengujian unit ke CalculatorTest

Dengan pengujian unit, Anda mengambil sedikit kode dalam aplikasi seperti metode atau kelas, dan mengisolasinya dari aplikasi, agar pengujian yang ditulis memastikan sedikit kode bekerja dengan semestinya. Biasanya pengujian unit memanggil metode dengan berbagai input yang berbeda, dan memverifikasi bahwa metode tertentu melakukan apa yang Anda harapkan dan menghasilkan apa yang Anda inginkan.

Dalam tugas ini Anda akan mempelajari lebih lanjut mengenai cara membangun pengujian unit. Anda akan menulis pengujian unit tambahan untuk metode dalam metode utilitas Kalkulator pada aplikasi SimpleCalc, dan menjalankan pengujian tersebut untuk memastikan pengujian menghasilkan keluaran yang diinginkan.

Catatan: Pengujian unit, development yang didorong pengujian, serta API JUnit 4 adalah topik yang besar dan kompleks dan berada di luar lingkup kursus ini. Lihat [Resources](#) untuk mendapatkan tautan ke informasi selengkapnya.

2.1 Menambahkan lebih banyak pengujian untuk metode add()

Walaupun tidak mungkin menguji setiap nilai yang mungkin, yang pernah dilihat metode add(), sebaiknya uji input yang mungkin tidak biasa. Misalnya, pertimbangkan apa yang terjadi jika metode add() mendapatkan argumen:

- Dengan operand negatif.
- Dengan angka titik-mengambang.
- Dengan angka yang sangat besar.
- Dengan operand tipe yang berbeda (yang mengambang dan ganda, misalnya)
- Dengan operand yang nol.
- Dengan operand yang tak terhingga.

Dalam tugas ini, kita akan menambahkan lebih banyak pengujian unit untuk metode add(), untuk menguji berbagai macam input yang berbeda.

1. Menambahkan metode baru CalculatorTest bernama addTwoNumbersNegative(). Gunakan kerangka ini:

```
@Test
public void addTwoNumbersNegative() {
}
```

Metode ini memiliki struktur yang serupa dengan `addTwoNumbers`: ini adalah metode publik, tanpa parameter, yang mengembalikan kosong. Ini dianotasi dengan `@Test`, yang menunjukkan pengujian unit tunggal.

Mengapa tidak menambahkan lebih banyak pernyataan ke `addTwoNumbers`? Mengelompokkan lebih dari satu pernyataan ke dalam satu metode tunggal dapat membuat pengujian lebih sulit di-debug jika salah satu pernyataan gagal, dan mengaburkan tes yang berhasil. Peraturan umum untuk pengujian unit adalah menyediakan metode pengujian untuk setiap pernyataan individual.

- Menjalankan semua pengujian di `CalculatorTests`, seperti sebelumnya.

Dalam jendela pengujian, `addTwoNumbers` dan `addTwoNumbersNegative` dicantumkan sebagai tersedia (dan lulus) pengujian pada panel kiri. Pengujian `addTwoNumbersNegative` akan tetap lulus meskipun tidak berisi kode apa pun, pengujian yang tidak melakukan apa pun dianggap berhasil.

- Menambahkan baris untuk memanggil metode `add()` dalam kelas `Kalkulator` dengan operand negatif.

```
double resultAdd = mCalculator.add(-1d, 2d);
```

Notasi "d" setelah setiap operand menunjukkan bahwa ini adalah angka tipe double. Karena metode `add()` didefinisikan dengan parameter ganda, floats atau ints juga akan berfungsi. Menunjukkan tipe secara eksplisit memungkinkan Anda untuk menguji tipe lain secara terpisah, jika perlu.

- Menambahkan pernyataan dengan `assertThat()`.

```
assertThat(resultAdd, is(equalTo(1d)));
```

Metode `assertThat()` adalah pernyataan JUnit4 yang mengklaim bahwa ekspresi dalam argumen pertama sama dengan yang ada dalam argumen kedua. JUnit yang lebih lama menggunakan metode pernyataan yang lebih spesifik (`assertEquals()`, `assertNull()`, `assertTrue()`), tetapi `assertThat()` adalah metode yang lebih fleksibel, mudah di-debug, dan sering kali lebih mudah untuk membaca format.

Metode `assertThat()` digunakan dengan *matcher*. Matcher adalah panggilan metode berantai dalam operand kedua pernyataan ini (`is(equalTo())`). Matcher yang tersedia, yang bisa Anda gunakan untuk membuat pernyataan, didefinisikan oleh framework hamcrest (Hamcrest adalah anagram untuk matcher.) Hamcrest menyediakan matcher dasar untuk pernyataan dasar. Anda juga bisa mendefinisikan matcher khusus untuk pernyataan yang lebih kompleks.

Dalam hal ini, pernyataan adalah hasil dari operasi `add()` ($-1 + 2$) sama dengan 1.

- Menambahkan pengujian unit baru ke `CalculatorTest` untuk angka titik-mengambang:

```
@Test
public void addTwoNumbersFloats() {
    double resultAdd = mCalculator.add(1.111f, 1.111d);
    assertThat(resultAdd, is(equalTo(2.222d)));
}
```

Lagi, pengujian yang sangat serupa dengan metode pengujian sebelumnya, tetapi dengan satu argumen untuk `ditambahkan()` yang secara eksplisit adalah tipe float, bukan double. Metode `add()` didefinisikan dengan parameter tipe double, sehingga Anda dapat memanggilnya dengan tipe float, dan angka tersebut dipromosikan ke double.

- Menjalankan semua pengujian di `CalculatorTests`, seperti sebelumnya.

- Klik **Run**  untuk menjalankan semua pengujian lagi.

Kali ini pengujiannya gagal dan bilah kemajuannya berwarna merah. Ini adalah bagian penting dari pesan kesalahan:

```
java.lang.AssertionError:  
Expected: is <2.222>  
but: was <2.2219999418258665>
```

Aritmetika dengan angka titik-mengambang tidak eksak, dan promosi menghasilkan efek samping presisi tambahan. Pernyataan dalam pengujian itu secara teknis salah: nilai yang diharapkan tidak sama dengan nilai yang sebenarnya.

Pertanyaannya adalah: saat Anda memiliki masalah presisi dengan argumen float promosi, apakah itu masalah dengan kode, atau masalah dengan pengujian? Dalam hal ini, kedua ke argumen input ke metode add() dari aplikasi Calculator selalu tipe double, sehingga pengujian ini bersifat arbitrer dan tidak realistik. Namun, jika aplikasi ditulis sedemikian rupa sehingga input ke metode add() bisa berupa double atau float dan Anda hanya peduli dengan beberapa presisi, Anda perlu menyediakan ruang toleransi pada pengujian agar "cukup dekat" diperhitungkan sebagai berhasil.

8. Mengubah metode assertThat() untuk menggunakan matcher closeTo():

```
assertThat(resultAdd, is(closeTo(2.222, 0.01)));
```

Untuk pengujian ini, daripada menguji persamaan eksak, Anda bisa menguji persamaan dalam delta yang spesifik. Dalam hal ini, metode matcher closeTo() memerlukan dua argumen: nilai yang diharapkan dan jumlah delta. Di sini, delta itu hanya dua poin desimal presisi.

2.2 Menambahkan pengujian unit untuk metode penghitungan lain

Gunakan yang Anda pelajari dalam tugas sebelumnya untuk mengisi pengujian unit untuk kelas Kalkulator.

1. Tambahkan pengujian unit bernama subTwoNumbers() yang menguji metode sub().
2. Tambahkan pengujian unit bernama subWorksWithNegativeResults() yang menguji metode sub() tempat penghitungan yang diberikan menghasilkan angka yang negatif.
3. Tambahkan pengujian unit bernama mulTwoNumbers() yang menguji metode mul().
4. Tambahkan pengujian unit bernama mulTwoNumbersZero() yang menguji metode mul dengan sedikitnya satu argumen yang nol.
5. Tambahkan pengujian unit bernama divTwoNumbers() yang menguji metode div() dengan dua argumen bukan nol.

Tantangan: Tambahkan pengujian unit bernama divByZero() yang menguji metode div() dengan dua argumen 0. Petunjuk: Coba ini dalam aplikasi terlebih dulu untuk melihat hasilnya.

Kode Solusi:

```

@Test
public void addTwoNumbers() {
    double resultAdd = mCalculator.add(1d, 1d);
    assertThat(resultAdd, is(equalTo(2d)));
}

@Test
public void addTwoNumbersNegative() {
    double resultAdd = mCalculator.add(-1d, 2d);
    assertThat(resultAdd, is(equalTo(1d)));
}

@Test
public void addTwoNumbersFloats() {
    double resultAdd = mCalculator.add(1.111f, 1.111d);
    assertThat(resultAdd, is(closeTo(2.222, 0.01)));
}

@Test
public void subTwoNumbers() {
    double resultSub = mCalculator.sub(1d, 1d);
    assertThat(resultSub, is(equalTo(0d)));
}

@Test
public void subWorksWithNegativeResult() {
    double resultSub = mCalculator.sub(1d, 17d);
    assertThat(resultSub, is(equalTo(-16d)));
}

@Test
public void multTwoNumbers() {
    double resultMul = mCalculator.mul(32d, 2d);
    assertThat(resultMul, is(equalTo(64d)));
}

@Test
public void divTwoNumbers() {
    double resultDiv = mCalculator.div(32d, 2d);
    assertThat(resultDiv, is(equalTo(16d)));
}

@Test
public void divTwoNumbersZero() {
    double resultDiv = mCalculator.div(32d, 0);
    assertThat(resultDiv, is(equalTo(Double.POSITIVE_INFINITY)));
}

```

Kode Solusi

Proyek Android Studio: [Pengujian SimpleCalc](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan 1: Membagi dengan nol selalu pantas diujikan, karena itu adalah kasus spesial dalam aritmetika. Jika Anda mencoba membagi dengan nol dalam versi terbaru aplikasi SimpleCalc, aplikasi akan berperilaku seperti yang didefinisikan Java: Membagi angka dengan akan mengembalikan konstanta "Infinity" (Double.POSITIVE_INFINITY). Membagi 0 dengan 0 menghasilkan konstanta bukan angka (Double.nan). Walaupun nilai ini benar untuk Java, nilai-nilai ini bukan berarti nilai yang berguna untuk pengguna dalam aplikasi itu sendiri. Bagaimana Anda mengubah aplikasi untuk lebih baik dalam menangani dibagi dengan nol? Untuk menyelesaikan tantangan ini, pertama mulailah dengan pengujian -- pertimbangkan perilaku apa yang benar, lalu tulis pengujian seakan-akan perilaku tersebut sudah ada. Lalu ubah atau tambahkan kode agar pengujian muncul berwarna hijau.

Tantangan 2: Kadang-kadang sulit mengisolasi unit kode dari semua dependensi eksternalnya. Daripada mengelola kode secara artifisial dengan cara yang rumit hanya supaya dapat diuji dengan lebih mudah, Anda bisa menggunakan framework palsu untuk membuat objek tiruan yang berpura-pura menjadi dependensi. Pelajari framework [Mockito](#), dan

pelajari cara menyiapkannya di Android Studio. Tulis kelas pengujian untuk metode calcButton() dalam SimpleCalc, dan gunakan Mockito untuk menyimulasikan konteks Android tempat pengujian akan dijalankan.

Rangkuman

- Android Studio memiliki fitur bawaan untuk pengujian unit lokal.
 - Pengujian unit lokal menggunakan JVM mesin lokal dan tidak menggunakan framework Android.
 - Pengujian unit ditulis dengan JUnit, framework pengujian unit yang umum untuk Java.
 - Folder "pengujian" (Tampilan Proyek Android Studio) adalah tempat di mana pengujian JUnit berada.
 - Pengujian unit lokal hanya memerlukan paket: org.junit, org.hamcrest dan android.test
 - Anotasi @RunWith(Unit4.class) memerintahkan runner pengujian untuk menjalankan pengujian dalam kelas ini.
 - Anotasi @SmallTest, @MediumTest, dan @LargeTest adalah konvensi yang memudahkan membundel grup pengujian yang serupa.
 - Anotasi @SmallTest menunjukkan bahwa semua pengujian dalam kelas adalah pengujian unit yang tidak memiliki dependensi dan berjalan dalam milidetik.
- Pengujian berinstrumen adalah pengujian yang bekerja pada perangkat Android atau emulator.
 - Pengujian berinstrumen memiliki akses ke framework Android.
- Runner pengujian adalah pustaka atau serangkaian alat yang memungkinkan pengujian terjadi dan hasilnya dicetak ke log.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Menguji Aplikasi Anda](#)

Ketahui Selengkapnya

- [Praktik Terbaik untuk Pengujian](#)
- [Memulai Pengujian](#)
- [Membangun Pengujian Unit Lokal](#)
- [Laman Beranda JUnit 4](#)
- [Referensi JUnit 4 API](#)
- [Laman Beranda Mockito](#)
- [Dukungan Pengujian Android - Pola Pengujian \(video\)](#)
- [Android Testing Codelab](#)
- [Tips untuk Ahli Alat Android: Anotasi Ukuran Pengujian](#)
- [Manfaat Menggunakan assertThat dibandingkan Metode Penegasan dalam Pengujian Unit](#)

3.3: Menggunakan Pustaka Dukungan Android

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1. Menyiapkan proyek Anda untuk mendukung pustaka
- Tugas 2. Mengimplementasikan perilaku tombol
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Android SDK menyertakan sejumlah pustaka secara kolektif yang disebut pustaka dukungan Android. Pustaka ini menyediakan sejumlah fitur yang tidak dibangun ke dalam framework Android, termasuk:

- Versi kompatibel-mundur dari komponen framework: Pustaka dukungan memungkinkan aplikasi berjalan pada versi platform Android lama untuk mendukung fitur-fitur yang tersedia pada versi platform yang lebih baru.
- Elemen layout dan antarmuka pengguna tambahan
- Dukungan untuk faktor bentuk yang berbeda, seperti TV atau perangkat yang dapat dikenakan
- Komponen untuk mendukung elemen desain bahan
- Beragam fitur lainnya seperti dukungan palet, anotasi, dimensi layout berdasarkan persentase, dan preferensi.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah memahami:

- Cara membuat proyek di Android Studio.
- Komponen utama proyek Android Studio (manifes, sumber daya, file Java, file pembangunan gradle).

Yang akan Anda PELAJARI

- Cara memverifikasi apakah pustaka dukungan Android tersedia di Android Studio.
- Cara menunjukkan kelas pustaka dukungan dalam aplikasi.
- Cara membedakan nilai untuk compileSdkVersion, targetSdkVersion, dan minSdkVersion.
- Cara mengenali API yang tidak digunakan lagi atau tidak tersedia dalam kode.
- Tempat menemukan informasi selengkapnya pada pustaka dukungan Android.

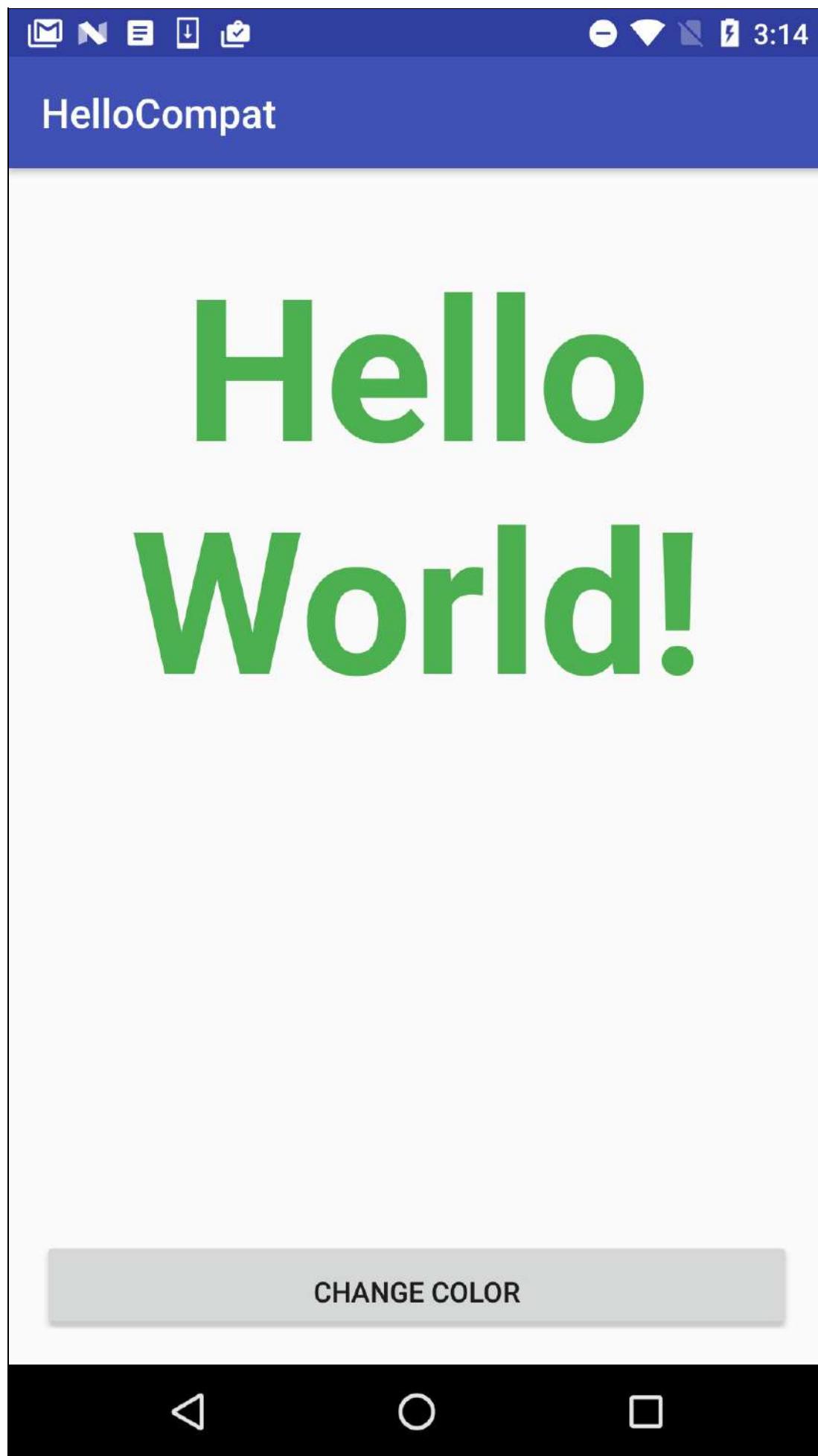
Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Membuat aplikasi baru dengan satu textview dan tombol.
- Memverifikasi bahwa Pustaka Dukungan Android tersedia di sistem.
- Menjelajahi build.gradle untuk aplikasi.
- Mengelola kelas atau panggilan metode yang tidak tersedia untuk versi Android yang didukung aplikasi.
- Menggunakan kelas kompatibilitas dari pustaka dukungan untuk menyediakan kompatibilitas-mundur untuk aplikasi Anda.

Ringkasan Aplikasi

Dalam praktik ini Anda akan membuat aplikasi bernama HelloCompat dengan satu textview yang menampilkan "Hello World" pada layar, dan satu tombol, yang mengubah warna teks. Ada 20 warna yang mungkin, didefinisikan sebagai sumber daya dalam file color.xml, dan setiap klik tombol memilih salah satu dari warna tersebut secara acak.



Metode untuk mendapatkan nilai warna dari sumber daya aplikasi telah berubah dengan versi yang berbeda untuk framework Android. Contoh ini menggunakan kelas ContextCompat, bagian dari pustaka dukungan Android, yang mengizinkan Anda menggunakan metode yang cocok untuk semua versi.

Tugas 1. Menyiapkan proyek

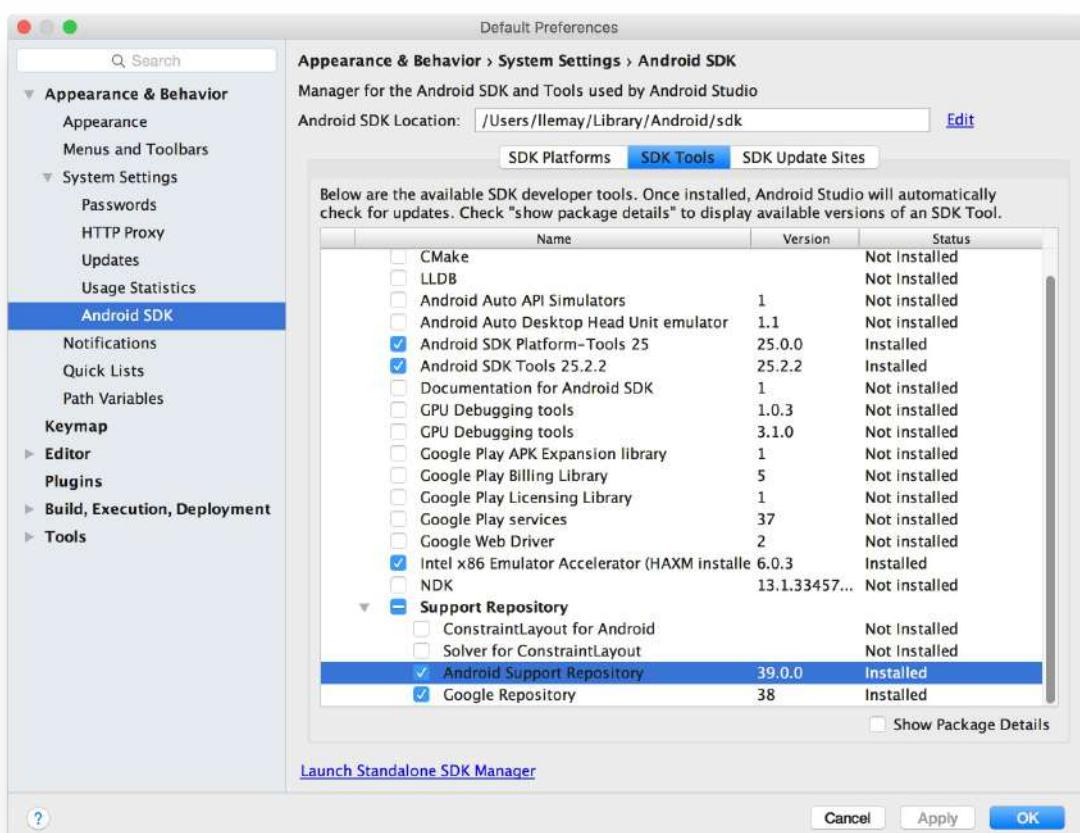
Untuk tugas ini Anda akan menyiapkan proyek baru untuk aplikasi HelloCompat dan mengimplementasikan layout serta perilaku dasar.

1.1 Memverifikasi bahwa Pustaka Dukungan Android tersedia

Pustaka dukungan Android diunduh berdasarkan bagian dari SDK, dan tersedia dalam Android SDK Manager. Dalam Android Studio, Anda akan menggunakan Android Support Repository, repositori lokal untuk pustaka pendukung, untuk mendapatkan akses ke pustaka di dalam file pembangunan gradle. Dalam tugas ini Anda akan memverifikasi bahwa Android Support Repository diunduh dan tersedia untuk proyek.

1. Dalam Android Studio, pilih **Tools > Android > SDK Manager**, atau klik ikon SDK Manager .

Panel preferensi SDK Manager muncul.

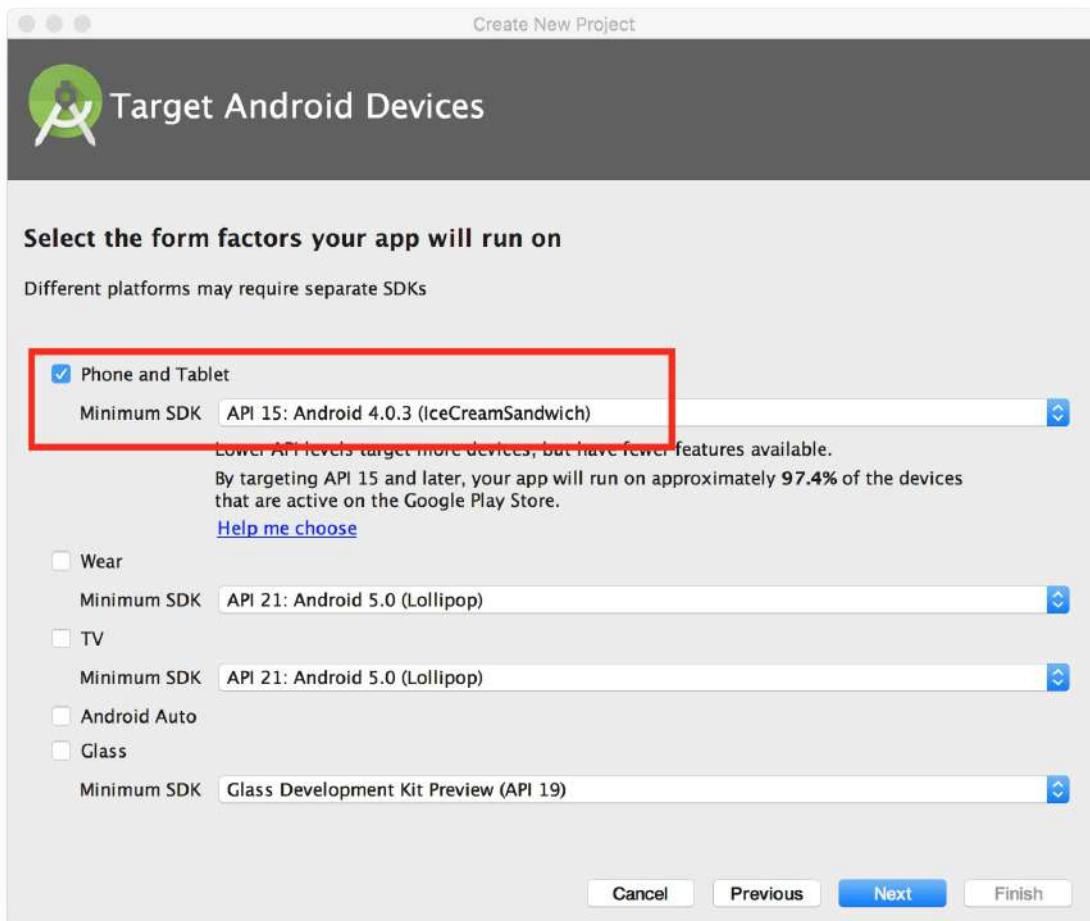


2. Klik tab **SDK Tools** dan perluas Support Repository.
3. Cari **Support Repository** dalam daftar.
4. Jika kata **Installed** muncul di kolom Status, Anda sudah siap. Klik **Cancel**.
5. Jika **Not installed** atau **Update Available** muncul, klik kotak centang di sebelah Android Support Repository. Ikon unduh seharusnya muncul di samping kotak centang. Klik **OK**.
6. Klik **OK** lagi, lalu klik **Finish** saat repositori pendukung sudah dipasang.

1.2 Menyiapkan Proyek dan memeriksa build.gradle

- Buat proyek baru bernama **HelloCompat**, dan pilih template Empty Activity.

Pada laman Perangkat Android Target, catat bahwa API 15: Android 4.0.3 (IceCreamSandwich) dipilih untuk SDK minimum. Seperti yang sudah Anda pelajari pada pelajaran sebelumnya, ini adalah versi terlama platform Android yang akan didukung oleh aplikasi Anda.



- Di Android Studio, pastikan panel **Project** dibuka dan tab **Android** diklik.
- Perluas **Gradle Scripts**, jika perlu, dan buka file **build.gradle (Module: app)**.

Perhatikan bahwa build.gradle untuk keseluruhan proyek (build.gradle (Project: app_name)) adalah file yang berbeda dari build.gradle untuk modul aplikasi.

- Temukan baris compileSdkVersion di dekat bagian atas file.

```
compileSdkVersion 24
```

Versi *kompilasi* adalah versi framework Android yang menyusun aplikasi Annda dalam Android Studio. Untuk proyek baru, versi kompilasi adalah serangkaian framework API terbaru yang Anda pasang. Nilai ini memengaruhi *hanya* Android Studio itu sendiri dan peringatan atau kesalahan yang Anda dapatkan dalam Android Studio, jika menggunakan API yang lebih lama atau lebih baru.

- Temukan baris minSdkVersion dalam bagian defaultConfig beberapa baris di bawah.

```
minSdkVersion 15
```

Versi *minimum* adalah versi API Android terlama tempat aplikasi Anda dijalankan. Ini sama dengan angka yang Anda pilih pada Langkah 1 saat membuat proyek. Google Play store menggunakan angka ini untuk memastikan aplikasi dapat berjalan pada perangkat pengguna yang diberikan Android Studio juga menggunakan angka ini untuk mengingatkan Anda tentang penggunaan API yang sudah tidak digunakan lagi.

6. Temukan baris targetSdkVersion dalam bagian defaultConfig.

```
targetSdkVersion 24
```

Versi *target* menunjukkan versi API yang didesain dan diuji untuk aplikasi Anda. Jika API platform Android lebih tinggi dari angka ini (yakni aplikasi Anda berjalan pada perangkat yang lebih baru), platform dapat mengaktifkan perilaku kompatibilitas untuk memastikan aplikasi terus bekerja seperti yang didesain. Contohnya, Android 6.0 (API 23) menyediakan model izin waktu proses baru. Jika aplikasi menargetkan level API yang lebih rendah, platform kembali ke model izin waktu proses yang lebih lama.

Walaupun SDK target bisa berupa nomor yang sama dengan SDK kompilasi, sering kali nomornya lebih rendah yang menunjukkan versi terbaru API yang sudah diuji untuk aplikasi Anda.

7. Temukan bagian `dependencies` build.gradle, dekat bagian akhir file.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile(
        'com.android.support.test.espresso:espresso-core:2.2.2',
        {
            exclude group: 'com.android.support',
            module: 'support-annotations'
        })
    compile 'com.android.support:appcompat-v7:24.2.1'
    testCompile 'junit:junit:4.12'
}
```

Bagian dependensi untuk proyek baru menyertakan beberapa dependensi untuk mengaktifkan pengujian dengan Espresso, JUnit, dan pustaka dukungan v7 appcompat. Catat bahwa nomor versi pustaka ini dalam proyek Anda bisa berbeda dengan yang ditunjukkan di sini.

Pustaka dukungan v7 appcompat menyediakan kompatibilitas-mundur untuk versi Android yang lebih lama hingga ke API 9. Ini juga termasuk pustaka v4 compat, jadi Anda tidak perlu menambahkan keduanya sebagai dependensi.

8. Perbarui nomor versi, jika perlu.

Jika nomor versi pustaka terbaru lebih rendah dari nomor versi pustaka yang saat ini tersedia, Android Studio akan menyorot baris dan memperingati Anda bahwa versi baru tersedia. ("a newer version of com.android.support:appcompat-v7 is available"). Edit nomor versi ke versi terbaru.

Tips: : Anda juga bisa mengeklik di mana saja pada baris yang disorot dan mengetikkan ****Alt-Enter**** (****Option-Enter**** di Mac). Pilih "Change to XX.X.X" dari menu, di mana XX.X.X adalah versi paling baru yang tersedia.

9. Perbarui nomor compileSdkVersion, jika perlu.

Nomor versi utama pustaka dukungan (nomor pertama) harus cocok dengan compileSdkVersion. Saat Anda memperbarui versi pustaka dukungan Anda mungkin juga perlu memperbarui compileSdkVersion agar sesuai.

10. Klik **Sync Now** untuk menyinkronkan file gradle terbaru dengan proyek, jika diminta.
11. Pasang file platform SDK yang hilang, jika perlu.

Jika Anda memperbarui compileSdkVersion, Anda mungkin perlu memasang komponen platform SDK agar cocok. Klik **Install missing platform(s) and sync project** untuk memulai proses ini.

1.3 Menambahkan layout dan warna

Dalam tugas ini, modifikasi layout untuk aplikasi.

1. Buka `res/layout/activity_main.xml`. Dalam Layout Editor, klik tab Text untuk melihat editor XML.
2. Modifikasi elemen `TextView` agar memiliki atribut berikut:

| Atribut | Nilai |
|------------------------------------|--------------------------------------------------|
| <code>android:id</code> | <code>"@+id/hello_textview"</code> |
| <code>android:layout_width</code> | <code>"match_parent"</code> |
| <code>android:layout_height</code> | <code>"wrap_content"</code> |
| <code>android:padding</code> | <code>"@dimen/activity_horizontal_margin"</code> |
| <code>android:gravity</code> | <code>"center"</code> |
| <code>android:textSize</code> | <code>"100sp"</code> |
| <code>android:textStyle</code> | <code>"bold"</code> |
| <code>android:text</code> | <code>"Hello World!"</code> |

3. Ekstrak string untuk "Hello World" ke dalam sumber daya string.
4. Tambahkan tampilan Tombol di bawah `TextView`, dan tambahkan atribut berikut:

| Atribut | Nilai |
|-----------------------------------------------|------------------------------------------------|
| <code>android:id</code> | <code>"@+id/color_button"</code> |
| <code>android:layout_width</code> | <code>"match_parent"</code> |
| <code>android:layout_height</code> | <code>"wrap_content"</code> |
| <code>android:layout_alignParentBottom</code> | <code>"true"</code> |
| <code>android:paddingTop</code> | <code>"@dimen/activity_vertical_margin"</code> |
| <code>android:text</code> | <code>"Change Color"</code> |
| <code>android:onClick</code> | <code>"changeColor"</code> |

5. Ekstrak string untuk "Change Color" ke dalam sumber daya string.

6. Buka `res/values/colors.xml`.

7. Tambahkan sumber daya warna berikut ke file:

```
<color name="red">#F44336</color>
<color name="pink">#E91E63</color>
<color name="purple">#9C27B0</color>
<color name="deep_purple">#673AB7</color>
<color name="indigo">#3F51B5</color>
<color name="blue">#2196F3</color>
<color name="light_blue">#03A9F4</color>
<color name="cyan">#00BCD4</color>
<color name="teal">#009688</color>
<color name="green">#4CAF50</color>
<color name="light_green">#8BC34A</color>
<color name="lime">#CDDC39</color>
<color name="yellow">#FFEB3B</color>
<color name="amber">#FFC107</color>
<color name="orange">#FF9800</color>
<color name="deep_orange">#FF5722</color>
<color name="brown">#795548</color>
<color name="grey">#9E9E9E</color>
<color name="blue_grey">#607D8B</color>
<color name="black">#000000</color>
```

Nilai dan nama warna berasal dari palet warna yang direkomendasikan untuk aplikasi Android yang didefinisikan di [Desain Material - Gaya - Warna](#). Kode menunjukkan nilai RGB warna dalam heksadesimal.

Kode Solusi (activity_main.xml)

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.hellocompat.MainActivity">

    <TextView
        android:id="@+id/hello_textview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="@dimen/activity_horizontal_margin"
        android:text="@string/hello_text_string"
        android:textSize="100sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/color_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:onClick="changeColor"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:text="@string/button_label" />
</RelativeLayout>

```

1.4 Menambahkan perilaku ke MainActivity

Dalam tugas ini Anda akan menyelesaikan persiapan proyek dengan menambahkan variabel privat dan mengimplementasikan onCreate() dan onSaveInstanceState().

1. Buka MainActivity.java.
2. Tambahkan variabel privat di bagian atas kelas untuk menampung objek TextView untuk textview Hello World.

```
private TextView mHelloTextView;
```

3. Tambahkan larik warna berikut ini setelah variabel privat:

```
private String[] mColorArray = {"red", "pink", "purple", "deep_purple",
    "indigo", "blue", "light_blue", "cyan", "teal", "green",
    "light_green", "lime", "yellow", "amber", "orange", "deep_orange",
    "brown", "grey", "blue_grey", "black"
};
```

Setiap nama warna cocol dengan nama sumber daya warna dari color.xml.

4. Dalam metode onCreate(), gunakan findViewById() untuk mendapatkan referensi tentang contoh TextView dan menetapkannya ke variabel privat tersebut:

```
mHelloTextView = (TextView) findViewById(R.id.hello_textview);
```

5. Juga pada onCreate(), pulihkan status instance tersimpan, jika ada:

```
// restore saved instance state (the text color)
if (savedInstanceState != null) {
    mHelloTextView.setTextColor(savedInstanceState.getInt("color"));
}
```

6. Tambahkan metode onSaveInstanceState() ke MainActivity untuk menyimpan warna teks:

```

@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    // save the current text color
    outState.putInt("color", mHelloTextView.getCurrentTextColor());
}

```

Kode Solusi (bukan seluruh kelas)

```

// Text view for Hello World.
private TextView mHelloTextView;
// array of color names, these match the color resources in color.xml
private String[] mColorArray = {"red", "pink", "purple", "deep_purple",
    "indigo", "blue", "light_blue", "cyan", "teal", "green",
    "light_green", "lime", "yellow", "amber", "orange", "deep_orange",
    "brown", "grey", "blue_grey", "black"
};
...
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Initialize the main text view
    mHelloTextView = (TextView) findViewById(R.id.hello_textview);

    // restore saved instance state (the text color)
    if (savedInstanceState != null) {
        mHelloTextView.setTextColor(savedInstanceState.getInt("color"));
    }
}
...
@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    // save the current text color
    outState.putInt("color", mHelloTextView.getCurrentTextColor());
}

```

Tugas 2. Mengimplementasikan perilaku tombol

Tombol Change Color dalam aplikasi HelloCompat memilih satu dari 20 warna dari file sumber daya color.xml secara acak dan menyetel warna teks ke warna tersebut. Dalam tugas ini Anda akan mengimplementasikan perilaku onClick() untuk handler ini.

2.1 Tambahkan handler onClick changeButton()

1. Buka `res/layout/activity_main.xml`, jika belum terbuka.
2. Klik di mana saja dalam atribut `android:onClick`, di dalam elemen `Button`.
3. Tekan **Alt-Enter (Option-Enter di Mac)**, dan pilih **Create onClick event handler**.
4. Pilih **MainActivity** klik **OK**.

Tindakan ini akan membuat stub metode placeholder untuk metode `changecolor()` dalam `MainActivity.java`.

2.2 Implementasikan tindakan tombol

1. Buka `MainActivity.java`, jika belum terbuka.
2. Dalam metode `changeColor()`, buat objek nomor secara acak.

```
Random random = new Random();
```

Gunakan kelas Random (kelas Java) untuk menghasilkan nomor acak sederhana.

3. Gunakan instance acak untuk memilih warna secara acak dari larik mColorArray:

```
String colorName = mColorArray[random.nextInt(20)];
```

Metode nextInt() dengan argumen 20 akan mendapatkan integer acak antara nomor 0 sampai 19. Gunakan integer itu sebagai indeks larik untuk mendapatkan nama warna.

4. Dapatkan identifier (integer) untuk nama warna dari sumber daya:

```
int colorResourceName = getResources().getIdentifier(colorName,
    "color", getApplicationContext().getPackageName());
```

Saat aplikasi Anda dikompilasi, sistem Android mengonversi definisi dalam file XML ke dalam sumber daya dengan ID integer internal. Ada ID terpisah untuk nama dan nilai. Baris ini cocok dengan string warna dari larik colorName dengan ID nama warna yang cocok dalam file sumber daya XML. Metode getResources() mendapatkan semua sumber daya untuk aplikasi Anda. Metode getIdentifier() mencari nama warna (string) dalam sumber daya warna ("color") untuk nama paket saat ini.

5. Dapatkan ID integer dari sumber daya dan tetapkan ke variabel colorRes:

```
int colorRes = getResources().getColor(colorResourceName);
```

Metode getResources mendapatkan serangkaian sumber daya untuk aplikasi Anda, dan metode getColor() mengambil warna spesifik dari sumber daya itu berdasarkan ID nama warna.

Perhatikan bahwa metode getColor() muncul dengan coretan dalam editor Android Studio. Jika Anda mengarahkan mouse ke getColor(), pesan kesalahan "getColor(int) is deprecated" muncul. Dalam API 23, metode getColor() dimodifikasi untuk menyertakan argumen kedua untuk tema aplikasi. Karena aplikasi Anda memiliki compileSdkVersion 24 (atau yang lebih tinggi), Android Studio memberikan peringatan bahwa Anda menggunakan metode lama yang tidak digunakan lagi.

Anda masih bisa mengompilasi aplikasi dan aplikasi akan tetap berjalan baik pada perangkat Android baru maupun perangkat yang lama. Peringatan penghentian penggunaan adalah peringatan, bukan kesalahan. Namun, sebaiknya jangan abaikan peringatan, jika ada, karena metode yang sudah tidak digunakan lagi bisa mengakibatkan perilaku tak terduga.

6. Ubah baris penetapan colorRes untuk menyertakan argumen kedua ke getColor():

```
int colorRes =
    getResources().getColor(colorResourceName, this.getTheme());
```

Anda bisa menggunakan metode getTheme() untuk mendapatkan tema untuk konteks aplikasi saat ini. Hanya dengan perubahan ini Anda sekarang akan mengingat bahwa getColor() memiliki sorotan bergaris bawah merah. Jika Anda mengarahkan cursor ke getColor(), Android Studio akan melaporkan: "Call requires API 23 (current min is 15)". Karena minSdkVersion Anda adalah 15, Anda akan mendapatkan pesan ini jika mencoba menggunakan API yang diperkenalkan setelah API 15. Anda masih bisa mengompilasi aplikasi, tetapi karena versi baru getColor() dengan dua argumen ini tidak tersedia pada perangkat sebelum API 23, Aplikasi akan crash saat pengguna mengetuk tombol Change.

Pada tahap ini Anda bisa memeriksa versi platform dan menggunakan versi getColor() yang benar tergantung di mana aplikasi berjalan (Anda akan tetap menerima peringatan untuk kedua panggilan di Android Studio). Cara yang lebih baik untuk mendukung API Android lama dan baru tanpa peringatan adalah dengan menggunakan salah satu dari

kelas kompatibilitas dalam pustaka dukungan.

7. Ubah baris penetapan colorRes untuk menggunakan kelas ContextCompat:

```
int colorRes = ContextCompat.getColor(this, colorResourceName);
```

ContextCompat menyediakan banyak metode kompatibilitas untuk menyelesaikan perbedaan API dalam konteks dan sumber daya aplikasi. Metode getColor() dalam ContextCompat memerlukan dua argumen: konteks saat ini (di sini, instance aktivitas, ini), dan nama warna.

Implementasi metode ini dalam pustaka dukungan menyembunyikan perbedaan implementasi dalam versi API yang berbeda. Anda bisa memanggil metode ini terlepas dari SDK kompilasi atau versi SDK minimum tanpa peringatan, kesalahan, atau crash.

8. Setel warna tampilan teks Hello World ke ID sumber daya warna:

```
mHelloTextView.setTextColor(colorRes);
```

9. Kompilasi dan jalankan aplikasi pada perangkat atau emulator.

Tombol Change Color sekarang seharusnya mengubah warna teks dalam Hello World.

Kode Solusi (hanya metode changeColor())

```
public void changeColor(View view) {
    // get a random color name from the color array (20 colors)
    Random random = new Random();
    String colorName = mColorArray[random.nextInt(20)];

    // get the color identifier that matches the color name
    int colorResourceName = getResources().getIdentifier(colorName, "color",
        getApplicationContext().getPackageName());

    // get the color ID from the resources
    int colorRes = ContextCompat.getColor(this, colorResourceName);

    // Set the text color
    mHelloTextView.setTextColor(colorRes);
}
```

Kode Solusi

Proyek Android Studio: [HelloCompat](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Sebagai ganti menggunakan ContextCompat untuk mendapatkan sumber daya warna, gunakan pengujian nilai dalam kelas [Build](#) untuk melakukan operasi yang berbeda jika aplikasi berjalan pada perangkat yang mendukung kurang dari API 23.

Rangkuman

Dalam praktik ini Anda telah mempelajari bahwa:

- Android menggunakan tiga arahan untuk menunjukkan bagaimana aplikasi harusnya berperilaku untuk versi API yang berbeda:

- minSdkVersion: versi API minimum untuk dukungan aplikasi Anda.
- compileSdkVersion: versi API yang harus dikompilasi dengan aplikasi Anda.
- targetSdkVersion: versi API untuk aplikasi Anda.
- Pustaka Dukungan Android dapat dipasang dalam SDK Manager
- Anda bisa menambahkan dependensi pustaka untuk pustaka dukungan dalam file gradle.build
- Kelas ContextCompat menyediakan metode untuk kompatibilitas dengan konteks dan metode terkait sumber daya untuk kedua level API lama dan baru.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Pustaka Dukungan Android](#)

Ketahui Selengkapnya

- [Pustaka Dukungan Android \(Pengantar\)](#)
- [Persiapan Pustaka Dukungan](#)
- [Fitur Pustaka Dukungan](#)
- [Mendukung Versi Platform Berbeda](#)
- [Memilih compileSdkVersion, minSdkVersion, dan targetSdkVersion](#)
- [All the Things Compat](#)
- [Referensi API](#)(semua paket yang dimulai dengan android.support)

4.1. Menggunakan Keyboard, Kontrol Masukan, Peringatan, dan Picker

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Bereksperimen dengan atribut keyboard entri teks](#)
- [Tugas 2: Mengubah tipe keyboard](#)
- [Tugas 3: Menambahkan kontrol masukan spinner untuk memilih label telepon](#)
- [Tugas 4: Menggunakan dialog untuk peringatan yang memerlukan keputusan](#)
- [Tugas 5: Menggunakan picker untuk masukan pengguna](#)
- [Tugas 6: Menggunakan tampilan gambar sebagai tombol](#)
- [Tugas 7: Menggunakan tombol radio](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Anda bisa menyesuaikan metode masukan untuk memudahkan pengguna memasukkan data.

Dalam praktik ini, Anda akan mempelajari cara:

- Menggunakan kontrol dan keyboard di layar yang berbeda untuk masukan pengguna.
- Menampilkan pesan peringatan yang bisa berinteraksi dengan pengguna.
- Memberikan elemen antarmuka untuk memilih tanggal dan waktu.
- Menggunakan gambar sebagai tombol untuk meluncurkan aktivitas.
- Menambahkan tombol radio bagi pengguna untuk memilih satu item dari serangkaian item.

Yang harus sudah Anda KETAHUI

Untuk praktik ini, Anda harus bisa:

- Membuat proyek Android Studio dari template dan membuat layout baru.
- Menjalankan aplikasi pada emulator atau perangkat yang terhubung.
- Membuat salin proyek aplikasi, dan mengubah nama aplikasi.
- Membuat dan mengedit elemen UI menggunakan Layout Editor dan kode XML.
- Mengakses elemen UI dari kode Anda menggunakan `findViewById()`.
- Mengonversi teks dalam tampilan menjadi string menggunakan `getText().toString()`.
- Menangani klik tombol.
- Menampilkan pesan toast.
- Memulai aktivitas dengan aplikasi lain menggunakan intent implisit.
- Menggunakan adaptor untuk menghubungkan data Anda ke tampilan, seperti RecyclerView di pelajaran sebelumnya.

Yang akan Anda PELAJARI

Dalam praktik ini, Anda akan mempelajari cara:

- Mengubah metode masukan untuk mengaktifkan saran ejaan, kapitalisasi otomatis, dan pengaburan sandi.
- Mengubah keyboard di layar generik menjadi keypad ponsel atau keyboard khusus lainnya.

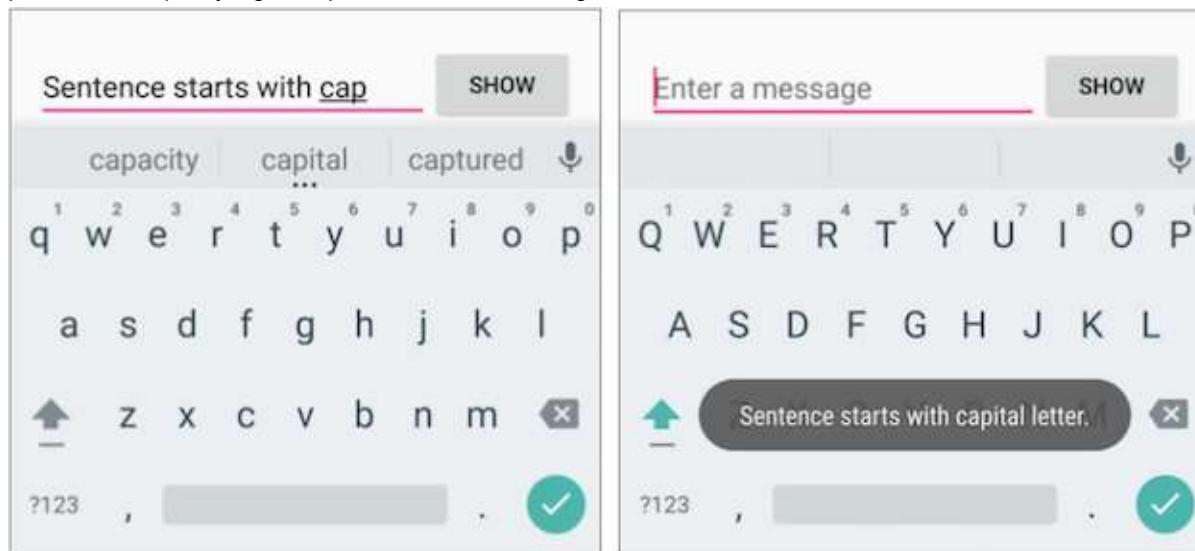
- Menambahkan kontrol masukan spinner untuk menampilkan menu tarik-turun dengan nilai, tempat pengguna bisa memilih salah satunya.
- Menambahkan peringatan dengan Oke dan Batal untuk mengambil keputusan pengguna.
- Menggunakan picker tanggal dan waktu dan merekam pilihan.
- Menggunakan gambar sebagai tombol untuk meluncurkan aktivitas.
- Menambahkan tombol radio bagi pengguna untuk memilih satu item dari serangkaian item.

Apa yang akan Anda LAKUKAN

- Membuat proyek Android Studio baru untuk menampilkan keyboard, spinner, peringatan, dan picker waktu dan tanggal.
- Memberikan saran ejaan saat pengguna memasukkan teks, dan otomatis mengubah huruf besar pada kalimat baru dengan bereksperimen dengan metode masukan.
- Bereksperimen dengan dengan atribut tipe masukan untuk mengubah keyboard di layar menjadi keyboard khusus untuk memasukkan alamat email, lalu menjadi keypad angka untuk memaksa entri angka.
- Menambahkan kontrol masukan spinner untuk bidang nomor telepon untuk memilih satu nilai dari serangkaian nilai.
- Membuat proyek baru dengan dialog peringatan untuk memberi tahu pengguna untuk membuat keputusan, misalnya Oke atau Batal.
- Menambahkan picker tanggal dan picker waktu ke proyek baru dan menggunakan listener untuk merekam pilihan pengguna.
- Membuat proyek baru agar untuk menggunakan gambar sebagai tombol.
- Membuat aktivitas kedua dan menambahkan tombol radio untuk memilih opsi.
- Menyetel handler `onClick` untuk gambar yang digunakan sebagai tombol untuk meluncurkan aktivitas kedua.

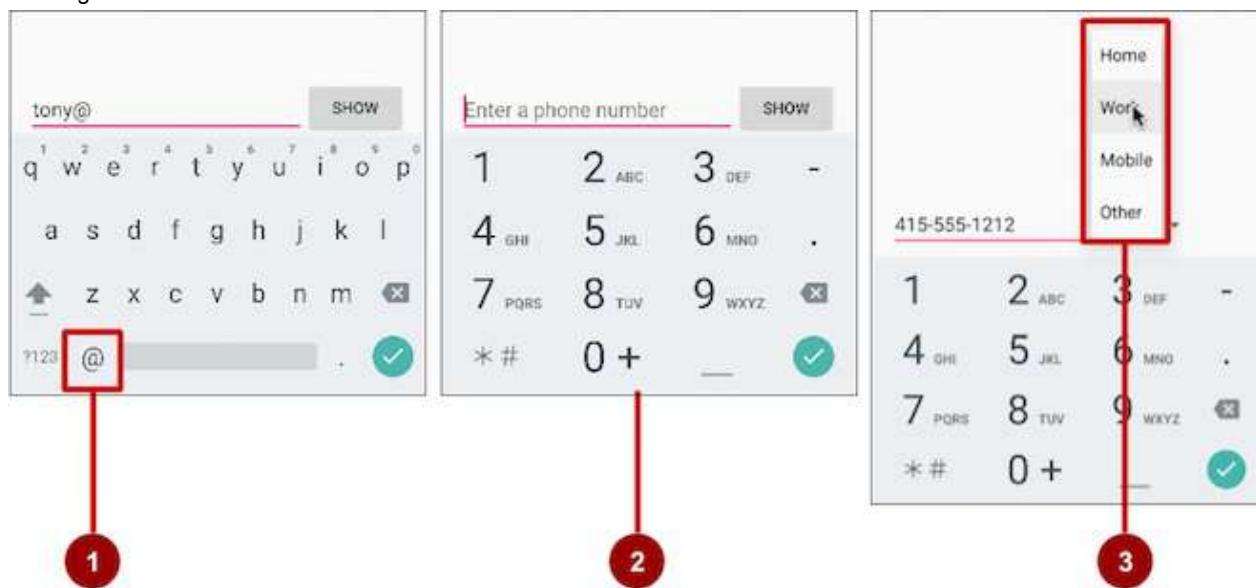
Ringkasan Aplikasi

Dalam praktik ini, Anda akan membuat dan membangun aplikasi baru yang bernama Keyboard Samples untuk bereksperimen dengan dengan atribut `android:inputType` untuk elemen UI `EditText`. Anda akan mengubah keyboard sehingga akan menyarankan pembetulan ejaan dan mengubah kapitalisasi setiap kalimat, seperti yang ditampilkan di sebelah kiri gambar di bawah ini. Untuk menyederhanakan aplikasi, Anda akan menampilkan teks yang dimasukkan dalam pesan `toast`, seperti yang ditampilkan di sebelah kanan gambar di bawah ini.



Anda juga akan mengubah keyboard menjadi keyboard yang menampilkan simbol "@" di lokasi yang umum untuk memasukkan alamat email, dan menjadi keypad ponsel untuk memasukkan nomor telepon, seperti yang ditampilkan di sebelah kiri dan di tengah gambar di bawah ini. Sebagai tantangan, Anda akan mengimplementasikan listener untuk kunci tindakan di keyboard untuk mengirimkan intent yang implisit ke aplikasi lain untuk memutar nomor telepon.

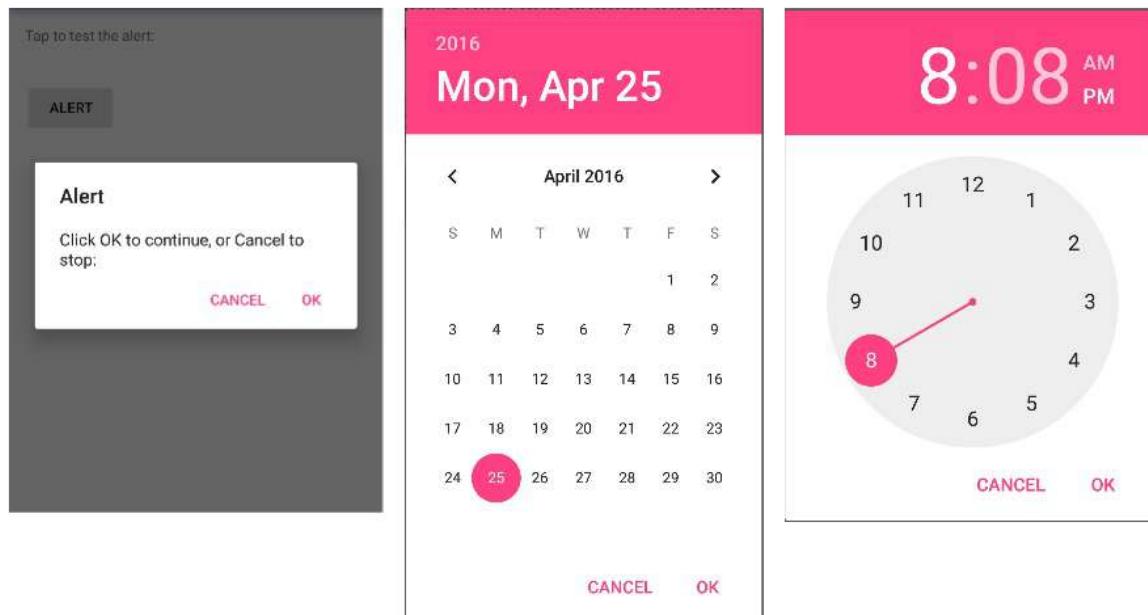
Anda kemudian akan menyalin aplikasi untuk membuat Spinner Nomor Telepon yang menawarkan kontrol masukan spinner untuk memilih label (Rumah, Kantor, Lainnya, Khusus) untuk nomor telepon, seperti yang ditampilkan di sebelah kanan gambar di bawah ini.

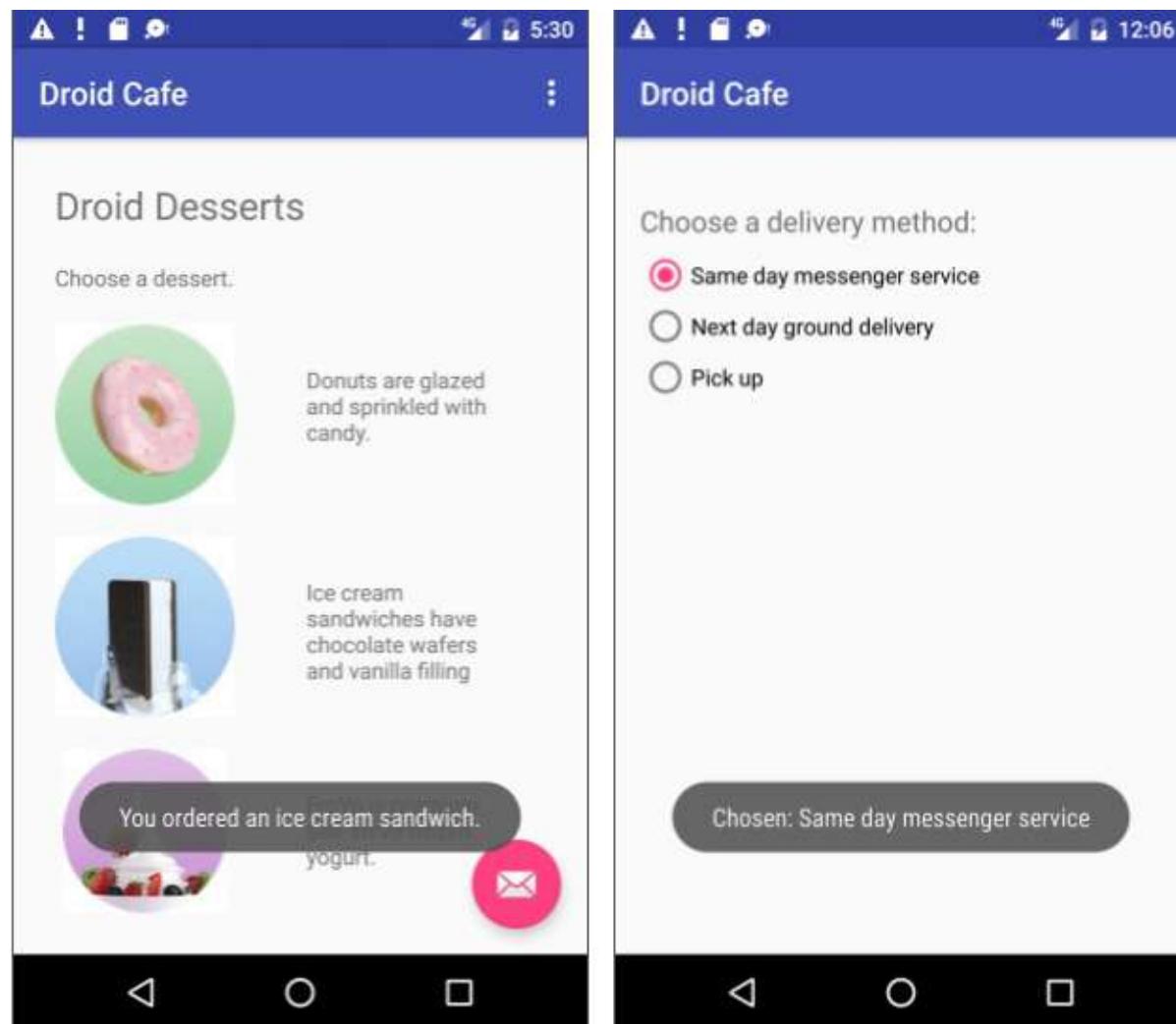


Gambar di atas menampilkan yang berikut:

1. Keyboard email dengan simbol "@" pada lokasi yang mudah ditemukan
2. Keypad ponsel
3. Spinner

Anda juga akan membuat Contoh Peringatan untuk bereksperimen dengan dialog peringatan, yang ditampilkan di sebelah kiri gambar di bawah ini, dan Picker Tanggal Waktu untuk bereksperimen dengan picker tanggal dan picker waktu, yang ditampilkan di tengah dan kiri gambar di bawah ini, dan menggunakan pemilihan waktu dan tanggal di aplikasi Anda.





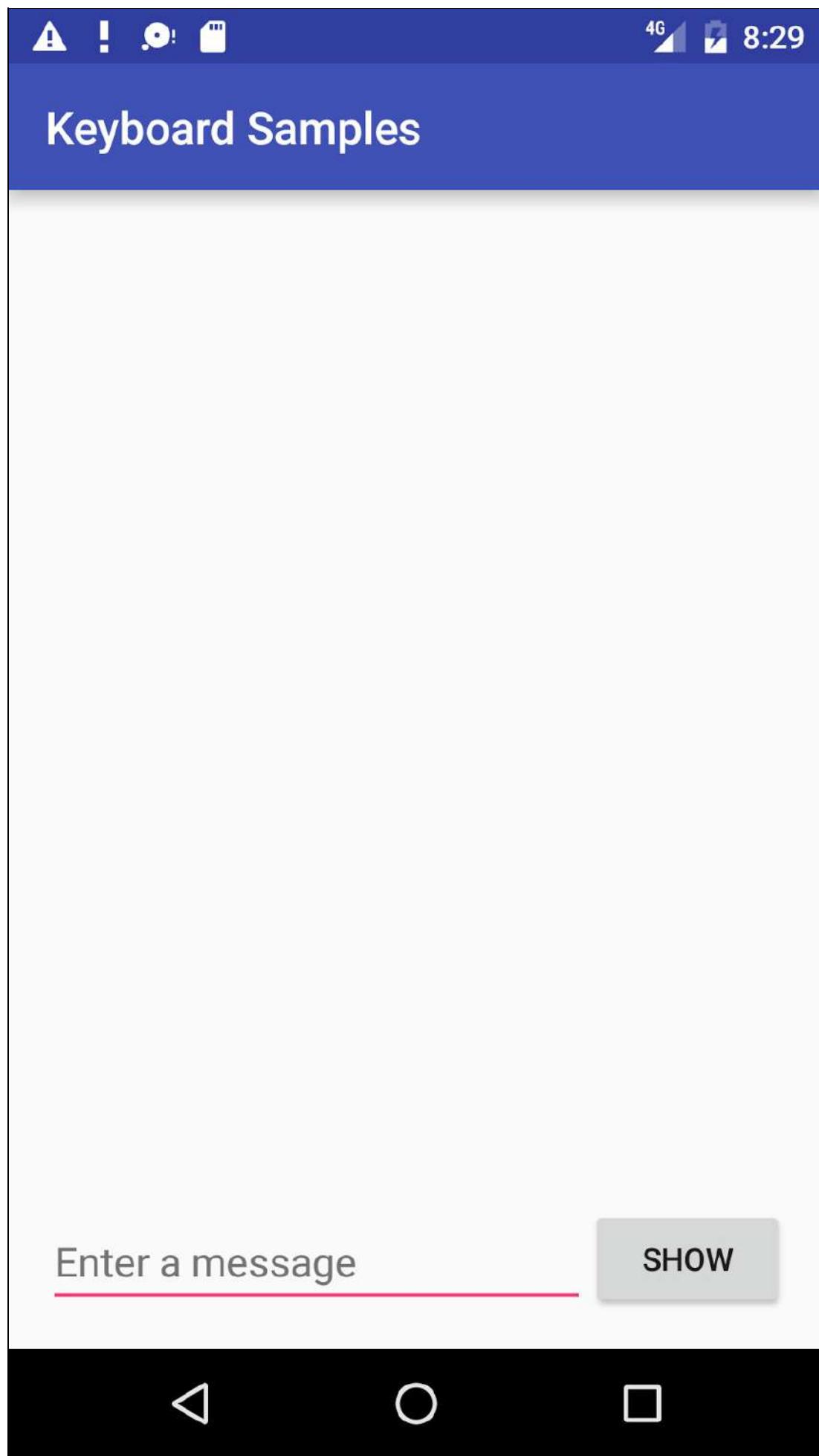
Tugas 1: Bereksperimen dengan atribut keyboard entri teks

Menyentuh bidang teks `EditText` yang bisa diedit akan menempatkan kursor pada bidang teks dan otomatis menampilkan keyboard di layar. Anda akan mengubah atribut bidang entri teks sehingga keyboard akan menyarankan pembetulan ejaan selagi Anda mengetik, dan otomatis memulai setiap kalimat baru dengan huruf kapital. Misalnya:

- `android:inputType="textCapSentences"` : Menyetel keyboard untuk menjadikan huruf besar di awal kalimat.
- `android:inputType="textAutoCorrect"` : Menyetel keyboard untuk menampilkan koreksi ejaan otomatis selagi Anda memasukkan karakter.
- `android:inputType="textMultiLine"` : Mengaktifkan kunci Return pada keyboard untuk mengakhiri baris dan membuat baris kosong baru tanpa menutup keyboard.
- `android:inputType="textPassword"` : Menyetel karakter yang dimasukkan pengguna agar menjadi titik-titik untuk menyembunyikan sandi yang dimasukkan.

1.1 Membuat layout utama dan metode showText

Anda akan menambahkan Tombol, dan mengubah elemen `TextView` menjadi elemen `EditText` sehingga pengguna bisa memasukkan teks. Layout aplikasi akan tampak seperti gambar berikut.



1. Buat proyek baru yang bernama **Keyboard Samples**, dan pilih template Empty Activity.
2. Buka file layout **activity_main.xml** untuk mengedit kode XML.
3. Tambahkan Tombol di atas elemen `TextView` yang sudah ada dengan atribut berikut:

| Atribut Tombol | Nilai Baru |
|-----------------------------------------------|--------------------|
| <code>android:id</code> | "@+id/button_main" |
| <code>android:layout_width</code> | "wrap_content" |
| <code>android:layout_height</code> | "wrap_content" |
| <code>android:layout_alignParentBottom</code> | "true" |
| <code>android:layout_alignParentRight</code> | "true" |
| <code>android:onClick</code> | "showText" |
| <code>android:text</code> | "Show" |

4. Ekstrak sumber daya string untuk nilai atribut `android:text` untuk membuat dan memasukkannya di `strings.xml`: Letakkan kursor pada **Show**, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract string resource**. Lalu ubah name sumber daya untuk nilai string ke **show**.

Anda mengekstrak sumber daya string karena menjadikan proyek aplikasi lebih fleksibel untuk mengubah string. Penetapan sumber daya string disimpan di file **strings.xml** (pada **app > res > values**). Anda bisa mengedit file ini untuk mengubah penetapan string sehingga aplikasi bisa dilokalkan dengan bahasa berbeda. Misalnya, nilai "Show" untuk sumber daya yang bernama `show` bisa diubah ke "Montrer" untuk aplikasi dalam versi bahasa Prancis.

5. Ubah elemen `TextView` yang sudah ada seperti berikut:
 - i. Hapus atribut `android:text` yang menetapkan "Hello World!".
 - ii. Ubah tag `TextView` menjadi tag `EditText` dan pastikan tag diakhiri dengan `>`.
 - iii. Tambahkan atau ubah atribut berikut:

| Atribut EditText | Nilai Lama TextView | Nilai Baru EditText |
|-----------------------------------------------|---------------------|----------------------|
| <code>android:id</code> | | "@+id/editText_main" |
| <code>android:layout_width</code> | "wrap_content" | "match_parent" |
| <code>android:layout_height</code> | "wrap_content" | "wrap_content" |
| <code>android:layout_alignParentBottom</code> | | "true" |
| <code>android:layout_toLeftOf</code> | | "@+id/button_main" |
| <code>android:hint</code> | | "Enter a message" |

Anda telah mempelajari tentang atribut `android:layout_toLeftOf` dan `android:layout_alignParentBottom` di pelajaran sebelumnya. Atribut yang terkait layout ini berfungsi dengan grup tampilan **RelativeLayout** untuk memosisikan tampilan turunan relatif terhadap yang lain atau terhadap induk. Atribut `android:hint` menyetel teks agar tampil dalam bidang yang menyediakan petunjuk bagi pengguna untuk menyediakan masukan, misalnya "Enter Enter a message"

6. Ekstrak sumber daya string untuk nilai atribut `android:hint` "Enter a message" ke nama sumber daya **enter**. File layout **activity_main.xml** sekarang harus terlihat seperti ini:

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.keyboardsamples.MainActivity">

    <Button
        android:id="@+id/button_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:onClick="showText"
        android:text="@string/show" />

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button_main"
        android:hint="@string/enter" />

</RelativeLayout>

```

7. Buka `MainActivity.java` dan masukkan metode `showText` berikut, yang mengambil informasi yang dimasukkan ke dalam elemen `EditText` dan menampilkannya di dalam pesan `toast`

```

public void showText(View view) {
    EditText editText = (EditText) findViewById(R.id.editText_main);
    if (editText != null) {
        String showString = editText.getText().toString();
        Toast.makeText(this, showString, Toast.LENGTH_SHORT).show();
    }
}

```

8. Buka `strings.xml` (dalam `app > res > values`) dan edit nilai `app_name` ke "**Keyboard Samples**" (pastikan untuk menyertakan spasi antara "Keyboard" dan "Samples").

9. Jalankan aplikasi dan periksa cara kerja keyboard.

Mengetuk tombol **Show** akan menampilkan pesan toast entri teks.

Untuk menutup keyboard di layar, ketuk panah yang mengarah ke bawah di baris bawah ikon.

Di layout keyboard standar, ikon centang dalam lingkaran hijau, ditampilkan di bawah, muncul di sudut kanan bawah



keypad. Ini dikenal sebagai tombol Return (atau Enter) dan digunakan untuk memasukkan baris baru:

Dengan atribut default untuk elemen `EditText`, mengetuk tombol Return akan menambahkan baris teks baru. Di bagian berikutnya, Anda akan mengubah keyboard sehingga dapat mengubah huruf besar pada huruf pertama kalimat selagi Anda mengetik. Sebagai hasil menyetel atribut `android:inputType`, atribut default untuk tombol Return berubah untuk mengalihkan fokus dari elemen `EditText` dan menutup keyboard.

1.2 Setel keyboard untuk menjadikan huruf besar di awal kalimat

1. Tambahkan atribut `android:inputType` ke elemen `EditText` menggunakan nilai `textCapSentences` untuk menyetel keyboard menjadi huruf besar di awal kalimat, sehingga pengguna bisa otomatis memulai kalimat dengan huruf besar:

```
    android:inputType="textCapSentences"
```

2. Jalankan aplikasi Anda.

Huruf besar sekarang akan tampil pada keyboard di awal kalimat. Saat Anda mengetuk tombol Return di keyboard, keyboard ditutup dan entri teks Anda berakhir. Anda masih bisa mengetuk bidang entri teks untuk menambahkan lebih banyak teks atau mengedit teks. Ketuk **Show** untuk menampilkan teks di pesan toast.

Untuk detail tentang atribut `android:inputType`, lihat [Menentukan Tipe Metode Masukan](#)

1.3 Setel keyboard untuk menyembunyikan sandi saat memasukkannya

1. Ubah elemen `EditText` untuk menggunakan nilai `textPassword` untuk atribut `android:inputType`.

```
    android:inputType="textPassword"
```

2. Ubah `android:hint` menjadi "**Enter your password**".

3. Jalankan aplikasi.

Karakter yang dimasukkan pengguna berubah menjadi titik-titik untuk menyembunyikan sandi yang dimasukkan. Untuk mendapatkan bantuan, lihat [Bidang Teks](#).

Kode solusi:

Proyek Android: [KeyboardSamples](#)

Tugas 2. Mengubah tipe keyboard

Setiap bidang teks mengharapkan tipe masukan teks tertentu, seperti alamat email, nomor telepon, sandi, atau hanya teks biasa. Penting untuk menetapkan tipe masukan untuk setiap bidang teks di aplikasi Anda sehingga sistem akan menampilkan metode masukan virtual yang sesuai, seperti:

- Keyboard di layar standar untuk teks biasa
- Keyboard untuk alamat email yang menyertakan simbol "@" di lokasi penting
- Keypad ponsel untuk nomor telepon

2.1 Menggunakan keyboard email

Ubah elemen `EditText` aktivitas utama agar menampilkan keyboard email, bukan keyboard standar:

1. Pada elemen `EditText` di file layout **activity_main.xml**, ubah atribut `android:inputType` menjadi yang berikut:

```
    android:inputType="textEmailAddress"
```

2. Ubah atribut `android:hint` menjadi "**Enter an email address**".

3. Ekstrak sumber daya string untuk nilai `android:hint` ke `enter_email`.

4. Jalankan aplikasi. Mengetuk bidang akan memunculkan keyboard email di layar dengan simbol "@" yang terletak di sebelah tombol spasi.

2.2 Menggunakan keypad ponsel

Ubah elemen `EditText` aktivitas utama agar menampilkan keypad ponsel, bukan keyboard standar:

1. Pada elemen `EditText` di file layout **activity_main.xml**, ubah atribut `android:inputType` menjadi yang berikut:

```
    android:inputType="phone"
```

2. Ubah atribut `android:hint` menjadi "**Enter a phone number**".
3. Ekstrak sumber daya string untuk nilai `android:hint` ke `enter_phone`.
4. Jalankan aplikasi.

Mengetuk bidang kini akan memunculkan keypad ponsel di layar sebagai ganti keyboard standar.

Catatan: Saat menjalankan aplikasi di emulator, bidang masih akan menerima teks, bukan angka yang Anda ketikkan di keyboard komputer. Akan tetapi, saat dijalankan di perangkat, bidang hanya menerima angka keypad.

Tugas 3: Menambahkan kontrol masukan spinner untuk memilih label telepon

Kontrol masukan adalah komponen interaktif dalam antarmuka pengguna aplikasi Anda. Android menyediakan aneka ragam kontrol yang bisa Anda gunakan dalam UI, seperti tombol, bilah pencarian, kotak centang, tombol zoom, tombol toggle, spinner, dan masih banyak lagi. (Untuk informasi selengkapnya tentang kontrol masukan, lihat [Kontrol Masukan](#).)

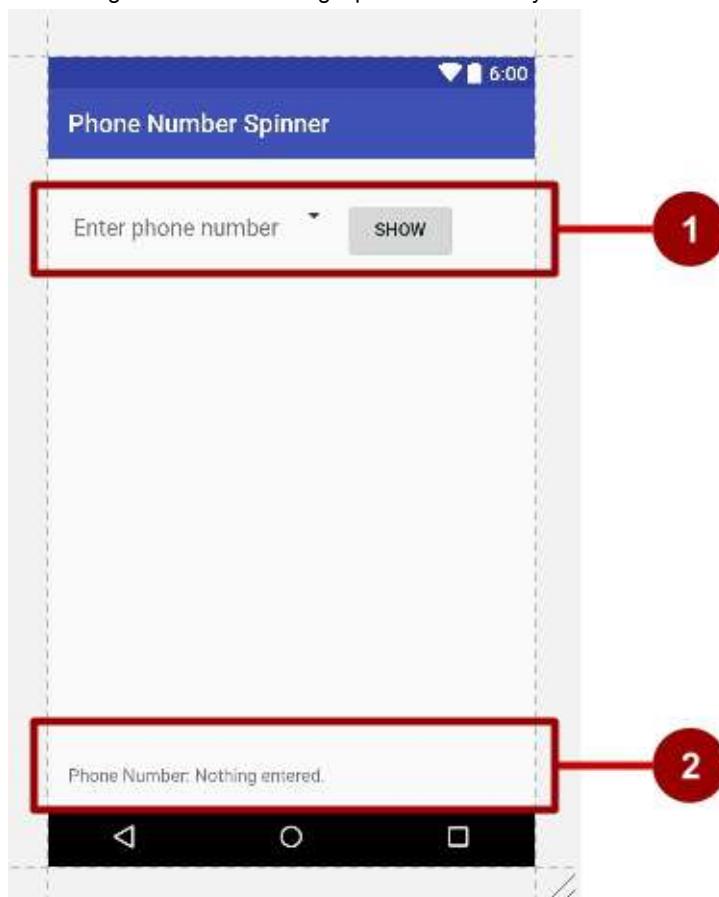
Spinner menyediakan cara cepat untuk memilih salah satu dari serangkaian nilai. Menyentuh spinner akan menampilkan daftar tarik-turun dengan semua nilai yang tersedia, yang bisa dipilih oleh pengguna. Jika hanya menyediakan dua atau tiga pilihan, Anda mungkin ingin menggunakan tombol radio untuk pilihan jika memiliki ruang di layout untuk tombol-tombol itu; akan tetapi, jika ada lebih dari tiga pilihan, spinner bekerja sangat baik, menggulir seperlunya untuk menampilkan item, dan hanya membutuhkan sedikit ruang di layout Anda.

Untuk informasi selengkapnya tentang spinner, lihat [Spinner](#).

Untuk memberikan cara memilih label untuk nomor telepon (misalnya Rumah, Kantor, Seluler, dan Lainnya), Anda bisa menambahkan spinner ke layout agar tampil tepat di sebelah bidang nomor telepon.

3.1 Salin proyek KeyboardSamples dan ubah layout

Gunakan gambar berikut sebagai panduan untuk layout aktivitas utama:



Pada gambar di atas:

1. LinearLayout pertama dengan tampilan EditText, ikon spinner, dan tombol **Show**.
2. LinearLayout kedua dengan dua TextView.

Ikuti langkah-langkah berikut:

1. Salin folder proyek **KeyboardSamples**, ubah namanya menjadi **PhoneNumberSpinner**, dan optimalkan kode untuk mengisikan nama baru di seluruh proyek aplikasi. (Lihat [Apendiks](#) untuk melihat petunjuk tentang menyalin proyek.)
2. Setelah mengoptimalkan kode, ubah nilai " di file **strings.xml** (di dalam **app > res > values**) menjadi **Phone Number Spinner** (berikut spasinya) sebagai nama aplikasi.
3. Bukan file layout **activity_main.xml**.
4. Kurung elemen `EditText` dan `Button` yang ada dari pelajaran sebelumnya di dalam `LinearLayout` dengan orientasi horizontal, dan menempatkan elemen `EditText` di atas `Button` :

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="@dimen/activity_vertical_margin"
    android:orientation="horizontal">

    <EditText
        ...
        <Button
        ...
    </EditText>
```

5. Buat perubahan berikut pada elemen `EditText` dan `Button`:

- i. Hapus atribut berikut ini dari elemen `EditText`:

- `android:layout_toLeftOf`
- `android:layout_alignParentBottom`

- ii. Hapus atribut berikut dari elemen `Button`:

- `android:layout_alignParentRight`
- `android:layout_alignParentBottom`

- iii. Ubah tiga atribut elemen `EditText` lainnya seperti berikut:

| Atribut <code>EditText</code> | Nilai |
|-----------------------------------|----------------------|
| <code>android:layout_width</code> | "wrap_content" |
| <code>android:inputType</code> | "phone" |
| <code>android:hint</code> | "Enter phone number" |

6. Tambahkan elemen `Spinner` di antara elemen `EditText` dan elemen `Button` :

```
<Spinner
    android:id="@+id/label_spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Spinner>
```

Elemen `Spinner` menyediakan daftar tarik-turun. Pada tugas selanjutnya, Anda akan menambahkan kode yang akan mengisi daftar spinner dengan nilai. Kode layout untuk elemen `EditText`, `Spinner`, dan `Button` di dalam `LinearLayout` seharusnya sekarang tampak seperti ini:

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="@dimen/activity_vertical_margin"
    android:orientation="horizontal">

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="phone"
        android:hint="Enter phone number" />

    <Spinner
        android:id="@+id/label_spinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </Spinner>

    <Button
        android:id="@+id/button_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="showText"
        android:text="Show" />

</LinearLayout>

```

7. Tambakan `LinearLayout` lagi di bawah `LinearLayout` yang baru saja Anda buat, dengan orientasi horizontal untuk mengurung dua elemen `TextView` secara bersisian — deskripsi teks dan bidang teks untuk menampilkan nomor telepon dan label telepon — dan meratakan `LinearLayout` ke bawah induknya (lihat gambar di atas):

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_alignParentBottom="true">

    <TextView
        ...
    <TextView
        ...
    </LinearLayout>

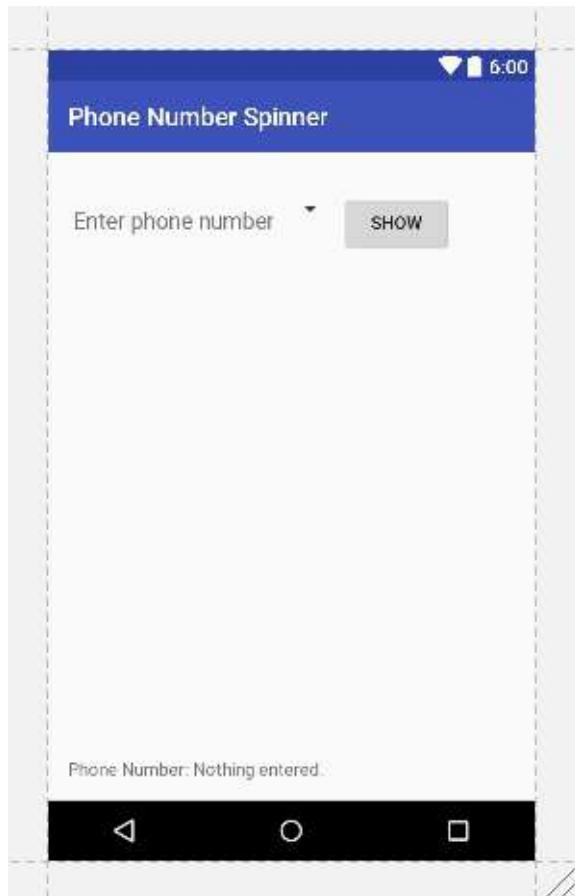
```

8. Tambahkan elemen `TextView` berikut ke dalam `LinearLayout`:

| Atribut <code>TextView</code> | Nilai |
|------------------------------------|----------------------------------------|
| <code>android:id</code> | " <code>@+id/title_phonelabel</code> " |
| <code>android:layout_width</code> | " <code>wrap_content</code> " |
| <code>android:layout_height</code> | " <code>wrap_content</code> " |
| <code>android:text</code> | "Phone Number: " |

| Atribut <code>TextView</code> | Nilai |
|------------------------------------|---------------------------------------|
| <code>android:id</code> | " <code>@+id/text_phonelabel</code> " |
| <code>android:layout_width</code> | " <code>wrap_content</code> " |
| <code>android:layout_height</code> | " <code>wrap_content</code> " |
| <code>android:text</code> | "Nothing entered." |

9. Periksa layout Anda dengan mengeklik tab Preview di sebelah kanan jendela layout.



Anda sekarang seharusnya melihat layar (lihat gambar di atas) yang menampilkan bidang entri telepon di atas sebelah kiri, spinner kerangka di sebelah bidang dan tombol

Show di sebelah kanan. Di bawah, akan muncul text "Phone Number:" yang diikuti oleh "Nothing entered"."

- Ekstrak string Anda ke dalam sumber daya string: Letakkan cursor di string hard-code, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract string resources**. Lalu edit nama Sumber Daya untuk nilai string. Ekstrak seperti berikut:

| Elemen | String | Sumber daya string |
|----------|----------------------|-----------------------------|
| EditText | "Enter phone number" | "@string/hint_phonenumber" |
| Button | "Show" | "@string/show_button" |
| TextView | "Phone Number: " | "@string/phonenumber_label" |
| TextView | "Nothing entered." | "@string/nothing_entered" |

3.2 Tambahkan kode untuk mengaktifkan spinner dan listener-nya

Pilihan untuk spinner label telepon ini adalah string statis yang didefinisikan dengan baik ("Home", "Work", dsb), sehingga Anda bisa menggunakan larik teks yang didefinisikan di strings.xml untuk menyimpan nilai untuknya.

Untuk mengaktifkan spinner dan listener-nya, implementasikan antarmuka `AdapterView.OnItemSelectedListener`, yang juga memerlukan penambahan metode callback `onItemSelected()` dan `onNothingSelected()`.

- Buka **strings.xml** untuk mendefinisikan nilai yang bisa dipilih (**Home**, **Work**, **Mobile**, dan **Other**) untuk spinner sebagai larik string `labels_array`.

```
<string-array name="labels_array">
    <item>Home</item>
    <item>Work</item>
    <item>Mobile</item>
    <item>Other</item>
</string-array>
```

- Untuk mendefinisikan callback pilihan untuk spinner, ubah kelas `MainActivity` Anda untuk mengimplementasikan

antarmuka `AdapterView.OnItemSelectedListener` seperti yang ditampilkan:

```
public class MainActivity extends AppCompatActivity implements
    AdapterView.OnItemSelectedListener {
```

Selagi Anda mengetik `AdapterView`, di pernyataan di atas, Android Studio otomatis mengimpor widget `AdapterView`. Alasan mengapa Anda memerlukan `AdapterView` adalah Anda memerlukan adaptor—khususnya `ArrayAdapter`—untuk menetapkan larik ke spinner. Sebuah *adaptor** menyambungkan data Anda—dalam hal ini, larik item spinner—ke tampilan spinner. Anda akan mempelajari selengkapnya tentang pola penggunaan adaptor untuk menyambungkan data seperti ini di pelajaran lain. Baris ini harus muncul di blok pernyataan impor:

```
import android.widget.AdapterView;
```

Setelah mengetikkan `OnItemSelectedListener` di pernyataan di atas, tunggu beberapa detik sampai bola lampu merah muncul di batas kiri.

3. Klik bola lampu dan pilih **Implement methods**. Metode `onItemSelected()` dan `onNothingSelected()` yang diperlukan untuk `OnItemSelectedListener`, seharusnya sudah disorot, dan opsi "Insert @Override" harus dicentang. Klik **OK**.

Langkah ini otomatis menambahkan metode callback `onItemSelected()` dan `onNothingSelected()` kosong ke bawah kelas `MainActivity`. Kedua metode tersebut menggunakan parameter `AdapterView<?>`. The `<?>` is a Java type wildcard, enabling the method to be flexible enough to accept any type of `AdapterView` sebagai argumen.

4. Buat instance objek spinner di metode `onCreate()` menggunakan elemen `Spinner` di layout (`label_spinner`), dan tetapkan listener-nya (`spinner.setOnItemSelectedListener`) di metode `onCreate()`. Tambahkan kode ke metode `onCreate()`:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    // Create the spinner.
    Spinner spinner = (Spinner) findViewById(R.id.label_spinner);
    if (spinner != null) {
        spinner.setOnItemSelectedListener(this);
    }
    ...
}
```

5. Lanjutkan mengedit metode `onCreate()`, tambahkan pernyataan yang membuat `ArrayAdapter` dengan larik string (`labels_array`) menggunakan layout spinner sederhana yang disediakan oleh Android untuk setiap item (`layout.simple_spinner_item`):

```
...
// Create ArrayAdapter using the string array and default spinner layout.
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
    R.array.labels_array, android.R.layout.simple_spinner_item);
...
```

Layout `simple_spinner_item` yang digunakan di langkah ini, dan layout `simple_spinner_dropdown_item` yang digunakan di langkah berikutnya, adalah layout default yang ditetapkan sebelumnya yang disediakan oleh Android di kelas `R.layout`. Anda harus menggunakan layout ini kecuali Anda ingin mendefinisikan layout Anda sendiri untuk item di spinner dan tampilan spinner.

6. Tetapkan layout untuk pilihan spinner menjadi `simple_spinner_dropdown_item`, lalu terapkan adaptor ke spinner:

```
...
// Specify the layout to use when the list of choices appears.
adapter.setDropDownViewResource
    (android.R.layout.simple_spinner_dropdown_item);
// Apply the adapter to the spinner.
if (spinner != null) {
    spinner.setAdapter(adapter);
}
...
```

3.3 Tambahkan kode untuk merespons pilihan pengguna

Saat pengguna memilih item di spinner, objek Spinner menerima kejadian on-item-selected. Untuk menangani kejadian ini, Anda telah mengimplementasikan antarmuka `AdapterView.OnItemSelectedListener` di langkah sebelumnya, dengan menambahkan metode callback `onItemSelected()` dan `onNothingSelected()` kosong.

Di langkah ini, Anda terlebih dulu harus mendeklarasikan `mSpinnerLabel` sebagai string untuk menampung item yang dipilih. Kemudian Anda mengisikan kode untuk metode `onItemSelected()` untuk mengambil item yang dipilih di spinner, menggunakan `getItemAtPosition()`, dan menetapkan item ke `mSpinnerLabel`.

1. Deklarasikan string `mSpinnerLabel` di awal definisi kelas `MainActivity`:

```
public class MainActivity extends AppCompatActivity implements
    AdapterView.OnItemSelectedListener {
    private String mSpinnerLabel = "";
    ...
}
```

2. Tambahkan kode ke metode callback `onItemSelected()` kosong, seperti yang ditampilkan di bawah ini, untuk mengambil item yang dipilih pengguna menggunakan `getItemAtPosition`, dan tetapkan ke `mSpinnerLabel`. Anda juga bisa menambahkan sebuah panggilan ke metode `showText()` yang sudah Anda tambahkan ke versi aplikasi sebelumnya:

```
public void onItemSelected(AdapterView<?> adapterView, View view, int
    i, long l) {
    mSpinnerLabel = adapterView.getItemAtPosition(i).toString();
    showText(view);
}
```

Tip: Dengan menambahkan metode `showText()` ke metode `onItemSelected()` di atas, Anda telah mengaktifkan listener pilihan spinner untuk menampilkan pilihan spinner bersama dengan nomor telepon, sehingga Anda tidak lagi memerlukan tombol **Show** yang memanggil metode `showText()`.

3. Tambahkan kode ke metode callback `onNothingSelected()` yang kosong, seperti yang ditampilkan di bawah ini, untuk menampilkan pesan logcat jika tidak satu pun yang dipilih:

```
public void onNothingSelected(AdapterView<?> adapterView) {
    Log.d(TAG, "onNothingSelected: ");
}
```

TAG di pernyataan di atas berwarna merah karena belum didefinisikan.

4. Ekstrak sumber daya string untuk `"onNothingSelected: "` ke `nothing_selected`.
5. Klik **TAG**, klik bola lampu merah, dan pilih **Create constant field 'TAG'** dari menu munculan. Android Studio menambahkan yang berikut di bawah deklarasi kelas `MainActivity`:

```
private static final String TAG = ;
```

1. Tambahkan `MainActivity.class.getSimpleName()` agar menggunakan nama sederhana kelas untuk TAG:

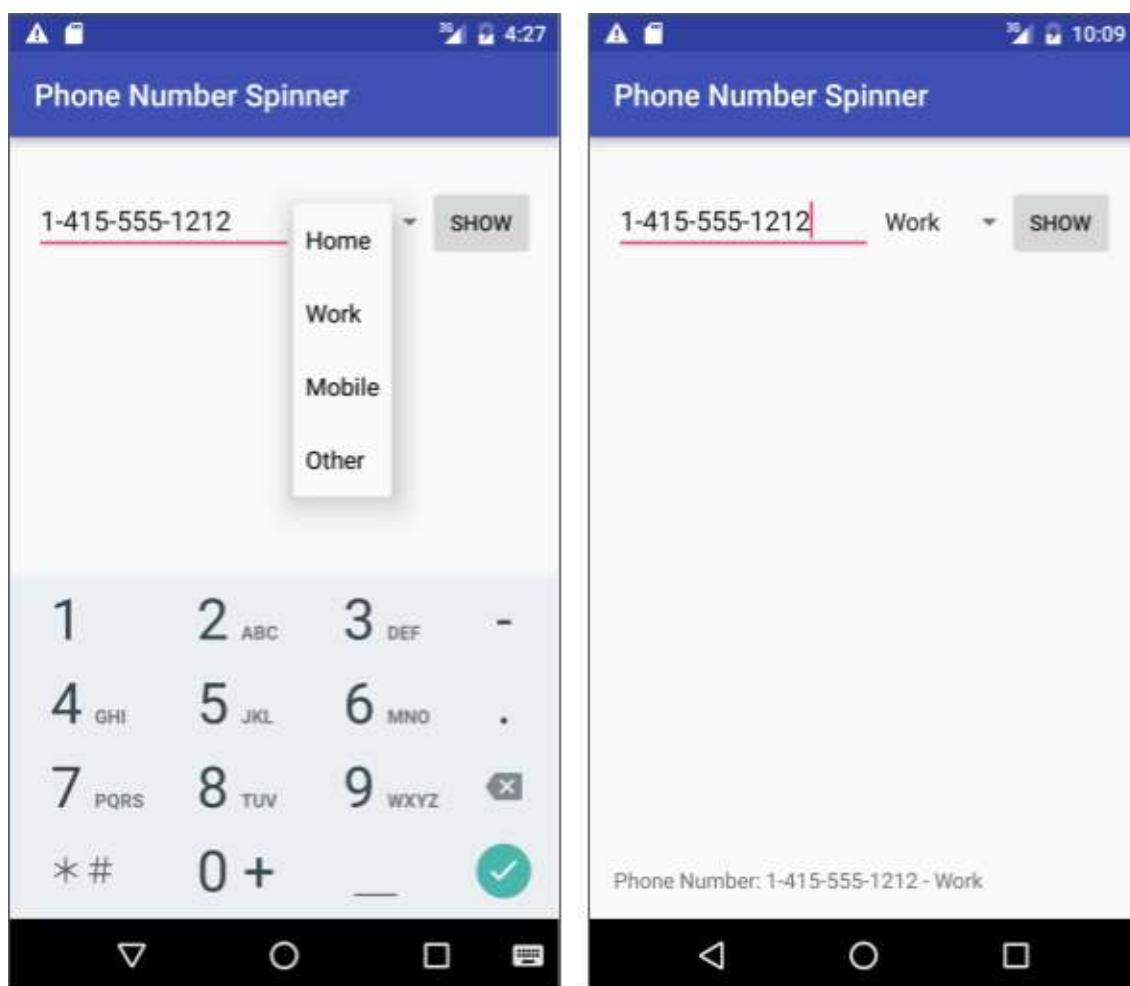
```
private static final String TAG = MainActivity.class.getSimpleName();
```

1. Ubah pernyataan `String showString` di metode `showText` untuk menampilkan string yang dimasukkan dan item spinner yang dipilih (`mSpinnerLabel`):

```
String showString = (editText.getText().toString() + " - " + mSpinnerLabel);
```

1. Jalankan aplikasi.

Spinner muncul di sebelah bidang entri telepon dan menampilkan pilihan pertama (Home). Mengetuk spinner akan memunculkan semua pilihan, seperti yang ditampilkan di sebelah kiri gambar di bawah ini. Setelah memasukkan nomor telepon dan memilih item spinner, sebuah pesan muncul bawah layar dengan nomor telepon dan item spinner yang dipilih, seperti yang ditampilkan di sebelah kanan gambar di bawah ini. (Anda juga bisa mengetuk tombol **Show** untuk menampilkan nomor telepon dan item spinner, tetapi karena ini berlebihan, Anda sekarang bisa menghilangkan tombol **Show**.)



Kode Solusi:

Proyek Android Studio: [PhoneNumberSpinner](#)

Tugas 4: Menggunakan dialog untuk peringatan yang memerlukan keputusan

Anda bisa menyediakan dialog untuk peringatan yang mengharuskan pengguna membuat keputusan. *Dialog* adalah jendela yang muncul di atas tampilan atau mengisi tampilan, menyela alur aktivitas.

Misalnya, sebuah dialog peringatan mungkin mengharuskan pengguna untuk mengeklik **Continue** setelah membacanya, atau memberi pilihan kepada pengguna untuk menyetujui suatu tindakan dengan mengeklik tombol positif (seperti **OK** atau **Accept**), atau untuk tidak menyetujui dengan mengeklik tombol negatif (seperti **Cancel**). Di Android, gunakan subkelas [AlertDialog](#) dari kelas [Dialog](#) untuk menampilkan dialog standar untuk peringatan.

Tip: Jangan gunakan terlalu banyak dialog karena bisa menyela alur kerja pengguna. Baca [Panduan desain dialog](#) untuk praktik desain terbaik, dan [Dialog](#) dalam dokumentasi developer Android untuk contoh kode.

Dalam praktik ini Anda akan menggunakan tombol untuk memicu dialog peringatan standar. Di aplikasi nyata, Anda bisa memicu dialog peringatan berdasarkan beberapa ketentuan, atau saat pengguna mengetuk sesuatu.

Proyek Android Studio: [AlertSample](#)

4.1 Buat proyek baru dengan layout untuk menampilkan dialog peringatan

Dalam latihan ini, Anda akan membangun sebuah peringatan dengan tombol OK dan Cancel, yang akan dipicu oleh klik pengguna pada tombol.

1. Buat proyek baru bernama **Alert Sample** berdasarkan template Empty Activity.
2. Bukan layout `activity_main.xml` dan buat perubahan berikut:

| Atribut TextView | Nilai |
|---------------------------|---------------------------------|
| <code>android:id</code> | <code>"@+id/top_message"</code> |
| <code>android:text</code> | "Tap to test the alert:" |

3. Ekstrak string `android:text` di atas ke dalam `tap_test` sumber daya untuk membuatnya lebih mudah diterjemahkan.
4. Tambahkan `Button` dengan atribut berikut:

| Atribut Tombol | Nilai |
|---------------------------------------|---------------------------------|
| <code>android:id</code> | <code>"@+button1"</code> |
| <code>android:layout_width</code> | <code>wrap_content</code> |
| <code>android:layout_height</code> | <code>wrap_content</code> |
| <code>android:layout_below</code> | <code>"@+top_message"</code> |
| <code>android:layout_marginTop</code> | <code>"36dp"</code> |
| <code>android:text</code> | "Alert" |
| <code>android:onClick</code> | <code>"onClickShowAlert"</code> |

5. Ekstrak string `android:text` di atas ke dalam `alert_button` sumber daya untuk membuatnya lebih mudah diterjemahkan.
6. Ekstrak nilai dimensi untuk `android:layout_marginTop` dengan cara yang sama: Letakkan kursor di `"36dp"`, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract dimension resource**. Lalu edit nama Sumber Daya untuk nilai ke `button_top_margin`.

Penetapan sumber daya dimensi disimpan di file `dimens.xml` (di `app > res > values > dimens`). Anda bisa mengedit file ini untuk mengubah penetapan, sehingga aplikasi bisa diubah untuk ukuran layar yang berbeda.

4.2 Menambahkan dialog peringatan ke aktivitas utama

Pola desain *builder* mempermudah pembuatan objek dari kelas yang membutuhkan banyak atribut wajib serta opsional dan karena itu memerlukan banyak parameter untuk dibangun. Tanpa pola ini, Anda tentunya harus membuat konstruktur untuk kombinasi atribut yang diperlukan dan atribut opsional; dengan pola ini, kode lebih mudah dibaca dan dikelola. Untuk informasi selengkapnya tentang pola desain builder, lihat [pola Builder](#)

Kelas builder biasanya merupakan kelas member statis dari kelas yang dibangunnya. Anda menggunakan `AlertDialog.Builder` untuk membangun dialog peringatan standar, menggunakan `setTitle` untuk menyetel judulnya, `setMessage` untuk menyetel pesannya, dan `setPositiveButton` dan `setNegativeButton` untuk menyetel tombolnya.

Untuk membuat peringatan, Anda perlu membuat objek `AlertDialog.Builder`. Anda akan menambahkan metode `onClickShowAlert()`, yang akan membuat objek ini sebagai urutan pertama bisnis.

Catatan: Agar memudahkan contoh ini dipahami, dialog peringatan dibuat di metode `onClickShowAlert()`. Ini hanya terjadi jika metode `onClickShowAlert` dipanggil, yang akan terjadi saat pengguna mengeklik tombol. Artinya, aplikasi membangun

dialog baru hanya saat tombol diklik, yang berguna jika dialog jarang dilihat (saat pengguna mengambil jalur tertentu di aplikasi). Akan tetapi, jika dialog sering mungkin, sebaiknya Anda membangun dialog sekali di metode `onCreate()`, lalu menjalankan dialog di metode `onClickShowAlert()`.

1. Tambahkan metode `onClickShowAlert()` ke `MainActivity.java` seperti berikut:

```
public void onClickShowAlert(View view) {
    AlertDialog.Builder myAlertDialog = new
        AlertDialog.Builder(MainActivity.this);
```

Catatan: Jika `AlertDialog.Builder` tidak dikenali saat Anda memasukkannya, klik ikon bola lampu merah dan pilih versi pustaka dukungan (`android.support.v7.app.AlertDialog`) untuk mengimportnya ke aktivitas Anda.

2. Setel judul dan pesan untuk dialog peringatan dalam `onClickShowAlert()` setel kode di langkah sebelumnya:

```
...
// Set the dialog title.
myAlertDialog.setTitle("Alert");
// Set the dialog message.
myAlertDialog.setMessage("Click OK to continue, or Cancel to stop:");
...
```

3. Ekstrak judul dan pesan ke dalam sumber daya string. Baris kode sebelumnya sekarang seharusnya menjadi:

```
...
// Set the dialog title.
myAlertDialog.setTitle(R.string.alert_title);
// Set the dialog message.
myAlertDialog.setMessage(R.string.alert_message);
...
```

4. Tambahkan tombol **OK** pada peringatan dengan `setPositiveButton()` dan menggunakan `onClickListener()`:

```
...
// Add the buttons.
myAlertDialog.setPositiveButton("OK", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        // User clicked OK button.
        Toast.makeText(getApplicationContext(), "Pressed OK",
            Toast.LENGTH_SHORT).show();
    }
});
...
```

Anda menyetel tombol positif (**OK**) dan negatif (**Cancel**) menggunakan metode `setPositiveButton()` dan `setNegativeButton()`. Setelah pengguna mengetuk tombol **OK** di peringatan, Anda bisa mengambil pilihan pengguna dan menggunakananya di kode Anda. Dalam contoh ini, Anda menampilkan pesan toast jika tombol **OK** diklik.

5. Ekstrak sumber daya string untuk `"OK"` dan untuk `"Pressed OK"`. Pernyataan sekarang seharusnya menjadi:

```
...
// Add the buttons.
myAlertDialog.setPositiveButton(R.string.ok, new
    DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        // User clicked OK button.
        Toast.makeText(getApplicationContext(), R.string.pressed_ok,
            Toast.LENGTH_SHORT).show();
    }
});
...
```

6. Tambahkan tombol **Cancel** pada peringatan dengan `setNegativeButton()` dan `onClickListener()`, tampilkan pesan toast jika tombol diklik, lalu batalkan dialog:

```
...
myAlertDialog.setNegativeButton("Cancel", new
    DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        // User cancelled the dialog.
        Toast.makeText(getApplicationContext(), "Pressed Cancel",
            Toast.LENGTH_SHORT).show();
    }
});
```

7. Ekstrak sumber daya string untuk "Cancel" dan "Pressed Cancel". Pernyataan sekarang seharusnya menjadi:

```
...
myAlertDialog.setNegativeButton(R.string.cancel, new
    DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        // User cancelled the dialog.
        Toast.makeText(getApplicationContext(), R.string.pressed_cancel,
            Toast.LENGTH_SHORT).show();
    }
});
```

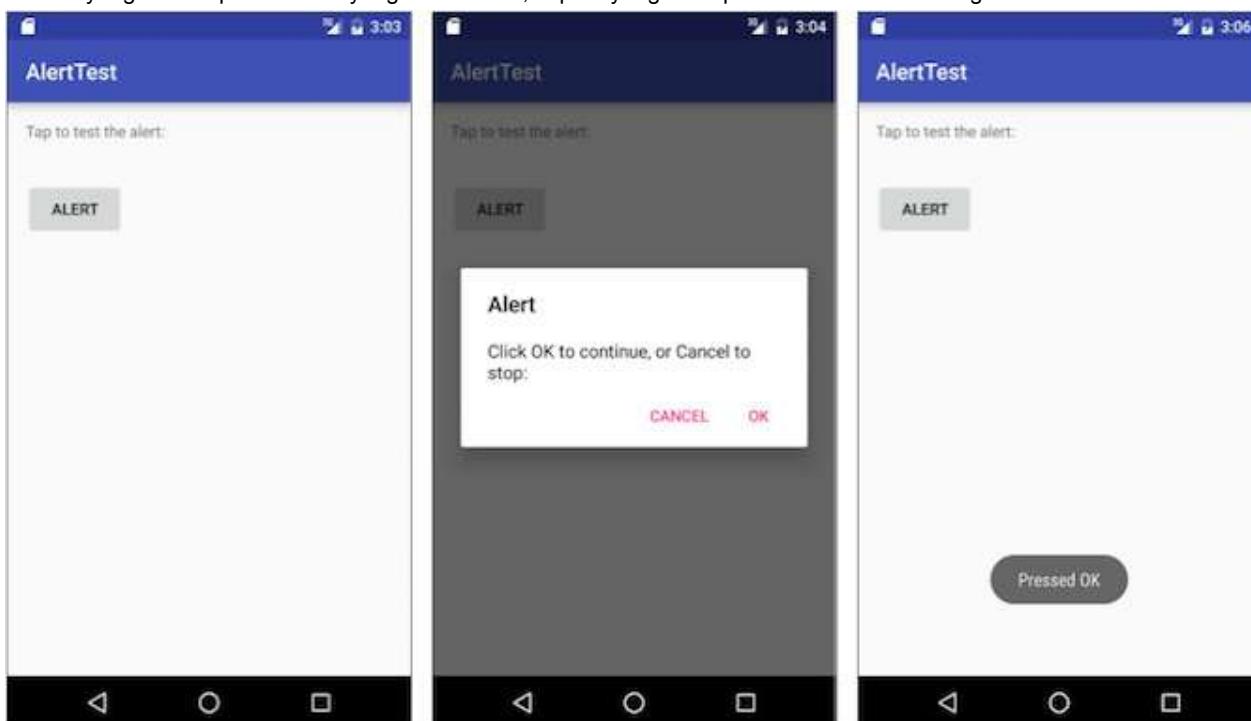
8. Tambahkan `show()` yang membuat lalu menampilkan dialog peringatan, ke akhir `onClickShowAlert()`:

```
...
// Create and show the AlertDialog.
myAlertDialog.show();
}
```

Tip: Untuk mengetahui selengkapnya tentang `onClickListener` dan listener lain, lihat [Antarmuka Pengguna: Kejadian Input](#).

9. Jalankan aplikasi.

Anda seharusnya bisa mengetuk tombol **Alert**, yang ditampilkan di sebelah kiri gambar berikut, untuk melihat dialog peringatan, yang ditampilkan di tengah gambar berikut ini. Dialog menampilkan tombol **OK** dan **Cancel** dan pesan toast muncul yang menampilkan mana yang Anda tekan, seperti yang ditampilkan di sebelah kanan gambar berikut ini.



Kode solusi:

Proyek Android Studio: [AlertSample](#)

Tugas 5: Menggunakan picker untuk masukan pengguna

Android menyediakan dialog yang siap digunakan, disebut *picker*, untuk memilih waktu atau tanggal. Anda bisa menggunakannya untuk memastikan pengguna Anda memilih waktu atau tanggal yang valid, yang diformat dengan benar dan disesuaikan dengan waktu dan tanggal lokal pengguna. Masing-masing picker menyediakan kontrol untuk memilih setiap bagian waktu (jam, menit, AM/PM) atau tanggal (bulan, tanggal, tahun). Anda bisa membaca semua tentang mempersiapkan picker di [Picker](#).

Dalam tugas ini, Anda akan membuat proyek baru, dan menambahkan picker tanggal dan picker waktu. Anda juga akan mempelajari cara menggunakan fragmen. *Fragmen* adalah perilaku atau bagian antarmuka pengguna dalam suatu aktivitas. Fragmen mirip seperti aktivitas mini di dalam aktivitas utama, dengan siklus hidupnya sendiri, dan digunakan untuk membangun picker. Semua pekerjaan dilakukan untuk Anda. Untuk mengetahui tentang fragmen, lihat [Fragmen](#) dalam Panduan API.

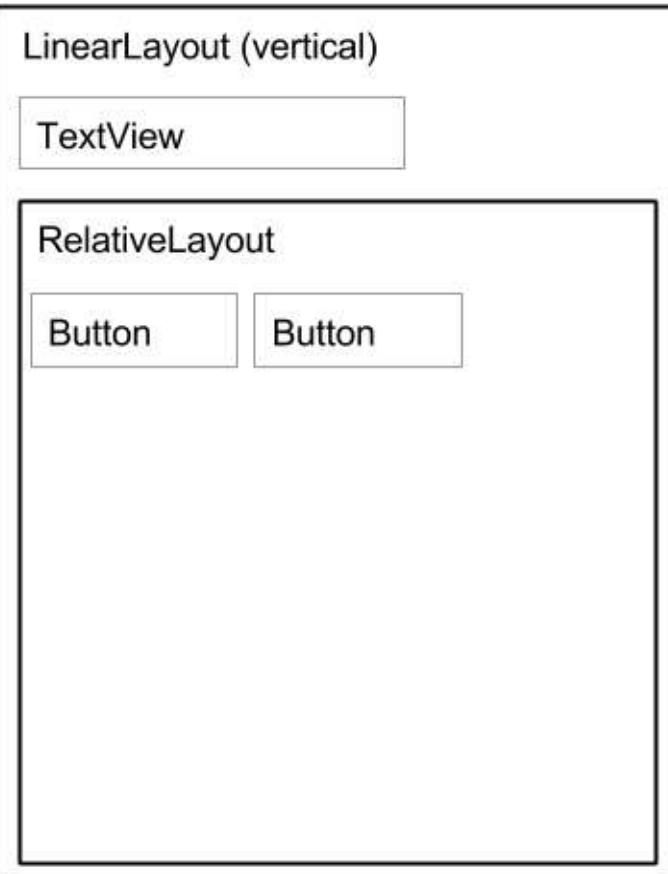
Salah satu manfaat menggunakan fragmen untuk picker adalah Anda bisa mengisolasi bagian kode untuk mengelola tanggal dan waktu untuk berbagai lokasi yang menampilkan tanggal dan waktu dengan cara yang berbeda. Praktik terbaik untuk menampilkan picker adalah dengan menggunakan instance [DialogFragment](#) yang merupakan subkelas Fragmen. DialogFragment menampilkan jendela dialog yang mengambang di atas jendela aktivitasnya. Dalam latihan ini, Anda akan menambahkan fragmen untuk setiap dialog picker dan menggunakan DialogFragment untuk mengelola daur hidup dialog.

Tip: Manfaat lain menggunakan fragmen untuk picker adalah Anda bisa mengimplementasikan konfigurasi layout yang berbeda, seperti dialog dasar pada tampilan berukuran handset atau bagian layout yang disematkan pada tampilan yang besar.

5.1 Membuat layout aktivitas utama

Untuk memulai tugas ini, buat layout aktivitas utama untuk menyediakan tombol untuk mengakses picker waktu dan tanggal. Periksa kode layout XML berikut:

1. Mulai proyek baru yang bernama **Date Time Pickers** menggunakan template Empty Activity.
2. Buka **activity_main.xml** untuk mengedit kode layout.
3. Ubah induk `RelativeLayout` agar menjadi `LinearLayout` dan tambahkan `android:orientation="vertical"` untuk mengatur orientasi layout menjadi vertikal. Tidak perlu mengkhawatirkan penampilan layout saat ini. Sasarannya adalah menggunakan layout yang menyematkan `RelativeLayout` di dalam `LinearLayout`:



4. Ubah teks elemen `TextView` pertama ke "**Choose the date and time:**" dan ekstrak teks ke sumber daya string `choose_datetime`.

| Atribut TextView | Nilai Lama | Nilai Baru |
|---------------------------|---------------|---------------------------|
| <code>android:text</code> | "Hello World" | "@string/choose_datetime" |

5. Tambahkan atribut `android:textSize` dan masukkan ukuran teks **20sp**. Ekstrak dimensi `android:textSize` ke `text_size`.

| Atribut TextView | Nilai Lama | Nilai Baru |
|-------------------------------|------------|--------------------|
| <code>android:textSize</code> | | "@dimen/text_size" |

6. Tambahkan turunan `RelativeLayout` di dalam `LinearLayout` untuk menampung elemen `Button`, dan terima lebar dan tinggi default `"match parent"`.

7. Tambahkan elemen `Button` pertama di dalam `RelativeLayout` dengan atribut berikut:

| Atribut Tombol Pertama | Nilai |
|---------------------------------------|------------------------|
| <code>android:layout_width</code> | "wrap_content" |
| <code>android:layout_height</code> | "wrap_content" |
| <code>android:id</code> | "@+id/button_date" |
| <code>android:layout_marginTop</code> | "12dp" |
| <code>android:text</code> | "Date" |
| <code>android:onClick</code> | "showDatePickerDialog" |

Tidak perlu mengkhawatirkan referensi `showDatePickerDialog` berwarna merah. Metode belum didefinisikan—Anda akan mendefinisikannya nanti.

8. Ekstrak string `"Date"` ke dalam sumber daya string `date_button`.

9. Ekstrak dimensi "12dp" untuk `android:layout_marginTop` ke `button_top_margin`.
10. Tambahkan elemen `Button` kedua di dalam turunan `RelativeLayout` atribut berikut:

| Atribut Tombol Kedua | Nilai |
|-----------------------------------------|----------------------------|
| <code>android:layout_width</code> | "wrap_content" |
| <code>android:layout_height</code> | "wrap_content" |
| <code>android:id</code> | "@+id/button_time" |
| <code>android:layout_marginTop</code> | "@dimen/button_top_margin" |
| <code>android:layout_alignBottom</code> | "@id/button_date" |
| <code>android:layout_toRightOf</code> | "@id/button_date" |
| <code>android:text</code> | "Time" |
| <code>android:onClick</code> | "showTimePickerDialog" |

Referensi `showTimePickerDialog` berwarna merah. Metode belum didefinisikan — Anda akan mendefinisikannya nanti.

11. Ekstrak string "Time" ke dalam sumber daya string `time_button`.
12. Jika Anda belum melakukannya, klik tab **Preview** untuk menampilkan pratinjau layout. Pratinjau seharusnya tampak seperti kode dan gambar berikut ini.

Kode solusi untuk layout utama:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.DateTimePicker.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="@dimen/text_size"
        android:text="@string/choose_datetime"/>

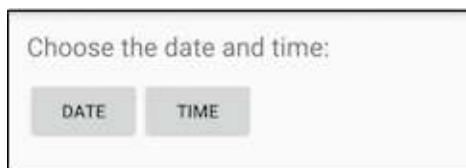
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/button_date"
            android:layout_marginTop="@dimen/button_top_margin"
            android:text="@string/date_button"
            android:onClick="showDatePickerDialog"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/button_time"
            android:layout_marginTop="@dimen/button_top_margin"
            android:layout_alignBottom="@+id/button_date"
            android:layout_toRightOf="@+id/button_date"
            android:text="@string/time_button"
            android:onClick="showTimePickerDialog"/>

    </RelativeLayout>
</LinearLayout>

```



5.2 Buat fragmen baru untuk picker tanggal

Dalam latihan ini, Anda akan menambahkan fragmen untuk picker tanggal. Fragmen mirip seperti aktivitas mini di dalam aktivitas utama, dengan daur hidupnya sendiri.

1. Luaskan `app > java > com.example.android.DateTimePicker` dan pilih `MainActivity`.
2. Pilih **File > New > Fragment > Fragment (Blank)**, dan beri nama fragmen `DatePickerFragment`. Kosongkan ketiga opsi kotak centang sehingga Anda *tidak* membuat XML layout, *jangan* sertakan metode pabrik fragmen, dan *jangan* sertakan callback antarmuka. Anda tidak perlu membuat layout untuk picker standar. Klik **Finish** untuk membuat fragmen.
3. Buka `DatePickerFragment` dan edit definisi kelas `DatePickerFragment` untuk memperluas `DialogFragment` dan implementasikan `DatePickerDialog.OnDateSetListener` untuk membuat picker tanggal standar dengan sebuah listener. Lihat [Picker](#) untuk informasi selengkapnya tentang memperluas `DialogFragment` untuk picker tanggal:

```

public class DatePickerFragment extends DialogFragment
    implements DatePickerDialog.OnDateSetListener {

```

Selagi Anda mengetik **DialogFragment** dan **DatePickerDialog.OnDateSetListener**, Android Studio otomatis menambahkan yang berikut di blok `import` di atas:

```
import android.app.DatePickerDialog;
import android.support.v4.app.DialogFragment;
```

Sebagai tambahan, ikon bola lampu merah muncul di margin kiri setelah beberapa detik.

4. Klik ikon bola lampu merah dan pilih **Implement methods** dari menu munculan. Sebuah dialog muncul dengan `onDateSet()` telah dipilih dan opsi "Insert @Override" dicentang. Klik **OK** untuk membuat metode `onDateSet()` kosong. Metode ini akan dipanggil saat pengguna menyetel tanggal. Setelah menambahkan metode `onDateSet()` kosong, Android Studio secara otomatis menambahkan yang berikut ini dalam blok `import` di atas.

```
import android.widget.DatePicker;
```

Parameter metode `onDateSet()` harus `int year`, dan `int month`, dan `int dayOfMonth`. Ubah parameter `dayOfMonth` ke `day` agar ringkas:

```
public void onDateSet(DatePicker view, int year, int month, int day)
```

5. Buang konstruktur publik kosong untuk `DatePickerFragment`.
6. Ganti `onCreateView()` dengan `onCreateDialog()` yang mengembalikan `Dialog`, dan anotasikan `onCreateDialog()` dengan `@NonNull` untuk menunjukkan bahwa nilai pengembalian `Dialog` tidak boleh null—upaya apa pun untuk mengacu pada nilai pengembalian `Dialog` harus dicentang null.

```
@NonNull
@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
    ...
}
```

7. Tambahkan kode berikut ke `onCreateDialog()` untuk membuat instance `year`, `month`, dan `day` dari `Calendar`, dan mengembalikan dialog dan nilai ini ke aktivitas utama. Saat Anda memasukkan `Calendar`, tetapkan import menjadi `java.util.Calendar`.

```
@NonNull
@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
    // Use the current date as the default date in the picker.
    final Calendar c = Calendar.getInstance();
    int year = c.get(Calendar.YEAR);
    int month = c.get(Calendar.MONTH);
    int day = c.get(Calendar.DAY_OF_MONTH);

    // Create a new instance of DatePickerDialog and return it.
    return new DatePickerDialog(getActivity(), this, year, month, day);
}
```

Kode solusi untuk `DatePickerFragment`:

```

public class DatePickerFragment extends DialogFragment
    implements DatePickerDialog.OnDateSetListener {

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Use the current date as the default date in the picker.
        final Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int day = c.get(Calendar.DAY_OF_MONTH);

        // Create a new instance of DatePickerDialog and return it.
        return new DatePickerDialog(getActivity(), this, year, month, day);
    }

    public void onDateSet(DatePicker view, int year, int month, int day) {
        // Do something with the date chosen by the user.
    }
}

```

5.3 Buat fragmen baru untuk picker waktu

Tambahkan fragmen ke proyek DateTimePickers untuk picker waktu:

1. Pilih **MainActivity** lagi.
2. Pilih **File > New > Fragment > Fragment (Blank)**, dan beri nama fragmen **TimePickerFragment**. Kosongkan ketiga opsi sehingga Anda *tidak* membuat XML layout, *jangan* sertakan metode pabrik fragmen, dan *jangan* sertakan callback antarmuka. Klik **Finish** untuk membuat fragmen.
3. Bukan ***TimePickerFrgment** dan ikuti prosedur yang sama dengan **DatePickerFragment**, mengimplementasikan metode kosong **onTimeSet()**, mengganti **onCreateView()** dengan **onCreateDialog()**, dan menghapus konstruktur publik untuk **TimePickerFrgment**. **TimePickerFragment** melakukan tugas yang sama dengan **DatePickerFragment**, tetapi dengan nilai waktu:
 - o Ini memperluas **DialogFragment** dan mengimplementasikan **TimePickerDialog.OnTimeSetListener** untuk membuat picker waktu standar dengan sebuah listener. Lihat [Picker](#) untuk informasi selengkapnya tentang memperluas **DialogFragment** untuk picker waktu:
 - o Ini menggunakan metode **onCreateDialog()** untuk melakukan inisialisasi **hour** dan **minute** dari **Calendar**, dan mengembalikan dialog dan nilai ini ke aktivitas utama menggunakan format tanggal 24 jam. Saat Anda memasukkan **Calendar**, tetapkan import menjadi **java.util.Calendar**.
 - o Ini juga mendefinisikan metode **onTimeSet()** kosong agar Anda bisa menambahkan kode untuk menggunakan **hourOfDay** dan **minute** yang dipilih pengguna. Metode ini akan dipanggil saat pengguna menyetel waktu:

```

public void onTimeSet(TimePicker view,
                      int hourOfDay, int minute) {
    // Do something with the time chosen by the user.
}

```

Catatan: Saat Anda membuat perubahan, Android Studio secara otomatis menambahkan yang berikut ini dalam blok import di bagian atas:

```

import android.support.v4.app.DialogFragment;
import android.app.TimePickerDialog;
import android.widget.TimePicker;
import java.util.Calendar;

```

Kode solusi untuk TimePickerFragment:

```

public class TimePickerFragment extends DialogFragment
    implements TimePickerDialog.OnTimeSetListener {

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Use the current time as the default values for the picker.
        final Calendar c = Calendar.getInstance();
        int hour = c.get(Calendar.HOUR_OF_DAY);
        int minute = c.get(Calendar.MINUTE);

        // Create a new instance of TimePickerDialog and return it.
        return new TimePickerDialog(getActivity(), this, hour, minute,
            DateFormat.is24HourFormat(getActivity()));
    }

    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        // Do something with the time chosen by the user.
    }
}

```

5.4 Modifikasi aktivitas utama

Meskipun kebanyakan kode di aktivitas utama tetap sama, Anda perlu menambahkan metode yang membuat instance `FragmentManager` untuk mengelola setiap fragmen dan menampilkan setiap picker.

1. Buat sumber daya string di `strings.xml`:

```

<string name="date_picker">datePicker</string>
<string name="time_picker">timePicker</string>

```

2. Buka **MainActivity**.

3. Tambahkan metode `showDatePickerDialog()` dan `showTimePickerDialog()`, dengan merujuk ke kode di bawah ini. Ini membuat instance `FragmentManager` untuk mengelola fragmen secara otomatis, dan untuk menampilkan picker. Untuk informasi selengkapnya tentang fragmen, lihat [Fragmen](#).

```

public class MainActivity extends AppCompatActivity {

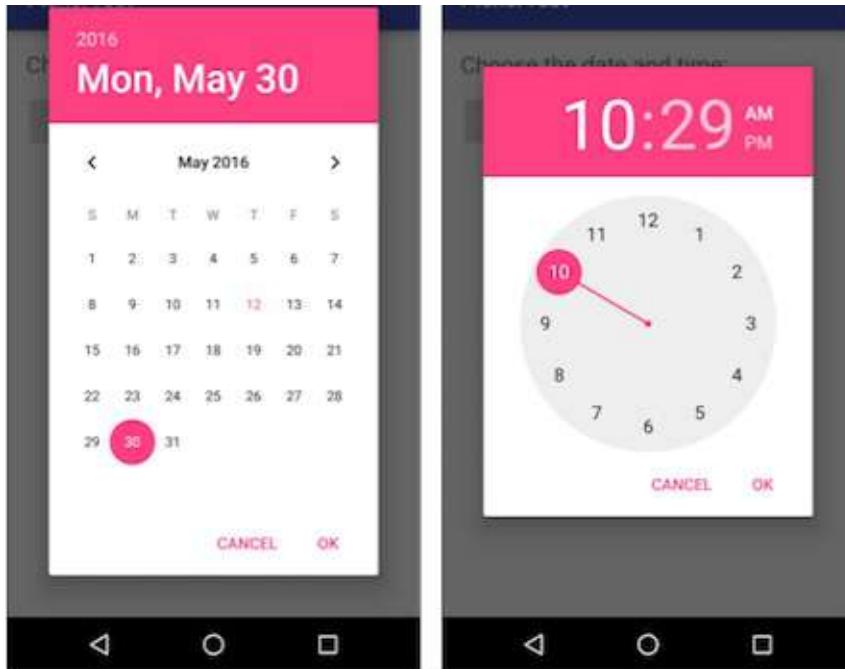
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void showDatePickerDialog(View v) {
        DialogFragment newFragment = new DatePickerFragment();
        newFragment.show(getSupportFragmentManager(),
            getString(R.string.date_picker));
    }

    public void showTimePickerDialog(View view) {
        DialogFragment newFragment = new TimePickerFragment();
        newFragment.show(getSupportFragmentManager(),
            getString(R.string.time_picker));
    }
}

```

4. Jalankan aplikasi. Anda akan melihat picker tanggal dan waktu setelah mengetuk tombol.



5.5 Gunakan tanggal dan waktu yang dipilih

Dalam latihan ini, Anda akan meneruskan tanggal dan waktu kembali ke MainActivity dan mengonversi tanggal dan waktu ke string yang bisa Anda tampilkan di pesan toast.

- Buka **MainActivity** dan tambahkan tanda tangan metode `processDatePickerResult()` yang mengambil `year`, `month`, dan `day` sebagai argumen:

```
public void processDatePickerResult(int year, int month, int day) {  
}
```

- Tambahkan kode berikut ke metode `processDatePickerResult()` untuk mengonversi `month`, `day`, dan `year` ke string yang terpisah:

```
String month_string = Integer.toString(month+1);  
String day_string = Integer.toString(day);  
String year_string = Integer.toString(year);
```

Tip: Integer `month` yang dikembalikan oleh picker tanggal mulai menghitung dari 0 untuk Januari, sehingga Anda perlu menambahkan 1 untuk mulai menampilkan bulan mulai dari 1.

- Tambahkan yang berikut ini setelah kode di atas untuk menggabungkan tiga string dan menyertakan garis miring untuk format tanggal A.S.:

```
String dateMessage = (month_string + "/" +  
                     day_string + "/" + year_string);
```

- Tambahkan yang berikut ini setelah pernyataan di atas untuk menampilkan pesan toast:

```
Toast.makeText(this, "Date: " + dateMessage,  
              Toast.LENGTH_SHORT).show();
```

- Ekstrak string `"Date: "` hard-code ke dalam sumber daya string bernama `date`. Hal ini secara otomatis mengganti string hard-code dengan `getString(R.string.date)`. Kode untuk metode `processDatePickerResult()` sekarang harus terlihat seperti ini:

```

public void processDatePickerResult(int year, int month, int day) {
    String month_string = Integer.toString(month + 1);
    String day_string = Integer.toString(day);
    String year_string = Integer.toString(year);
    // Assign the concatenated strings to dateMessage.
    String dateMessage = (month_string + "/" +
                          day_string + "/" + year_string);
    Toast.makeText(this, getString(R.string.date) + dateMessage,
                  Toast.LENGTH_SHORT).show();
}

```

6. Buka **DatePickerFragment** dan tambahkan yang berikut ini ke metode `onDataSet()` untuk menjalankan metode `processDatePickerResult()` di `MainActivity` dan meneruskan `year`, `month`, dan `day` ke metode:

```

public void onDataSet(DatePicker view, int year, int month, int day) {
    // Set the activity to the Main Activity.
    MainActivity activity = (MainActivity) getActivity();
    // Invoke Main Activity's processDatePickerResult() method.
    activity.processDatePickerResult(year, month, day);
}

```

Anda menggunakan `getActivity()`, yang jika digunakan dalam fragmen, akan mengembalikan aktivitas yang saat ini dikaitkan dengan fragmen. Anda memerlukannya karena Anda tidak bisa memanggil suatu metode dalam `MainActivity` tanpa konteks `MainActivity` (Anda harus menggunakan `intent` sebagai ganti, seperti yang Anda pelajari pada pembelajaran sebelumnya). Aktivitas mewarisi konteks, sehingga Anda bisa menggunakannya sebagai konteks untuk memanggil metode (seperti dalam `activity.processDatePickerResult`).

7. `TimePickerFragment` menggunakan logika yang sama. 1.Buka **MainActivity** dan tambahkan tanda tangan metode `processTimePickerResult()` yang mengambil `hourOfDay` dan `minute` sebagai argumen:

```

public void processTimePickerResult(int hourOfDay, int minute) {
}

```

8. Tambahkan kode berikut ke metode `processTimePickerResult()` untuk mengonversi `hourOfDay` dan `minute` ke string yang terpisah:

```

String hour_string = Integer.toString(hourOfDay);
String minute_string = Integer.toString(minute);

```

9. Tambahkan yang berikut ini setelah kode di atas untuk menggabungkan string dan menyertakan titik dua untuk format waktu:

```

String timeMessage = (hour_string + ":" + minute_string);

```

10. Tambahkan yang berikut ini setelah pernyataan di atas untuk menampilkan pesan toast:

```

Toast.makeText(this, "Time: " + timeMessage,
              Toast.LENGTH_SHORT).show();

```

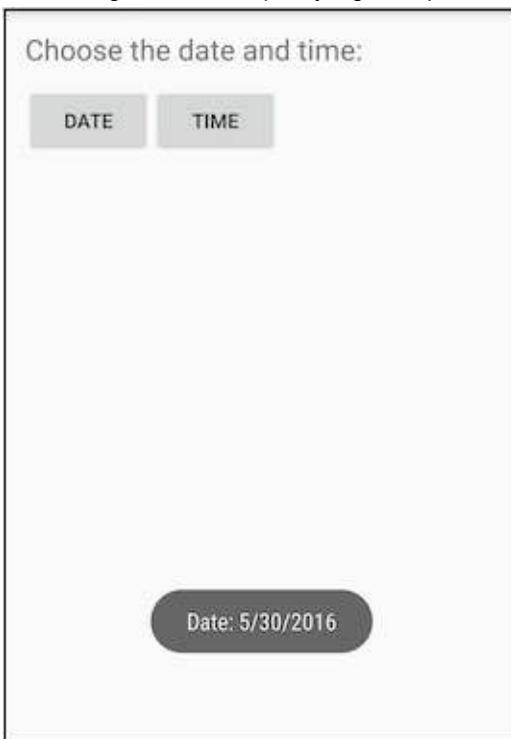
11. Ekstrak string `"Time: "` hard-code ke dalam sumber daya string bernama `time`. Hal ini secara otomatis mengganti string hard-code dengan `getString(R.string.time)`. Kode untuk metode `processDatePickerResult()` sekarang harus terlihat seperti ini:

```
public void processTimePickerResult(int hourOfDay, int minute) {  
    // Convert time elements into strings.  
    String hour_string = Integer.toString(hourOfDay);  
    String minute_string = Integer.toString(minute);  
    // Assign the concatenated strings to timeMessage.  
    String timeMessage = (hour_string + ":" + minute_string);  
    Toast.makeText(this, getString(R.string.time) + timeMessage,  
        Toast.LENGTH_SHORT).show();  
}
```

12. Buka **TimePickerFrgment** dan tambahkan yang berikut ini ke metode `onTimeSet()` untuk memanggil metode `processTimePickerResult()` di `MainActivity` dan meneruskan `hourOfDay` dan `minute` ke metode:

```
public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
    // Set the activity to the Main Activity.  
    MainActivity activity = (MainActivity) getActivity();  
    // Invoke Main Activity's processTimePickerResult() method.  
    activity.processTimePickerResult(hourOfDay, minute);  
}
```

13. Anda sekarang bisa menjalankan aplikasi. Setelah memilih tanggal atau waktu, tanggal atau waktu muncul di pesan toast di bagian bawah, seperti yang ditampilkan dalam gambar di bawah ini.



Kode Solusi:

Proyek Android Studio: [DateTimePickers](#)

Tugas 6: Menggunakan tampilan gambar sebagai tombol

Anda bisa membuat tampilan agar bisa diklik, sebagai tombol, dengan menambahkan atribut `android:onClick` di layout XML. Misalnya, Anda bisa membuat gambar agar berfungsi sebagai tombol dengan menambahkan `android:onClick` ke [ImageView](#).

Tip: Jika Anda menggunakan beberapa gambar sebagai gambar yang bisa diklik, atur gambar dalam sebuah viewgroup sehingga dikelompokkan bersama.

Dalam tugas ini, Anda akan membuat prototipe aplikasi untuk memesan pencuci mulut dari kafe. Setelah memulai proyek baru berdasarkan template Basic Activity, Anda akan memodifikasi TextView "Hello World" dengan teks yang sesuai, dan menambahkan gambar yang digunakan untuk tombol "Add to order".

6.1 Mulai proyek baru

1. Mulai proyek Android Studio baru dengan nama aplikasi **Droid Cafe**. Pilih template **Basic Activity**, terima setelan default, dan klik **Finish**. Proyek terbuka dengan dua layout dalam folder **res > layout: activity_main.xml**, dan **content_main.xml**.
2. Buka **content_main.xml** dan ekstrak string "Hello World" di **TextView** untuk menggunakan nama sumber daya **intro_text**. Lalu buka **strings.xml** dan ubah definisi sumber daya **intro_text** untuk menggunakan lebih banyak teks deskriptif, seperti **Droid Desserts**:

```
<string id="intro_text">Droid Desserts</string>
```

3. Ubah **TextView** di layout agar menggunakan ukuran teks yang lebih besar **24sp** dan padding **10dp**, dan tambahkan atribut **android:id** dengan id **textintro**.
4. Ekstrak sumber daya dimensi untuk atribut **android:padding** ke nama sumber daya **padding_regular**, dan atribut **android:textSize** ke sumber daya **text_heading**. Anda akan menggunakan nama sumber daya ini di langkah-langkah selanjutnya.
5. Tambahkan **TextView** lagi di bawah **textintro** **TextView** dengan atribut berikut:

| Atribut TextView | Nilai |
|-----------------------|--------------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:padding | "@dimen/padding_regular" |
| android:id | "@+id/choose_dessert" |
| android:layout_below | "@+id/textintro" |
| android:text | "Choose a dessert." |

6. Ekstrak sumber daya string untuk atribut **android:text** ke nama sumber daya **choose_a_desert**.

6.2 Tambahkan gambar

1. Gambar bernama circle.jpg, froyo_circle.jpg, dan icecream_circle.jpg disediakan dengan aplikasi starter di file [4_1_P_starter_images.zip](#) yang bisa Anda unzip di komputer. Untuk menyalin gambar ke proyek Anda, ikuti langkah-langkah ini:
 - i. Tutup proyek Anda.
 - ii. Salin file gambar ke dalam folder **drawable** proyek Anda. Temukan folder **drawable** di proyek dengan menggunakan jalur ini: **project_name > app > src > main > res > drawable**
 - iii. Buka kembali proyek Anda.
2. Buka file **content_main.xml** lagi dan tambahkan sebuah **ImageView** untuk gambar donat ke layout di bawah tampilan **choose_desert**, menggunakan atribut berikut:

| Atribut ImageView untuk donat | Nilai |
|-------------------------------|-----------------------------------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:padding | "@dimen/padding_regular" |
| android:id | "@+id/donut" |
| android:layout_below | "@id/choose_dessert" |
| android:contentDescription | "Donuts are glazed and sprinkled with candy." |
| android:src | "@drawable/donut_circle" |

3. Ekstrak nilai atribut `android:contentDescription` ke sumber daya string `donuts`. Anda akan menggunakan sumber daya string ini di langkah berikutnya.
4. Tambahkan `TextView` yang akan muncul di sebelah gambar donat sebagai deskripsi, atribut berikut:

| Atribut TextView | Nilai |
|--------------------------|-----------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:padding | "35dp" |
| android:layout_below | "@+id/choose_dessert" |
| android:layout_toRightOf | "@id/donut" |
| android:text | "@string/donuts" |

5. Ekstrak sumber daya dimensi untuk atribut `android:padding` ke nama sumber daya `padding_wide`. Anda akan menggunakan nama sumber daya ini di langkah-langkah selanjutnya.
6. Tambahkan `ImageView` kedua ke layout untuk sandwich es krim, menggunakan atribut berikut:

| Atribut ImageView untuk ice_cream | Nilai |
|-----------------------------------|-------------------------------------------------------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:padding | "@dimen/padding_regular" |
| android:id | "@+id/ice_cream" |
| android:layout_below | "@id/donut" |
| android:contentDescription | "Ice cream sandwiches have chocolate wafers and vanilla filling." |
| android:src | "@drawable/icecream_circle" |

7. Ekstrak nilai atribut `android:contentDescription` ke sumber daya string `ice_cream_sandwiches`.
8. Tambahkan `TextView` yang akan muncul di sebelah sandwich es krim sebagai deskripsi, dengan atribut berikut:

| Atribut TextView | Nilai |
|--------------------------|--------------------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:padding | "@dimen/padding_wide" |
| android:layout_below | "@+id/donut" |
| android:layout_toRightOf | "@id/ice_cream" |
| android:text | "@string/ice_cream_sandwiches" |

9. Tambahkan `ImageView` ketiga ke layout untuk froyo, menggunakan atribut berikut:

| Atribut ImageView untuk ice_cream | Nilai |
|-----------------------------------|----------------------------------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:padding | "@dimen/padding_regular" |
| android:id | "@+id/froyo" |
| android:layout_below | "@+id/ice_cream" |
| android:contentDescription | "FroYo is premium self-serve frozen yogurt." |
| android:src | "@drawable/froyo_circle" |

10. Ekstrak nilai atribut `android:contentDescription` ke sumber daya string `froyo`.

11. Tambahkan `TextView` yang akan muncul di sebelah gambar froyo sebagai deskripsi, dengan atribut berikut:

| Atribut TextView | Nilai |
|--------------------------|-----------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:padding | "@dimen/padding_wide" |
| android:layout_below | "@+id/ice_cream" |
| android:layout_toRightOf | "@+id/froyo" |
| android:text | "@string/froyo" |

6.3 Tambahkan metode onClick untuk tampilan gambar

Anda bisa menambahkan atribut `android:onClick` ke View apa pun untuk membuatnya bisa diklik sebagai tombol. Di langkah ini, Anda akan menambahkan `android:onClick` ke gambar di layout **content_main.xml**. Anda juga perlu menambahkan metode untuk atribut `android:onClick` yang akan dipanggil. Metode ini, untuk tugas ini, menampilkan pesan toast yang menampilkan gambar mana yang diketuk. (Di tugas berikutnya, Anda akan memodifikasi metode untuk meluncurkan aktivitas lain yang disebut **OrderActivity**.)

1. Tambahkan sumber daya string berikut ke file **strings.xml** yang akan ditampilkan di pesan toast:

```
<string name="donut_order_message">You ordered a donut.</string>
<string name="ice_cream_order_message">You ordered an ice cream sandwich.</string>
<string name="froyo_order_message">You ordered a FroYo.</string>
```

2. Tambahkan metode `displayToast()` berikut untuk menampilkan pesan toast:

```
public void displayToast(String message) {
    Toast.makeText(getApplicationContext(), message,
        Toast.LENGTH_SHORT).show();
}
```

3. Tambahkan metode `showFoodOrder()` berikut di akhir **MainActivity*** (sebelum tanda kurung penutup). Untuk tugas ini, gunakan metode `displayToast()` untuk menampilkan pesan toast:

```
/**
 * Displays a toast message for the food order
 * and starts the OrderActivity activity.
 * @param message Message to display.
 */
public void showFoodOrder(String message) {
    displayToast(message);
}
```

Tip: Empat baris pertama adalah komentar dalam format Javadoc, yang memudahkan pemahaman kode dan juga membantu membuat dokumentasi untuk kode Anda jika Anda menggunakan [Javadoc](#). Menambahkan komentar seperti itu ke setiap metode baru yang Anda buat merupakan praktik terbaik. Untuk informasi selengkapnya tentang cara menulis komentar, lihat [Cara Menulis Komentar untuk Alat Javadoc](#).

Meskipun Anda mungkin telah menambahkan metode ini di posisi mana pun di dalam **MainActivity**, sebaiknya tempatkan metode Anda sendiri *di bawah* metode yang telah disediakan di dalam **MainActivity** oleh template.

1. Tambahkan metode berikut di akhir **MainActivity** (Anda bisa menambahkannya sebelum `showFoodOrder()`):

```
/**
 * Shows a message that the donut image was clicked.
 */
public void showDonutOrder(View view) {
    showFoodOrder(getString(R.string.donut_order_message));
}

/**
 * Shows a message that the ice cream sandwich image was clicked.
 */
public void showIceCreamOrder(View view) {
    showFoodOrder(getString(R.string.ice_cream_order_message));
}

/**
 * Shows a message that the froyo image was clicked.
 */
public void showFroyoOrder(View view) {
    showFoodOrder(getString(R.string.froyo_order_message));
}
```

2. Tambahkan atribut `android:onClick` ke tiga ImageView di `content_main.xml`:

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:id="@+id/donut"
    android:layout_below="@+id/choose_dessert"
    android:contentDescription="@string/donut"
    android:src="@drawable/donut_circle"
    android:onClick="showDonutOrder"/>
. .
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:id="@+id/ice_cream"
    android:layout_below="@+id/donut"
    android:contentDescription="@string/ice_cream_sandwich"
    android:src="@drawable/icecream_circle"
    android:onClick="showIceCreamOrder"/>
. .
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:id="@+id/froyo"
    android:layout_below="@+id/ice_cream"
    android:contentDescription="@string/froyo"
    android:src="@drawable/froyo_circle"
    android:onClick="showFroyoOrder"/>
```

3. Jalankan aplikasi.

Mengeklik gambar donat, sandwich es krim, atau froyo akan menampilkan pesan toast tentang pesanan, seperti yang ditampilkan dalam gambar di bawah ini.

The screenshot shows a mobile application interface for "Droid Cafe". At the top, there is a blue header bar with the text "Droid Cafe" on the left and three vertical dots on the right. The main content area has a white background. A large title "Droid Desserts" is centered at the top. Below it, the text "Choose a dessert." is displayed. There are three dessert items listed, each with an image and a description:

- Donuts are glazed and sprinkled with candy.
- Ice cream sandwiches have chocolate wafers and vanilla filling
- You ordered an ice cream sandwich.
FroYo is premium self-serve frozen yogurt.

A red circular button with a white envelope icon is located on the right side of the third item's description. The bottom of the screen features a black navigation bar with standard Android icons: a triangle pointing left, a circle, and a square.

Droid Cafe

Droid Desserts

Choose a dessert.

Donuts are glazed and sprinkled with candy.

Ice cream sandwiches have chocolate wafers and vanilla filling

You ordered an ice cream sandwich.
FroYo is premium self-serve frozen yogurt.

4G 5:30

◀ ○ □

Tugas 7: Menggunakan tombol radio

Tombol radio adalah kontrol masukan yang berguna untuk memilih hanya satu opsi dari serangkaian opsi. Anda harus menggunakan tombol radio jika Anda ingin pengguna melihat semua opsi yang tersedia secara berdampingan. Jika tidak perlu menampilkan semua opsi secara berdampingan, sebaiknya gunakan spinner.

Nanti dalam praktik ini, Anda akan menambahkan aktivitas dan layout layar lainnya untuk menyetel opsi pengantaran untuk pesanan makanan, dan menggunakan tombol radio untuk pilihan pengantaran.

Untuk ringkasan dan kode contoh lainnya untuk tombol radio, lihat [Tombol Radio](#).

7.1 Tambahkan aktivitas lain

Sesuai yang telah Anda pelajari di pelajaran sebelumnya, sebuah *aktivitas* merepresentasikan satu layar di aplikasi Anda tempat pengguna Anda bisa melakukan satu tugas yang terfokus. Anda sudah memiliki satu aktivitas, `MainActivity.java`. Anda sekarang akan menambahkan aktivitas lain untuk menyetel opsi pengiriman untuk pesanan, dan menggunakan intent eksplisit untuk meluncurkan aktivitas kedua.

1. Klik kanan folder `com.example.android.droidcafe` di kolom kiri dan pilih **New > Activity > Empty Activity**. Edit Nama Aktivitas agar menjadi **OrderActivity** dan Nama Layout menjadi **activity_order**. Jangan ubah opsi lain, dan klik **Finish**.

Kelas `OrderActivity` sekarang seharusnya dicantumkan di bawah `MainActivity` di folder `java` dan `activity_order.xml` sekarang seharusnya dicantumkan di folder `layout`. Template Empty Activity ditambahkan di file berikut.

2. Buka **MainActivity**. Ubah metode `showFoodOrder()` untuk membuat intent eksplisit untuk memulai `OrderActivity` :

```
public void showFoodOrder(String message) {
    displayToast(message);
    Intent intent = new Intent(this, OrderActivity.class);
    startActivity(intent);
}
```

3. Jalankan aplikasi. Mengeklik tombol gambar sekarang akan meluncurkan aktivitas kedua, yaitu layar kosong. (Pesan toast muncul sebentar di layar kosong.)

7.2 Tambahkan layout untuk tombol radio

Untuk membuat setiap opsi tombol radio, Anda akan membuat elemen `RadioButton` di file layout `activity_order.xml`, yang ditautkan ke `OrderActivity`. Setelah mengedit file layout, layout untuk tombol radio di `OrderActivity` harus tampak

Choose a delivery method:

- Same day messenger service
- Next day ground delivery
- Pick up

seperti gambar di bawah ini.

Karena pilihan tombol radio bersifat eksklusif secara mutual, Anda akan mengelompokkan tombol-tombol ini di dalam `RadioGroup`. Dengan mengelompokkan tombol radio, sistem Android memastikan hanya satu tombol radio yang bisa dipilih dalam satu waktu.

Catatan: Urutan cantuman elemen `'RadioButton'` menentukan urutan tampilannya di layar.

1. Buka `activity_order.xml` dan tambahkan elemen `Textview` dengan id `order_intro_text` :

| Atribut TextView | Nilai |
|-----------------------------|-----------------------------|
| android:id | "@+id/order_intro_text" |
| android:layout_width | "match_parent" |
| android:layout_height | "wrap_content" |
| android:layout_marginTop | "24dp" |
| android:layout_marginBottom | "6dp" |
| android:textSize | "18sp" |
| android:text | "Choose a delivery method." |

2. Ekstrak sumber daya string untuk "Choose a delivery method:" menjadi `choose_delivery_method`.
3. Ekstrak sumber daya dimensi untuk nilai margin:
 4. "24dp" ke `text_margin_top`
 5. "6dp" ke `text_margin_bottom`
 6. "18sp" ke `intro_text_size`
7. Tambahkan `RadioGroup` ke layout di bawah `TextView` yang baru saja Anda tambahkan:

```
<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_below="@+id/order_intro_text">
</RadioGroup>
```

8. Tambahkan tiga elemen `RadioButton` berikut di dalam `RadioGroup` menggunakan atribut berikut. Entri `"onRadioButtonClicked"` untuk atribut `onClick` akan disorot sampai Anda menambahkan metode tersebut di tugas berikutnya.

| Atribut #1 RadioButton | Nilai |
|------------------------|------------------------------|
| android:id | "@+id/sameday" |
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:text | "Same day messenger service" |
| android:onClick | "onRadioButtonClicked" |

| Atribut #2 RadioButton | Nilai |
|------------------------|----------------------------|
| android:id | "@+id/nextday" |
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:text | "Next day ground delivery" |
| android:onClick | "onRadioButtonClicked" |

| Atribut #3 RadioButton | Nilai |
|------------------------|------------------------|
| android:id | "@+id/pickup" |
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:text | "Pick up" |
| android:onClick | "onRadioButtonClicked" |

9. Ekstrak tiga sumber daya string untuk atribut `android:text` ke nama-nama berikut, sehingga string bisa dengan mudah diterjemahkan:

- o `same_day_messenger_service`
- o `next_day_ground_delivery`
- o `pick_up`

7.3 Tambahkan handler klik tombol radio

Atribut `android:onClick` untuk setiap elemen tombol radio menetapkan metode `onRadioButtonClicked()` untuk menangani kejadian klik. Dengan demikian, Anda perlu menambahkan metode `onRadioButtonClicked()` baru di kelas `OrderActivity`.

Biasanya, aplikasi Anda akan menampilkan pesan mengenai tipe pengantaran seperti apa yang dipilih. Anda akan membuat ini dengan pesan toast dengan membuat metode yang bernama `displayToast()` di `OrderActivity`.

Di metode `onRadioButtonClicked()`, Anda akan menggunakan blok `switch case` untuk memeriksa apakah tombol radio telah diklik. Di akhir blok `switch case`, Anda akan menambahkan pernyataan `default` yang menampilkan pesan `log` jika tidak satu pun tombol radio yang dicentang.

1. Buka `strings.xml` dan buat sumber daya string berikut:
 - i. Sebuah sumber daya yang bernama `chosen` untuk string "Chosen: " (sertakan spasi setelah titik dua dan tanda tanya).
 - ii. Sebuah sumber daya yang bernama `nothing_clicked` untuk string " onRadioButtonClicked: Nothing clicked."
2. Buka `OrderActivity` dan tambahkan pernyataan berikut untuk mendefinisikan `TAG_ACTIVITY` untuk pesan `log`:

```
private static final String TAG_ACTIVITY =
    OrderActivity.class.getSimpleName();
```

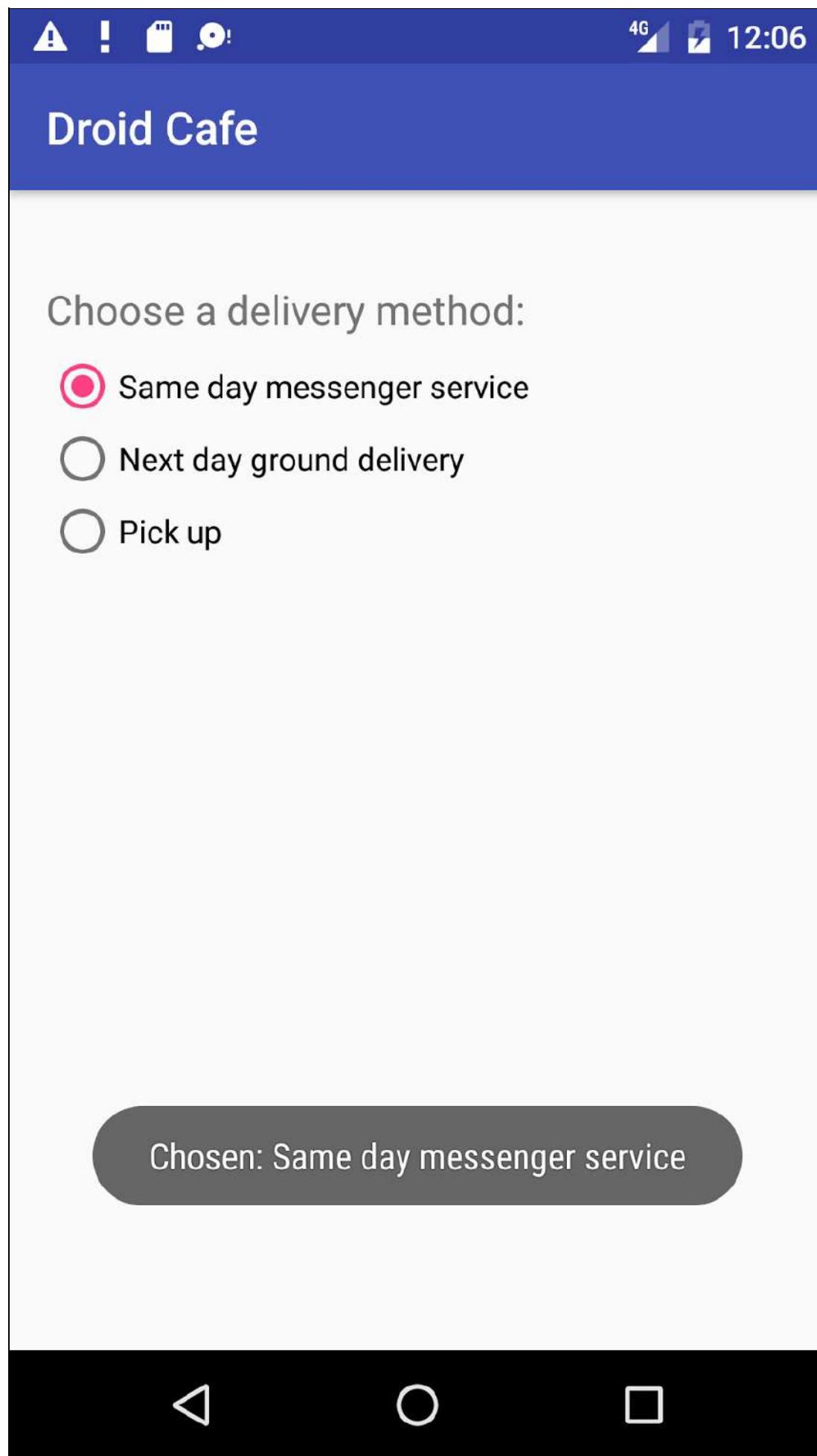
3. Tambahkan metode `displayToast` berikut ke `OrderActivity`:

```
public void displayToast(String message) {
    Toast.makeText(getApplicationContext(), message,
        Toast.LENGTH_SHORT).show();
}
```

4. Tambahkan metode `onRadioButtonClicked()` berikut, yang memeriksa untuk melihat apakah tombol radio telah dicentang, dan menggunakan blok `switch case` untuk menentukan item tombol radio mana yang dipilih, untuk menyetel `message` yang sesuai untuk item tersebut untuk digunakan bersama `displayToast()`:

```
public void onRadioButtonClicked(View view) {  
    // Is the button now checked?  
    boolean checked = ((RadioButton) view).isChecked();  
    // Check which radio button was clicked  
    switch(view.getId()) {  
        case R.id.sameday:  
            if (checked)  
                // Same day service  
                displayToast(getString(R.string.chosen) +  
                            getString(R.string.same_day_messenger_service));  
            break;  
        case R.id.nextday:  
            if (checked)  
                // Next day delivery  
                displayToast(getString(R.string.chosen) +  
                            getString(R.string.next_day_ground_delivery));  
            break;  
        case R.id.pickup:  
            if (checked)  
                // Pick up  
                displayToast(getString(R.string.chosen) +  
                            getString(R.string.pick_up));  
            break;  
        default:  
            Log.d(TAG_ACTIVITY, getString(R.string.nothing_clicked));  
            break;  
    }  
}
```

5. Jalankan aplikasi. Ketuk sebuah pesan untuk melihat aktivitas OrderActivity, yang menampilkan pilihan pengantaran. Ketuk pilihan pengantaran dan Anda akan melihat pesan toast di bagian bawah layar dengan pilihan, seperti yang ditampilkan dalam gambar di bawah ini.



Kode Solusi

Proyek Android Studio: [DroidCafe Part 1](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Anda juga bisa melakukan tindakan langsung dari keyboard dan mengganti kunci Return (Enter) dengan kunci "kirim", seperti untuk memutar nomor telepon: 

Untuk tantangan ini, gunakan atribut `android:imeOptions` untuk komponen `EditText` dengan nilai `actionSend`:

```
    android:imeOptions="actionSend"
```

Di metode `onCreate()` untuk aktivitas utama ini, Anda bisa menggunakan `setOnEditorActionListener()` untuk menyetel listener untuk tampilan `EditText` untuk mendeteksi apakah tombol ditekan:

```
EditText editText = (EditText) findViewById(R.id.editText_main);
if (editText != null)
    editText.setOnEditorActionListener(new TextView.OnEditorActionListener() {
        ...
});
```

Untuk membantu menyetel listener, lihat "Menetapkan Tindakan Masukan" di (<https://developer.android.com/training/keyboard-input/style.html#Action>) dan "Menetapkan Tindakan Keyboard" di [Bidang Teks](#).

Langkah berikutnya adalah mengganti `onEditorAction()` dan menggunakan konstanta `IME_ACTION_SEND` di kelas `EditorInfo` untuk merespons tombol yang ditekan. Dalam contoh berikut, tombol digunakan untuk memanggil metode `dialNumber()` untuk memanggil nomor telepon:

```
@Override
public boolean onEditorAction(TextView textView, int actionId, KeyEvent keyEvent) {
    boolean mHandled = false;
    if (actionId == EditorInfo.IME_ACTION_SEND) {
        dialNumber();
        mHandled = true;
    }
    return mHandled;
}
```

Untuk menyelesaikan tantangan, buat metode `dialNumber()` yang menggunakan intent implisit dengan `ACTION_DIAL` untuk meneruskan nomor telepon ke aplikasi lain yang bisa memanggil nomor. Akan terlihat seperti ini:

```

private void dialNumber() {
    EditText editText = (EditText) findViewById(R.id.editText_main);
    String mPhoneNum = null;
    if (editText != null) mPhoneNum = "tel:" + editText.getText().toString();
    Log.d(TAG, "dialNumber: " + mPhoneNum);
    Intent intent = new Intent(Intent.ACTION_DIAL);
    intent.setData(Uri.parse(mPhoneNum));
    if (intent.resolveActivity(getApplicationContext()) != null) {
        startActivity(intent);
    } else {
        Log.d("ImplicitIntents", "Can't handle this!");
    }
}

```

*Rangkuman

Dalam praktik ini, Anda telah mempelajari cara:

- Menyiapkan atribut layout XML untuk mengontrol keyboard untuk elemen `EditText`.
 - Menggunakan nilai `textAutoCorrect` untuk atribut `android:inputType` untuk mengubah keyboard sehingga akan menyarankan koreksi ejaan.
 - Menggunakan nilai `textCapSentences` untuk atribut `android:inputType` untuk memulai setiap kalimat baru dengan huruf besar.
 - Menggunakan nilai `textPassword` untuk atribut `android:inputType` untuk menyembunyikan sandi saat memasukkannya.
 - Menggunakan nilai `textEmailAddress` untuk atribut `android:inputType` untuk menampilkan keyboard email, bukan keyboard standar.
 - Menggunakan nilai `phone` untuk atribut `android:inputType` untuk menampilkan keypad telepon, bukan keyboard standar.
 - Tantangan: Menggunakan atribut `android:imeOptions` dengan nilai `actionSend` untuk melakukan tindakan langsung dari keyboard dan mengganti tombol Return dengan tombol tindakan, seperti intent implisit ke aplikasi lain untuk memutar nomor telepon.
- Menggunakan kontrol masukan `spinner` untuk menyediakan menu tarik-turun dan menulis kode untuk mengontrolnya:
 - Menggunakan `ArrayAdapter` untuk menetapkan larik nilai teks sebagai item menu spinner.
 - Implementasikan antarmuka `AdapterView.OnItemSelectedListener`, yang juga memerlukan penambahan metode callback `onItemSelected()` dan `onNothingSelected()` untuk mengaktifkan spinner dan listenernya.
 - Menggunakan metode callback `onItemSelected` untuk mengambil item yang dipilih di menu spinner menggunakan `getItemAtPosition`.
- Menggunakan `AlertDialog.Builder`, sebuah subkelas `AlertDialog` untuk membangun dialog peringatan standar, menggunakan `setTitle` untuk menyetel judulnya, `setMessage` untuk menyetel pesannya, dan `setPositiveButton` dan `setNegativeButton` untuk memilih tombolnya.
- Menggunakan picker tanggal dan waktu standar:
 - Tambahkan fragmen untuk picker tanggal dan untuk meluaskan kelas `DialogFragment` untuk mengimplementasikan `DatePickerDialog.OnDateSetListener` untuk picker tanggal standar dengan sebuah listener.
 - Tambahkan fragmen untuk picker waktu dan untuk meluaskan kelas `DialogFragment` untuk mengimplementasikan `TimePickerDialog.OnTimeSetListener` untuk picker waktu standar dengan sebuah listener.
 - Mengimplementasikan metode `onDateSet()`, `onTimeSet()`, dan `onCreateDialog()`.
 - Menggunakan metode `onFinishDateDialog()` dan `onFinishTimeDialog()` untuk mengambil tanggal dan waktu yang dipilih.
- Menggunakan gambar dalam proyek:
 - Menyalin gambar ke dalam proyek, dan mendefinisikan elemen `ImageView` untuk menggunakannya.
 - Tambahkan atribut `android:onClick` untuk menjadikan elemen `ImageView` bisa diklik seperti tombol. Anda bisa membuat View apa pun dapat diklik dengan dengan atribut `android:onClick`.
- Menggunakan tombol radio:

- Membuat aktivitas kedua.
- Tambahkan elemen `RadioButton` di dalam `RadioGroup` di aktivitas kedua.
- Membuat handler tombol radio.
- Meluncurkan aktivitas kedua dari klik gambar.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Kontrol Masukan Pengguna](#)

Ketahui selengkapnya

- Panduan Android API, bagian "Kembangkan":
 - [Menetapkan Tipe Metode Masukan](#)
 - [Bidang Teks](#)
 - [Kontrol Masukan](#)
 - [Spinner](#)
 - [Dialog](#)
 - [Fragmen](#)
 - [Kejadian Input](#)
 - [Picker](#)
 - [DateFormat](#)
 - [ImageView](#)
 - [Tombol Radio](#) (bagian Antarmuka Pengguna)
- Spesifikasi Desain Material:
 - [Panduan desain dialog](#)

4.2: Menggunakan Menu Opsi

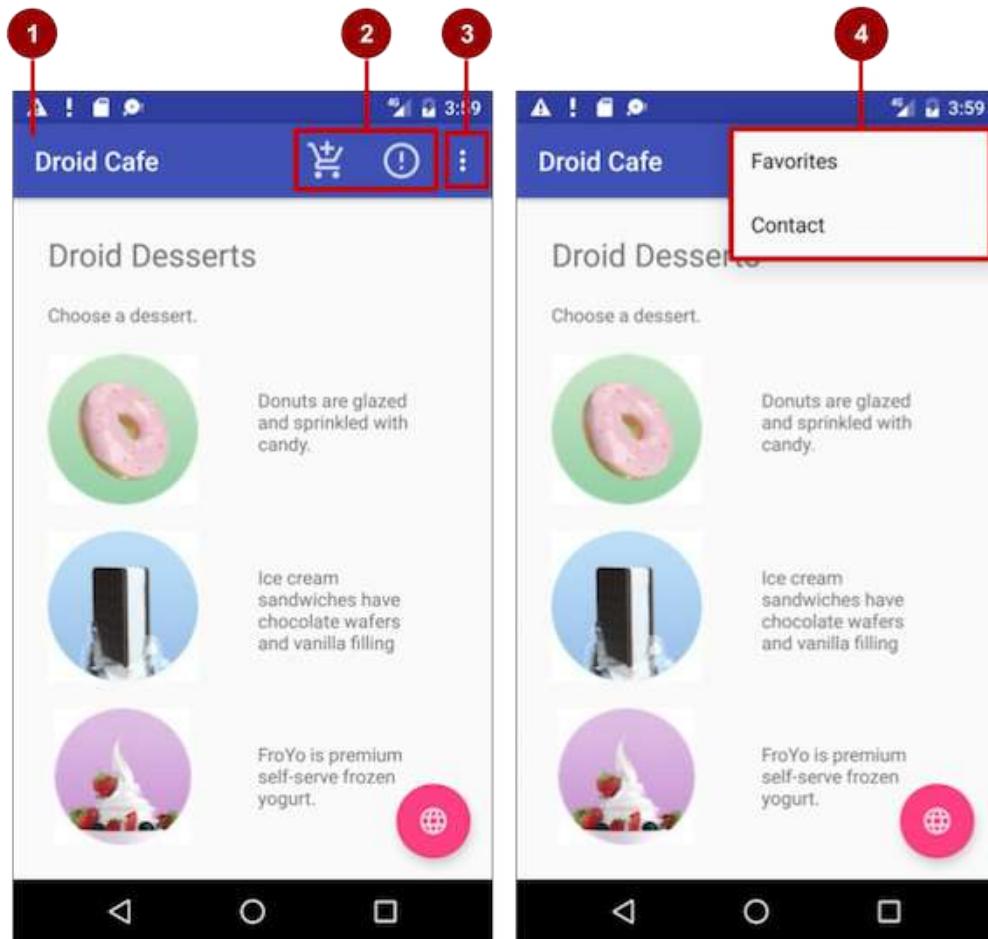
Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Menambahkan item ke menu opsi opsi](#)
- [Tugas 2: Menambahkan ikon untuk item menu](#)
- [Tantangan Penyusunan Kode #1](#)
- [Tugas 3: Menangani item menu yang dipilih](#)
- [Tantangan Penyusunan Kode #2](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Bilah aplikasi (disebut juga *bilah tindakan*) adalah ruang khusus di bagian atas setiap layar aktivitas. Bila Anda membuat aktivitas dari suatu template (seperti Template Aktivitas Dasar), bilah aplikasi secara otomatis disertakan untuk aktivitas dalam grup tampilan akar CoordinatorLayout di bagian atas hierarki tampilan.

Menu opsi dalam bilah aplikasi menyediakan navigasi ke aktivitas lain dalam aplikasi, atau opsi utama yang memengaruhi penggunaan aplikasi itu sendiri—namun bukan yang menjalankan tindakan pada elemen di layar. Misalnya, menu opsi Anda mungkin menyediakan pilihan bagi pengguna untuk mengarahkan ke aktivitas lain, seperti menempatkan urutan, atau untuk tindakan yang memiliki efek global pada aplikasi, seperti mengubah setelan atau informasi profil.

Dalam praktik ini, Anda akan mempelajari tentang menyetel bilah aplikasi dan menu opsi di aplikasi Anda (ditampilkan di gambar di bawah).



Pada gambar di atas:

- Bilah aplikasi.** Bilah aplikasi menyertakan judul aplikasi, menu opsi, dan tombol luapan.
- Ikon tindakan menu opsi.** Dua item menu opsi pertama muncul sebagai ikon dalam bilah aplikasi.
- Tombol luapan.** Tombol luapan (tiga titik vertikal) membuka menu yang menampilkan item menu opsi selengkapnya.
- Menu luapan opsi.** Setelah mengeklik tombol luapan, item menu opsi selengkapnya akan muncul dalam menu luapan.

Item menu opsi muncul di menu luapan opsi (lihat gambar di atas). Akan tetapi, Anda bisa menempatkan beberapa item sebagai ikon—sebanyak yang bisa dimuat—di bilah aplikasi. Menggunakan bilah aplikasi untuk menu opsi akan membuat aplikasi Anda konsisten dengan aplikasi Android lainnya, memungkinkan pengguna untuk dengan cepat memahami cara mengoperasikan aplikasi Anda dan memiliki pengalaman hebat.

Tip: Untuk menyediakan pengalaman pengguna yang sudah akrab dan konsisten, Anda harus menggunakan Menu API untuk menyajikan tindakan dan opsi lain dalam aktivitas kepada pengguna. Lihat [Menu](#) untuk detailnya.

Yang harus sudah Anda KETAHUI

Dari bab sebelumnya, Anda seharusnya sudah memahami cara melakukan yang berikut:

- Membuat dan menjalankan aplikasi di Android Studio.
- Membuat dan mengedit elemen UI menggunakan Layout Editor, memasukkan kode XML secara langsung, dan mengakses elemen dari kode Java Anda.
- Menambahkan fungsionalitas onClick ke tombol.

Yang akan Anda PELAJARI

- Menambahkan item menu ke menu opsi.
- Menambahkan ikon untuk item di menu opsi.
- Menyetel item menu untuk ditampilkan di bilah tindakan.
- Menambahkan handler kejadian untuk klik item menu.

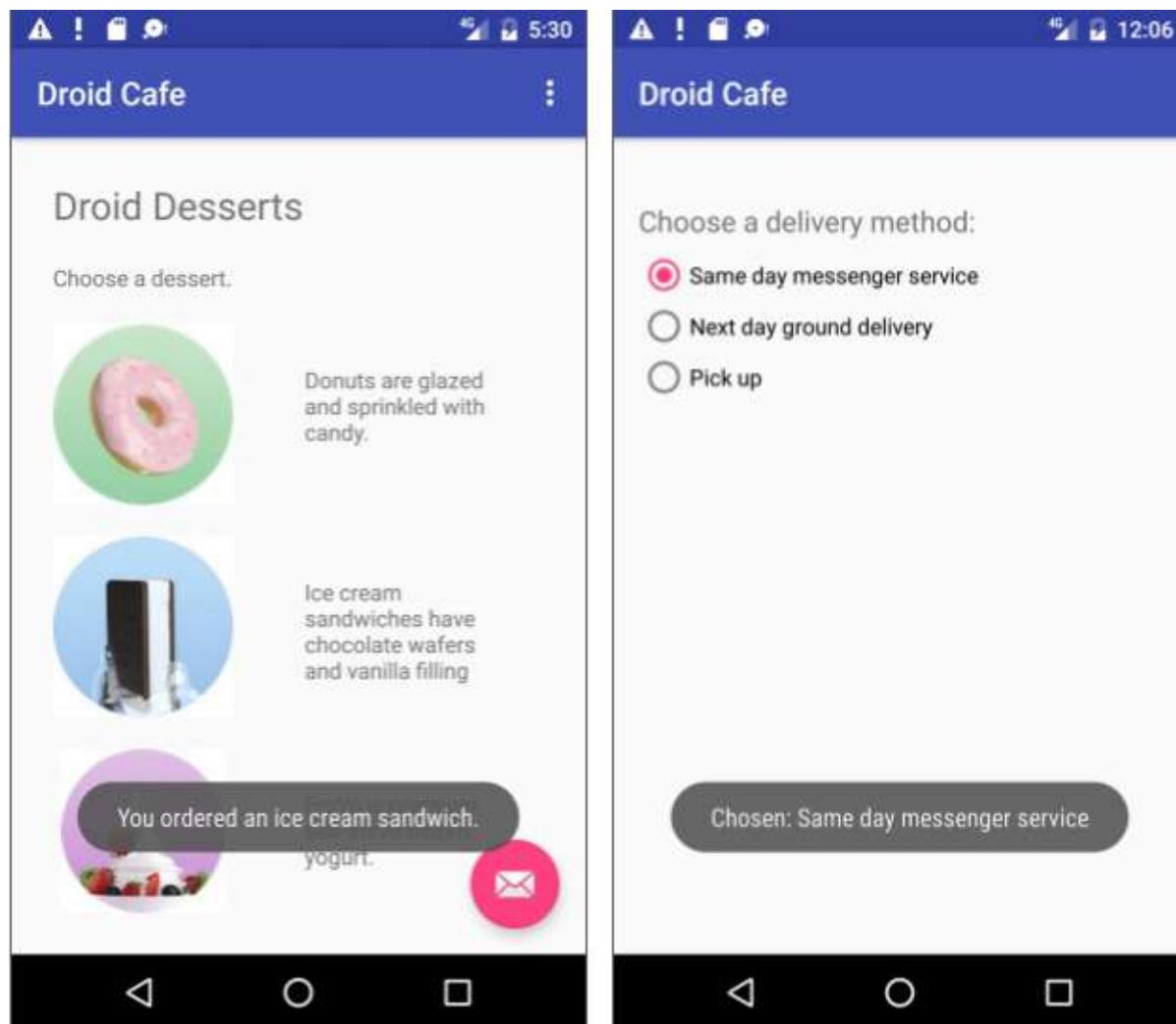
Apa yang akan Anda LAKUKAN

- Melanjutkan menambahkan fitur pada proyek Droid Cafe dari praktik sebelumnya.
- Menambahkan item menu ke menu opsi.
- Menambahkan ikon untuk item menu agar tampil di bilah tindakan.
- Menghubungkan klik item menu ke handler kejadian yang memproses kejadian klik.

Ringkasan Aplikasi

Di praktik sebelumnya, Anda telah membuat aplikasi yang bernama Droid Cafe, yang ditampilkan di gambar di bawah, menggunakan template Basic Activity. Template ini juga menyediakan kerangka menu opsi di bilah aplikasi di bagian atas layar. Anda akan mempelajari cara:

- Mempersiapkan bilah aplikasi.
- Memodifikasi menu opsi.
- Menambahkan ikon untuk beberapa item menu.
- Menampilkan ikon untuk item menu di bilah aplikasi, bukannya menu luapan.
- Menampilkan item di menu luapan, bergantung pada ukuran dan orientasi layar.



Untuk latihan ini, Anda akan menggunakan [Bilah Alat](#) pustaka dukungan `v7 appcompat` sebagai bilah aplikasi. Ada cara lain untuk mengimplementasikan bilah aplikasi. Misalnya, beberapa tema mempersiapkan sebuah [ActionBar](#) sebagai bilah aplikasi secara default. Namun menggunakan [Toolbar](#) appcompat memudahkan penyiapan bilah aplikasi yang bekerja pada banyak perangkat, dan juga memberi Anda ruang untuk menyesuaikan bilah aplikasi Anda seiring pengembangan aplikasi.

Untuk membaca lebih lanjut tentang pertimbangan desain untuk penggunaan bilah aplikasi, lihat [Bilah Aplikasi](#) di Spesifikasi Desain Material.

Untuk memulai proyek dari posisi terakhir yang Anda tinggalkan di praktik sebelumnya, unduh:

[Proyek Android Studio: DroidCafe Part 1](#)

Tugas 1: Menambahkan item ke menu opsi opsi

Anda akan membuka proyek Droid Cafe dari praktik sebelumnya, dan menambahkan item menu dari menu opsi di bilah aplikasi di bagian atas layar.

1.1 Periksa kode bilah aplikasi

1. Buka proyek Droid Cafe dari praktik sebelumnya. Proyek menyertakan file layout berikut di folder `res > layout`:
 - i. `activity_main.xml`: Layout utama untuk MainActivity, layar pertama yang dilihat pengguna.
 - ii. `content_main.xml`: Layout untuk konten layar MainActivity, yang (seperti akan segera Anda lihat) *disertakan di dalam* `activity_main.xml`**.

- iii. **activity_order.xml**: Layout untuk OrderActivity, yang Anda tambahkan di praktik sebelumnya.
- Buka **content_main.xml**: Di praktik sebelumnya, Anda telah menambahkan TextView dan ImageView ke grup view root, yang merupakan `RelativeLayout`.

Perilaku layout untuk `RelativeLayout` disetel ke `@string/appbar_scrolling_view_behavior`, yang mengontrol perilaku pengguliran layar terkait bilah aplikasi di bagian atas. Klik kanan (Control-klik) sumber daya string ini dan pilih **Go To Declaration** untuk melihat nilai sumber daya string sebenarnya, yang didefinisikan dalam sebuah file yang bernama "values.xml". File ini dibuat oleh Android Studio, tidak terlihat di Project: Android view dan tidak boleh diedit. Nilai `@string/appbar_scrolling_view_behavior` sebenarnya di `values.xml` adalah `"android.support.design.widget.AppBarLayout$ScrollingViewBehavior"`.

Untuk informasi selengkapnya tentang perilaku pengguliran, lihat [entri blog Android Design Support Library](#) di Blog Developer Android. Untuk praktik desain yang melibatkan menu bergulir, lihat [Teknik Pengguliran](#) di Spesifikasi Desain Material.

- Buka **activity_main.xml** untuk melihat layout utama, yang menggunakan layout CoordinatorLayout dengan layout `AppBarLayout` yang disematkan. Tag `CoordinatorLayout` dan `AppBarLayout` memerlukan nama yang sepenuhnya memenuhi kualifikasi yang ditetapkan `android.support.design`, yaitu Android Design Support Library.

`AppBarLayout` adalah `LinearLayout` yang menggunakan kelas `Toolbar` di pustaka dukungan, sebagai ganti `ActionBar`, untuk mengimplementasikan bilah aplikasi. Bilah aplikasi adalah bagian di atas layar yang bisa menampilkan judul aktivitas, navigasi, dan item interaktif lainnya. `ActionBar` asli berperilaku berbeda bergantung pada versi Android yang berjalan pada perangkat. Oleh karena itu, jika Anda menambahkan menu opsi, Anda harus menggunakan `Toolbar` pustaka dukungan `v7 appcompat` sebagai bilah aplikasi. Menggunakan `Toolbar` akan memudahkan penyiapan bilah aplikasi yang bekerja pada beragam perangkat, serta memberi Anda ruang untuk menyesuaikan bilah aplikasi Anda nanti seiring pengembangan aplikasi. Bilah alat menyertakan berbagai fitur terbaru, dan bekerja pada perangkat apa pun yang bisa menggunakan pustaka dukungan.

Toolbar di dalam layout ini memiliki id `toolbar`, dan juga ditetapkan, seperti `AppBarLayout`, dengan nama yang berkualifikasi sepenuhnya (`android.support.v7.widget`):

```
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay" />

</android.support.design.widget.AppBarLayout>
```

Untuk detail selengkapnya tentang kelas `AppBarLayout`, lihat [AppBarLayout](#) di Referensi Developer Android. Untuk detail selengkapnya tentang bilah alat, lihat [Toolbar](#) di Referensi Developer Android.

Tip: Layout **activity_main.xml** juga menggunakan pernyataan `include layout` untuk menyertakan keseluruhan layout yang didefinisikan di **content_main.xml**. Pemisahan definisi layout ini memudahkan perubahan *konten* layout yang terpisah dari definisi bilah alat layout dan layout koordinator. Ini adalah praktik terbaik untuk memisahkan konten Anda (yang mungkin perlu diterjemahkan) dari format layout Anda.

- Jalankan aplikasi. Perhatikan bilah di bagian atas layar yang menampilkan nama aplikasi (Droid Cafe). Ini juga menampilkan tombol *luapan tindakan* (tiga titik vertikal) di sebelah kanan. Ketuk tombol luapan untuk melihat menu opsi, yang di tahap ini hanya memiliki satu menu opsi, **Settings**.
- Periksa file **AndroidManifest.xml**. Aktivitas `MainActivity` disetel untuk menggunakan tema `NoActionBar`:

```
    android:theme="@style/AppTheme.NoActionBar"
```

Tema `NoActionBar` didefinisikan di file `styles.xml` (luaskan expand `app > res > values > styles.xml` untuk melihatnya). Gaya dicakup di pelajaran lainnya, tetapi Anda bisa melihat bahwa tema `NoActionBar` menyetel atribut `windowActionBar` ke `false` (tanpa bilah tindakan jendela) dan atribut `windowNoTitle` ke `true` (tanpa judul).

Alasan mengapa nilai ini disetel adalah Anda mendefinisikan bilah aplikasi di layout (`activity_main.xml`) dengan `AppBarLayout`, bukan menggunakan `ActionBar`. Menggunakan salah satu tema `NoActionBar` mencegah aplikasi menggunakan kelas `ActionBar` asli untuk menyediakan bilah aplikasi. Kelas `ActionBar` asli berperilaku berbeda, bergantung pada versi sistem Android yang sedang digunakan perangkat. Sebaliknya, fitur terbaru ditambahkan ke pustaka dukungan versi `Toolbar`, dan tersedia pada setiap perangkat apa pun yang bisa menggunakan pustaka dukungan. Dengan alasan ini, Anda harus menggunakan pustaka dukungan kelas `Toolbar` untuk mengimplementasikan aktivitas bilah aplikasi Anda, bukannya `ActionBar`. Menggunakan Bilah Alat pustaka dukungan memastikan aplikasi Anda akan memiliki perilaku yang konsisten pada rangkaian perangkat yang paling luas.

6. Lihat **MainActivitMainActivity** yang memperluas `AppCompatActivity` dan dimulai dengan metode `onCreate()` :

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    ...
}
```

Setelah menyetel tampilan konten ke layout `activity_main.xml`, metode `onCreate()` menyetel `toolbar` menjadi `Toolbar` yang didefinisikan di layout `activity_main.xml`. Ini kemudian memanggil metode `setSupportActionBar()` aktivitas dan meneruskan `toolbar` padanya, menyetel `toolbar` yang didefinisikan di `activity_main.xml` sebagai bilah aplikasi untuk aktivitas.

Untuk praktik terbaik tentang menambahkan bilah aplikasi di aplikasi Anda, lihat [Menambahkan Bilah Aplikasi](#) di Praktik Terbaik untuk Antarmuka Pengguna.

1.2 Menambahkan lebih banyak item menu ke menu opsi

Anda akan menambahkan item menu berikut ke menu opsi aplikasi Droid Cafe:

- **Order:** Masuk ke layar Order Activity untuk melihat pesanan makanan.
- **Status:** Memeriksa status pesanan makanan.
- **Favorites:** Menampilkan makanan favorit.
- **Contact:** Menghubungi restoran. Karena Anda tidak memerlukan item **Settings** yang ada, anda akan mengubah **Settings** ke **Contact**.

Android menyediakan format XML standar untuk mendefinisikan item menu. Sebagai ganti membangun menu di kode aktivitas Anda, Anda bisa mendefinisikan menu dan semua item menunya dalam sumber daya menu XML. Anda kemudian bisa memekarkan sumber daya menu (muat sebagai objek Menu) di aktivitas atau fragmen Anda:

1. Lihat `menu_main.xml` (luaskan `res > menu` di tampilan Project). Ini mendefinisikan item menu dengan `<item>` `</item>` di dalam `<menu> </menu>` block. Satu-satunya item menu yang tersedia dari template adalah `action_settings` (pilihan **Settings**) yang didefinisikan sebagai:

```
<item
    android:id="@+id/action_settings"
    android:orderInCategory="100"
    android:title="@string/action_settings"
    app:showAsAction="never" />
```

Di Android Studio, atribut `android:title` menampilkan nilai string "Settings" meskipun string didefinisikan sebagai sumber daya. Android Studio menampilkan nilai sehingga Anda bisa melihat sekilas berapa nilainya tanpa harus membuka file sumber daya `strings.xml`. Jika Anda mengeklik string ini, string berubah untuk menampilkan sumber daya string "@string/action_settings".

2. Ubah atribut item `action_settings` berikut untuk membuatnya sebagai item `action_contact` (jangan ubah atribut `android:orderInCategory` yang sudah ada):

| Atribut | Nilai |
|-------------------------------|-----------------------|
| <code>android:id</code> | "@+id/action_contact" |
| <code>android:title</code> | "Contact" |
| <code>app:showAsAction</code> | "never" |

3. Ekstrak string "Contact" yang di-hardcode ke dalam sumber daya string `action_contact`.
 4. Tambahkan item menu baru menggunakan blok `<item> </item>` di dalam `<menu> </menu>`, dan berikan atribut berikut ke item:

| Atribut | Nilai |
|--------------------------------------|---------------------|
| <code>android:id</code> | "@+id/action_order" |
| <code>android:orderInCategory</code> | "10" |
| <code>android:title</code> | "Order" |
| <code>app:showAsAction</code> | "never" |

Atribut `android:orderInCategory` menetapkan urutan kemunculan item menu di menu, dengan angka terkecil muncul lebih tinggi dalam menu. Item Contact disetel ke 100, yang merupakan nomor besar untuk menetapkan bahwa ini akan ditampilkan di bagian bawah, bukan di bagian atas. Anda menyetel item Order ke 10, yang menempatkannya di atas Contact, dan meninggalkan cukup banyak ruang di menu untuk lebih banyak item.

5. Ekstrak string "Order" yang di-hardcode ke dalam sumber daya string `action_order`.
 6. Tambahkan dua item menu lagi dengan cara yang sama dengan atribut berikut:

| Atribut Item Status | Nilai |
|--------------------------------------|----------------------|
| <code>android:id</code> | "@+id/action_status" |
| <code>android:orderInCategory</code> | "20" |
| <code>android:title</code> | "Status" |
| <code>app:showAsAction</code> | "never" |

| Atribut Item Favorit | Nilai |
|--------------------------------------|-------------------------|
| <code>android:id</code> | "@+id/action_favorites" |
| <code>android:orderInCategory</code> | "40" |
| <code>android:title</code> | "Favorites" |
| <code>app:showAsAction</code> | "never" |

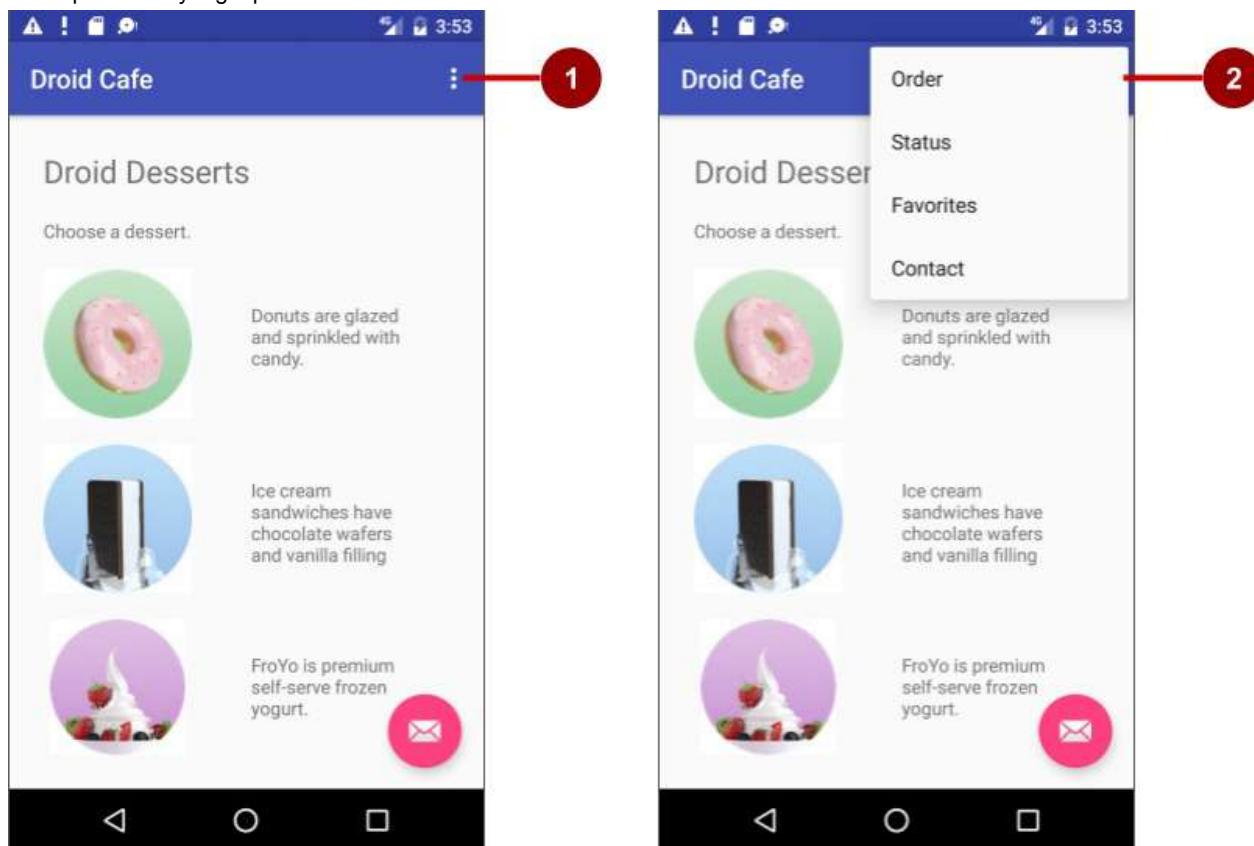
7. Ekstrak "Status" ke dalam sumber daya `action_status` dan "Favorites" ke dalam sumber daya `action_favorites`.
 8. Anda akan menampilkan pesan toast dengan pesan tindakan bergantung pada item menu apa yang dipilih pengguna. Tambahkan nama dan nilai string berikut di `strings.xml` untuk pesan ini:

```
<string name="action_order_message">You selected Order.</string>
<string name="action_status_message">You selected Status.</string>
<string name="action_favorites_message">You selected Favorites.</string>
<string name="action_contact_message">You selected Contact.</string>
```

9. Buka **MainActivity** dan ubah pernyataan `if` di metode `onOptionsItemSelected()` dengan mengganti id `action_settings` dengan id `action_order` baru:

```
if (id == R.id.action_order)
```

Jalankan aplikasi dan ketuk ikon luapan tindakan, yang ditampilkan di sebelah kiri gambar di bawah ini, untuk melihat menu opsi, yang ditampilkan di sebelah kanan gambar di bawah ini. Anda akan segera menambahkan callback untuk merespons item yang dipilih dari menu ini.



Dalam gambar di atas:

1. Ketuk ikon luapan di bilah aplikasi untuk melihat menu opsi.
2. Menu opsi akan ditarik menurun dari bilah aplikasi.

Perhatikan urutan item di menu opsi. Anda menggunakan atribut `android:orderInCategory` untuk menetapkan prioritas item menu di menu: Item Order adalah 10, diikuti oleh Status (20) dan Favorites (40), dan Contact berada di urutan terakhir (100). Tabel berikut menampilkan prioritas item di menu:

| Item Menu | Atribut <code>orderInCategory</code> |
|-----------|--------------------------------------|
| Order | 10 |
| Status | 20 |
| Favorites | 40 |
| Contact | 100 |

Tugas 2. Menambahkan ikon untuk item menu

Jika memungkinkan, Anda bisa menampilkan tindakan yang paling sering digunakan menggunakan ikon di bilah aplikasi, sehingga pengguna bisa mengekliknya tanpa harus mengeklik ikon luapan terlebih dahulu. Dalam tugas ini, Anda akan menambahkan ikon untuk beberapa item menu dan menampilkan beberapa item menu di bilah aplikasi di bagian atas layar sebagai ikon.

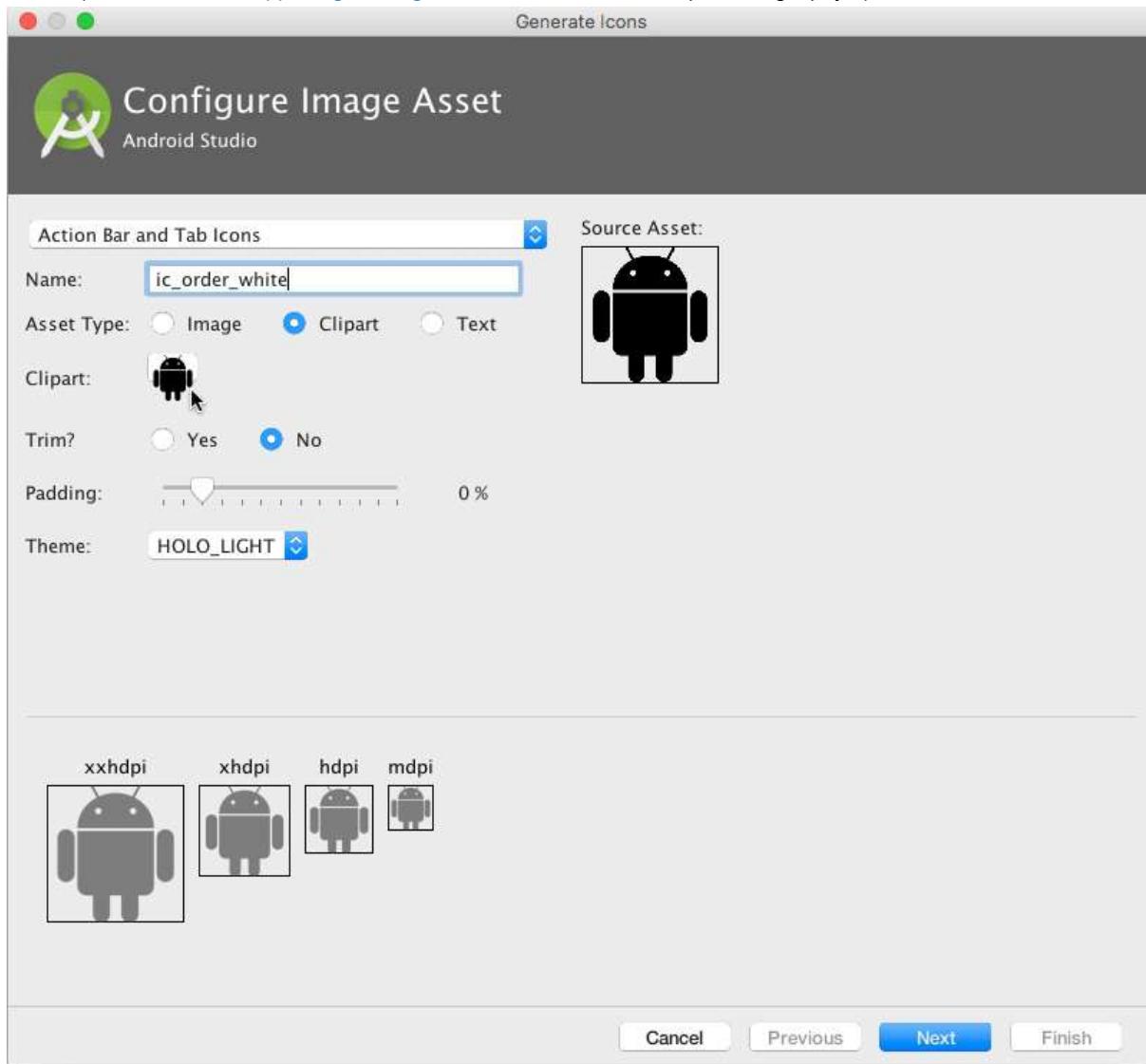
Dalam contoh ini, misalkan tindakan **Order** dan **Status** **dianggap sebagai paling sering digunakan**. **Favorites kadang-kadang digunakan**, dan **Contact*** paling jarang digunakan. Anda bisa menyetel ikon untuk tindakan ini dan menentukan yang berikut:

- **Order** dan **Status** harus selalu ditampilkan di bilah aplikasi.
- **Favorites** harus ditampilkan di bilah aplikasi jika pas ukurannya, jika tidak, harus tampil di menu luapan.
- **Contact** harus tidak muncul di bilah aplikasi, dan hanya muncul di menu luapan.

2.1 Menambahkan ikon untuk item menu

Untuk menetapkan ikon untuk tindakan, Anda terlebih dulu perlu menambahkan ikon sebagai aset gambar di folder **drawable**.

1. Luaskan **res** di tampilan Project, dan klik kanan (atau Kontrol-klik) **drawable**.
2. Pilih **New > Image Asset**. Dialog **Configure Image Asset** akan muncul.
3. Pilih **Action Bar and Tab Items** di menu tarik-turun.
4. Ubah **ic_action_name** ke **ic_order_white** (untuk tindakan **Order**). Layar **Configure Image Asset** akan tampak seperti berikut (lihat [Ikon Create App dengan Image Asset Studio](#) untuk deskripsi selengkapnya.)



5. Klik gambar clipart (logo Android di sebelah "Clipart") untuk memilih gambar clipart sebagai ikon. Laman ikon akan muncul. Klik ikon yang ingin Anda gunakan untuk tindakan **Order** (misalnya, ikon keranjang belanja mungkin sesuai).
6. Pilih **HOLO_DARK** dari menu tarik-turun Theme. Ini akan menyetel ikon menjadi putih dengan latar belakang berwarna gelap (atau hitam). Klik **Next**.
7. Klik **Finish**, dalam dialog Confirm Icon Path.
8. Ulang langkah-langkah di atas untuk ikon **Status** dan **Favorites**, dan beri nama masing-masing **ic_status_white** dan **ic_favorites_white**. Anda mungkin ingin menggunakan ikon i dalam lingkaran untuk **Status** (umumnya digunakan untuk Info) dan ikon hati untuk **Favorites**.

2.2 Tampilkan item menu sebagai ikon di bilah aplikasi

Untuk menampilkan item menu sebagai ikon di bilah aplikasi, gunakan atribut `app:showAsAction` di `menu_main.xml`. Nilai berikut untuk atribut akan menetapkan apakah tindakan akan muncul atau tidak di bilah aplikasi sebagai ikon:

- "always" : Selalu muncul di bilah aplikasi. (Jika tidak cukup ruang, ikon bisa menutupi ikon menu lainnya.)
- "ifRoom" : Muncul di bilah aplikasi jika ada ruang.
- "never" : Tidak pernah muncul di bilah aplikasi; teksnya muncul di menu luapan.

Ikuti langkah-langkah berikut untuk menampilkan beberapa item menu sebagai ikon:

1. Buka `menu_main.xml` lagi dan tambahkan atribut berikut ke item **Order**, **Status**, dan **Favorites** sehingga dua item pertama (**Order** dan **Status**) selalu muncul, dan item **Favorites** hanya muncul jika terdapat ruang untuknya:

| Atribut Item Order | Nilai Lama | Nilai Baru |
|--------------------|------------|----------------------------|
| android:icon | | "@drawable/ic_order_white" |
| app:showAsAction | "never" | "always" |

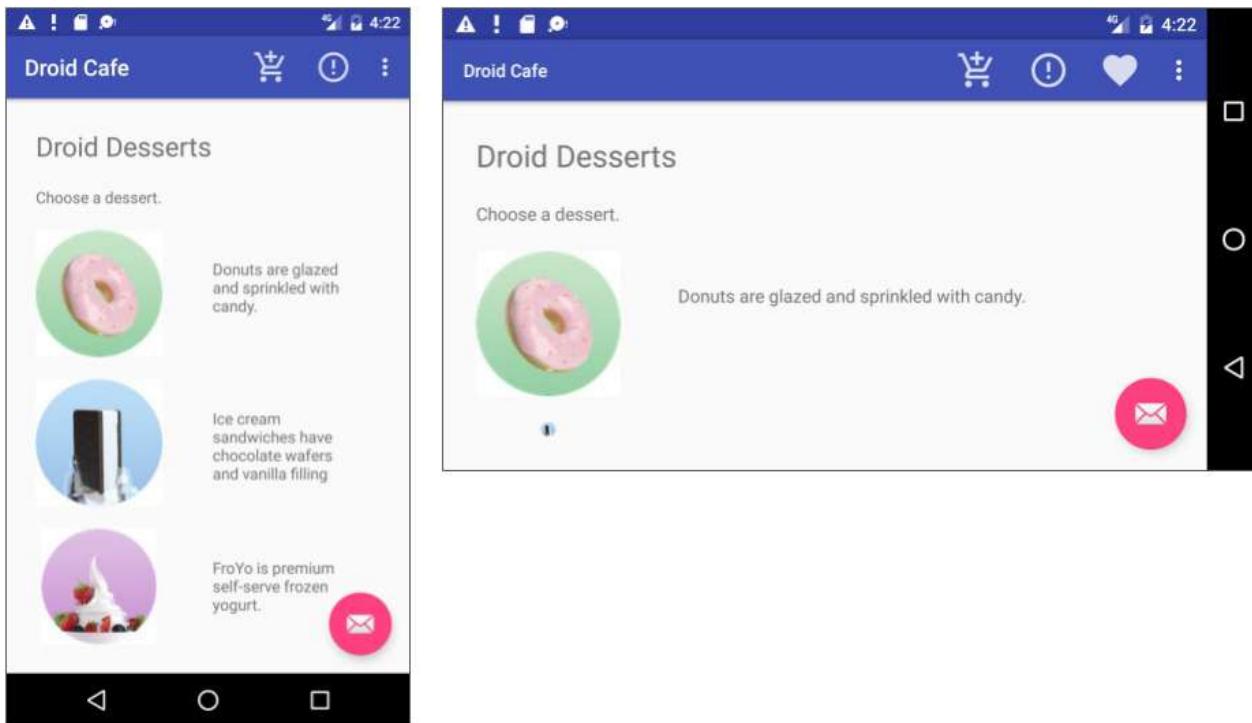
| Atribut Item Status | Nilai Lama | Nilai Baru |
|---------------------|------------|-----------------------------|
| android:icon | | "@drawable/ic_status_white" |
| app:showAsAction | "never" | "always" |

| Atribut Item Favorit | Nilai Lama | Nilai Baru |
|----------------------|------------|--------------------------------|
| android:icon | | "@drawable/ic_favorites_white" |
| app:showAsAction | "never" | "ifRoom" |

2. Jalankan aplikasi. Anda sekarang seharusnya melihat paling tidak dua ikon di bilah aplikasi: ikon untuk **Order** dan ikon untuk **Status** seperti yang ditampilkan dalam gambar di bawah ini. Jika perangkat atau emulator Anda menampilkan orientasi vertikal, opsi **Favorites** dan **Contact** muncul di menu luapan.
3. Putar perangkat Anda ke orientasi horizontal, atau jika Anda menjalankan emulator, klik ikon **Rotete Left** atau **Rotate Right** untuk memutar layar ke orientasi horizontal. Anda seharusnya melihat ketiga ikon di bilah aplikasi untuk **Order**, **Status**, dan **Favorites**.

Tip: Berapa banyak tombol tindakan yang dapat masuk di bilah aplikasi? Bergantung pada orientasi dan ukuran layar perangkat. Lebih sedikit tombol akan muncul di orientasi vertikal, seperti yang ditampilkan di sebelah kiri gambar di bawah, dibandingkan dengan orientasi horizontal seperti yang ditampilkan di sebelah kanan gambar di bawah. Tombol tindakan mungkin tidak menempati lebih dari separuh lebar bilah aplikasi utama.

Tip: Berapa banyak tombol tindakan yang dapat masuk di bilah aplikasi? Bergantung pada orientasi dan ukuran layar perangkat. Lebih sedikit tombol akan muncul di orientasi vertikal, seperti yang ditampilkan di sebelah kiri gambar di bawah, dibandingkan dengan orientasi horizontal seperti yang ditampilkan di sebelah kanan gambar di bawah. Tombol tindakan mungkin tidak menempati lebih dari separuh lebar bilah aplikasi utama.



Tantangan Penyusunan Kode #1

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan 1: Saat Anda meneklik tombol tindakan mengambang dengan ikon email yang muncul di bagian bawah layar, kode di MainActivity menampilkan panel samping yang membuka dan menutup, yang disebut sebagai **snackbar**. Snackbar memberikan umpan balik mengenai sebuah operasi—yaitu menampilkan pesan singkat di bagian bawah layar ponsel cerdas atau di bagian sudut kiri bawah di perangkat yang lebih besar. Untuk informasi selengkapnya, lihat [Snackbar](#).

Pelajari bagaimana aplikasi lain mengimplementasikan tombol tindakan mengambang. Misalnya, aplikasi Gmail menyediakan tombol tindakan mengambang untuk membuat pesan email baru dan aplikasi Kontak menyediakan tombol untuk membuat kontak baru. Untuk informasi selengkapnya tentang tombol tindakan mengambang, lihat [FloatingActionButton](#).

Setelah Anda mengetahui cara menambahkan ikon untuk item menu, gunakan teknik yang sama untuk menambahkan ikon lain dan menetapkan ikon tersebut pada tombol tindakan mengambang, yang menggantikan ikon email. Misalnya, Anda mungkin ingin tombol aksi mengambang memulai sesi chat, yang dalam hal ini Anda mungkin ingin menggunakan ikon yang menampilkan wajah manusia.

Petunjuk: Tombol Aksi Mengambang didefinisikan di `activity_main.xml`.

Saat menambahkan ikon, ubah juga teks yang muncul di snackbar setelah mengetuk tombol aksi mengambang. Anda akan menemukan teks ini di pernyataan `Snackbar.make` di aktivitas utama. Ekstrak sumber daya string untuk teks ini agar menjadi `snackbar_text`.

Tugas 3. Menangani item menu yang dipilih

Dalam tugas ini, Anda akan menambahkan metode untuk menampilkan pesan tentang item menu mana yang diketuk dan menggunakan metode `onOptionsItemSelected` untuk menentukan item menu mana yang diketuk.

2. Jika Anda belum menambahkan metode berikut (di pelajaran sebelumnya) untuk menampilkan pesan toast, tambahkan sekarang:

```
public void displayToast(String message) {
    Toast.makeText(getApplicationContext(), message,
        Toast.LENGTH_SHORT).show();
}
```

Metode `displayToast()` mengambil `message` dari string yang sesuai (misalnya `action_contact_message`).

****3.2 Gunakan handler kejadian onOptionsItemSelected**

Metode `onOptionsItemSelected` menangani pilihan dari menu opsi. Anda akan menambahkan blok `switch case` untuk menentukan item menu mana yang dipilih dan `message` mana yang akan dibuat untuk setiap item yang dipilih. (Sebagai ganti membuat `message` untuk setiap item, Anda bisa mengimplementasikan handler kejadian untuk setiap item yang melakukan sebuah tindakan, misalnya memulai aktivitas lainnya, seperti yang ditampilkan nanti di pelajaran ini.)

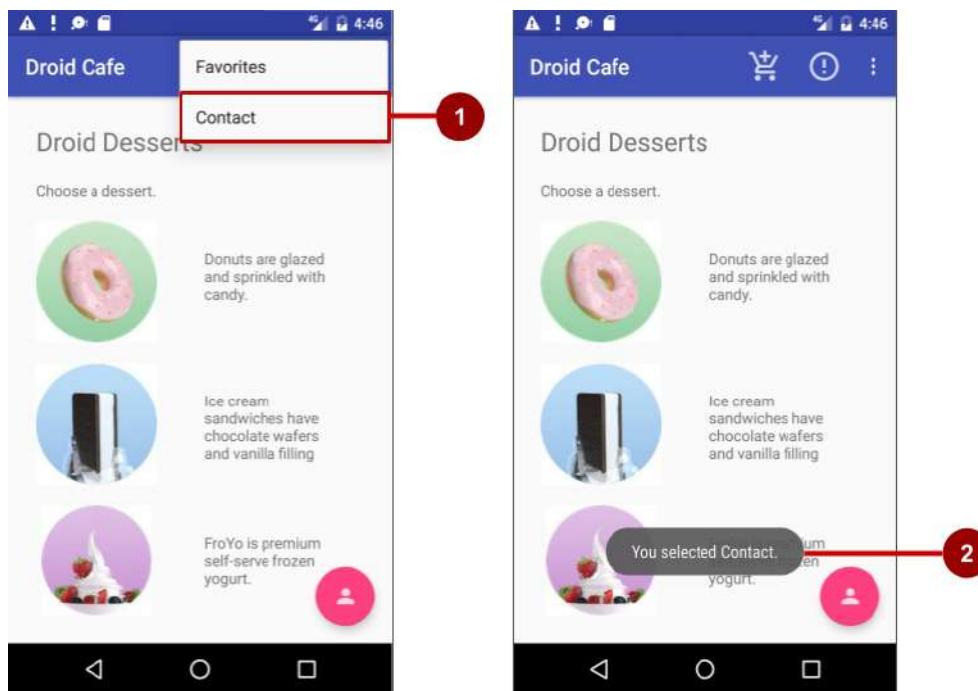
1. Temukan metode `onOptionsItemSelected()`. Pernyataan `if` di metode yang disediakan oleh template ini, menentukan apakah item menu tertentu diklik, menggunakan `id` menu item (`action_order` di contoh berikut):

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_order) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```

2. Ganti pernyataan `if` dan penetapannya ke `id` dengan blok `switch case` berikut yang menyetel `message` yang sesuai berdasarkan `id` item menu:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_order:
            displayToast(getString(R.string.action_order_message));
            return true;
        case R.id.action_status:
            displayToast(getString(R.string.action_status_message));
            return true;
        case R.id.action_favorites:
            displayToast(getString(R.string.action_favorites_message));
            return true;
        case R.id.action_contact:
            displayToast(getString(R.string.action_contact_message));
            return true;
        default:
            // Do nothing
    }
    return super.onOptionsItemSelected(item);
}
```

3. Jalankan aplikasi. Anda sekarang seharusnya melihat pesan toast berbeda di layar, seperti yang ditampilkan di sebelah kanan gambar di bawah, berdasarkan item menu mana yang Anda pilih.



Di gambar di atas:

1. Memilih item **Contact** di menu opsi.
2. Pesan toast yang muncul.

Kode solusi (termasuk tantangan penyusunan kode #1)

Proyek Android Studio: [DroidCafe Part 2](#)

Tantangan Penyusunan Kode #2

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan 2: Di tantangan sebelumnya, Anda telah mengubah ikon untuk tombol aksi mengambang yang muncul di bagian bawah layar `MainActivity` di aplikasi Anda.

Untuk tantangan ini:

1. Ubah ikon untuk tombol aksi mengambang lagi, tetapi kali ini menjadi ikon yang sesuai untuk peta, misalnya ikon dunia.
2. Di `MainActivity`, ganti tindakan untuk menampilkan snack bar dengan intent implisit untuk meluncurkan aplikasi Maps saat tombol aksi mengambang diketuk.
3. Tambahkan koordinat khusus berikut (untuk kantor pusat Google) dan tingkat zoom (12) ke sebuah string yang bernama `google_mtv_coord_zoom12`:

```
<string name="google_mtv_coord_zoom12">geo:37.422114,-122.086744?z=12</string>
```

1 Tambahkan metode berikut untuk memulai aplikasi Maps, yang meneruskan string di atas sebagai data menggunakan intent implisit:

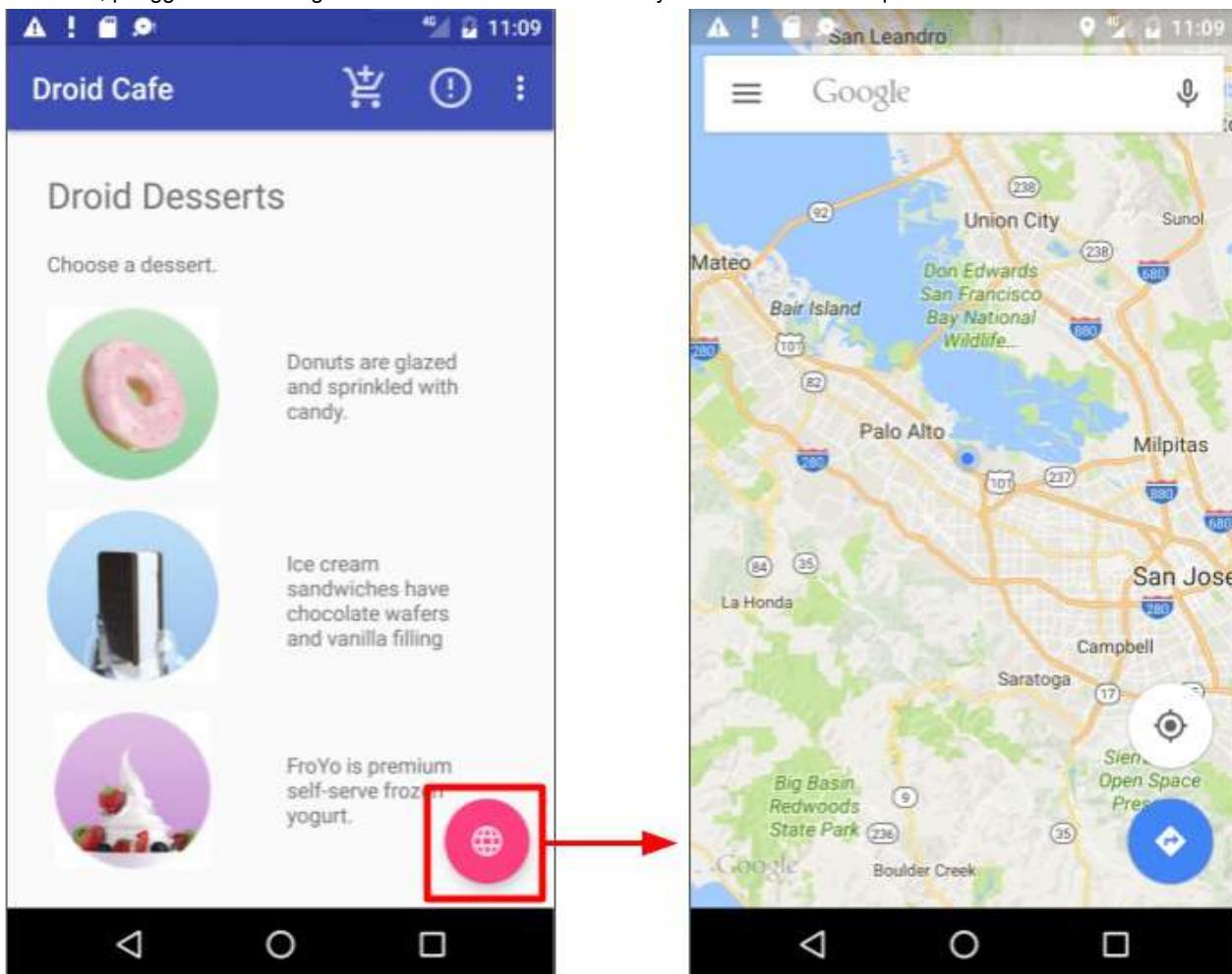
```

public void displayMap() {
    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_VIEW);
    // Using the coordinates for Google headquarters.
    String data = getString(R.string.google_mtv_coord_zoom12);
    intent.setData(Uri.parse(data));
    if (intent.resolveActivity(getApplicationContext()) != null) {
        startActivity(intent);
    }
}

```

Contoh intent implisit, termasuk membuka aplikasi Maps, lihat [Intent Implisit Umum](#) di GitHub.

Setelah mengetuk tombol aksi mengambang untuk beralih ke aplikasi Maps, seperti yang ditampilkan dalam gambar di bawah ini, pengguna bisa mengetuk tombol **Kembali** di bawah layar untuk kembali ke aplikasi Anda.



Kode solusi (termasuk tantangan penyusunan kode #2)

Proyek Android Studio: [DroidCafe Part 3](#)

Anda akan menyelesaikan aplikasi DroidCafe di pelajaran berikutnya.

Rangkuman

Dalam praktik ini, Anda telah mempelajari cara:

- Menyiapkan menu opsi di bilah aplikasi:
 - Menggunakan template Basic Activity untuk secara otomatis menyiapkan menu opsi dan tombol aksi

- mengambang.
- Menggunakan `@string/appbar_scrolling_view_behavior` untuk menyediakan perilaku peng guliran standar pada menu opsi bilah aplikasi.
 - Menggunakan grup tampilan `CoordinatorLayout` dengan kelas `AppBarLayout` untuk membuat menu opsi di bilah aplikasi.
 - Menggunakan pernyataan `include layout` di sebuah file layout XML untuk menyertakan keseluruhan layout yang didefinisikan di file XML lain.
 - Menggunakan tema `NoActionBar` untuk mencegah aplikasi agar tidak menggunakan atribut kelas `ActionBar`, untuk menyetel atribut `windowActionBar` ke `false` (tanpa bilah aplikasi tindakan jendela) dan atribut `windowNoTitle` ke `true` (tanpa judul).
 - Menggunakan metode `onCreate()` untuk memanggil metode `setSupportActionBar()` aktivitas untuk menyetel bilah alat yang didefinisikan di layout sebagai bilah aplikasi untuk aktivitas.
 - Mendefinisikan menu dan semua item dalam sumber daya XML, kemudian memekarkan sumber daya menu dalam aktivitas atau fragmen untuk memuatnya sebagai objek Menu.
 - Menggunakan atribut `android:orderInCategory` untuk menetapkan urutan kemunculan item menu di menu, dengan angka terkecil muncul lebih tinggi dalam menu.
 - Menggunakan atribut `app:showAsAction` untuk menampilkan item menu sebagai ikon di bilah aplikasi.
 - Menambahkan handler kejadian untuk item menu opsi dan menggunakan metode `onOptionsItemSelected()` untuk mengambil pilihan dari menu opsi.
- Menggunakan ikon di proyek:
 - Menambahkan ikon ke proyek dan menggunakananya untuk menampilkan item menu di bilah aplikasi.
 - Tantangan: Mengubah ikon untuk tombol aksi mengambang dan mengubah kode `Snackbar.make`.
 - Tantangan: Membuat intent implisit untuk meluncurkan aplikasi Maps dengan koordinat khusus.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Menu](#)

Ketahui selengkapnya

- Referensi Developer Android:
 - [AppBarLayout](#)
 - [Bilah Alat](#)
 - [Menu](#)
- Android Developers Blog: [Pustaka Dukungan Desain Android](#)
- Spesifikasi Desain Material:
 - [Bilah Aplikasi](#)
 - [Teknik Pengguliran](#)
- Praktik Terbaik untuk Antarmuka Pengguna: [Menambahkan Bilah Aplikasi](#)
- GitHub: [Intent Implisit Umum](#)
- Gambar dan ikon:
 - [Image Asset Studio](#)
 - [Membandingkan Ikon untuk Drawable](#)
 - [Ikon dan sumber daya yang bisa diunduh lainnya](#)

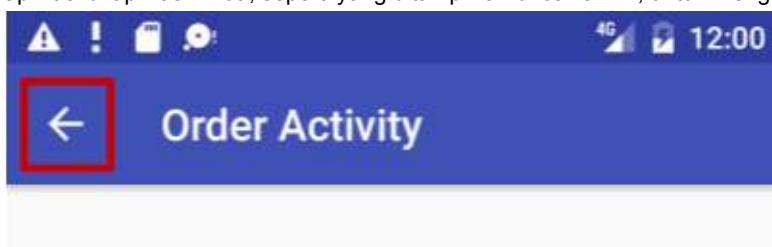
4.3: Menggunakan Bilah Aplikasi dan Tab untuk Navigasi

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Menambahkan tombol Atas untuk navigasi ancestral](#)
- [Tugas 2: Menambahkan navigasi tab dengan tampilan gesek\]\(#task2intro\)](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Pada tahap awal pengembangan aplikasi, Anda harus menentukan jalur yang harus digunakan pengguna melalui aplikasi Anda untuk melakukan sesuatu, seperti menyerahkan pesanan atau menjelajahi materi. Masing-masing jalur memungkinkan pengguna melakukan navigasi ke seluruh bagian, ke dalam, dan kembali dari tugas serta potongan materi yang berbeda-beda dalam aplikasi.

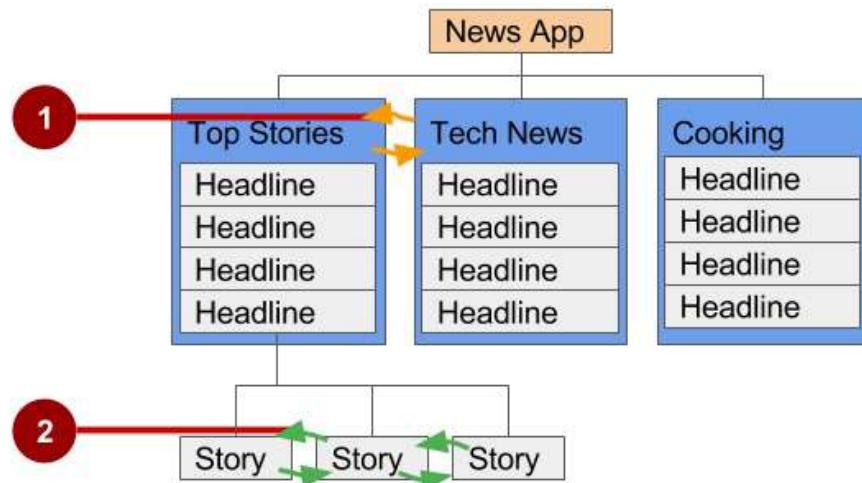
Dalam praktik ini, Anda akan mempelajari cara menambahkan tombol *Naik (panah yang menghadap ke kiri) ke bilah aplikasi di aplikasi Anda, seperti yang ditampilkan di bawah ini, untuk mengarahkan dari layar anak ke atas ke layar induk.



Tombol **Naik** selalu digunakan untuk mengarahkan ke layar induk di hierarki. Tombol ini berbeda dengan tombol **Kembali** (segitiga di bagian bawah layar), yang menyediakan navigasi ke layar apa pun yang sebelumnya dilihat pengguna.

Praktik ini juga memperkenalkan *navigasi tab* yang di situ, tab muncul di sepanjang bagian atas layar, yang menyediakan navigasi ke layar lain. Navigasi tab merupakan solusi yang populer untuk navigasi lateral dari satu layar anak ke layar anak lain yang seinduk, seperti yang ditampilkan di diagram berikut. Tab menyediakan navigasi ke dan dari layar seinduk Top Stories, Tech News, dan Cooking tanpa harus beralih ke induknya. Tab juga bisa menyediakan navigasi ke dan dari story, yang merupakan layar seinduk di bawah induk Top Stories.

Tab paling sesuai untuk empat layar seinduk atau kurang. Pengguna bisa mengetuk tab untuk melihat layar yang berbeda, atau menggesek ke kiri atau ke kanan untuk melihat layar yang berbeda.



Dalam gambar di atas:

1. Navigasi lateral dari satu layar kategori ke layar kategori lainnya
2. Navigasi lateral dari satu layar cerita ke layar cerita lainnya

Yang harus sudah Anda KETAHUI

Dari bab sebelumnya, Anda harus sudah bisa:

- Membuat dan menjalankan aplikasi dalam Android Studio.
- Membuat dan mengedit elemen UI menggunakan Layout Editor, memasukkan kode XML secara langsung, dan mengakses elemen dari kode Java Anda.
- Menambahkan item dan ikon menu ke menu opsi di bilah aplikasi.

Yang akan Anda PELAJARI

Dalam praktik ini, Anda akan mempelajari cara:

- Menambahkan tombol **Naik** ke bilah aplikasi.
- Menyiapkan aplikasi dengan navigasi tab dan tampilan gesek.

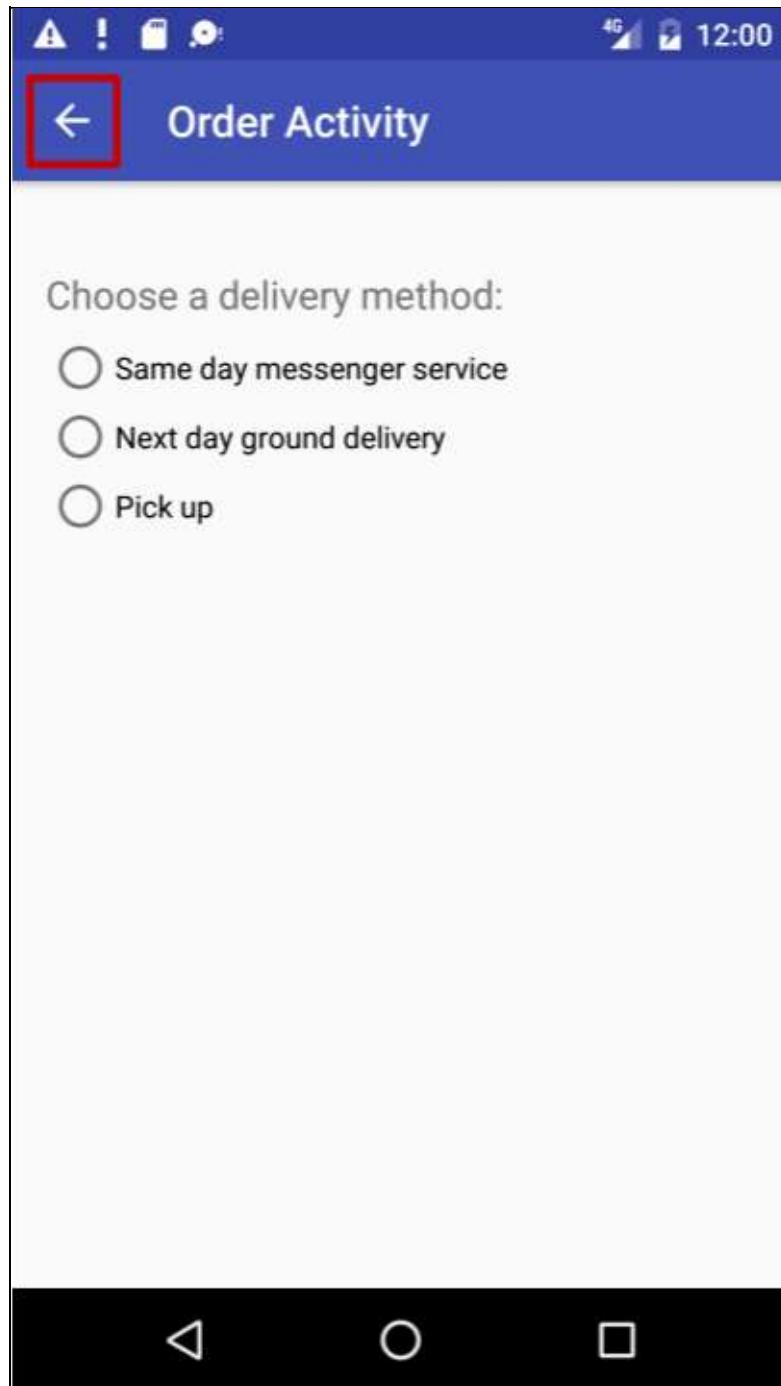
Yang akan Anda LAKUKAN

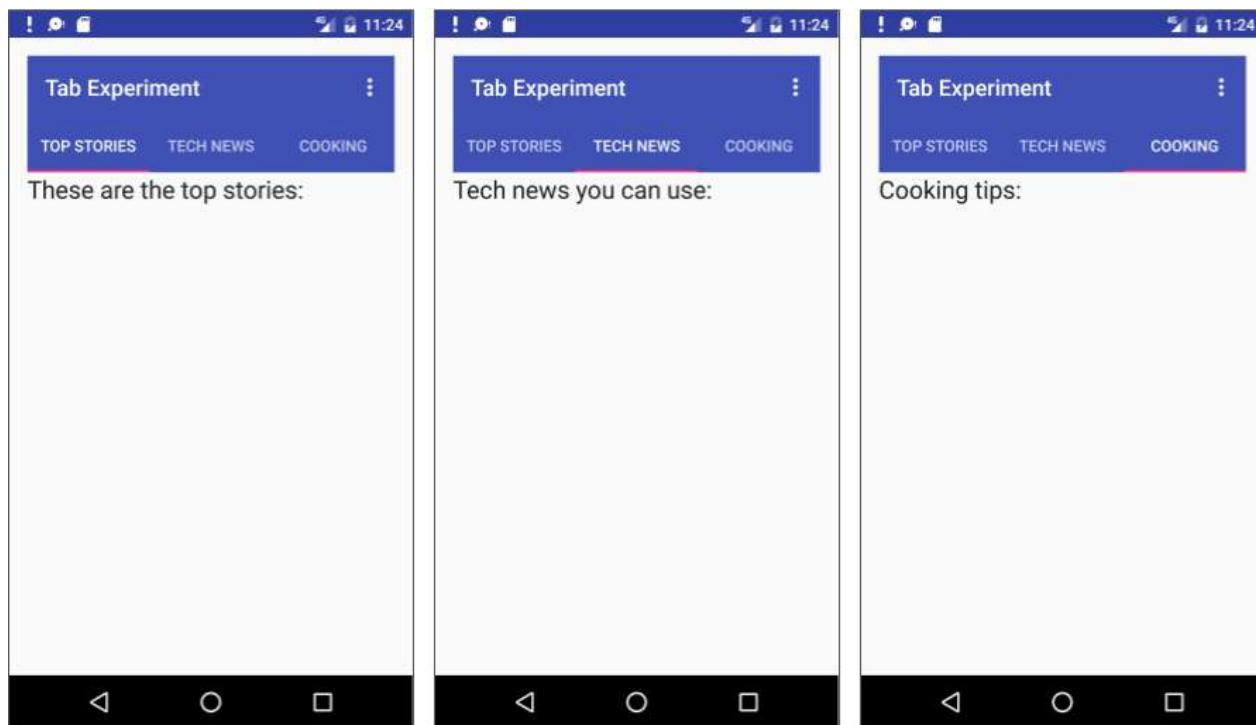
- Melanjutkan menambahkan fitur pada proyek Droid Cafe dari praktik sebelumnya.
- Menyediakan tombol **Naik** ke bilah aplikasi untuk mengarahkan ke layar sebelumnya di dalam sebuah aktivitas.
- Membuat aplikasi baru dengan tab untuk mengarahkan layar aktivitas yang juga bisa digesek.

Ringkasan Aplikasi

Di praktik sebelumnya, Anda telah membuat aplikasi yang bernama Droid Cafe dalam tiga bagian, menggunakan template Basic Activity. Template ini juga menyediakan bilah aplikasi di bagian atas layar. Anda akan mempelajari cara menambahkan tombol **Naik** (panah yang menghadap ke kiri) ke bilah aplikasi untuk navigasi naik dari aktivitas kedua (`orderActivity`) ke aktivitas utama (`MainActivity`). Ini akan menyelesaikan aplikasi Droid Cafe.

Untuk memulai proyek dari posisi terakhir yang Anda tinggalkan di praktik sebelumnya, unduh proyek Android Studio



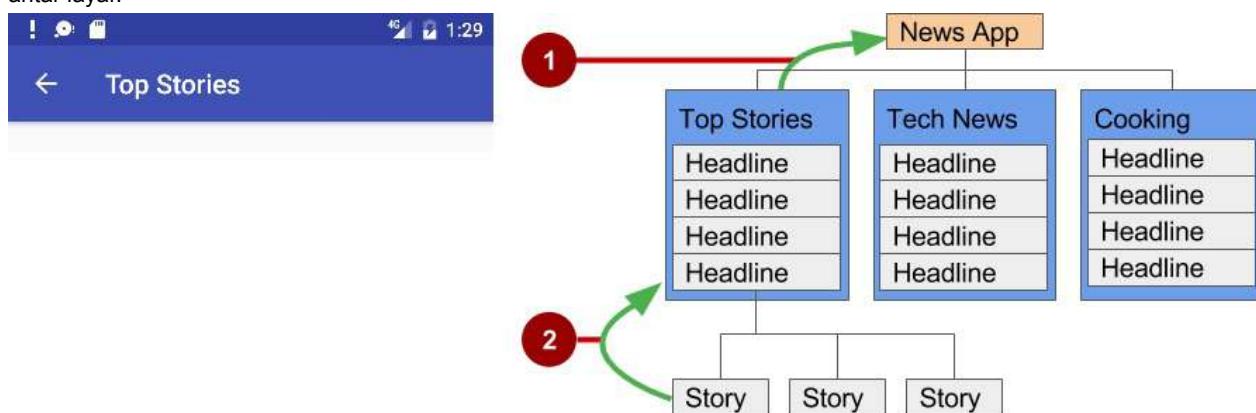


Tugas 1. Menambahkan tombol Naik untuk navigasi ancestral

Aplikasi Anda harus memudahkan pengguna untuk menemukan jalan kembali ke layar utama aplikasi, dengan aktivitas induknya. Salah satu cara untuk melakukannya adalah dengan menyediakan tombol **Naik** di bilah aplikasi untuk semua aktivitas yang menjadi anak aktivitas induk.

Tombol **Naik** menyediakan navigasi "ke atas" ancestral, yang memungkinkan pengguna untuk naik "ke atas" dari laman anak ke laman induk. Tombol **Naik** adalah panah yang menghadap ke kiri di sebelah kiri bilah aplikasi, seperti yang ditampilkan pada sebelah kiri gambar di bawah.

Saat pengguna menyentuh tombol **Naik**, aplikasi akan mengarah ke aktivitas induk. Diagram di sebelah kanan gambar di bawah menampilkan cara tombol **Naik** digunakan untuk mengarahkan di dalam aplikasi berdasarkan hubungan hierarkis antar layar.



Dalam gambar di atas:

1. Mengarahkan dari saudara level pertama ke induk.
2. Mengarahkan dari saudara level kedua ke layar anak level pertama yang berfungsi sebagai layar induk.

Tip: Tombol **Kembali** (segitiga di bagian bawah layar) berbeda dengan tombol **Naik**. Tombol **Kembali** menyediakan navigasi ke layar mana pun yang Anda tampilkan sebelumnya. Jika Anda memiliki sejumlah layar anak yang bisa disusuri oleh pengguna menggunakan pola navigasi lateral (seperti yang dijelaskan di bagian berikutnya), tombol **Kembali** akan

Tip: Tombol **Kembali** (segitiga di bagian bawah layar) berbeda dengan tombol **Naik**. Tombol **Kembali** menyediakan navigasi ke layar mana pun yang Anda tampilkan sebelumnya. Jika Anda memiliki sejumlah layar anak yang bisa disusuri oleh pengguna menggunakan pola navigasi lateral (seperti yang dijelaskan di bagian berikutnya), tombol **Kembali** akan mengirim pengguna kembali ke layar anak sebelumnya, bukan ke layar induk. Gunakan tombol **Naik** jika Anda ingin menyediakan navigasi ancestral dari layar anak kembali ke layar induk. Untuk informasi selengkapnya tentang navigasi Naik, lihat [Menyediakan Navigasi Naik](#).

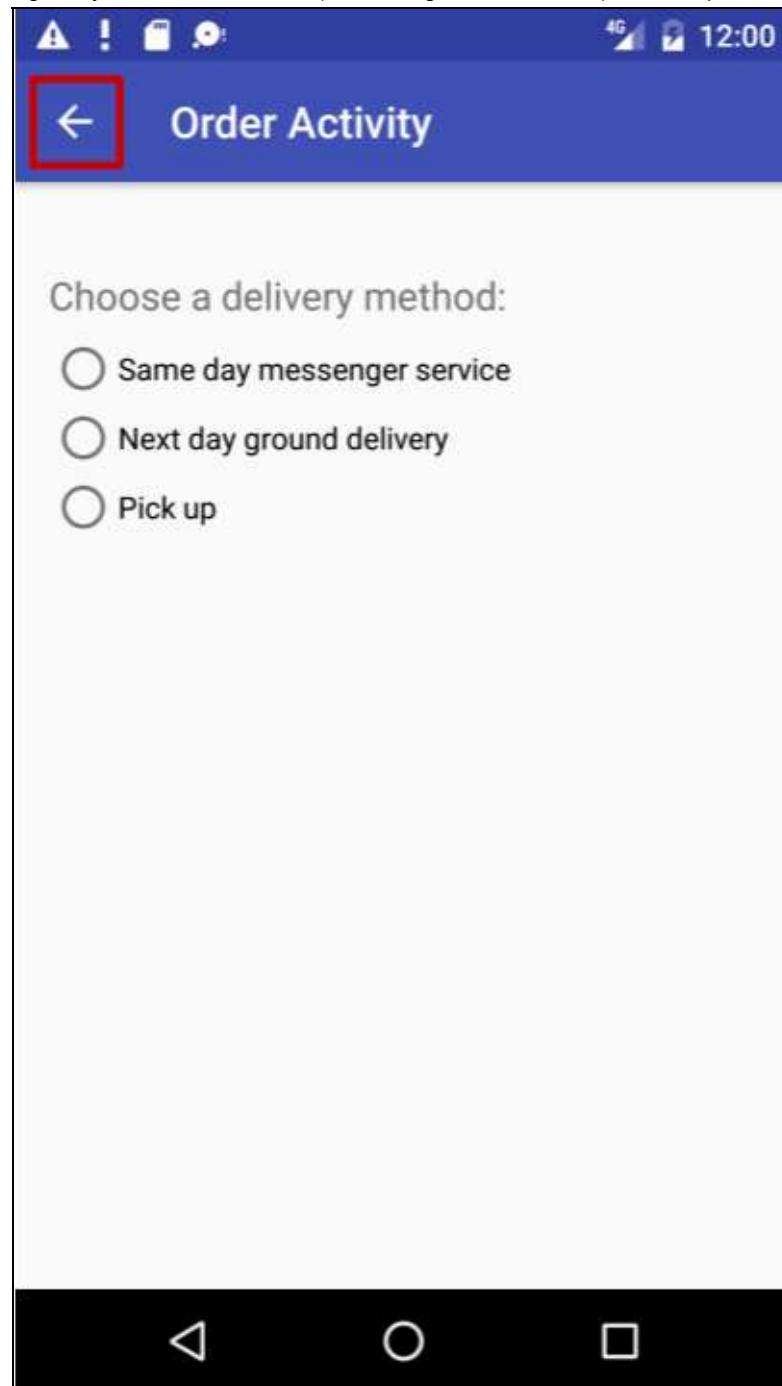
Seperti yang telah Anda pelajari sebelumnya, saat menambahkan aktivitas ke aplikasi, Anda bisa menambahkan navigasi tombol **Naik** ke aktivitas anak, seperti `OrderActivity` dengan mendeklarasikan induk aktivitas menjadi `MainActivity` di file **AndroidManifest.xml**. Anda juga bisa menyetel atribut `android:label` untuk judul layar aktivitas, seperti `"Order Activity"`:

1. Jika Anda belum memiliki aplikasi DroidCafe yang terbuka dari praktik sebelumnya, unduh proyek Android Studio [DroidCafe Bagian 3](#) dan ganti nama proyek menjadi **DroidCafe**.
2. Buka proyek **DroidCafe**.
3. Buka **AndroidManifest.xml**.
4. Ubah elemen aktivitas untuk `OrderActivity` ke yang berikut ini:

```
<activity android:name=".OrderActivity"
    android:label="Order Activity"
    android:parentActivityName="com.example.android.
        droidcafe.MainActivity">
<meta-data
    android:name="android.support.PARENT_ACTIVITY"
    android:value=".MainActivity"/>
</activity>
```

5. Ekstrak nilai `android:label "Order Activity"` ke sumber daya string yang bernama `title_activity_order`.
6. Jalankan aplikasi.

Layar Order Activity sekarang menyertakan tombol **Naik** (disorot di gambar di bawah) di bilah aplikasi untuk mengarahkan



kembali ke aktivitas induk.

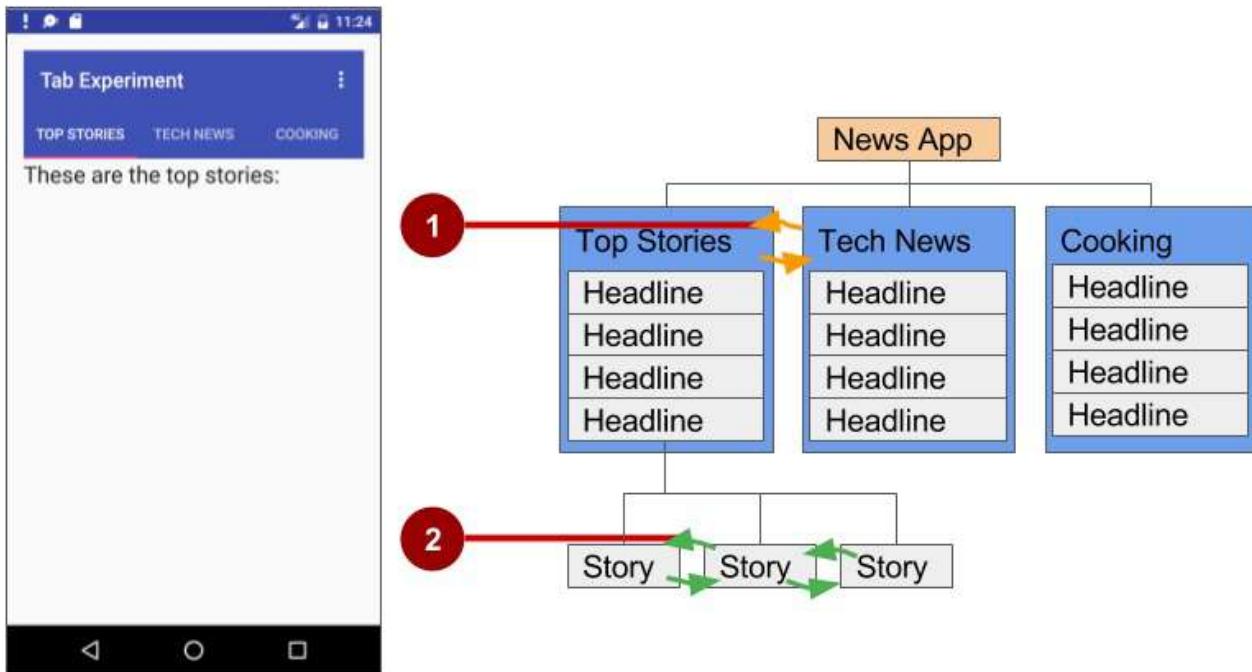
Kode solusi:

Proyek Android Studio: [DroidCafe](#)

Tugas 2. Menambahkan navigasi tab dengan tampilan gesek

Dengan navigasi lateral, Anda memungkinkan pengguna pergi dari satu saudara ke saudara yang lain (pada level yang sama dalam hierarki multilayer). Misalnya, jika aplikasi Anda menyediakan sejumlah kategori cerita (seperti Top Stories, Tech News, dan Cooking, seperti yang ditampilkan dalam gambar di bawah ini), Anda mungkin ingin menyediakan bagi pengguna Anda kemampuan untuk beralih dari satu kategori ke kategori berikutnya, tanpa harus mengarahkan kembali ke

layer induk. Contoh navigasi lateral lainnya adalah kemampuan untuk menggesek ke kiri atau kanan pada percakapan Gmail untuk menampilkan percakapan baru atau lama dalam Inbox yang sama.



Dalam gambar di atas:

1. Navigasi lateral dari satu layar kategori ke layar kategori lainnya
2. Navigasi lateral dari satu layar cerita ke layar cerita lainnya

Anda bisa mengimplementasikan navigasi lateral dengan *tab* yang mewakili masing-masing layar. Tab muncul sepanjang bagian atas layar, seperti yang ditampilkan pada sebelah kiri gambar di atas, untuk menyediakan navigasi ke layar lainnya. Navigasi tab adalah solusi yang terpopuler untuk navigasi lateral dari satu layar anak ke layar anak lainnya yang *bersaudara*—di posisi yang sama dalam hierarki dan memiliki layar induk yang sama. Navigasi tab sering dikombinasikan dengan kemampuan untuk menggesek layar anak dari kiri ke kanan atau kanan ke kiri.

Kelas utama yang digunakan untuk menampilkan tab adalah [TabLayout](#) di Pustaka Dukungan Desain Android. Kelas ini menyediakan layout horizontal untuk menampilkan tab. Anda bisa menampilkan tab di bawah bilah aplikasi, dan menggunakan kelas [PagerAdapter](#) untuk mengisi "laman" layar di dalam [ViewPager](#). ViewPager adalah pengelola layout yang memungkinkan pengguna membalik layar ke kiri dan ke kanan. Ini adalah pola umum untuk menyajikan layar konten yang berbeda di dalam sebuah aplikasi—gunakan *adaptor* untuk mengisi layar konten untuk ditampilkan di aktivitas dan sebuah *pengelola layout* yang mengubah layar konten bergantung pada tab mana yang dipilih.

Sediakan implementasi [PagerAdapter](#) untuk menghasilkan layar yang akan ditunjukkan oleh tampilan. ViewPager paling sering digunakan bersama dengan [Fragment](#). Dengan menggunakan fragmen, Anda mudah mengelola daur hidup setiap "laman" layar.

Untuk menggunakan Pustaka Dukungan Android, tambahkan `com.android.support:design:xx.xx.x` (`xx.xx.xx` merupakan versi terbaru) ke file **build.gradle (Module: app)**.

Berikut ini adaptor standar untuk menggunakan fragmen bersama ViewPager:

- [FragmentPagerAdapter](#): Didesain untuk navigasi antar layar (laman) saudara yang menyatakan jumlah layar yang tetap dan sedikit.
- [FragmentStatePagerAdapter](#)(<https://developer.android.com/reference/android/support/v4/app/FragmentStatePagerAdapter.html>): Didesain untuk membuka layar ke semua kumpulan layar (laman) dengan jumlah layar tidak ditentukan. Hal ini akan memusnahkan fragmen saat pengguna beralih ke layar lain, sehingga meminimalkan penggunaan memori. Aplikasi untuk tantangan praktis ini menggunakan FragmentStatePagerAdapter.

2.1 Buat layout untuk navigasi tab

1. Buat proyek baru menggunakan template Empty Activity. Beri nama aplikasi **Tab Experiment**.
2. Edit file `build.gradle` (Module: app) dan tambahkan baris berikut (jika belum ditambahkan) ke bagian `dependencies` :

```
compile 'com.android.support:design:25.0.1'  
compile 'com.android.support:support-v4:25.0.1'
```

Jika Android Studio menyarankan versi yang lebih tinggi, edit baris di atas untuk memperbarui versi. Selain itu, jika Android Studio menyarankan versi baru `compileSdkVersion`, `buildToolsVersion`, dan/atau `targetSdkVersion`, edit semuanya untuk memperbarui versi.

3. Untuk menggunakan bilah alat dan bukan bilah tindakan dan judul aplikasi, tambah pernyataan berikut ke file `res > values > styles.xml` untuk menyembunyikan bilah tindakan dan judul:

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
...  
<item name="windowActionBar">false</item>  
<item name="windowNoTitle">true</item>  
</style>
```

4. Di layout **main_activity.xml**, buang `TextView` yang disediakan oleh template, dan tambahkan `Toolbar`, `TabLayout`, dan `ViewPager` di dalam layout root. Layout seharusnya tampak seperti kode berikut.

Selagi Anda mengetikkan atribut `app:popupTheme` untuk `Toolbar`, seperti yang ditampilkan di bawah ini, `app` akan berwarna merah jika Anda tidak menambahkan pernyataan berikut ini ke `RelativeLayout` :

```
<RelativeLayout xmlns:app="http://schemas.android.com/apk/res-auto"
```

Anda bisa mengeklik `app` dan menekan **Option-Return** dan Android Studio otomatis menambahkan pernyataan.

Kode solusi:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.tabexperiment.MainActivity">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:background="?attr/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>

    <android.support.design.widget.TabLayout
        android:id="@+id/tab_layout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/toolbar"
        android:background="?attr/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"/>

    <android.support.v4.view.ViewPager
        android:id="@+id/pager"
        android:layout_width="match_parent"
        android:layout_height="fill_parent"
        android:layout_below="@id/tab_layout"/>

</RelativeLayout>

```

2.2 Buat layout dan kelas untuk setiap fragmen

1. Tambahkan fragmen yang mewakili setiap layar bertab: **TabFragment1**, **TabFragment2**, dan **TabFragment3**. Untuk menambahkan setiap fragmen:
 - i. Klik **com.example.android.tabexperiment** di tampilan proyek.
 - ii. Pilih **File > New > Fragment > Fragment (Blank)**.
 - iii. Beri nama fragmen **TabFragment1**.
 - iv. Periksa opsi "Create layout XML?" dan ubah Nama Layout Fragmen untuk file XML ke **tab_fragment1**.
 - v. *Hapus centang* opsi "Include fragment factory methods?" dan "include interface callbacks?". Anda tidak memerlukan metode ini.
 - vi. Klik **Finish**.
 - vii. Ulangi langkah-langkah di atas, menggunakan **TabFragment2** dan **TabFragment3** untuk Langkah C dan **tab_fragment2** dan **tab_fragment3** untuk Langkah D.

Setiap fragmen (**TabFragment1**, **TabFragment2**, dan **TabFragment3**) dibuat dengan definisi kelasnya disetel untuk meluaskan `Fragment`. Selain itu, setiap fragmen *memekarkan* layout yang dikaitkan dengan layar (`tab_fragment1`, `tab_fragment2`, dan `tab_fragment3`), menggunakan pola desain pemekaran sumber daya yang telah Anda pelajari di bab sebelumnya dengan menu opsi.

Misalnya **TabFragment1** tampak seperti ini:

```

public class TabFragment1 extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater,
        ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.tab_fragment1, container, false);
    }
}

```

Android Studio otomatis menyertakan pernyataan impor berikut:

```

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

```

2. Edit setiap file XML layout fragmen (**tab_fragment1**, **tab_fragment2**, dan **tab_fragment3**):
 - i. Ubah Root Tag ke `RelativeLayout`.
 - ii. Tambahkan `TextView` dengan teks, seperti "These are the top stories".
 - iii. Atur penampilan teks dengan `android:textAppearance="?android:attr/textAppearanceLarge"`.
 - iv. Ulangi langkah-langkah dia atas untuk setiap file XML layout, dengan memasukkan teks yang berbeda untuk `TextView` di langkah B.
3. Periksa setiap file XML layout fragmen. Misalnya, **tab_fragment1** seharusnya tampak seperti ini:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="These are the top stories:"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

</RelativeLayout>

```

4. Di file XML layout fragmen **tab_fragment1**, ekstrak string untuk "These are the top stories:" ke dalam sumber daya `string tab_1`. Lakukan yang sama untuk string di **tab_fragment2** dan **tab_fragment3**.

2.3 Tambahkan PagerAdapter

Pola pengelola layout adaptor memungkinkan Anda memberikan layar konten yang berbeda di dalam akan dipilih—gunakan *adaptor* untuk mengisi layar konten untuk menampilkan akan dipilih dan *pengelola layout* yang mengubah layar konten bergantung pada tab mana yang di `TextView`.

1. Tambahkan kelas `PagerAdapter` baru pada aplikasi yang meluaskan `FragmentStatePagerAdapter` dan mendefinisikan jumlah tab (`mNumTabs`):

```

public class PagerAdapter extends FragmentStatePagerAdapter {
    int mNumTabs;
    public PagerAdapter(FragmentManager fm, int NumOfTabs) {
        super(fm);
        this.mNumOfTabs = NumOfTabs;
    }
}

```

Saat memasukkan kode di atas, Android Studio otomatis mengimpor:

```
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentStatePagerAdapter;
```

Jika FragmentManager dalam kode di atas berwarna merah, sebuah ikon bola lampu merah akan muncul saat Anda mengekliknya. Klik ikon bola lampu merah dan pilih **Import class**. Pilihan impor akan muncul. Pilih opsi impor berikut:

FragmentManager (android.support.v4)

Memilih yang di atas akan mengimpor:

```
import android.support.v4.app.FragmentManager;
```

Selain itu, Android Studio menggarisbawahi definisi kelas untuk PagerAdapter, jika Anda mengeklik **PagerAdapter**, ikon bola lampu merah akan muncul. Klik ikon dan pilih **Implement Methods** lalu klik **OK** untuk mengimplementasikan metode `getItem()` dan `getCount()` yang telah dipilih.

2. Ubah metode `getItem()` yang baru ditambahkan ke yang berikut ini, yang menggunakan blok `switch case` untuk mengembalikan fragmen yang akan ditampilkan berdasarkan tab mana yang diklik

```
@Override
public Fragment getItem(int position) {
    switch (position) {
        case 0:
            return new TabFragment1();
        case 1:
            return new TabFragment2();
        case 2:
            return new TabFragment3();
        default:
            return null;
    }
}
```

3. Ubah metode `getCount()` yang baru ditambahkan menjadi yang berikut ini ke mengembalikan jumlah tab:

```
@Override
public int getCount() {
    return mNumOfTabs;
}
```

2.4 Mekarkan Toolbar dan TabLayout

Karena Anda menggunakan tab yang pas di bawah bilah aplikasi, Anda telah menyiapkan bilah aplikasi dan `Toolbar` di layout `activity_main.xml` di langkah pertama tugas ini. Sekarang Anda perlu memekarkan `Toolbar` (menggunakan metode yang sama yang dijelaskan di bab sebelumnya tentang menu opsi) dan membuat instance `TabLayout` untuk mengatur posisi tab.

1. Mekarkan `Toolbar` di metode `onCreate()` di `MainActivity.java`:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    // Create an instance of the tab layout from the view.
    ...
}
```

Dalam kode di atas, `Toolbar` berwarna merah dan ikon bola lampu merah akan muncul. Klik ikon bola lampu merah dan pilih **Import class**. Pilihan impor akan muncul. Pilih **Toolbar (android.support.v7.widget.Toolbar)** dan pernyataan import berikut muncul di kode Anda:

```
import android.support.v7.widget.Toolbar;
```

2. Buka `strings.xml` dan buat sumber daya string berikut:

```
<string name="tab_label1">Top Stories</string>
<string name="tab_label2">Tech News</string>
<string name="tab_label3">Cooking</string>
```

3. Di akhir metode `onCreate()`, buat instance layout tab dari elemen `tab_layout` di layout dan setel teks untuk setiap tab menggunakan `addTab()`:

```
...
// Create an instance of the tab layout from the view.
TabLayout tabLayout = (TabLayout) findViewById(R.id.tab_layout);
// Set the text for each tab.
tabLayout.addTab(tabLayout.newTab().setText(R.string.tab_label1));
tabLayout.addTab(tabLayout.newTab().setText(R.string.tab_label2));
tabLayout.addTab(tabLayout.newTab().setText(R.string.tab_label3));
// Set the tabs to fill the entire layout.
tabLayout.setTabGravity(TabLayout.GRAVITY_FILL);

// Use PagerAdapter to manage page views in fragments.
...
```

2.5 Gunakan PagerAdapter untuk mengelola tampilan layar

1. Berikut ini kode yang Anda tambahkan ke metode `onCreate()` di tugas sebelumnya, tambahkan kode berikut ini untuk menggunakan `PagerAdapter` untuk mengelola tampilan layar (laman) di fragmen:

```
...
// Using PagerAdapter to manage page views in fragments.
// Each page is represented by its own fragment.
// This is another example of the adapter pattern.
final ViewPager viewPager = (ViewPager) findViewById(R.id.pager);
final PagerAdapter adapter = new PagerAdapter
        (getSupportFragmentManager(), tabLayout.getTabCount());
viewPager.setAdapter(adapter);
// Setting a listener for clicks.
...
```

2. Di akhir metode `onCreate()`, setel sebuah listener (`TabLayoutOnPageChangeListener`) untuk mendeteksi jika sebuah tab diklik dan buat metode `onTabSelected()` untuk menyetel `viewPager` ke layar bertab yang sesuai. Kode akan terlihat seperti berikut:

```

...
// Setting a listener for clicks.
viewPager.addOnPageChangeListener(new
    TabLayout.TabLayoutOnPageChangeListener(tabLayout));
tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
    public void onTabSelected(TabLayout.Tab tab) {
        viewPager.setCurrentItem(tab.getPosition());
    }

    @Override
    public void onTabUnselected(TabLayout.Tab tab) {

    }

    @Override
    public void onTabReselected(TabLayout.Tab tab) {
    }
});
}

```

- Jalankan aplikasi. Ketuk setiap tab untuk melihat setiap "laman" (layar). Anda juga bisa menggesek ke kiri dan ke kanan untuk mengunjungi "laman" yang berbeda.

Kode solusi

Proyek Android Studio: [Eksperimen Tab](#) (termasuk tantangan penyusunan kode 1)

Proyek Android Studio: [Eksperimen NavDrawer](#) (tantangan penyusunan kode 2)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan 1: Saat Anda membuat layout untuk navigasi tab di langkah pertama pelajaran sebelumnya, Anda telah membuat `Toolbar` untuk bilah aplikasi di file layout `activity_main.xml`. Tambahkan menu opsi di bilah aplikasi sebagai tantangan.

Untuk memulai, buat file `menu_main.xml` dan tambahkan item menu untuk menu opsi. Anda harus menambahkan minimal satu item menu, seperti **Settings**.

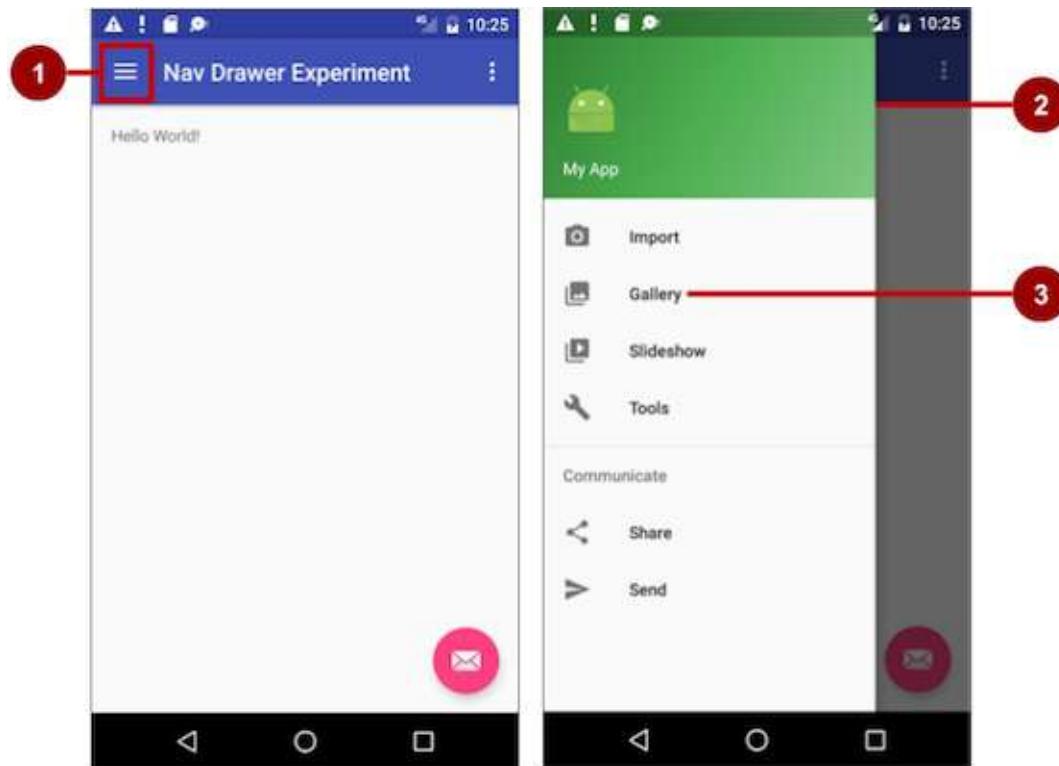
Anda bisa memekarkan menu opsi di `Toolbar` dengan menambahkan metode `onCreateOptionsMenu()` seperti yang Anda lakukan di pelajaran sebelumnya tentang menggunakan menu opsi.

Akhirnya, Anda bisa mendeteksi item menu opsi mana yang dicentang menggunakan metode `onOptionsItemSelected()`.

Tantangan 2: Buat aplikasi baru dengan panel samping navigasi. Saat pengguna mengetuk pilihan panel samping navigasi, tutup panel samping navigasi dan tampilkan pesan toast yang menampilkan opsi mana yang dipilih.

Panel samping navigasi adalah panel yang biasanya menampilkan opsi navigasi pada tepi kiri layar, seperti yang ditampilkan pada sisi kanan gambar di bawah. Panel ini paling sering tersembunyi, namun ditampilkan bila pengguna menggesekkan jari dari tepi kiri layar atau menyentuh ikon navigasi dalam bilah aplikasi, seperti yang ditampilkan pada

samping kiri gambar di bawah ini.



Dalam gambar di atas:

1. Ikon navigasi dalam bilah aplikasi
2. Panel samping navigasi
3. Item menu panel samping navigasi

Untuk membuat panel samping navigasi dalam aplikasi, Anda perlu melakukan yang berikut ini:

1. Buat layout berikut:
 - o Panel samping navigasi sebagai tampilan root layout aktivitas.
 - o Tampilan navigasi untuk panel samping itu sendiri.
 - o Layout bilah aplikasi yang akan menyertakan tombol ikon navigasi.
 - o Layout materi untuk aktivitas yang menampilkan panel samping navigasi.
 - o Layout untuk header panel samping navigasi.
2. Isilah menu panel samping navigasi dengan judul item dan ikon.
3. Siapkan panel samping navigasi dan listener item dalam kode aktivitas.
4. Tangani pemilihan item menu navigasi.

Untuk membuat layout panel samping navigasi, gunakan

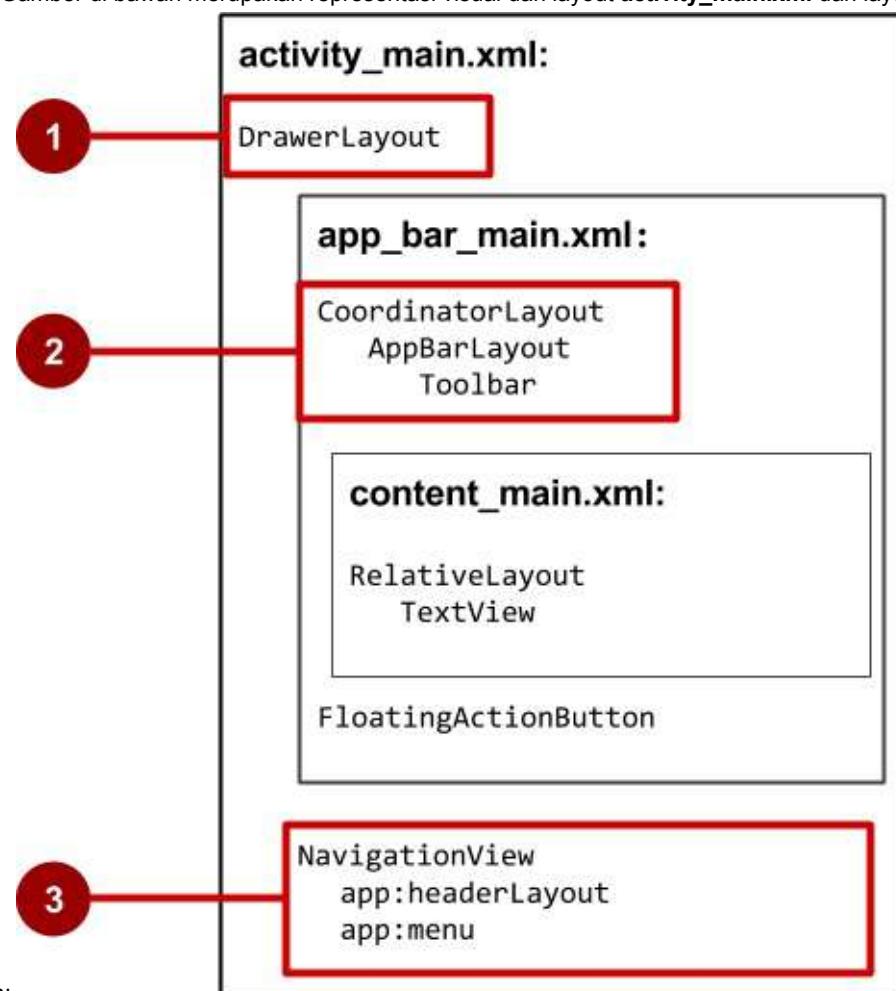
`DrawerLayout`(<https://developer.android.com/reference/android/support/v4/widget/DrawerLayout.html>) API yang tersedia dalam Pustaka Dukungan(<https://developer.android.com/tools/support-library/index.html>).

Untuk spesifikasi desain, ikuti prinsip desain panel samping navigasi dalam panduan desain Panel Samping

Navigasi(<https://developer.android.com/design/patterns/navigation-drawer.html>).

Untuk menambahkan panel samping navigasi, gunakan `DrawerLayout` sebagai tampilan root layout aktivitas Anda. Di dalam `DrawerLayout`, tambahkan satu tampilan yang berisi konten utama untuk layar (layout utama Anda saat panel samping disembunyikan) dan tampilan lain, umumnya `NavigationView`, yang berisi konten panel samping navigasi **Tip:**

Untuk membuat layout Anda lebih mudah dipahami, gunakan tag `include` untuk menyertakan layout XML dalam layout XML lainnya. Gambar di bawah merupakan representasi visual dari layout **activity_main.xml** dan layout XML yang



disertakannya:

Dalam gambar di atas:

1. `DrawerLayout` adalah tampilan root layout aktivitas.
 2. `app_bar_main` yang disertakan menggunakan `CoordinatorLayout` sebagai root-nya, dan mendefinisikan layout bilah aplikasi dengan `Bilah Alat` yang akan menyertakan ikon navigasi untuk membuka panel samping.
 3. `NavigationView` mendefinisikan layout panel samping navigasi dan header-nya, serta menambahkan item menu ke panel.

Rangkuman

- Tambahkan navigasi tombol Naik ke aktivitas anak dengan mendeklarasikan induk aktivitas di file AndroidManifest.xml.
 - Siapkan navigasi tab:
 - Tab adalah solusi yang baik untuk "navigasi lateral" antara tampilan seinduk.
 - Kelas utama yang digunakan untuk tab adalah [TabLayout](#) di pustaka dukungan desain.
 - Anda harus menggunakan pola adaptor saat mengisikan tab (laman) dengan data.
 - ViewPager adalah pengelola layout yang memungkinkan pengguna membalik ke kiri dan ke kanan melalui laman data.
 - ViewPager paling sering digunakan bersama dengan fragmen.
 - Ada dua adaptor standar untuk menggunakan ViewPager: [FragmentPagerAdapter](#) dan [FragmentStatePagerAdapter](#).

[Konsep terkait]

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Navigasi Layar](#)

Ketahui selengkapnya

- Referensi Developer Android:
 - [Menyediakan Navigasi Naik](#)
 - [TabLayout](#)
 - [Membuat Tampilan Gesek dengan Tab](#)
 - [Panel Samping Navigasi](#)
 - [DrawerLayout](#)
 - [Pustaka Dukungan](#)
- Android Developers Blog: [Pustaka Dukungan Desain Android](#)
- Lainnya
 - [AndroidHive: Android Material Design working with Tabs](#)
 - [Truiton: Android Tabs Example – With Fragments and ViewPager](#)

4.4 Membuat RecyclerView

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Membuat dan mengonfigurasi proyek WordList](#)
- [Tugas 2: Membuat kumpulan data](#)
- [Tugas 3: Membuat RecyclerView](#)
- [Tugas 4: Menambahkan onClick ke daftar item](#)
- [Tugas 5: Menambahkan FAB untuk menyisipkan item](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Menampilkan dan mengubah daftar item data yang mirip yang bisa digulir, seperti yang telah Anda lakukan di praktik tampilan pengguliran, adalah fitur umum aplikasi. Misalnya, kontak, daftar putar, foto, kamus, daftar belanja, indeks dokumen, atau cantuman game yang disimpan, semua adalah contoh daftar yang bisa digulir.

Di awal kelas, Anda menggunakan ScrollView untuk melakukan pengguliran View lainnya. ScrollView mudah digunakan, tetapi tidak disarankan untuk penggunaan produksi, khususnya daftar item yang bisa digulir yang panjang.

RecyclerView adalah subkelas ViewGroup dan lebih efisien dalam hal sumber daya untuk menampilkan daftar yang bisa digulir. Sebagai ganti membuat tampilan untuk setiap item, apakah bisa dilihat atau tidak, RecyclerView membuat daftar item dengan jumlah terbatas dan menggunakan item daftar kembali untuk konten yang bisa dilihat.

Dalam seri praktik ini, Anda akan menggunakan RecyclerView untuk:

- Menampilkan daftar item yang bisa digulir.
- Menambahkan handler klik ke setiap item.
- Menambahkan item ke daftar menggunakan tombol aksi mengambang (FAB), tombol merah muda di tangkapan layar berikut. Tombol aksi mengambang bisa digunakan untuk tindakan umum atau tindakan yang dipromosikan, yaitu, tindakan yang Anda ingin agar dilakukan pengguna.

Yang harus sudah Anda KETAHUI

Untuk praktik ini, Anda harus sudah mengenal cara:

- Membuat aplikasi Hello World dengan Android Studio.
- Mengimplementasikan layout yang berbeda untuk aplikasi.
- Membuat dan menggunakan sumber daya string.
- Menambahkan handler onClick ke tampilan.

Yang akan Anda PELAJARI

Dalam praktik ini, Anda akan mempelajari cara:

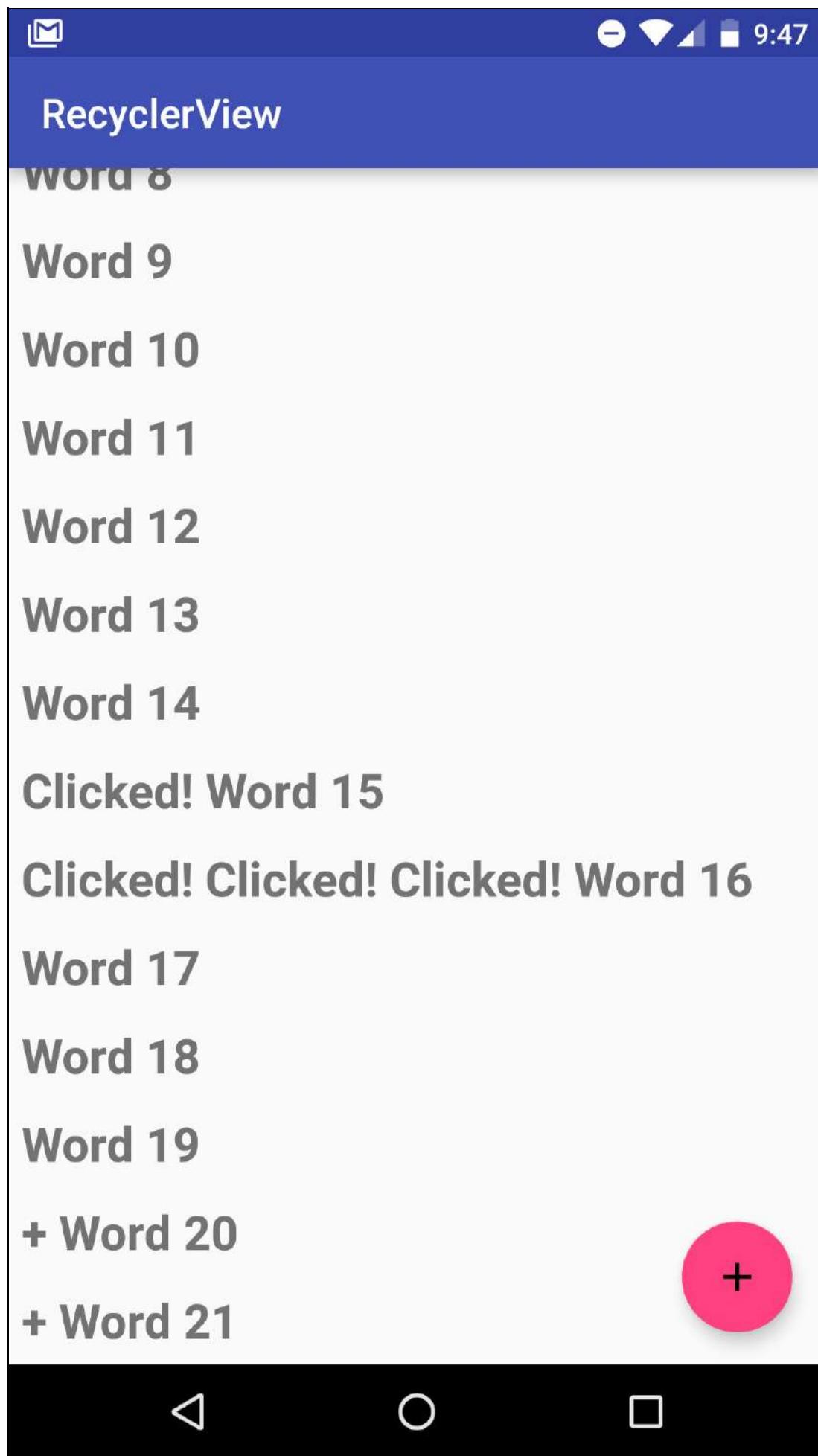
- Menggunakan kelas RecyclerView untuk menampilkan item di daftar yang bisa digulir.
- Secara dinamis menambahkan item ke RecyclerView saat terlihat melalui pengguliran.
- Melakukan tindakan saat pengguna mengetuk item tertentu
- Menampilkan tombol aksi mengambang dan melakukan tindakan saat pengguna mengetuknya.

Yang akan Anda LAKUKAN

Membuat aplikasi yang menggunakan [RecyclerView](#) untuk menampilkan daftar item sebagai daftar yang bisa digulir dan mengaitkan perilaku klik dengan item daftar ini. Menggunakan tombol aksi mengambang untuk memungkinkan pengguna menambahkan item ke RecyclerView.

Ringkasan Aplikasi

Aplikasi "RecyclerView" akan menampilkan daftar kata yang panjang.



- Mengetuk item akan menandainya sebagai diklik.
- Mengetuk tombol aksi mengambang akan menambahkan item.
- Tidak ada masukan pengguna berupa kata untuk aplikasi ini.

Tugas 1. Membuat dan mengonfigurasi proyek baru

Dalam tugas ini, Anda akan membuat dan mengonfigurasi proyek baru untuk aplikasi contoh RecyclerView.

1.1. proyek

1. Mulai Android Studio dan buat proyek baru dengan parameter berikut:

| Atribut | Nilai |
|------------------------------|----------------------------------------------|
| Application Name | RecyclerView |
| Company Name | com.example.android atau domain Anda sendiri |
| Phone and Tablet Minimum SDK | API15: Android 4.0.3 IceCreamSandwich |
| Template | Empty Activity |
| Generate Layout file box | Checked |

2. Jalankan aplikasi Anda di emulator atau perangkat keras. Anda akan melihat judul "RecyclerView" dan "Hello World" dalam tampilan kosong.

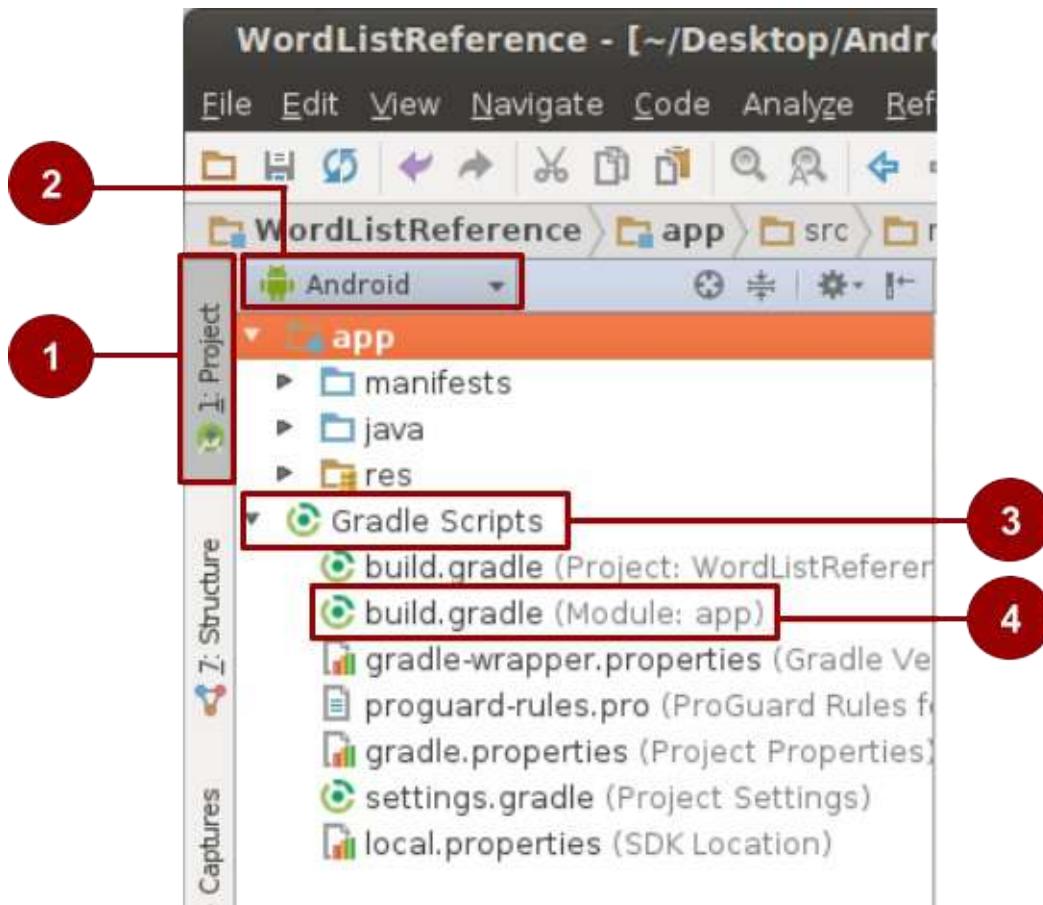
1.2 Tambahkan pustaka dukungan ke file build

Untuk menggunakan RecyclerView dan tombol aksi mengambang (FAB), Anda perlu menambahkan [Pustaka Dukungan Android](#) yang terkait ke build Anda.

Mengapa: Pustaka Dukungan Android menyediakan versi Android Framework API yang kompatibel dengan versi lama, komponen UI tambahan, dan serangkaian utilitas yang berguna. Kelas RecyclerView terletak di paket Android Support; dua dependensi harus disertakan di proses build Gradle agar dapat menggunakannya.

Ikuti langkah-langkah berikut dan lihat tangkapan layar:

1. Di Android Studio, di proyek Anda, pastikan Anda berada dalam panel **Project** (1) dan di tampilan **Android** (2).
2. Di hierarki file, temukan folder **Gradle Scripts** (3).
3. Luaskan **Gradle Scripts**, jika perlu, dan buka file **build.gradle (Module: app)** (4).



4. Di akhir file **build.gradle (Module: app)**, temukan bagian dependensi.
5. Tambahkan kedua dependensi ini sebagai dua baris di dalam bagian dependensi:

```
compile 'com.android.support:recyclerview-v7:23.1.1'
compile 'com.android.support:design:23.1.1'
```

- Mungkin terdapat baris yang sudah ada seperti ini, tetapi jumlahnya mungkin berbeda: `compile 'com.android.support:appcompat-v7:23.1.1'`
 - Tambahkan baris Anda bawah baris tersebut.
 - **Pastikan nomor versi baris Anda sesuai dengan nomor versi baris tersebut.**
 - Pastikan nomor versi semua pustaka sama dan cocok dengan `compiledSdkVersion` di bagian atas file. Jika tidak cocok, Anda akan mendapatkan kesalahan build time.
6. Jika diminta, sinkronkan aplikasi Anda sekarang.
 7. Jalankan aplikasi Anda. Anda seharusnya melihat aplikasi "RecyclerView" yang sama, menampilkan "Hello World". Jika Anda mendapatkan kesalahan gradle, sinkronkan proyek Anda. Anda tidak perlu memasang plugin tambahan.

Solusi:

Ini adalah contoh bagian dependensi file build.gradle. File Anda mungkin sedikit berbeda dan entri Anda mungkin memiliki nomor versi yang berbeda.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:recyclerview-v7:23.1.1'
    compile 'com.android.support:design:23.1.1'
}
```

Tugas 2. Membuat kumpulan data

Sebelum Anda bisa menampilkan apa pun, Anda memerlukan data untuk ditampilkan. Di aplikasi yang lebih rumit, data Anda bisa berasal dari penyimpanan internal (file, database SQLite, preferensi yang disimpan), dari aplikasi lain (Kontak, Foto), atau dari internet (storage awan, Google Spreadsheet, atau sumber data apa pun dengan API). Untuk latihan ini, Anda akan menyimulasikan data dengan membuatnya di metode `onCreate()` aktivitas utama.

Mengapa: Menyimpan dan mengambil data adalah topik terpisah yang dibahas dalam bab storage data. Anda berkesempatan untuk meningkatkan aplikasi Anda untuk menggunakan data nyata di pelajaran tersebut.

2.1 Tambahkan kode untuk membuat data

Dalam tugas ini, Anda akan secara dinamis membuat daftar tertaut yang berisi dua puluh string kata yang berakhiran dengan angka yang bertambah, misalnya ["Word 1", "Word 2", "Word 3", ...].

Anda harus menggunakan `LinkedList` untuk praktik ini. Lihat kode solusi, jika Anda memerlukan bantuan.

1. Buka file `MainActivity.java`.
2. Tambahkan variabel anggota privat untuk daftar tertaut `mWordList`.
3. Tambahkan variabel `mCount` penghitung integer untuk melacak angka kata.
4. Tambahkan kode yang mengisikan `mWordList` dengan kata. Gabungkan string "Word" dengan nilai `mCount`, lalu tingkatkan hitungan angka.
5. Karena Anda belum bisa menampilkan kata untuk pengujian, tambahkan pernyataan log yang memverifikasi bahwa kata ditambahkan ke daftar tertaut dengan benar.
6. Jalankan aplikasi Anda untuk memastikan tidak ada kesalahan.

UI aplikasi tidak berubah, tetapi Anda seharusnya melihat daftar pesan log di logcat, seperti: `android.example.com.wordlist D/WordList: Word 1`.

Solusi:

Variabel kelas:

```
private final LinkedList<String> mWordList = new LinkedList<>();
private int mCount = 0;
```

Di metode `onCreate` `MainActivity`:

```
for (int i = 0; i < 20; i++) {
    mWordList.addLast("Word " + mCount++);
    Log.d("WordList", mWordList.getLast());
}
```

Tugas 3: Membuat RecyclerView

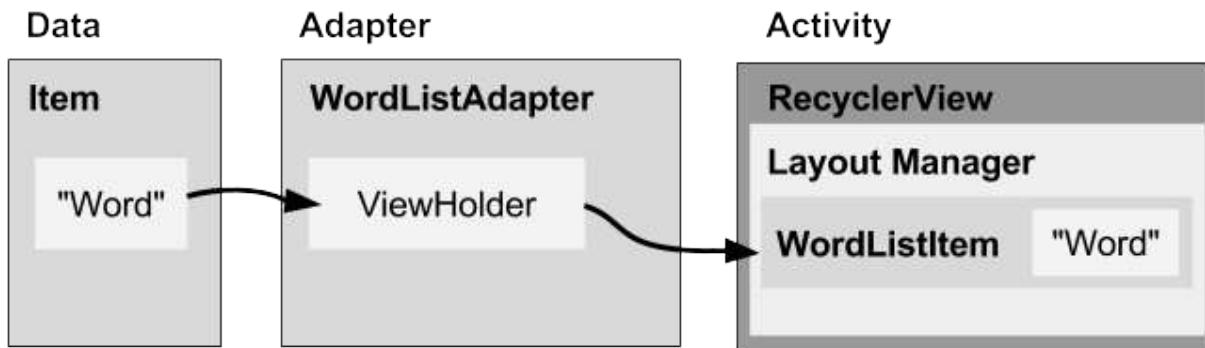
Dalam praktik ini Anda akan menampilkan data di `RecyclerView`. Karena ada beberapa bagian pada proses membuat `RecyclerView` yang berfungsi, pastikan Anda langsung memperbaiki kesalahan apa pun yang Anda lihat di Android Studio.

Untuk menampilkan data dalam `RecyclerView`, Anda memerlukan bagian berikut:

- **Data.** Anda akan menggunakan `mWordList`.
- **RecyclerView.** Daftar gulir yang berisi item daftar.
- **Layout untuk satu item data.** Semua item daftar tampak sama.
- **Pengelola layout.** Pengelola layout menangani pengelolaan (layout) komponen antarmuka pengguna di sebuah tampilan. Anda telah menggunakan `LinearLayout` di praktik sebelumnya tempat sistem Android menangani layout untuk Anda. `RecyclerView` memerlukan pengelola layout eksplisit untuk mengelola susunan item daftar yang terdapat di dalamnya. Layout ini bisa vertikal, horizontal, atau berupa petak. Anda akan menggunakan pengelola layout linear vertikal yang disediakan oleh Android.
- **Adaptor.** Adaptor menghubungkan data Anda dengan `RecyclerView`. Ini menyiapkan data dalam holder tampilan.

- Anda akan membuat adaptor yang disisipkan ke dan memperbarui kata yang dihasilkan di tampilan Anda.
- **Holder tampilan.** Di dalam adaptor, Anda akan membuat kelas ViewHolder yang berisi informasi tampilan untuk menampilkan satu item dari layout item.

Diagram berikut menampilkan hubungan antara data, adaptor, holder view, dan pengelola layout.



Ringkasan langkah-langkah implementasi

Untuk mengimplementasikan bagian-bagian ini, Anda perlu:

1. Membuat layout XML untuk aplikasi "RecyclerView" (activity_main.xml).
 2. Membuat layout XML yang digunakan untuk menata letak satu item daftar, yaitu WordListItem (worlist_item.xml).
 3. Membuat adaptor (WordListAdapter) dengan holder tampilan (WordViewHolder). Implementasikan metode yang mengambil data, menempatkannya dalam holder tampilan, dan memberi tahu pengelola layout cara menampilkannya.
 4. Dalam metode onCreate method, buat RecyclerView dan inisialisasi dengan adapter serta pengelola layout standar.
- Mari kita lakukan satu per satu.

3.1 Buat layout utama di activity_main.xml

Di aplikasi sebelumnya, Anda menggunakan LinearLayout untuk menata tampilan Anda. Untuk mengakomodasi RecyclerView dan tombol aksi mengambang yang nanti Anda tambahkan, Anda perlu menggunakan grup tampilan yang berbeda yang disebut sebagai layout koordinator. [CoordinatorLayout](#) lebih fleksibel dari LinearLayout saat menata tampilan. Misalnya, tampilan seperti tombol aksi mengambang bisa terhampar di atas tampilan lain.

Di main_activity.xml, ganti kode yang dibuat oleh Empty Activity dengan kode untuk CoordinatorLayout, lalu tambahkan RecyclerView:

1. Buka build.gradle (Module: app) dan verifikasi apakah dependensi tampilan recycler tersedia.

```
compile 'com.android.support:recyclerview-v7:24.1.1'
```

2. Buka activity_main.xml.
3. Pilih semua kode di activity_main.xml dan ganti dengan kode ini:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
</android.support.design.widget.CoordinatorLayout>
```

4. Periksa kode dan catat hal-hal berikut ini:
 - Properti yang ditetapkan untuk grup tampilan ini sama seperti untuk LinearLayout karena beberapa properti dasar, seperti layout_width dan layout_height diperlukan untuk semua tampilan dan grup tampilan.
 - Karena CoordinatorLayout berada di pustaka dukungan, Anda harus menetapkan jalur lengkap ke pustaka dukungan. Anda harus melakukan tindakan yang sama untuk RecyclerView.

5. Tambahkan kode RecyclerView di dalam CoordinatorLayout:

- o Anda perlu menetapkan jalur lengkap, karena RecyclerView adalah bagian dari pustaka dukungan.

```
<android.support.v7.widget.RecyclerView>
</android.support.v7.widget.RecyclerView>
```

6. Berikan properti berikut pada RecyclerView Anda:

| Atribut | Nilai |
|-----------------------|---------------------|
| android:id | "@+id/recyclerview" |
| android:layout_width | match_parent |
| android:layout_height | match_parent |

7. Jalankan aplikasi Anda dan pastikan tidak ada kesalahan yang ditampilkan di logcat. Anda hanya akan melihat layar kosong karena Anda belum menempatkan item apa pun ke dalam RecyclerView.

Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerview"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </android.support.v7.widget.RecyclerView>

</android.support.design.widget.CoordinatorLayout>
```

3.2 Buat layout untuk satu item daftar

Adaptor memerlukan layout untuk satu item di daftar. Semua item menggunakan layout yang sama. Anda perlu menetapkan layout item daftar tersebut dalam file sumber daya layout terpisah, karena ini digunakan oleh adaptor, yang terpisah dari RecyclerView.

Buat layout item kata sederhana menggunakan LinearLayout vertikal dengan TextView:

1. Klik kanan folder **app/res/layout** dan pilih **New > Layout resource file**.
2. Beri nama file `wordlist_item` dan klik **OK**.
3. Dalam mode Text, ubah LinearLayout yang dibuat dengan file agar sesuai dengan atribut berikut. Ekstrak sumber daya selagi Anda bekerja.

| Atribut | Nilai |
|-----------------------|----------------|
| android:layout_width | "match_parent" |
| android:layout_height | "wrap_content" |
| android:orientation | "vertical" |
| android:padding | "6dp" |

4. Tambahkan TextView untuk kata ke LinearLayout:

| Atribut | Nilai |
|-----------------------|----------------|
| android:id | "@+id/word" |
| android:layout_width | "match_parent" |
| android:layout_height | "wrap_content" |
| android:textSize | "24sp" |
| android:textStyle | "bold" |

3.3 Buat gaya dari atribut TextView

Anda bisa menggunakan gaya untuk mengizinkan elemen berbagi grup atribut tampilan. Satu cara mudah untuk membuat gaya adalah dengan mengekstrak gaya elemen UI yang sudah Anda buat. Ekstrak informasi gaya untuk tampilan teks kata:

1. Saat wordlist_item.xml terbuka, arahkan mouse ke atas bagian TextView yang baru Anda buat dan **Klik kanan > Refactor > Extract > Style**.
2. Dalam dialog **Extract Android Style** dialog,
 - o Beri nama gaya Anda dengan **word_title**.
 - o Biarkan semua kotak tidak tercentang.
 - o Centang kotak **Launch 'Use Style Where Possible'**.
 - o Klik **OK**.
3. Saat diminta, terapkan gaya pada **Whole Project**.
4. Temukan dan periksa gaya **word_title** di **values/styles.xml**.
5. Kembali ke wordlist_item.xml. Tampilan teks sekarang mengacu ke gaya, sebagai ganti menggunakan properti penataan gaya individual.
6. Jalankan aplikasi Anda. Karena Anda telah membuang tampilan teks "Hello World" default, Anda akan melihat judul "RecyclerView" dan tampilan kosong.

Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="6dp">

    <TextView
        android:id="@+id/word"
        style="@style/word_title" />

</LinearLayout>
```

3.4. Buat adapter dengan holder tampilan

Android menggunakan adaptor (dari [kelas Adapter](#)) untuk menghubungkan data dengan tampilannya. Ada berbagai tipe adaptor yang tersedia. Anda juga bisa menulis adaptor khusus Anda sendiri. Dalam tugas ini, Anda akan membuat adaptor yang mengaitkan daftar kata Anda dengan tampilan item daftar kata.

Untuk menghubungkan data dengan tampilan, adaptor perlu mengetahui tampilan tujuan penempatan data. Dengan demikian, adaptor berisi holder tampilan (dari kelas [ViewHolder](#)) yang menjelaskan tampilan item dan posisinya di dalam RecyclerView.

Dalam tugas ini, Anda akan membangun adaptor dengan holder tampilan yang menghubungkan celah antara data di daftar kata Anda dengan RecyclerView yang menampilkannya.

1. Klik kanan `java/com.android.example.recyclerview` dan pilih **New >> Java Class**.

2. Beri nama kelas sebagai WordListAdapter.
3. Berikan tanda tangan berikut pada WordListAdapter:

```
public class WordListAdapter extends
    RecyclerView.Adapter<WordListAdapter.WordViewHolder> {}
```

WordListAdapter menambah sebuah adaptor generik untuk RecyclerView agar menggunakan holder tampilan yang khusus untuk aplikasi Anda dan didefinisikan di dalam WordListAdapter. WordViewHolder menampilkan kesalahan karena Anda belum mendefinisikannya.

4. Klik deklarasi kelas (WordListAdapter) lalu klik bola lampu merah di sebelah kiri panel. Pilih **Implement methods**. Ini akan memunculkan kotak dialog yang meminta Anda memilih metode mana yang akan diimplementasikan. Pilih ketiga metode dan klik **OK**.

Ini akan membuat placeholder kosong untuk semua metode yang harus Anda implementasikan. Perhatikan bagaimana onCreateViewHolder dan onBindViewHolder mengacu pada WordViewHolder, yang belum diimplementasikan.

2.5 Buat holder tampilan

1. Di dalam kelas WordListAdapter, tambahkan kelas dalam WordViewHolder dengan tanda tangan ini:

```
class WordViewHolder extends RecyclerView.ViewHolder {}
```

2. Anda akan melihat kesalahan tentang konstruktor default yang tidak ada. Anda bisa melihat detail tentang kesalahan dengan mengarahkan mouse ke atas kode sumber yang digaris bawah merah atau ke atas baris horizontal merah di margin kanan panel open-files.

3. Tambahkan variabel ke kelas dalam WordViewHolder untuk tampilan teks dan adaptor:

```
public final TextView wordItemView;
final WordListAdapter mAdapter;
```

4. Dalam kelas WordViewHolder, tambahkan sebuah konstruktor yang melakukan inisialisasi tampilan teks holder tampilan dari sumber daya XML dan menyetel adaptornya:

```
public WordViewHolder(View itemView, WordListAdapter adapter) {
    super(itemView);
    wordItemView = (TextView) itemView.findViewById(R.id.word);
    this.mAdapter = adapter;
}
```

5. Jalankan aplikasi Anda untuk memastikan tidak ada kesalahan. Anda masih akan melihat tampilan kosong. Perhatikan peringatan **E/RecyclerView: No adapter attached; skipping layout** di logcat.

3.6 Menyimpan data Anda di adaptor

1. Untuk menampung data Anda di adaptor, buat daftar string yang tertaut dan privat di WordListAdapter dan beri nama mWordList.

```
private final LinkedList<String> mWordList;
```

2. Anda sekarang bisa mengisi metode getItemCount() untuk mengembalikan ukuran mWordList .

```
@Override
public int getItemCount() {
    return mWordList.size();
}
```

Selanjutnya, WordListAdapter memerlukan konstruktor yang menginisialisasi daftar kata dari data. Untuk membuat tampilan dari item daftar, WordListAdapter perlu memekarkan XML untuk sebuah item daftar. Anda harus

menggunakan `layout inflater` untuk pekerjaan ini. `LayoutInflater` membaca deskripsi XML dan mengonversinya menjadi tampilan yang sesuai.

- Buat variabel anggota untuk inflater di `WordListAdapter`.

```
private LayoutInflater mInflater;
```

- Implementasikan constructor untuk `WordListAdapter`. Constructor perlu memiliki parameter konteks dan daftar kata teraut dengan data aplikasi. Metode perlu membuat instance layout inflater untuk `mInflater` dan menyetel `mWordList` ke data yang diteruskan.

```
public WordListAdapter(Context context, LinkedList<String> wordList) {
    mInflater = LayoutInflater.from(context);
    this.mWordList = wordList;
}
```

- Isikan metode `onCreateViewHolder()` dengan kode berikut. Metode `onCreateViewHolder` sama dengan metode `onCreate`. Metode ini memekarkan layout item dan mengembalikan holder tampilan dengan layout dan adaptor.

```
@Override
public WordViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View mItemView = mInflater.inflate(R.layout.wordlist_item, parent, false);
    return new WordViewHolder(mItemView, this);
}
```

- Isikan metode `onBindViewHolder` dengan kode berikut. Metode `onBindViewHolder` menghubungkan data Anda ke holder tampilan.

```
@Override
public void onBindViewHolder(WordViewHolder holder, int position) {
    String mCurrent = mWordList.get(position);
    holder.wordItemView.setText(mCurrent);
}
```

- Jalankan aplikasi Anda untuk memastikan tidak ada kesalahan. Anda masih akan melihat peringatan "E/RecyclerView: No adapter attached; skipping layout". Anda akan memperbaiknya di tugas selanjutnya.

3.7 Buat RecyclerView di Main Activity

Setelah memiliki adaptor dengan holder tampilan, Anda akhirnya bisa membuat `RecyclerView` dan menghubungkan semua bagian untuk menampilkan data.

- Buka `MainActivity.java`
- Tambahkan variabel anggota ke `MainActivity` untuk `RecyclerView` dan adaptor.

```
private RecyclerView mRecycler;
private WordListAdapter mAdapter;
```

- Dalam metode `onCreate` aktivitas, tambahkan kode berikut yang membuat `RecyclerView` dan menghubungkannya dengan adaptor dan data. Baca komentar kode! Perhatikan bahwa Anda harus menyisipkan kode ini setelah inisialisasi `mWordList`.

```
// Get a handle to the RecyclerView.
mRecycler = (RecyclerView) findViewById(R.id.recyclerview);
// Create an adapter and supply the data to be displayed.
mAdapter = new WordListAdapter(this, mWordList);
// Connect the adapter with the RecyclerView.
mRecycler.setAdapter(mAdapter);
// Give the RecyclerView a default layout manager.
mRecycler.setLayoutManager(new LinearLayoutManager(this));
```

- Jalankan aplikasi Anda. Anda seharusnya melihat daftar kata ditampilkan dan Anda bisa mengulir daftar.

Tugas 4. Membuat daftar agar interaktif

Melihat daftar item memang menarik, tetapi akan lebih menyenangkan dan berguna jika pengguna Anda bisa berinteraksi dengannya.

Untuk melihat bagaimana RecyclerView bisa merespons masukan pengguna, Anda akan menyisipkan handler klik ke setiap item secara terprogram. Saat item diketuk, handler klik dieksekusi dan teks item tersebut akan berubah.

4.1 Buat item merespons klik

1. Buka file WordListAdapter.java.
2. Ubah tanda tangan kelas WordViewHolder untuk mengimplementasikan View.OnClickListener.

```
class WordViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener
```

3. Klik header kelas dan bola lampu merah untuk mengimplementasikan stub untuk metode yang diperlukan, yang dalam hal ini hanyalah metode onClick().
4. Tambahkan kode berikut pada isi metode onClick().

```
// Get the position of the item that was clicked.  
int mPosition = getLayoutPosition();  
// Use that to access the affected item in mWordList.  
String element = mWordList.get(mPosition);  
// Change the word in the mWordList.  
mWordList.set(mPosition, "Clicked! " + element);  
// Notify the adapter, that the data has changed so it can  
// update the RecyclerView to display the data.  
mAdapter.notifyDataSetChanged();
```

5. Hubungkan onClickListener dengan tampilan. Tambahkan kode ini ke constructor WordViewHolder (di bawah baris "this.mAdapter = adapter"):

```
itemView.setOnClickListener(this);
```

6. Jalankan aplikasi Anda. Klik item untuk melihat perubahan teksnya.

Kode solusi: [WordListAdapter.java](#) dan [Activity.java](#)

Tugas 5. Menambahkan FAB untuk menyisipkan item

Ada berbagai cara untuk menambahkan perilaku tambahan ke daftar dan item daftar. Salah satunya adalah menggunakan tombol aksi mengambang (FAB). Misalnya, di Gmail, FAB digunakan untuk menulis email baru. Dalam tugas ini, Anda akan mengimplementasikan FAB untuk menambahkan item ke daftar kata.

Mengapa? Anda telah melihat bahwa Anda bisa mengubah konten item daftar. Daftar item yang ditampilkan RecyclerView bisa dimodifikasi secara dinamis-- ini bukanlah daftar item yang statis.

Untuk praktik ini, Anda akan membuat kata baru untuk disisipkan ke dalam daftar. Untuk aplikasi yang lebih berguna, Anda akan mendapatkan data dari pengguna Anda.

5.1. Menambahkan Tombol Aksi Mengambang (FAB)

FAB adalah kontrol standar dari Spesifikasi Desain Material dan merupakan bagian dari Pustaka Dukungan Android. Anda akan mempelajari selengkapnya di bab tentang Desain Material. Kontrol UI ini memiliki properti yang telah didefinisikan sebelumnya. Untuk membuat FAB untuk aplikasi Anda, tambahkan kode berikut di dalam layout koordinator activity_main.xml.

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="16dp"
    android:clickable="true"
    android:src="@drawable/ic_add_24dp" />
```

Perhatikan yang berikut ini:

- `@+id/fab` FAB umumnya diberi id "fab".
- `android:layout_gravity="bottom|end"`. FAB umumnya ditempatkan di bagian bawah dan di bagian akhir alur membaca/menulis.
- `android:src="@drawable/ic_add_black_24dp"`. Ditandai dengan warna merah oleh Android Studio karena sumber daya tidak ada.

Android menyediakan pustaka ikon untuk ikon Android standar. `ic_add_black_24dp` adalah salah satu dari ikon standar. Anda harus menambahkannya ke sumber daya yang dapat digambar untuk menggunakannya.

1. Klik kanan folder drawable Anda.
2. Pilih **New > Vector Asset**
3. Pastikan **Asset Type** adalah **Material Icon**.
4. Klik tombol ikon di samping **Icon**.
5. Di bagian Content, temukan tanda +. Nama sumber daya adalah **ic_add_black_24dp**.
6. Biarkan semuanya tidak dicentang dan klik **Next**.
7. Klik **Finish**.
8. Jalankan aplikasi Anda.

Catatan: Karena ini adalah penggambaran vektor, ini disimpan sebagai file XML. Penggambaran vektor otomatis diskalakan, sehingga Anda tidak perlu menyimpan bitmap untuk setiap resolusi layar. Ketahui selengkapnya:[Android Vector Asset Studio](#).

5.2 Tambahkan perilaku ke FAB

Dalam tugas ini, Anda akan menambahkan listener onClick ke FAB yang melakukan tindakan berikut:

- Menambahkan kata di akhir daftar kata.
- Memberi tahu adaptor bahwa data telah berubah.
- Menggulir ke item yang disisipkan.
- Di MainActivity.java, di akhir metode onCreate(), tambahkan kode berikut:

```
// Add a floating action click handler for creating new entries.
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int wordListSize = mWordList.size();
        // Add a new word to the end of the wordList.
        mWordList.addLast("+" + Word + wordListSize);
        // Notify the adapter, that the data has changed so it can
        // update the RecyclerView to display the data.
        mRecyclerView.getAdapter().notifyItemInserted(wordListSize);
        // Scroll to the bottom.
        mRecyclerView.smoothScrollToPosition(wordListSize);
    }
});
```

- Jalankan aplikasi Anda. Untuk menguji aplikasi Anda, lakukan yang berikut:
 1. Gulir daftar kata.
 2. Klik item.

3. Tambahkan item dengan mengeklik FAB. 1 Apa yang terjadi jika Anda memutar layar? Anda akan mempelajari di pelajaran nanti cara mempertahankan kondisi aplikasi saat layar diputar.

Kode solusi

Proyek Android Studio: [RecyclerView](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Membuat listener klik untuk setiap item di daftar cukup mudah, tetapi bisa membebani kinerja aplikasi Anda jika Anda memiliki banyak sekali data. Pelajari bagaimana Anda bisa mengimplementasikan ini secara lebih efisien. Ini adalah tantangan tingkat lanjut. Mulai dengan memikirkannya secara konseptual, lalu carilah contoh implementasi.

Rangkuman

- RecyclerView adalah cara efisien jika dilihat dari sumber daya, untuk menampilkan daftar item yang bisa digulir.
- Untuk menggunakan RecyclerView, Anda harus mengaitkan data ke Adapter/ViewHolder yang Anda buat dan ke pengelola layout pilihan Anda.
- Listener klik bisa dibuat untuk mendeteksi klik mouse di RecyclerView.
- Pustaka dukungan Android berisi versi framework Android yang kompatibel dengan versi sebelumnya.
- Pustaka dukungan Anda berisi rangkaian utilitas yang berguna untuk aplikasi Anda.
- Dependensi build ditambahkan ke file build.gradle (Module app).
- Layout bisa ditetapkan sebagai file sumber daya.layout
- Sebuah LayoutInflater membaca file sumber daya layout dan membuat objek View dari file tersebut.
- Tombol Aksi Mengambang (FAB) bisa memodifikasi item dalam RecyclerView secara dinamis.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [RecyclerView](#)

Ketahui selengkapnya

- [CoordinatorLayout](#)
- [RecyclerView](#)
- [Android Support Library](#)
- [Membuat Daftar dan Kartu](#)
- [Animasi RecyclerView dan di Balik Layar \(Android Dev Summit 2015\)](#)

5.1: Drawable, Gaya, dan Tema

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1: Membuat Proyek Scorekeeper
- Tugas 2: Membuat sumber daya Drawable
- Tugas 3: Menerapkan gaya ke tampilan Anda
- Tugas 4: Memperbarui tema dari bilah menu
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Dalam bab ini, Anda akan mempelajari cara mengurangi kode Anda, meningkatkan keterbacaan kode dan kemudahan pengelolaan dengan menerapkan gaya umum ke tampilan Anda, menggunakan sumber daya Drawable, dan menerapkan tema ke aplikasi Anda.

Yang harus sudah Anda KETAHUI

Dari bab sebelumnya, Anda seharusnya sudah mengenal konsep dasar daur hidup Aktivitas dan cara:

- Membuat dan menjalankan aplikasi dalam Android Studio.
- Membuat dan mengedit elemen UI menggunakan Layout Editor, XML, dan kode.
- Menambahkan fungsionalitas onClick ke tombol.
- Memperbarui tampilan pada waktu proses.
- Menambahkan item menu dengan fungsionalitas onClick.
- Meneruskan data antar aktivitas menggunakan Intent.

Yang akan Anda PELAJARI

Dalam praktik ini, Anda akan mempelajari cara:

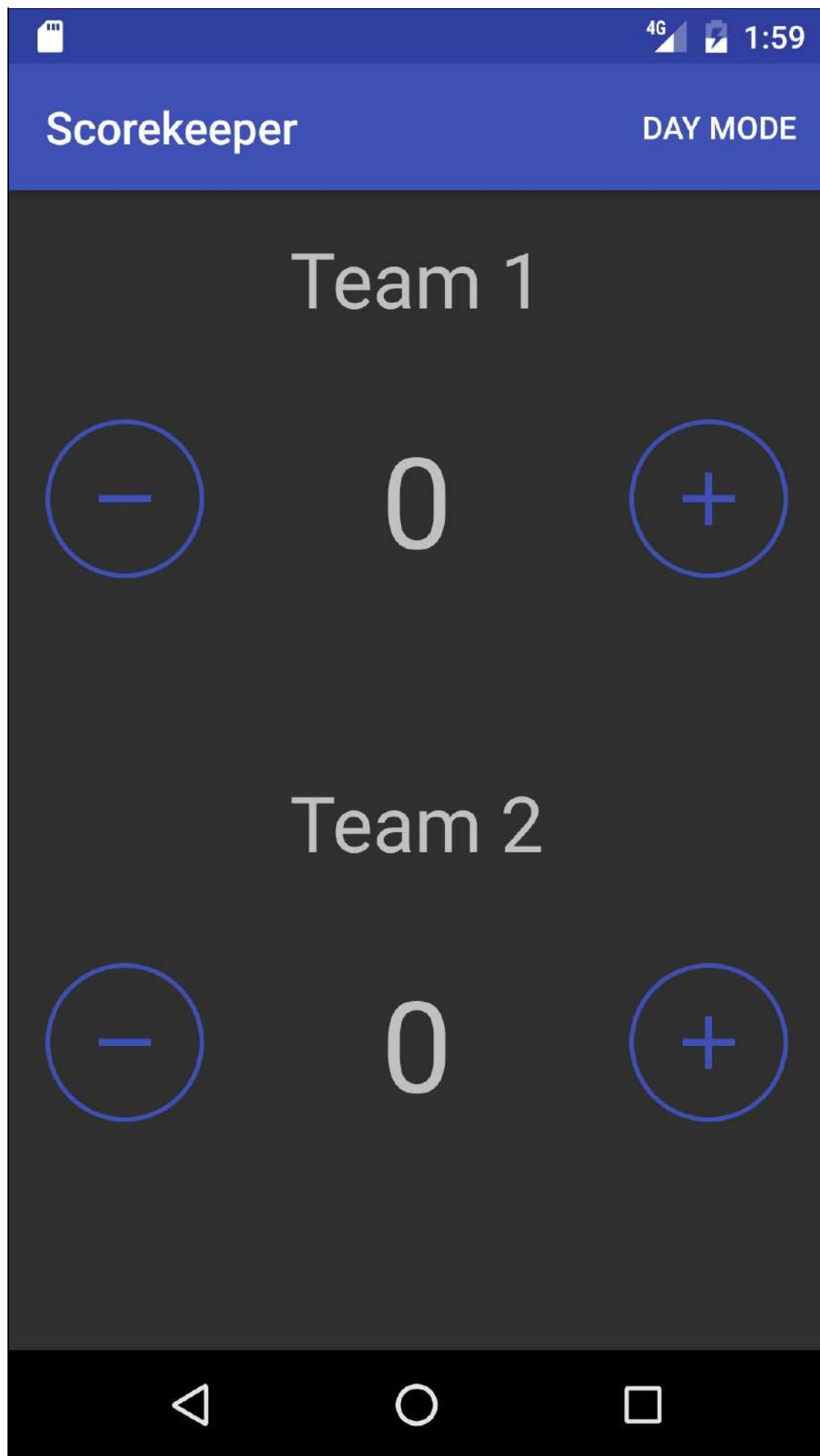
- Mendefinisikan gaya.
- Menerapkan gaya pada tampilan.
- Menerapkan tema pada aktivitas atau aplikasi baik dalam XML atau secara terprogram.
- Menggunakan sumber daya Drawable.

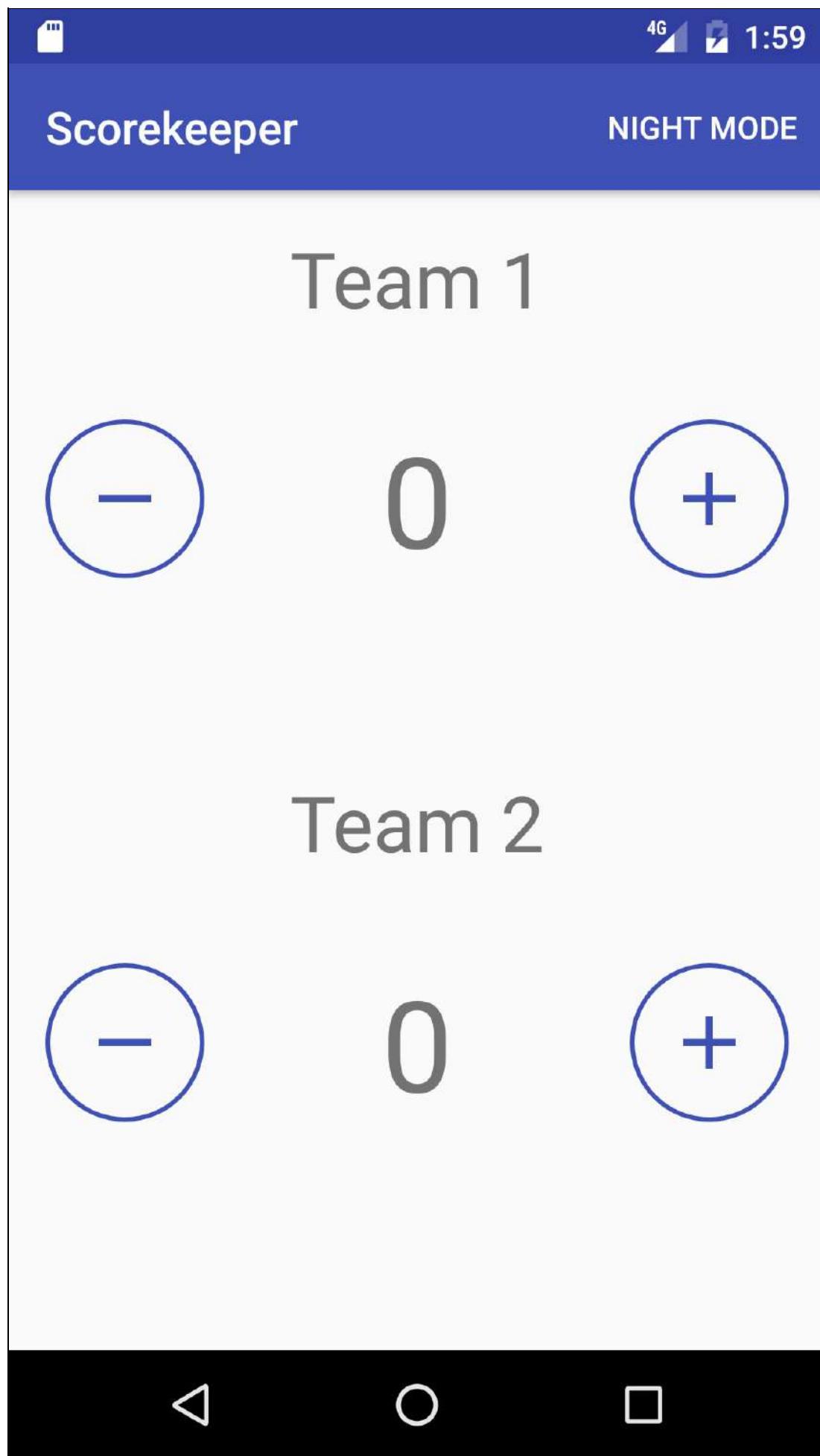
Yang akan Anda LAKUKAN

- Membuat aplikasi Android baru dan menambahkan tombol dan tampilan TextView ke layout.
- Membuat sumber daya Drawable di XML dan menggunakannya sebagai latar belakang untuk tombol Anda.
- Menerapkan gaya ke elemen UI aplikasi.
- Menambahkan item menu yang mengubah tema aplikasi ke "mode malam" yang berkontras rendah.

Ringkasan Aplikasi

Aplikasi "Scorekeeper" terdiri atas dua rangkaian tombol dan dua tampilan teks yang digunakan untuk melacak skor game berbasis poin apa pun dengan dua pemain.





Tugas 1: Membuat Aplikasi Scorekeeper

Dalam bagian ini, Anda akan membuat proyek Android Studio, memodifikasi layout, dan menambahkan fungsionalitas onClick pada tombolnya.

1.1 Buat proyek "Scorekeeper"

1. Mulai Android Studio dan buat proyek Android Studio baru.
 - Beri nama proyek Anda "Scorekeeper"
 - Terima default untuk Company Domain dan Project location.
2. Terima SDK Minimum default.
3. Pilih template **Empty Activity**. 1 Terima nama default untuk aktivitas, pastikan **Generate Layout File** dicentang dan klik **Finish**.

1.2 Buat layout untuk aktivitas utama

Definisikan tampilan root:

1. Buka file layout untuk aktivitas utama.
2. Hapus TextView yang berisi "Hello World."
3. Ubah tampilan root menjadi LinearLayout dan tambahkan atribut berikut (tanpa menghapus atribut yang sudah ada):

| Atribut | Nilai |
|---------------------|------------|
| android:orientation | "vertical" |

Definisikan kontainer skor:

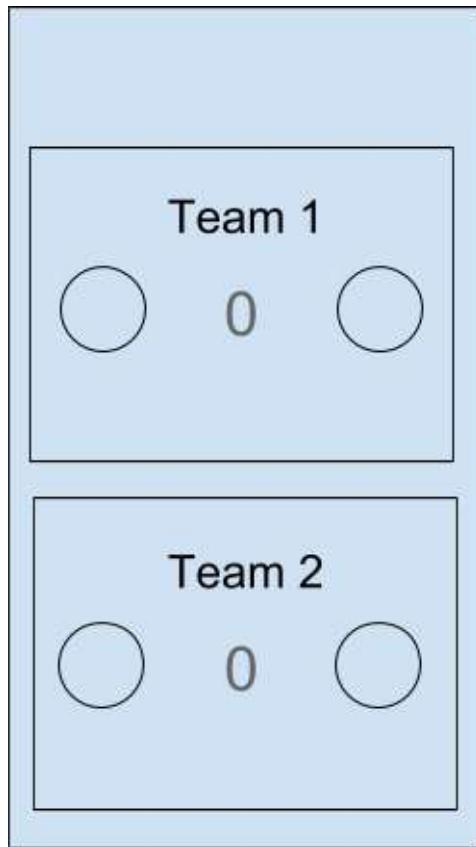
1. Di dalam LinearLayout, tambahkan dua grup tampilan RelativeLayout (satu untuk menampung skor untuk setiap tim) dengan atribut berikut:

| Atribut | Nilai |
|-----------------------|----------------|
| android:layout_width | "match_parent" |
| android:layout_height | "0dp" |
| android:layout_weight | "1" |

Anda mungkin tidak menyangka saat melihat atribut layout_height disetel ke 0dp di tampilan ini. Ini karena kita menggunakan atribut "layout_weight" untuk menentukan seberapa luas ruang yang diperlukan tampilan ini di layout induk. Lihat [Dokumentasi LinearLayout](#) untuk informasi selengkapnya.

Tambahkan tampilan ke UI

1. Tambahkan dua [tampilan ImageButton](#) (satu untuk meningkatkan skor dan satu untuk menurunkan skor) dan sebuah TextView untuk menampilkan skor di antara tombol ke setiap RelativeLayout.
2. Tambahkan atribut `android:id` ke TextView skor dan semua ImageButton.
3. Tambahkan satu lagi TextView ke setiap RelativeLayout di atas skor untuk menunjukkan Nama Tim.



Tambahkan aset vektor

1. Pilih **File > New > Vector Asset** untuk membuka Vector Asset Studio.
2. Klik ikon untuk mengubahnya menjadi daftar file ikon material. Pilih kategori **Content**.
3. Pilih ikon plus dan klik **OK**.
4. Ubah nama file sumber daya "ic_plus" dan centang kotak **Override** di sebelah opsi ukuran.
5. Ubah ukuran ikon ke 40dp x 40dp.
6. Klik **Next**, lalu **Finish**.
7. Ulangi proses ini untuk menambahkan ikon "minus" dan beri nama file "ic_minus".

Tambahkan atribut ke tampilan Anda

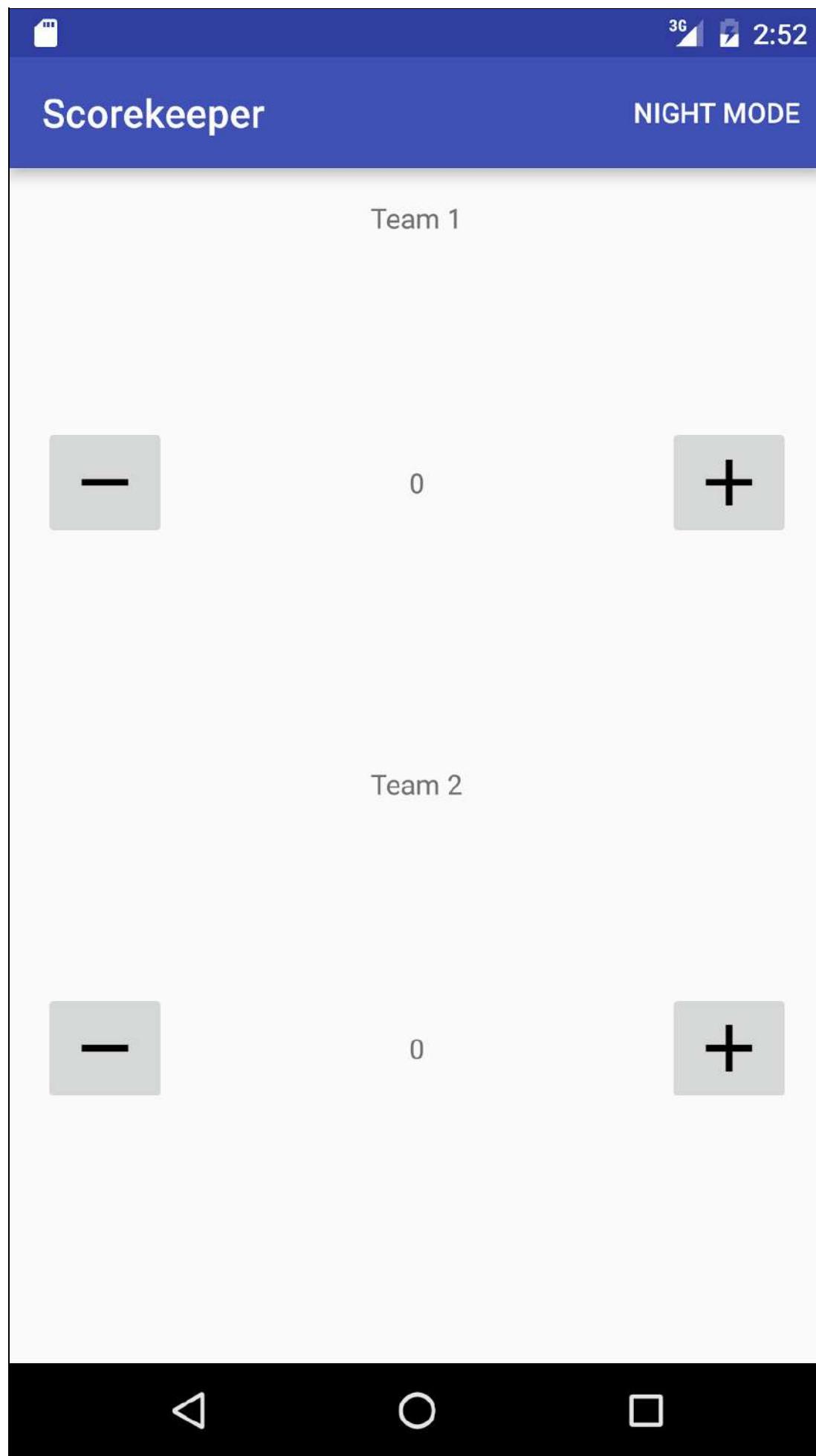
1. Ubah tampilan teks skor agar terbaca "0" dan tampilan teks tim agar terbaca "Team 1" dan "Team 2".
2. Tambahkan atribut berikut ini ke tombol gambar kiri Anda:

```
    android:src="@drawable/ic_minus"  
    android:contentDescription="Minus Button"
```

3. Tambahkan atribut berikut ini ke tombol gambar kanan Anda:

```
    android:src="@drawable/ic_plus"  
    android:contentDescription="Plus Button"
```

4. Ekstrak semua sumber daya string Anda. Proses ini membuang semua string Anda dari kode Java dan menempatkannya ke dalam file string.xml. Ini memungkinkan aplikasi Anda lebih mudah dilokalkan ke bahasa yang berbeda. Untuk mempelajari cara mengekstrak sumber daya string, lihat bagian [Mengekstrak Sumber Daya](#) di [Apendiks](#).



Kode Solusi:

Catatan: Kode Anda mungkin sedikit berbeda karena terdapat beberapa cara untuk membuat layout yang sama.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.scorekeeper.MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:text="@string/team_1"/>

        <ImageButton
            android:id="@+id/decreaseTeam1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_centerVertical="true"
            android:contentDescription="@string/minus_button"
            android:src="@drawable/ic_minus" />

        <TextView
            android:id="@+id/score_1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:text="@string/initial_count"/>

        <ImageButton
            android:id="@+id/increaseTeam1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:contentDescription="@string/plus_button"
            android:src="@drawable/ic_plus"/>
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:text="@string/team_2"/>

        <ImageButton
            android:id="@+id/decreaseTeam2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:contentDescription="@string/minus_button"
            android:src="@drawable/ic_minus" />
    </RelativeLayout>

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:contentDescription="@string/minus_button"
        android:src="@drawable/ic_minus"/>

    <TextView
        android:id="@+id/score_2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/initial_count"/>

    <ImageButton
        android:id="@+id/increaseTeam2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="true"
        android:layout_alignRight="true"
        android:layout_centerVertical="true"
        android:contentDescription="@string/plus_button"
        android:src="@drawable/ic_plus"/>
</RelativeLayout>
</LinearLayout>
```

1.3 Inisialisasi TextView Anda dan variabel hitungan skor

Untuk mencatat skor, Anda memerlukan dua hal:

- Variabel integer agar Anda bisa melacak skor.
- Referensi ke TextView skor Anda di MainActivity sehingga Anda bisa memperbarui skor.
- Dalam metode `onCreate()` MainActivity, temukan TextView skor Anda menurut id dan tetapkan TextView ke variabel anggota.
- Buat dua variabel anggota integer, yang menunjukkan skor setiap tim, dan inisialisasi ke 0.

1.4 Implementasikan fungsionalitas onClick untuk tombol Anda.

1. Di MainActivity Anda, implementasi dua metode onClick: `increaseScore()` dan `decreaseScore()`.

Catatan: semua metode onClick memiliki tanda tangan yang sama - metode ini mengembalikan void dan mengambil View sebagai argumen.

2. Tombol kiri akan mengurangi TextView skor, sedangkan tombol kanan akan menambahnya.

Kode Solusi:

Catatan: Anda juga harus menambahkan atribut `android:onClick` ke setiap tombol di file `activity_main.xml`. Anda bisa mengidentifikasi tombol mana yang diklik dengan memanggil `view.getId()` di metode `onClick`.

```

/**
 * Method that handles the onClick of both the decrement buttons
 * @param view The button view that was clicked
 */
public void decreaseScore(View view) {
    //Get the ID of the button that was clicked
    int viewID = view.getId();
    switch (viewID){
        //If it was on Team 1
        case R.id.decreaseTeam1:
            //Decrement the score and update the TextView
            mScore1--;
            mScoreText1.setText(String.valueOf(mScore1));
            break;
        //If it was Team 2
        case R.id.decreaseTeam2:
            //Decrement the score and update the TextView
            mScore2--;
            mScoreText2.setText(String.valueOf(mScore2));
    }
}

/**
 * Method that handles the onClick of both the increment buttons
 * @param view The button view that was clicked
 */
public void increaseScore(View view) {
    //Get the ID of the button that was clicked
    int viewID = view.getId();
    switch (viewID){
        //If it was on Team 1
        case R.id.increaseTeam1:
            //Increment the score and update the TextView
            mScore1++;
            mScoreText1.setText(String.valueOf(mScore1));
            break;
        //If it was Team 2
        case R.id.increaseTeam2:
            //Increment the score and update the TextView
            mScore2++;
            mScoreText2.setText(String.valueOf(mScore2));
    }
}

```

Tugas 2: Membuat sumber daya Drawable

Anda sekarang sudah memiliki aplikasi Scorekeeper yang berfungsi! Akan tetapi, layout-nya tidak menarik dan tidak menyampaikan fungsi tombol. Untuk memperjelasnya, latar belakang abu-abu standar tombol bisa diubah.

Di Android, grafik sering kali ditangani oleh sebuah sumber daya yang disebut **Drawable**. Di latihan berikut, Anda akan mempelajari cara membuat tipe drawable tertentu yang disebut **ShapeDrawable** dan menerapkannya ke tombol Anda sebagai latar belakang.

Untuk informasi selengkapnya tentang Drawable, lihat [Dokumentasi Sumber Daya Drawable](#).

2.1 Buat Drawable Bentuk

ShapeDrawable adalah bentuk geometris primitif yang didefinisikan di file xml oleh sejumlah atribut, yang mencakup warna, bentuk, padding, dan lainnya. Ini mendefinisikan grafik vektor, yang bisa diskalakan naik dan turun tanpa kehilangan definisi.

1. Klik kanan folder **drawable** di direktori sumber daya.
2. Pilih **New > Drawable resource file**.
3. Beri nama file "button_background" dan klik **OK**.

- Buang semua kode, kecuali:

```
<?xml version="1.0" encoding="utf-8"?>
```

- Tambahkan kode berikut yang membuat bentuk oval dengan garis luar:

```
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval">
    <stroke
        android:width="2dp"
        android:color="@color/colorPrimary"/>
</shape>
```

2.2 Terapkan drawable bentuk sebagai latar belakang

- Buka file layout untuk aktivitas utama Anda.
- Untuk semua tombol, tambah drawable sebagai latar belakang: `android:background="@drawable/button_background"`. Perhatikan bahwa latar belakang otomatis diskalakan agar pas dengan ukuran tampilan.
- Ukuran tombol besarnya harus sedemikian rupa sehingga bisa dirender dengan benar di semua perangkat. Ubah "layout_height" dan "layout_width" untuk setiap tombol menjadi 70dp, yaitu ukuran yang baik untuk kebanyakan perangkat. Menggunakan dimensi yang di-hardcode bukanlah praktik terbaik, tetapi menggunakan ketebalan dengan layout linear yang dilapiskan untuk mendapatkan ukuran yang diinginkan akan melibatkan terlalu banyak detail untuk praktik ini.

```
    android:layout_width="70dp"
    android:layout_height="70dp"
```

- Ekstrak sumber daya dimensi sehingga Anda bisa mengaksesnya di satu lokasi. Untuk informasi cara melakukannya, lihat [Appendix](#).
- Jalankan aplikasi Anda.

Tugas 3: Tata gaya tampilan Anda

Seiring Anda terus menambahkan tampilan dan atribut ke layout Anda, kode Anda mulai menjadi lebih besar dan repetitif, khususnya jika Anda menerapkan atribut yang sama pada berbagai elemen yang mirip. Gaya bisa menentukan properti umum, seperti padding, font, warna font, ukuran font, dan warna latar belakang. Atribut yang berorientasi layout, seperti tinggi, lebar dan lokasi relatif harus berada di file sumber daya layout.

Di latihan berikut, Anda akan mempelajari cara membuat gaya dan menerapkannya ke beberapa tampilan dan layout, yang memungkinkan atribut umum diperbarui secara bersamaan dari satu lokasi.

Catatan: Gaya ditujukan untuk atribut yang memodifikasi rupa tampilan. Parameter layout seperti tinggi, tebal, dan lokasi relatif harus tetap berada di file layout.

3.1 Buat gaya tombol

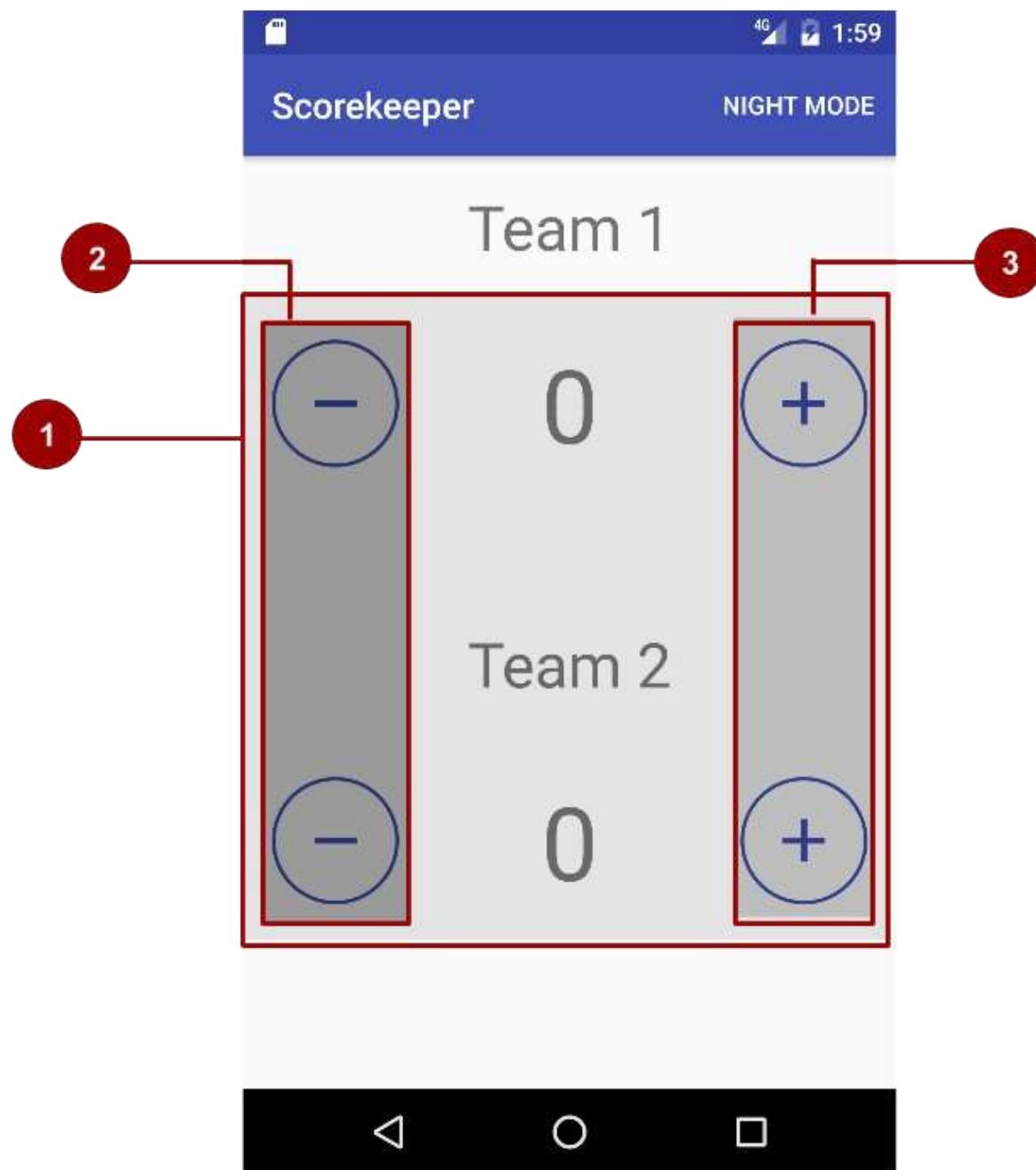
Di Android, gaya bisa mewarisi properti dari gaya lain. Anda bisa mendeklarasikan induk untuk gaya Anda menggunakan parameter "parent" opsional dan memiliki properti berikut:

- Atribut gaya apa pun yang didefinisikan oleh gaya induk otomatis disertakan di dalam gaya anak.
- Atribut gaya yang didefinisikan dalam gaya induk dan anak menggunakan definisi gaya anak (anak mengganti induk).
- Gaya anak bisa mencakup atribut tambahan yang tidak didefinisikan oleh induk.

Misalnya, keempat tombol dalam contoh ini memiliki drawable latar belakang umum yang sama, tetapi dengan ikon yang berbeda untuk plus dan minus. Terlebih lagi, dua tombol untuk menambah memiliki ikon yang sama, seperti halnya dengan dua tombol untuk mengurangi. Dengan demikian, Anda bisa membuat 3 gaya:

1. Gaya tombol skor untuk semua tombol, yang menyertakan properti default tampilan ImageButton dan juga latar belakang drawable.
2. Gaya tombol minus untuk tombol kurang, yang mewarisi atribut gaya sebelumnya dan juga menyertakan ikon minus.
3. Gaya tombol plus untuk tombol tambah, juga mewarisi gaya tombol skor dan juga menyertakan ikon plus.

Gaya ini ditunjukkan dalam gambar di bawah.



Lakukan yang berikut:

1. Dalam direktori sumber daya, cari dan buka file "values/styles.xml". Di file inilah semua kode gaya Anda ditempatkan. Gaya "AppTheme" selalu ditambahkan secara otomatis dan Anda bisa melihat bahwa gaya ini diluaskan dari "Theme.AppCompat.Light.DarkActionBar".

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
```

Perhatikan bahwa atribut "parent", yaitu tempat Anda menentukan gaya induk Anda menggunakan XML. Atribut nama, dalam hal ini, "AppTheme" mendefinisikan nama gaya. Atribut induk, dalam hal ini, "Theme.AppCompat.Light.DarkActionBar", mendeklarasikan atribut gaya induk yang diwarisi oleh "AppTheme". Dalam hal ini, ini adalah tema default Android dengan latar belakang warna cerah dan bilah tindakan gelap. Tema adalah gaya yang diterapkan pada keseluruhan aktivitas atau aplikasi, bukan pada tampilan tunggal. Ini memungkinkan gaya yang konsisten pada keseluruhan aktivitas atau aplikasi (misalnya tampilan dan nuansa yang konsisten untuk Bilah Aplikasi di setiap bagian aplikasi Anda).

2. Di antara tag `<resources>`, tambahkan gaya baru dengan atribut berikut untuk membuat gaya umum untuk semua tombol:

- o Setel gaya induk menjadi "Widget.AppCompat.Button" untuk mempertahankan atribut default tombol.
- o Tambahkan atribut yang mengubah latar belakang drawable menjadi latar belakang yang Anda buat di tugas sebelumnya.

```
<style name="ScoreButtons" parent="Widget.AppCompat.Button">
    <item name="android:background">@drawable/button_background</item>
</style>
```

3. Buat gaya untuk tombol plus dengan mewujudkan gaya "ScoreButtons":

```
<style name="PlusButtons" parent="ScoreButtons">
    <item name="android:src">@drawable/ic_plus</item>
    <item name="android:contentDescription">@string/plus_button</item>
</style>
```

Catatan: Atribut contentDescription adalah untuk pengguna yang menyandang gangguan visual. Ini berfungsi sebagai label yang digunakan oleh perangkat aksesibilitas tertentu untuk mengucapkan untuk menyediakan konteks tentang makna elemen UI.

4. Buat gaya untuk tombol minus:

```
<style name="MinusButtons" parent="ScoreButtons">
    <item name="android:src">@drawable/ic_minus</item>
    <item name="android:contentDescription">@string/minus_button</item>
</style>
```

5. Di file layout untuk aktivitas utama, buang semua atribut yang Anda definisikan di gaya untuk setiap tombol dan ganti dengan gaya yang sesuai:

```
style="@style/MinusButtons"
style="@style/PlusButtons"
```

Catatan: Atribut gaya tidak menggunakan namespace "android:" karena ini adalah bagian dari atribut XML default.

3.2 Buat gaya TextView

Tampilan teks untuk menampilkan nama tim dan skor juga bisa ditata gayanya karena warna dan font-nya sama. Lakukan yang berikut:

1. Tambahkan atribut berikut ke semua TextView:

```
android:textAppearance="@style/TextAppearance.AppCompat.Headline"
```

2. Klik kanan di mana saja di atribut TextView skor pertama dan pilih **Refactor > Extract > Style...**
3. Beri nama gaya "ScoreText" dan centang kotak textAppearance (atribut yang baru Anda tambahkan) serta **Launch 'Use Styles Where Possible' refactoring after the style is extracted** (menggunakan kotak centang). Ini akan memindai file layout untuk tampilan dengan atribut yang sama dan menerapkan gaya untuk Anda. Jangan ekstrak atribut yang terkait dengan layout.
4. Pilih **OK**.

5. Pastikan scope disetel ke file layout activity_main.xml dan klik **OK**.
6. Sebuah panel di bagian bawah Android Studio akan terbuka jika gaya yang sama ditemukan di tampilan lain. Pilih **Do Refactor** untuk menerapkan gaya baru ke tampilan dengan atribut yang sama.
7. Jalankan aplikasi Anda. Seharusnya tidak ada perubahan apa pun, kecuali semua kode penataan gaya Anda sekarang berada di file sumber daya dan file layout Anda lebih ringkas.

Kode Solusi:**styles.xml**

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

    <style name="ScoreButtons" parent="AppTheme">
        <item name="android:background">@drawable/button_background</item>
    </style>

    <style name="PlusButtons" parent="ScoreButtons">
        <item name="android:src">@drawable/ic_plus</item>
        <item name="android:contentDescription">@string/plus_button</item>
    </style>

    <style name="MinusButtons" parent="ScoreButtons">
        <item name="android:src">@drawable/ic_minus</item>
        <item name="android:contentDescription">@string/minus_button</item>
    </style>

    <style name="ScoreText">
        <item name="android:textAppearance">@style/TextAppearance.AppCompat.Headline</item>
    </style>
</resources>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:weightSum="2"
    tools:context="com.example.android.scorekeeper.MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <TextView
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:text="@string/team_1"
            style="@style/ScoreText" />

        <ImageButton
            android:id="@+id/decreaseTeam1"
```

```
        android:layout_height="@dimen/button_size"
        android:layout_width="@dimen/button_size"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:onClick="decreaseScore"
        style="@style/MinusButtons"/>

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"
        android:id="@+id/score_1"
        android:text="@string/initial_count"
        style="@style/ScoreText" />

    <ImageButton
        android:id="@+id/increaseTeam1"
        android:layout_height="@dimen/button_size"
        android:layout_width="@dimen/button_size"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:onClick="increaseScore"
        style="@style/PlusButtons"/>

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1">

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="@string/team_2"
        style="@style/ScoreText" />

    <ImageButton
        android:id="@+id/decreaseTeam2"
        android:layout_height="@dimen/button_size"
        android:layout_width="@dimen/button_size"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:onClick="decreaseScore"
        style="@style/MinusButtons"/>

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"
        android:id="@+id/score_2"
        android:text="@string/initial_count"
        style="@style/ScoreText" />

    <ImageButton
        android:id="@+id/increaseTeam2"
        android:layout_height="@dimen/button_size"
        android:layout_width="@dimen/button_size"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:onClick="increaseScore"
        style="@style/PlusButtons"/>
</RelativeLayout>
</LinearLayout>
```

3.3 Memperbarui gaya

Manfaat menggunakan gaya tampak jelas ketika Anda ingin membuat perubahan pada beberapa elemen gaya yang sama. Anda bisa membuat teks lebih besar, lebih tebal, dan lebih cerah, dan mengubah ikon menjadi warna latar belakang tombol.

Buat perubahan berikut di file styles.xml Anda:

1. Tambahkan atau modifikasi masing-masing atribut berikut di blok gaya yang ditetapkan:

| Atribut | Blok Gaya |
|------------------------------------------|--------------|
| @color/colorPrimary | ScoreButtons |
| @style/TextAppearance.AppCompat.Display3 | ScoreText |

Catatan: Nilai `colorPrimary` otomatis dihasilkan oleh Android Studio saat Anda membuat proyek dan bisa ditemukan di file values/colors.xml. Atribut `TextAppearance.AppCompat.Display3` adalah gaya teks yang telah ditetapkan dan disediakan oleh Android.

2. Buat gaya baru yang disebut "TeamText" dengan atribut berikut:

```
<item name="android:textAppearance">@style/TextAppearance.AppCompat.Display1</item>
```

3. Ubah atribut gaya TextView nama tim menjadi gaya TeamText yang baru dibuat di activity_main.xml.
4. Jalankan aplikasi Anda. Dengan perubahan ini saja di file style.xml, semua tampilan diperbarui untuk merefleksikan perubahan.

Tugas 4: Tema dan Sentuhan Akhir

Anda telah melihat tampilan dengan karakteristik yang sama yang bisa ditata gayanya secara bersamaan di file "styles.xml". Tetapi, bagaimana jika Anda ingin mendefinisikan gaya untuk keseluruhan aktivitas atau bahkan aplikasi? Hal ini dapat dilakukan dengan menggunakan "Tema". Untuk menyetel tema untuk Aktivitas atau serangkaian Aktivitas, Anda harus memodifikasi file AndroidManifest.xml.

Dalam tugas ini, Anda akan menambahkan tema "mode malam" ke aplikasi Anda, yang akan memungkinkan pengguna menggunakan versi kontras rendah pada aplikasi Anda yang lebih nyaman untuk mata di malam hari, serta membuat beberapa sentuhan akhir pada Antarmuka Pengguna.

4.1 Jelajahi tema

1. Di file manifes Android, temukan atribut `<application>` tag dan ubah `android:theme` ke:

```
android:theme="@style/Theme.AppCompat.Light.NoActionBar"
```

Ini adalah tema yang ditentukan sebelumnya yang membuang bilah tindakan dari aktivitas Anda.

2. Jalankan aplikasi Anda. Bilah alat menghilang!
3. Ubah tema aplikasi kembali ke AppTheme yang merupakan anak tema `Theme.Appcompat.Light.DarkActionBar` seperti yang bisa dilihat di styles.xml.

Untuk menerapkan tema ke aktivitas, bukan ke keseluruhan aplikasi, tempatkan atribut tema di tag aktivitas, bukan di tag aplikasi. Untuk informasi selengkapnya tentang Tema dan Gaya, lihat [Panduan Gaya dan Tema](#).

4.2 Tambahkan tombol tema ke menu

Satu yang digunakan untuk menyetel tema aplikasi adalah untuk menyediakan pengalaman visual alternatif untuk menjelajah di malam hari. Dalam kondisi demikian, sering kali lebih baik menggunakan layout gelap dengan kontras rendah. Framework Android menyediakan tema yang memang dirancang untuk ini: Tema DayNight. Tema ini memiliki beberapa opsi bawaan yang memungkinkan Anda mengontrol warna di aplikasi Anda secara terprogram: baik dengan menyetelnya untuk berubah secara otomatis sesuai waktu, atau dengan perintah pengguna.

Dalam latihan ini, Anda akan menambahkan tombol menu yang akan mengalihkan aplikasi dari atau ke tema reguler dan tema "mode malam".

1. Klik kanan direktori "res" dan pilih **New > Android resource file**.
2. Beri nama file "main_menu", ubah **Resource Type** ke **Menu**, dan klik **OK**.
3. Tambahkan item menu dengan atribut berikut:

```
<item
    android:id="@+id/night_mode"
    android:title="@string/night_mode"/>
```

4. Buka "strings.xml" dan buat dua sumber daya string:

```
<string name="night_mode">Night Mode</string>
<string name="day_mode">Day Mode</string>
```

5. Tekan **Ctrl - O** untuk membuka menu Override Method di file Java aktivitas utama Anda dan pilih metode **onCreateOptionsMenu** yang terletak di bawah kategori "android.app.Activity". Klik **OK**.
6. Mekarkan menu yang baru Anda buat di dalam metode `onCreateOptionsMenu()` :

```
getMenuInflater().inflate(R.menu.main_menu, menu);
```

4.3 Ubah tema dari menu

Tema DayNight menggunakan kelas AppCompatDelegate untuk menyetel opsi mode malam di aktivitas Anda. Untuk mengetahui selengkapnya tentang tema ini, kunjungi [entri blog](#).

1. Di file styles.xml, ubah induk AppTheme menjadi "Theme.AppCompat.DayNight.DarkActionBar".
2. Ganti metode `onOptionsItemSelected()` di MainActivity dan periksa item menu mana yang diklik:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    //Check if the correct item was clicked
    if(item.getItemId()==R.id.night_mode){}
}
```

3. Sebagai respons terhadap klik pada tombol menu, Anda bisa memverifikasi setelan mode malam saat ini dengan memanggil `AppCompatDelegate.getDefaultNightMode()`.
4. Jika mode malam diaktifkan, ubah menjadi status nonaktif:

```
//Get the night mode state of the app
int nightMode = AppCompatDelegate.getDefaultNightMode();
//Set the theme mode for the restarted activity
if(nightMode == AppCompatDelegate.MODE_NIGHT_YES) {
    AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO);
}
```

5. Jika tidak, aktifkan mode ini:

```
else {
    AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_YES);
}
```

6. Tema hanya bisa berubah saat aktivitas sedang dibuat, sehingga panggil `recreate()` agar perubahan tema bisa diterapkan.

7. Metode `onOptionsItemSelected()` Anda harus mengembalikan true, karena klik item ditangani.
8. Jalankan aplikasi Anda. Menu "Mode Malam" sekarang akan mengalihkan tema aktivitas Anda. Anda mungkin memperhatikan bahwa label untuk item menu Anda selalu tampak sebagai "Night Mode" yang mungkin membingungkan pengguna jika aplikasi Anda sudah dalam tema gelap.
9. Tambahkan kode berikut di metode `onCreateOptionsMenu()`:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    //Inflate the menu from XML
    getMenuInflater().inflate(R.menu.main_menu, menu);

    //Change the label of the menu based on the state of the app
    int nightMode = AppCompatDelegate.getDefaultNightMode();
    if(nightMode == AppCompatDelegate.MODE_NIGHT_YES){
        menu.findItem(R.id.night_mode).setTitle(R.string.day_mode);
    } else{
        menu.findItem(R.id.night_mode).setTitle(R.string.night_mode);
    }
    return true;
}
```

10. Jalankan aplikasi Anda. Label tombol menu sekarang berubah dengan tema.

4.4 SaveInstanceState

Anda telah belajar di pelajaran sebelumnya bahwa Anda harus bersiap karena aktivitas Anda dihapus dan dibuat ulang pada waktu yang tidak diduga, misalnya saat layar Anda diputar. Di aplikasi ini, `TextView` yang berisi skor akan disetel ulang ke nilai awal 0 saat perangkat diputar. Untuk memperbaiki ini, lakukan yang berikut:

1. Ganti metode `onSaveInstanceState()` di `MainActivity` agar mempertahankan nilai dua `TextView` skor:

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    //Save the scores
    outState.putInt(STATE_SCORE_1, mScore1);
    outState.putInt(STATE_SCORE_2, mScore2);
    super.onSaveInstanceState(outState);
}
```

2. Di metode `onCreate()` `MainActivity.java`, periksa apakah terdapat `savedInstanceState`. Jika ada, pulihkan skor ke tampilan teks:

```
if (savedInstanceState != null) {
    mScore1 = savedInstanceState.getInt(STATE_SCORE_1);
    mScore2 = savedInstanceState.getInt(STATE_SCORE_2);

    //Set the score text views
    mScoreText1.setText(String.valueOf(mScore1));
    mScoreText2.setText(String.valueOf(mScore2));
}
```

Cukup demikian! Selamat, Anda sekarang memiliki Aplikasi Scorekeeper dengan gaya.

Kode solusi

Proyek Android Studio: [Scorekeeper](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Sekarang, perilaku tombol Anda belum terlalu mudah dipahami karena penampilan tombol tidak berubah saat ditekan. Android memiliki tipe drawable lain yang disebut **StateListDrawable** yang memungkinkan grafik yang berbeda digunakan, bergantung pada status objek.

Untuk masalah tantangan ini, buat sumber daya drawable yang mengubah latar belakang tombol menjadi warna yang sama dengan batasnya saat status tombol "ditekan". Anda juga harus menyetel warna teks di dalam tombol menjadi selector yang membuatnya berwarna putih saat tombol "ditekan".

Rangkuman

- [ShapeDrawable](#) adalah bentuk geometris primitif yang didefinisikan di file xml oleh sejumlah atribut, yang mencakup warna, bentuk, padding, dan lainnya.
 - Drawable meningkatkan tampilan aplikasi.
- Gaya bisa menetapkan properti utama, seperti tinggi, padding, warna font, ukuran font, warna latar belakang, dan sebagainya.
 - Penggunaan gaya bisa mengurangi jumlah kode umum untuk komponen UI Anda.
 - Gaya harus tidak menyertakan informasi yang terkait layout.
 - Gaya bisa diterapkan pada Tampilan, Aktivitas, atau Aplikasi.
 - Gaya yang diterapkan pada Aktivitas atau Aplikasi harus didefinisikan di file XML Manifes Android.
 - Gaya bisa diwarisi dengan mengidentifikasi gaya induk menggunakan XML.
- Saat Anda menerapkan gaya pada kumpulan Tampilan di dalam Aktivitas atau di keseluruhan aplikasi Anda, ini dikenal sebagai Tema.
 - Android:theme adalah atribut yang gayanya perlu Anda setel pada rangkaian Tampilan dalam Aktivitas atau Aplikasi.
 - Platform Android menyediakan banyak sekali kumpulan gaya dan tema.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Drawable, Gaya, dan Tema](#)

Ketahui selengkapnya

Dokumentasi Developer

- [Panduan LinearLayout](#)
- [Panduan Sumber Daya Drawable](#)
- [Panduan Gaya dan Tema](#)
- [Panduan Tema DayNight](#)

Video

- [Udacity - Themes and Styles](#)

5.2: Desain Material Daftar, Kartu, dan Warna

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1: Mengunduh Kode Starter
- Tugas 2: Menambahkan CardView and Gambar
- Tugas 3: Animasi RecyclerView dan Tampilan Detail
- Tugas 4: FAB dan Palet Warna Material
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Bab ini memperkenalkan konsep dari panduan [Desain Material](#) Google, serangkaian praktik terbaik untuk membuat aplikasi yang indah secara visual dan mudah digunakan. Anda akan mendapatkan pemahaman tentang cara menambahkan dan menggunakan CardView dan Tampilan Tombol Aksi Mengambang, menggunakan gambar secara efisien, serta menerapkan praktik terbaik desain untuk membuat pengalaman pengguna yang menyenangkan.

Yang harus sudah Anda KETAHUI

Dari bab sebelumnya, Anda harus sudah bisa:

- Membuat dan menjalankan aplikasi dalam Android Studio.
- Membuat dan mengedit elemen mengedit Layout Editor, XML, dan secara terprogram.
- Menggunakan RecyclerView untuk menampilkan daftar.

Yang akan Anda PELAJARI

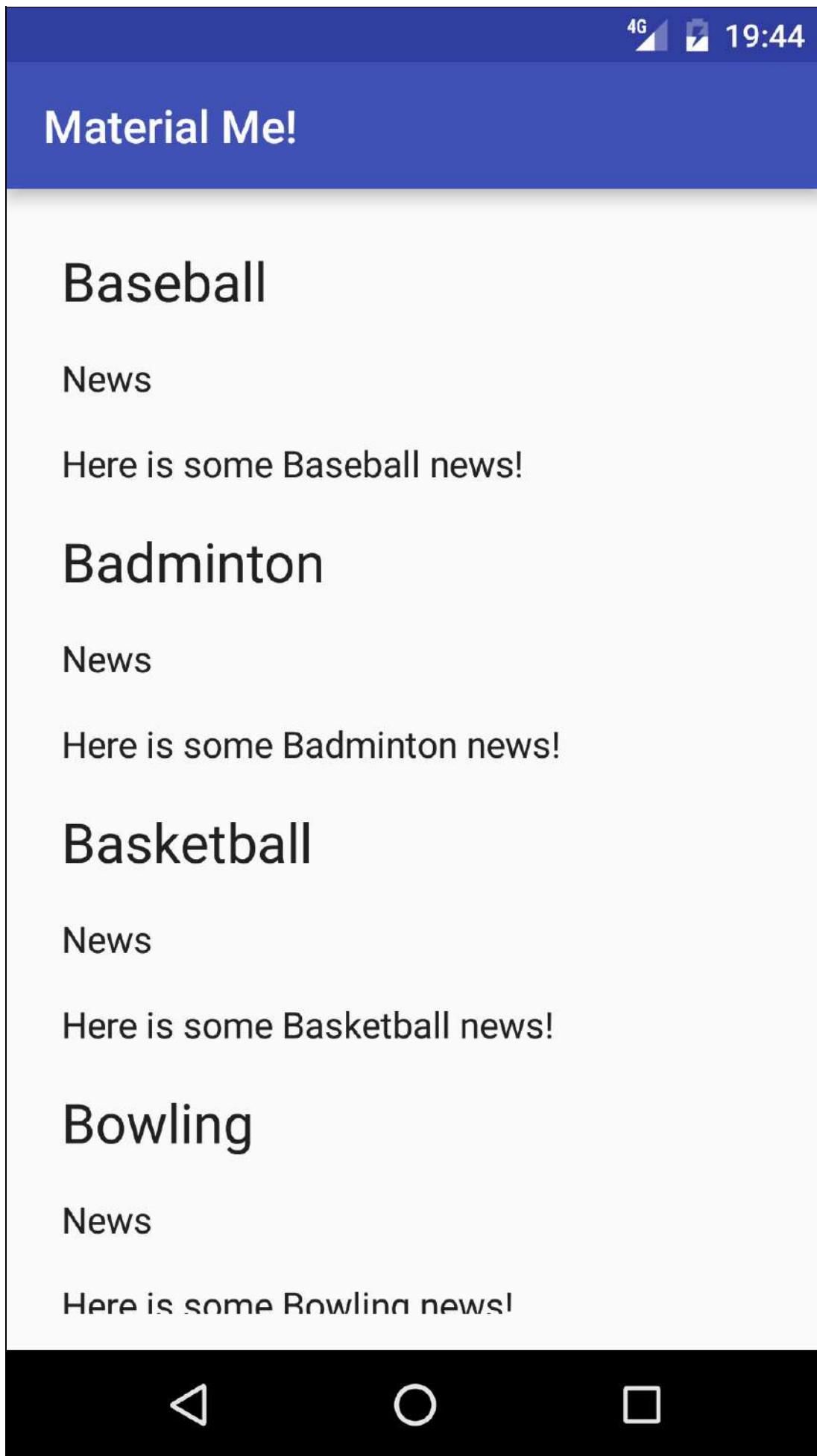
- Penggunaan widget material (Tombol Aksi Mengambang, CardView) yang disarankan.
- Cara menggunakan gambar di aplikasi Anda secara efisien.
- Praktik terbaik yang disarankan untuk mendesain layout yang mudah digunakan menggunakan warna yang berani.

Yang akan Anda LAKUKAN

- Memodifikasi aplikasi untuk mengikuti panduan Desain Material.
- Menambahkan gambar dan gaya pada daftar RecyclerView.
- Mengimplementasikan ItemTouchHelper untuk menambahkan fungsionalitas Seret dan Lepas di aplikasi Anda.

Ringkasan Aplikasi

Aplikasi "Material Me!" adalah aplikasi berita olahraga tiruan dengan implementasi desain yang sangat buruk. Anda akan memperbaikinya agar mematuhi panduan desain untuk membuat pengalaman pengguna yang menyenangkan! Berikut ini beberapa tangkapan layar aplikasi sebelum dan setelah mendapatkan peningkatan Desain Material.



The image shows a mobile application interface titled "Material Me!" with a green header bar. The status bar at the top indicates "4G" signal, battery level, and the time "19:50".

Bowling

News

Here is some Bowling news!

Cycling

News

Here is some Cycling news!

Golf

The bottom navigation bar features three icons: a triangle pointing left, a circle, and a square.

Tugas 1: Mengunduh Kode Starter

Proyek aplikasi starter lengkap untuk praktik ini tersedia di [MaterialMe-Starter](#). Dalam tugas ini, Anda akan memuat proyek ke dalam Android Studio dan menjelajahi beberapa fitur utama aplikasi.

1.1 Buka dan Jalankan Proyek Material Me

1. Unduh dan buka zip [file MaterialMe-Starter](#).
2. Buka aplikasi di Android Studio.
3. Bangun dan jalankan aplikasi.

Aplikasi menampilkan daftar nama olahraga dengan beberapa teks berita placeholder untuk setiap olahraga. Layout dan gaya aplikasi saat ini membuatnya hampir tidak bisa dipakai: setiap baris data tidak dipisahkan dengan jelas dan tidak ada gambar atau warna untuk membuat pengguna tertarik.

1.2 Jelajahi Aplikasi

1. Sebelum membuat modifikasi pada aplikasi, jelajahi strukturnya saat ini. Strukturnya berisi elemen berikut:

- **Sport.java**

Kelas ini mewakili model data untuk setiap baris data di RecyclerView. Sekarang, ini berisi sebuah bidang untuk judul olahraga dan sebuah bidang untuk beberapa informasi tentang olahraga.

- **SportsAdapter.java**

Ini adalah adaptor untuk RecyclerView. Ini menggunakan ArrayList objek olahraga sebagai datanya dan mengisi setiap baris dengan data ini.

- **MainActivity.java**

MainActivity melakukan inisialisasi RecyclerView dan adaptor, dan membuat data dari file sumber daya.

- **strings.xml**

File sumber daya ini berisi semua data untuk aplikasi, termasuk judul dan informasi untuk setiap olahraga.

- **list_item.xml**

File layout ini mendefinisikan setiap baris RecyclerView. Ini terdiri dari tiga TextView, satu untuk setiap bagian data (judul dan info untuk setiap olahraga) dan satu digunakan sebagai label.

Tugas 2: Menambahkan CardView dan Gambar

Salah satu prinsip dasar Material Design adalah pengguna gambar yang berani untuk meningkatkan pengalaman pengguna. Menambahkan gambar ke item daftar RecyclerView adalah awal yang baik untuk membuat pengalaman pengguna yang dinamis dan memukau.

Saat menyajikan informasi yang memiliki media campuran (seperti gambar dan teks), panduan Desain Material menyarankan penggunaan **CardView**, yang merupakan **FrameLayout** dengan beberapa fitur ekstra (seperti elevasi dan sudut tumpul) yang memberikan tampilan dan nuansa yang konsisten di berbagai aplikasi dan platform. CardView adalah komponen UI yang ditemukan di Android Support Library.

Dalam bagian ini, Anda akan memindahkan setiap item daftar ke dalam CardView dan menambahkan Gambar untuk membuat aplikasi mematuhi panduan Material.

2.1 Tambahkan CardView

CardView tidak disertakan di Android SDK default, jadi Anda harus menambahkannya sebagai dependensi build.gradle. Lakukan yang berikut:

1. Di file build.gradle tingkat aplikasi Anda, tambahkan baris berikut ke blok dependensi:

```
compile 'com.android.support:cardview-v7:24.1.1'
```

Catatan: Versi pustaka dukungan mungkin telah berubah sejak penulisan praktik ini. Perbarui ke versi paling baru dan sinkronkan file gradle Anda.

2. Di file list_item.xml, kurung LinearLayout root dengan CardView dengan atribut berikut:

| Atribut | Nilai |
|-----------------------|----------------|
| android:layout_width | "match_parent" |
| android:layout_height | "wrap_content" |
| android:layout_margin | "8dp" |

Catatan: Anda perlu memindahkan deklarasi skema (`xmlns:android="http://schemas.android.com/apk/res/android"`) dari LinearLayout ke CardView yang sekarang menjadi View tingkat tinggi file layout Anda.

3. Jalankan aplikasi. Sekarang setiap item baris ditampung di dalam CardView, yang dielevasikan di atas layer bawah dan memunculkan bayangan.

2.2 Unduh gambar

CardView tidak dimaksudkan untuk dipakai secara eksklusif dengan teks biasa: CardView paling baik untuk menampilkan campuran konten. Anda bisa membuat aplikasi informasi olahraga ini lebih menarik dengan menambahkan gambar spanduk ke setiap baris!

Menggunakan gambar sangat berat untuk aplikasi Anda: framework Android harus memuat seluruh gambar ke dalam memori dengan resolusi penuh, meskipun aplikasi hanya menampilkan gambar kecil.

Dalam bagian ini, Anda akan mempelajari cara menggunakan pustaka Glide untuk memuat gambar berukuran besar secara efisien tanpa menghabiskan sumber daya atau bahkan membuat aplikasi aplikasi Anda crash karena pengecualian 'Out of Memory'.

1. Unduh [file zip gambar spanduk](#).
2. Salin file ini ke direktori res > drawable di aplikasi Anda.

Catatan: Salin file menggunakan file explorer, bukan Android Studio. Buka direktori tempat semua Proyek Android Anda disimpan (yaitu /AndroidStudioProjects) dan masuk ke MaterialMe/app/src/main/res/drawable dan tempelkan gambar di sini.

Anda akan memerlukan laris dengan jalur ke setiap gambar sehingga Anda bisa menyertakannya di dalam objek java Sports. Untuk melakukannya:

3. Definisikan laris yang berisi semua jalur ke drawable sebagai item di file string.xml Anda. Pastikan laris dalam urutan yang sama dengan laris judul olahraga:

```
<array name="sports_images">
    <item>@drawable/img_baseball</item>
    <item>@drawable/img_badminton</item>
    <item>@drawable/img_basketball</item>
    <item>@drawable/img_bowling</item>
    <item>@drawable/img_cycling</item>
    <item>@drawable/img_golf</item>
    <item>@drawable/img_running</item>
    <item>@drawable/img_soccer</item>
    <item>@drawable/img_swimming</item>
    <item>@drawable/img_tabletennis</item>
    <item>@drawable/img_tennis</item>
</array>
```

2.3 Modifikasi objek Sport

Objek Sport.java perlu menyertakan sumber daya yang dapat digambar yang berkaitan dengan olahraga. Untuk mencapai ini:

1. Tambahkan variabel anggota integer ke objek Sport yang akan berisi sumber daya yang dapat digambar:

```
private final int imageResource;
```

2. Modifikasi konstruktor sehingga akan mengambil sebuah integer sebagai parameter dan menetapkannya ke variabel anggota:

```
public Sport(String title, String info, int imageResource) {
    this.title = title;
    this.info = info;
    this.imageResource = imageResource;
}
```

3. Buat getter untuk integer sumber daya:

```
public int getImageResource() {
    return imageResource;
}
```

2.4 Perbaiki metode initializeData()

Di MainActivity.java, metode `initializeData()` sekarang rusak, karena konstruktor untuk objek Sport meminta sumber daya gambar sebagai parameter ketiga.

Struktur data yang mudah adalah menggunakan `TypedArray`. `TypedArray` memungkinkan menyimpan larik sumber daya XML lainnya. Dengan menggunakan `TypedArray`, Anda akan bisa memperoleh sumber daya gambar, serta judul dan informasi olahraga dengan menggunakan pengindeksan di loop yang sama.

1. Dalam metode `initializeData()`, dapatkan `TypedArray` id sumber daya dengan memanggil

```
getResources().obtainTypedArray() , dan meneruskan nama larik sumber daya yang dapat digambar, yang Anda definisikan di file strings.xml:
```

```
TypedArray sportsImageResources =
    getResources().obtainTypedArray(R.array.sports_images);
```

Anda bisa mengakses elemen pada indeks i di `TypedArray` dengan menggunakan metode "get" yang sesuai, bergantung pada tipe sumber daya di larik. Dalam kasus spesifik ini, ini berisi ID sumber daya, sehingga Anda bisa menggunakan metode `getResourceId()`.

2. Perbaiki kode di loop yang membuat objek Sport, dengan menambahkan ID sumber daya yang dapat digambar yang sesuai sebagai parameter ketiga dengan memanggil `getResourceId()` di `TypedArray`:

```
for(int i=0;i<sportsList.length;i++){
    mSportsData.add(new Sport(sportsList[i],sportsInfo[i],
        sportsImageResources.getResourceId(i,0)));
}
```

3. Bersihkan data di `TypedArray` setelah Anda membuat `ArrayList` data Sport:

```
sportsImageResources.recycle();
```

2.5 Tambahkan ImageView ke item daftar

1. Ubah `LinearLayout` di dalam file `list_item.xml` menjadi `RelativeLayout`, dan hapus atribut orientasi.
2. Tambahkan `ImageView` dengan atribut berikut:

| Atribut | Nilai |
|--------------------------|--------------------|
| android:layout_width | "match_parent" |
| android:layout_height | "wrap_content" |
| android:id | "@+id/sportsImage" |
| android:adjustViewBounds | "true" |

Atribut `adjustViewBounds` membuat ImageView menyesuaikan batasnya untuk mempertahankan rasio aspek gambar.

3. Tambahkan atribut berikut ke TextView yang ada:

| TextView id: title | Atribut | Nilai |
|------------------------|----------------------------|--------------------------------------|
| | android:layout_alignBottom | "@+id/sportsImage" |
| | android:theme | "@style/ThemeOverlay.AppCompat.Dark" |
| TextView id: newsTitle | Atribut | Nilai |
| | android:layout_below | "@+id/sportsImage" |
| | android:textColor | "?android:textColorSecondary" |
| TextView id: subTitle | android:layout_below | "@+id/newsTitle" |

Catatan: Tanda tanya di atribut textColor di atas ("?android:textColorSecondary") artinya adalah framework akan menerapkan nilai dari tema yang saat ini diterapkan. Dalam hal ini, atribut ini diwarisi dari tema "Theme.AppCompat.Light.DarkActionBar" yang mendefinisikannya sebagai warna abu-abu muda, yang sering digunakan untuk subjudul.

2.6 Muat gambar menggunakan Glide

Setelah mengunduh gambar dan menyiapkan ImageView, langkah berikutnya adalah memodifikasi SportsAdapter untuk memuat gambar ke dalam ImageView di `onBindViewHolder()`. Jika Anda mengambil pendekatan ini, Anda akan menemukan bahwa aplikasi Anda crash karena kesalahan "Out of Memory". Framework Android harus terus-menerus memuat gambar ke dalam memori dengan resolusi penuh, apa pun ukuran layar ImageView.

Ada sejumlah cara untuk mengurangi konsumsi memori saat memuat gambar, tetapi salah satu pendekatan yang paling mudah adalah menggunakan Image Loading Library seperti [Glide](#), yang akan Anda lakukan di langkah ini. Glide menggunakan pemrosesan latar belakang, serta beberapa pemrosesan kompleks lain, untuk mengurangi kebutuhan memori pemuatan gambar. Ini juga menyertakan beberapa fitur yang berguna, seperti menampilkan gambar placeholder saat gambar yang diinginkan sedang dimuat.

Catatan: Anda bisa mengetahui selengkapnya tentang mengurangi konsumsi memori di aplikasi Anda di [panduan Menampilkan Bitmap](#).

1. Tambahkan dependensi berikut untuk Glide, di file build.gradle tingkat aplikasi Anda:

```
compile 'com.github.bumptech.glide:glide:3.5.2'
```

2. Tambahkan variabel di kelas SportsAdapter, kelas ViewHolder untuk ImageView, dan lakukan inisialisasi di konstruktur ViewHolder:

```
mSportsImage = (ImageView) itemView.findViewById(R.id.sportsImage);
```

3. Tambahkan baris kode berikut ke `onBindViewHolder()` untuk mendapatkan sumber daya gambar dari objek Sport dan muat ke dalam ImageView menggunakan Glide:

```
Glide.with(mContext).load(currentSport.getImageResource()).into(holder.mSportsImage);
```

Beginilah cara memuat gambar dengan Glide. Glide juga memiliki beberapa fitur tambahan yang memungkinkan Anda

- mengubah ukuran, mentransformasi, dan memuat gambar dengan berbagai cara. Kunjungi [laman GitHub Glide](#) untuk mengetahui selengkapnya.
- Jalankan aplikasi, item daftar Anda sekarang seharusnya memiliki grafik yang tegas yang membuat pengalaman pengguna dinamis dan menyenangkan!

Tugas 3: Membuat CardView Anda bisa digesek, dipindahkan, dan diklik

Saat melihat kartu Anda di aplikasi, pengguna memiliki harapan bagaimana kartu akan berperilaku. [Panduan Desain Material](#) mengatakan demikian:

- Sebuah kartu telah ditutup biasanya dengan menggeseknya.
- Sebuah daftar kartu bisa diubah urutannya dengan menahan dan menyeret kartu.
- Mengetuk kartu akan memberikan detail lebih lanjut.

Anda sekarang akan mengimplementasikan sumber daya ini di aplikasi Anda.

3.1 Implementasikan gesek untuk menutup

Android SDK menyertakan sebuah kelas bernama `ItemTouchHelper` yang digunakan untuk mendefinisikan apa yang terjadi ke item daftar RecyclerView saat pengguna melakukan berbagai tindakan sentuh, seperti menggesek atau menyeret dan melepas. Beberapa kasus penggunaan umum telah diimplementasikan dalam serangkaian metode dalam `ItemTouchHelper.SimpleCallback`.

`ItemTouchHelper.SimpleCallback` memungkinkan Anda mendefinisikan arah mana yang didukung untuk menggesek dan memindahkan item daftar dan mengimplementasikan perilaku penggesekan dan pemindahan.

Lakukan yang berikut:

- Buat objek `ItemTouchHelper` baru, di metode `onCreate()` `MainActivity.java`. Untuk argumennya, buat instance baru `ItemTouchHelper.SimpleCallback` dan tekan **Enter** untuk membuat Android Studio mengisi metode yang diperlukan: `onMove()` dan `onSwiped()`.

Catatan: Jika metode yang diperlukan tidak otomatis ditambahkan, klik bola lampu merah dan pilih **Implement methods**.

- Konstruktur `SimpleCallback` akan diberi garis bawah merah karena Anda belum menyediakan parameter yang diperlukan: arah yang Anda rencanakan untuk mendukung pemindahan dan penggesekan item daftar.

Karena kita hanya mengimplementasikan gesek untuk menutup sekarang, Anda harus meneruskan `0` untuk arah pemindahan yang didukung dan `ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT` untuk arah penggesekan yang didukung:

```
ItemTouchHelper helper = new ItemTouchHelper(new ItemTouchHelper
    .SimpleCallback(0, ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT) {})
```

- Sekarang Anda harus mengimplementasikan perilaku yang diinginkan di `onSwiped()`. Dalam hal ini, menggesek kartu ke kiri atau ke kanan akan menghapusnya dari daftar. Panggil `remove()` di rangkaian data, dengan meneruskan indeks yang sesuai dengan mendapatkan posisi dari `ViewHolder`:

```
mSportsData.remove(viewHolder.getAdapterPosition());
```

- Agar RecyclerView dapat menggerakkan penghapusan dengan benar, Anda juga harus memanggil `notifyItemRemoved()`, lagi dengan meneruskan indeks yang sesuai dengan mendapatkan posisi dari `ViewHolder`:

```
mAdapter.notifyItemRemoved(viewHolder.getAdapterPosition());
```

- Setelah membuat objek `ItemTouchHelper` baru di metode `onCreate()` `MainActivity`, panggil `attachToRecyclerView()` di

instance ItemTouchHelper untuk menambahkannya ke RecyclerView Anda:

```
helper.attachToRecyclerView(mRecyclerView);
```

- Jalankan aplikasi Anda, Anda sekarang bisa menggesek item daftar ke kiri dan ke kanan untuk menghapusnya!

3.2 Implementasikan seret dan lepas

Anda juga bisa mengimplementasikan fungsionalitas seret dan lepas menggunakan SimpleCallback yang sama. Argumen SimpleCallback pertama menentukan arah mana yang didukung oleh ItemTouchHelper untuk memindahkan objek.

Lakukan yang berikut:

- Ubah argumen pertama SimpleCallback dari 0 untuk menyertakan setiap arah, karena kita ingin bisa melakukan seret dan lepas di mana pun:

```
ItemTouchHelper helper = new ItemTouchHelper(new ItemTouchHelper.SimpleCallback(ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT | ItemTouchHelper.DOWN | ItemTouchHelper.UP, ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT) {}
```

- Dalam metode `onMove()`, dapatkan indeks asli dan target dari argumen kedua dan ketiga yang diteruskan (sesuai dengan ViewHolder asli dan target).

```
int from = viewHolder.getAdapterPosition();
int to = target.getAdapterPosition();
```

- Pindahkan item di kumpulan data dengan memanggil `collections.swap()` dan meneruskan kumpulan data dan indeks awal dan akhir:

```
Collections.swap(mSportsData, from, to);
```

- Beri tahu adaptor bahwa item sudah dipindahkan, dengan meneruskan indeks lama dan baru:

```
mAdapter.notifyItemMoved(from, to);
```

- Jalankan aplikasi Anda. Sekarang Anda bisa menghapus item daftar Anda dengan menggeseknya ke kira atau ke kanan atau mengubah urutannya dengan menekan lama untuk mengaktifkan mode Seret dan Lepas.

3.3 Implementasikan tampilan detail

Menurut [panduan Desain Material](#), kartu digunakan untuk menyediakan titik masuk ke informasi yang lebih detail. Anda mungkin akan perlu mengetuk kartu untuk melihat informasi selengkapnya tentang olahraga, karena begitulah ekspektasi perilaku kartu. Di bagian ini, Anda akan menambahkan aktivitas detail yang akan diluncurkan saat item daftar apa pun ditekan. Untuk praktik ini, aktivitas detail akan berisi nama dan gambar item daftar yang Anda klik, tetapi hanya akan berisi teks detail placeholder generik, sehingga Anda tidak harus membuat detail khusus untuk setiap item daftar.

- Buat aktivitas baru dengan masuk ke **File > New > Activity > Empty Activity**.
- Beri nama DetailActivity, dan terima semua default.
- Di file layout yang baru dibuat, buang padding dari RelativeLayout rootview.
- Salin semua dari tampilan TextView dan ImageView dari file `list_item.xml` ke file `activity_detail.xml`.
- Tambahkan kata "Detail" ke setiap referensi ke sebuah id, untuk membedakannya dengan id `list_item`. Misalnya, ubah id ImageView dari `sportsImage` ke `sportsImageDetail`, serta semua referensi ke id ini untuk penepatan relatif, seperti `layout_below`.
- Untuk textView `subTitleDetail`, buang semua string teks placeholder dan tempelkan paragraf teks generik untuk menggantikan semua teks detail (Misalnya, beberapa paragraf [Lorem Ipsum](#)).
- Ubah padding TextView menjadi 16dp.

8. Bungkus keseluruhan activity_detail.xml di ScrollView dan ubah atribut layout_height RelativeLayout ke "wrap_content".

Catatan: Atribut untuk ScrollView mungkin berwarna merah terlebih dulu. Ini karena Anda harus menambahkan atribut yang mendefinisikan namespace Android. Ini adalah atribut yang muncul di semua file layout Anda secara default:

```
xmlns:android="http://schemas.android.com/apk/res/android" .
```

Cukup pindahkan deklarasi ini ke tampilan tingkat atas dan warna merah pun akan hilang.

9. Dalam kelas SportsAdapter, buat kelas dalam ViewHolder agar mengimplementasikan View.OnClickListener dan implementasikan metode yang diperlukan (`onClick()`).
10. Setel OnClickListener ke itemview di konstruktur:

```
itemView.setOnClickListener(this);
```

11. Dalam metode `onClick()`, dapatkan objek Sport untuk item yang diklik menggunakan `getAdapterPosition()`.

```
Sport currentSport = mSportsData.get(getAdapterPosition());
```

12. Buat sebuah Intent yang meluncurkan aktivitas Detail, dan letakkan judul dan sumber daya gambar sebagai ekstra di Intent:

```
Intent detailIntent = new Intent(mContext, DetailActivity.class);
detailIntent.putExtra("title", currentSport.getTitle());
detailIntent.putExtra("image_resource", currentSport.getImageResource());
```

13. Panggil `startActivity()` di variabel mContext, dengan meneruskan Intent baru.

14. Dalam DetailActivity.java, lakukan inisialisasi ImageView dan TextView judul di `onCreate()`:

```
TextView sportsTitle = (TextView) findViewById(R.id.titleDetail);
ImageView sportsImage = (ImageView) findViewById(R.id.sportsImageDetail);
```

15. Dapatkan judul dari Intent yang masuk dan setel ke TextView:

```
sportsTitle.setText(getIntent().getStringExtra("title"));
```

16. Gunakan Glide untuk memuat gambar ke dalam ImageView:

```
Glide.with(this).load(getIntent().getIntExtra("image_resource", 0))
    .into(sportsImage);
```

17. Jalankan aplikasi. Mengetuk item daftar sekarang meluncurkan aktivitas detail.

Tugas 4: Tambahkan FAB dan pilih Palet Warna Material

Salah satu prinsip Material Design adalah menggunakan elemen yang konsisten di semua aplikasi dan platform sehingga pengguna mengenali pola dan mengetahui cara menggunakannya. Anda telah menggunakan salah satu elemen tersebut: [Tombol Aksi Mengambang](#). FAB adalah tombol bulat yang mengambang di atas UI. Ini digunakan untuk mempromosikan tindakan tertentu kepada pengguna, tindakan yang paling mungkin digunakan pada aktivitas tertentu. Dalam tugas ini, Anda akan membuat FAB yang menyetel ulang kumpulan data ke status aslinya.

4.1 Tambahkan FAB

Tombol Aksi Mengambang adalah bagian dari [Pustaka Dukungan Desain](#).

1. Tambahkan baris kode berikut ke file build.gradle tingkat aplikasi untuk menambahkan dependensi pustaka dukungan desain:

```
compile 'com.android.support:design:24.2.1'
```

2. Gunakan studio aset vektor untuk mengunduh ikon untuk digunakan di FAB. Tombol akan menyetel ulang konten di

RecyclerView sehingga ikon ini akan melakukan:  Mengubah nama ke ic_reset.

3. Di activity_main.xml, tambahkan tampilkan Tombol Aksi Mengambang dengan parameter berikut:

| Atribut | Nilai |
|----------------------------------|----------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:layout_alignParentBottom | "true" |
| android:layout_alignParentRight | "true" |
| android:layout_margin | "16dp" |
| android:src | "@drawable/ic_reset" |
| android:onClick | resetSports |

4. Definisikan metode `resetSports()` di MainActivity.java agar cukup memanggil `initializeData()` untuk menyetel ulang data.

5. Jalankan aplikasi. Anda sekarang bisa menyetel ulang data dengan menekan FAB.

Catatan: Karena aktivitas dihapus dan dibuat ulang saat konfigurasi berubah, memutar perangkat akan menyetel ulang data pada implementasi ini. Agar perubahan menjadi persisten (seperti kasus mengubah urutan dan membuang data), Anda harus mengimplementasikan `onSaveInstanceState()` atau menuliskan perubahan ke sumber yang persisten (seperti database atau SharedPreferences).

4.2 Pilih Palet Material

Jika Anda menjalankan aplikasi, Anda mungkin memperhatikan bahwa FAB memiliki warna yang tidak Anda definisikan di mana pun. Selain itu, bilah aplikasi (bilah yang berisi judul aplikasi Anda) memiliki warna yang tidak Anda setel secara eksplisit. Di mana warna ini didefinisikan?

1. Buka file styles.xml Anda (yang ada di direktori values). Gaya AppTheme mendefinisikan tiga warna secara default: colorPrimary, colorPrimaryDark, dan colorAccent. Gaya ini didefinisikan oleh nilai dari file colors.xml. Desain Material menyarankan agar memilih setidaknya tiga warna ini untuk aplikasi Anda:
2. Warna primer. Ini otomatis digunakan untuk mewarnai bilah aplikasi Anda.
3. Warna gelap primer. Warna lebih gelap dari warna yang sama. Ini digunakan untuk bilah status di atas bilah aplikasi, di antara yang lain.
4. Warna aksen. Warna yang cukup kontras dengan warna primer. Ini digunakan untuk beragam sorotan, tetapi juga menjadi warna default FAB. Anda bisa menggunakan [Panduan Warna Material](#) untuk memilih beberapa warna untuk bereksperimen.
5. Ambil warna dari panduan untuk digunakan sebagai warna utama Anda. Warna harus dalam rentang 300-700 sehingga Anda masih bisa memilih warna aksen dan gelap yang sesuai. Modifikasi nilai hex colorPrimary di file colors.xml Anda agar sesuai dengan warna yang Anda pilih.
6. Pilih tingkat warna yang lebih gelap dari warna yang sama untuk digunakan sebagai warna gelap primer. Modifikasi warna hex colors.xml lagi agar sesuai dengan colorPrimaryDark.
7. Pilih warna aksen untuk FAB Anda dari warna yang nilainya dimulai dengan A dan yang warnanya cukup kontras dengan warna primer (seperti Orange A200). Ubah nilai colorAccent di colors.xml agar menyesuaikan.
8. Tambahkan Tambahkan atribut android:tint ke FAB dan setel agar sama dengan #FFFFFF (putih) untuk mengubah warna ikon ke putih.
9. Jalankan aplikasi. Bilah Aplikasi dan FAB berubah untuk mencerminkan palet warna baru!

Catatan: Jika Anda ingin mengubah warna FAB ke warna yang berbeda dengan warna tema, gunakan atribut `app:backgroundTint`. Perhatikan bahwa ini menggunakan namespace `app`: dan Android Studio akan memperingatkan Anda untuk menambahkan pernyataan untuk mendefinisikan namespace.

Kode solusi

Proyek Android Studio: [MaterialMe](#)

Tantangan penyusunan kode

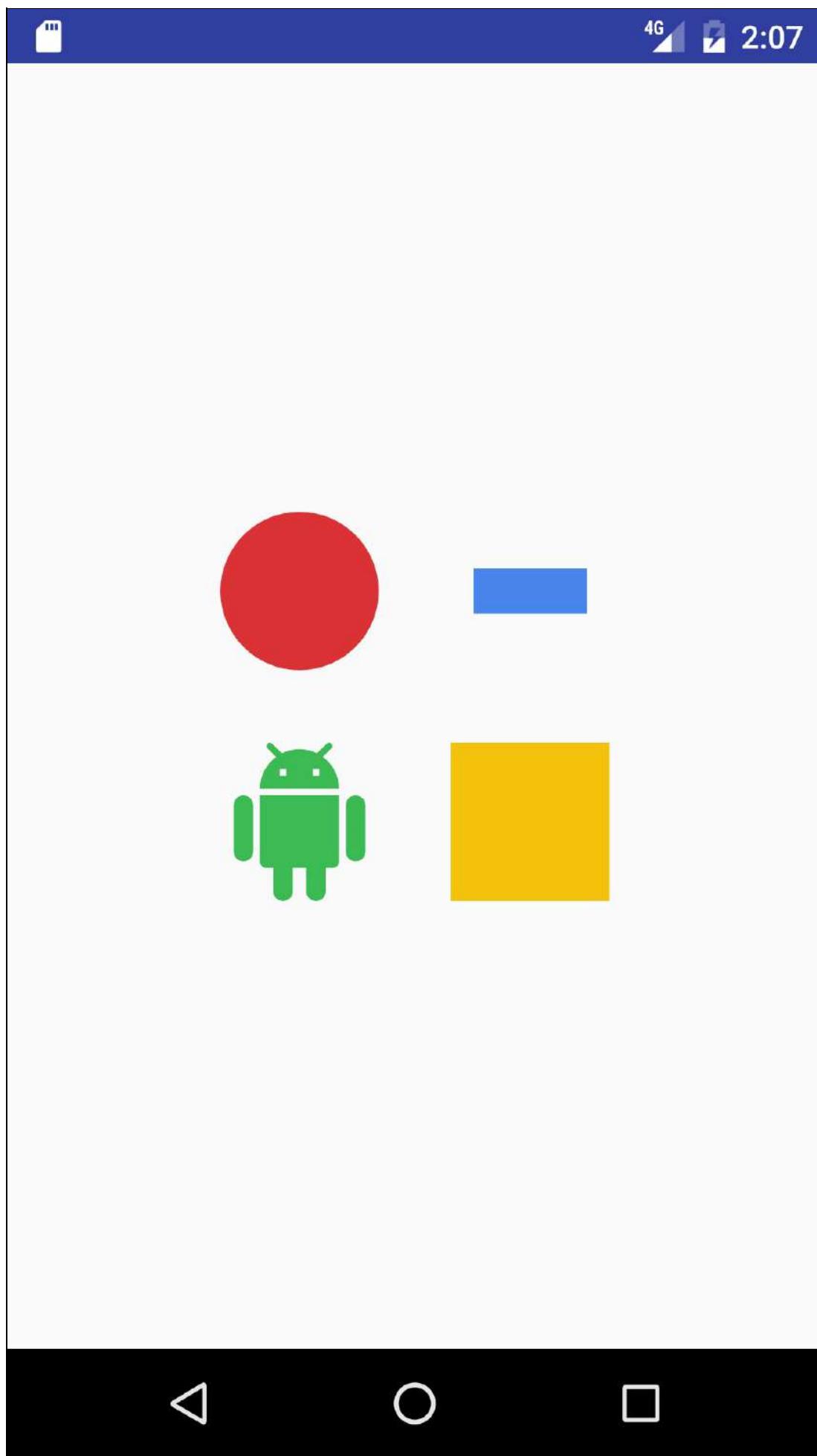
Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan 1: Tantangan ini terdiri dari beberapa peningkatan kecil pada aplikasi Anda:

- Tambahkan detailnya nyata pada objek Sport dan teruskan detailnya ke tampilan detail.
- Implementasikan cara untuk memastikan bahwa status aplikasi bersifat persisten di semua perubahan orientasi.

Tantangan 2: Buat aplikasi dengan 4 gambar yang diatur dalam grid di tengah layout Anda. Buat tiga latar belakang berwarna solid pertama dengan bentuk yang berbeda (kotak, lingkaran, dan garis) dan keempatnya adalah [Ikon Desain Material Android](#). Buat setiap gambar tersebut merespons klik seperti berikut:

1. Salah satu blok yang berwarna menjalankan Aktivitas menggunakan animasi [Explode](#) untuk transisi masuk dan keluar.
2. Buka ulang Aktivitas dari blok berwarna lain, kali ini menggunakan transisi [Fade](#).
3. Menyentuh blok berwarna ketiga memulai animasi tampilan dalam tempat (misalnya rotasi).
4. Akhirnya, menyentuh ikon Android akan memulai aktivitas kedua dengan Transisi Shared Element yang menukar blok Android dengan salah satu blok lain.



Catatan: Anda harus menyetel tingkat SDK minimum Anda ke 21 atau lebih tinggi agar bisa mengimplementasikan transisi elemen bersama.

Rangkuman

- CardView adalah layout yang bagus untuk menyajikan informasi yang berisi media campuran (misalnya gambar dan teks)
- CardView adalah komponen UI yang ditemukan di Android Support Library
- CardView tidak dirancang hanya untuk View anak teks.
- Memuat gambar langsung ke dalam ImageView sangat memakan banyak memori. Semua gambar dimuat dalam resolusi penuh.
- Gunakan pustaka pemuatan gambar, seperti Glide, untuk memuat gambar ke dalam aplikasi secara efisien.
- Android SDK memiliki kelas yang disebut ItemTouchHelper yang membantu mendapatkan informasi pengetukan, penggesekan, atau seret dan letakkan untuk UI Anda.
- Tombol Aksi Mengambang (FAB) berfokus pada tindakan tertentu dan "mengambang" di UI Anda.
- Desain Material menyarankan 3 warna untuk aplikasi Anda: warna primer, warna gelap primer, dan warna aksen.
- Panduan Desain Material adalah serangkaian prinsip arahan yang bertujuan membuat aplikasi yang konsisten, mudah dipahami, dan menyenangkan.
- Desain Material mempromosikan penggunaan gambar dan warna yang berani untuk meningkatkan pengalaman pengguna.
- Desain Material juga mempromosikan elemen yang konsisten di semua platform (misalnya CardView dan FAB).
- Desain Material harus digunakan untuk gerakan yang bermakna dan intuitif, seperti kartu yang bisa ditutup atau diatur ulang.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Desain Material](#)

Ketahui selengkapnya

- [Pedoman Desain Material](#)
- [Generator Palet Material](#)
- [Panduan Kartu dan Daftar](#)
- [Referensi Tombol Aksi Mengambang](#)
- [Mendefinisikan Animasi Khusus](#)
- [Animasi Tampilan](#)

5.3 Mendukung Lanskap, Beberapa Ukuran Layar, dan Pelokalan

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Mendukung Orientasi Lanskap](#)
- [Tugas 2: Mendukung Tablet](#)
- [Tugas 3: Melokalkan Aplikasi Anda](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Setelah menggunakan aplikasi Material Me! yang telah Anda buat dalam praktik terakhir, Anda mungkin memperhatikan bahwa penggunaan aplikasi tidak optimal saat orientasi perangkat diputar dari mode potret ke mode lanskap. Sama juga, jika Anda menguji pada tablet, ukuran font terlalu kecil dan ruang tidak dipakai secara efisien. Framework Android memiliki cara untuk menyelesaikan kedua masalah ini. Qualifier sumber daya memungkinkan Android Runtime menggunakan file sumber daya alternatif (.xml) bergantung pada konfigurasi perangkat, seperti, orientasi, lokal dan "qualifier" lain. Untuk daftar lengkap qualifier yang tersedia, kunjungi panduan [Menyediakan Sumber Daya](#). Dalam praktik ini Anda akan mengoptimalkan penggunaan ruang di aplikasi olahraga Material sehingga aplikasi Anda bekerja dengan baik dalam mode lanskap, serta di tablet.

Yang harus sudah Anda KETAHUI

Dari bab sebelumnya, Anda harus sudah bisa:

- Menemukan dan mengedit file sumber daya.
- Mengekstrak sumber daya.
- Membuat instance ponsel atau tablet virtual menggunakan emulator.

Yang akan Anda PELAJARI

Dalam praktik ini, Anda akan mempelajari cara:

- Menyediakan sumber daya alternatif untuk mode lanskap.
- Menyediakan sumber daya alternatif untuk tablet.
- Menyediakan sumber daya alternatif untuk lokal yang berbeda.

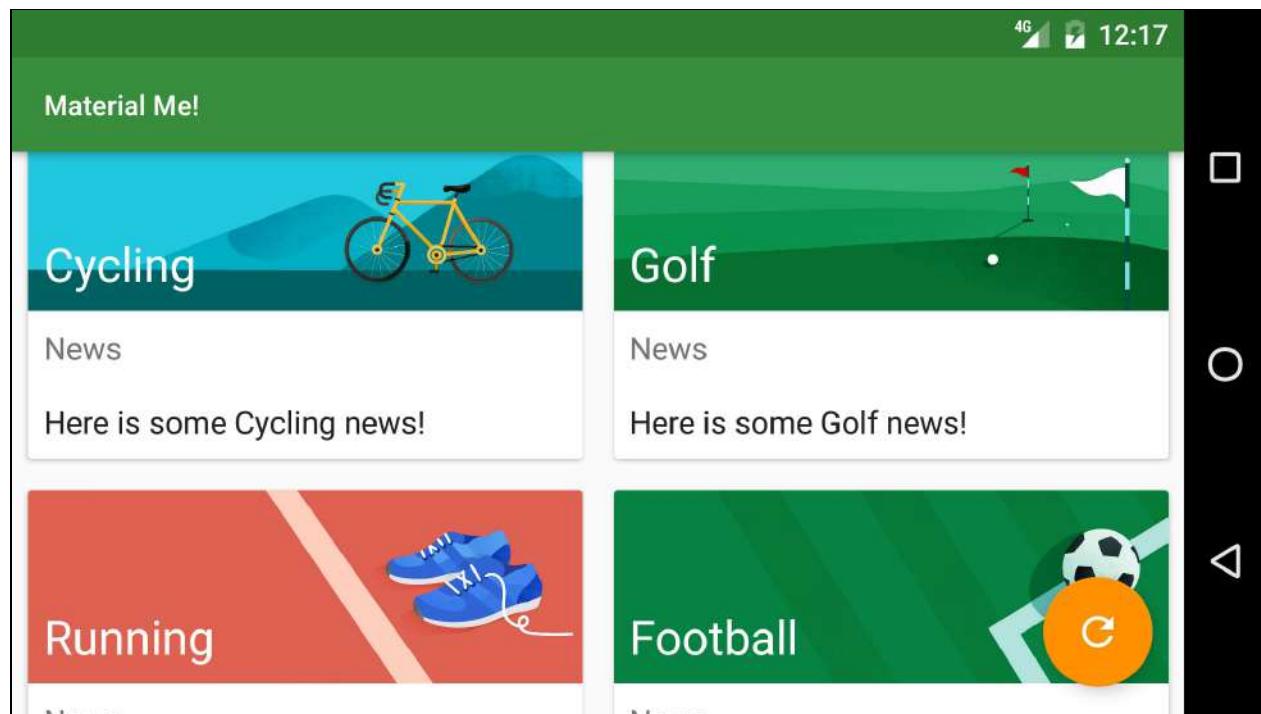
Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Memperbarui aplikasi Material Me! agar menggunakan ruang dengan baik dalam mode lanskap.
- Menambahkan layout alternatif untuk tablet.
- Melokalkan konten aplikasi Anda.

Ringkasan Aplikasi

Aplikasi Material Me! yang ditingkatkan akan menyertakan layout yang disempurnakan saat digunakan dalam mode lanskap, pada tablet dan menawarkan konten yang dilokalkan untuk pengguna di luar AS.



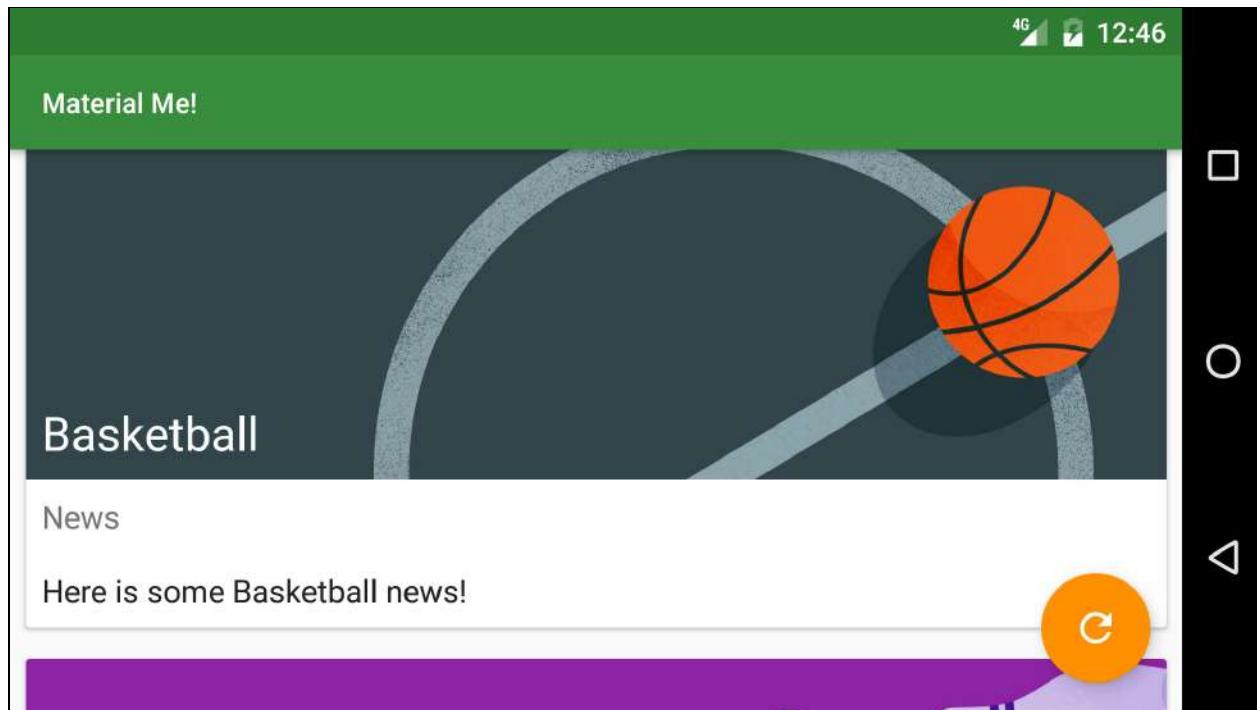


Praktik ini melanjutkan aplikasi "Material Me!" dari praktik sebelumnya.

1. Lanjutkan versi aplikasi "Material Me!" Anda, atau [unduh di sini](#).

Tugas 1: Mendukung Orientasi Lanskap

Anda mungkin ingat bahwa saat pengguna mengubah orientasi perangkat, framework Android akan menghapus dan membuat ulang aktivitas saat ini. Orientasi baru sering kali memiliki persyaratan layout yang berbeda dari yang asli. Misalnya, aplikasi Material Me! tampak bagus dalam mode potret, tetapi tidak mengoptimalkan pengguna layar di mode lanskap. Dengan lebar yang lebih panjang dalam mode lanskap, gambar dalam setiap item daftar melebihi teks, sehingga pengalaman pengguna menurun.



Dalam tugas ini, Anda akan membuat file sumber daya alternatif yang akan mengubah penampilan aplikasi saat digunakan dalam orientasi lanskap.

1.1 Ubah ke GridLayoutManager

Layout yang berisi item daftar sering terlihat tidak seimbang dalam mode lanskap saat item daftar menyertakan gambar dengan lebar penuh. Salah satu solusi yang baik adalah menggunakan grid, bukan daftar linear saat menampilkan CardViews dalam mode lanskap. Ingat bahwa item dalam daftar RecyclerView ditempatkan menggunakan LayoutManager; sampai sekarang, Anda telah menggunakan [LinearLayoutManager](#) yang menata letak setiap item dalam daftar pengguliran vertikal dan horizontal. [GridLayoutManager](#) adalah pengelola layout lain yang menampilkan item dalam grid, bukan dalam daftar. Saat Anda membuat GridLayoutManager, Anda harus memberikan dua parameter: konteks aplikasi dan integer yang menunjukkan jumlah kolom. Anda bisa mengubah jumlah kolom secara terprogram sehingga Anda lebih fleksibel dalam merancang layout responsif. Dalam hal ini, jumlah integer kolom harus 1 dalam orientasi potret (kolom tunggal) dan 2 saat dalam mode lanskap. Perhatikan bahwa jumlah kolom adalah 1, GridLayoutManager berperilaku sama dengan LinearLayoutManager.

- Buat file sumber daya baru yang bernama integers.xml. Masuk ke direktori sumber daya Anda, klik kanan nama direktori nama dan pilih **New > Values resource file**.
- Beri nama file dengan integers.xml dan klik **OK**.
- Buat konstanta integer antara tag `<resources>` yang bernama "grid_column_count" dan setel sama dengan 1:

```
<integer name="grid_column_count">1</integer>
```

- Buat file sumber daya nilai lain, yang bernama integers.xml tetapi dengan karakteristik yang berbeda.

Perhatikan opsi "Available qualifiers" di dialog untuk membuat file sumber daya. Karakteristik ini disebut sebagai "resource qualifiers" dan digunakan untuk melabeli konfigurasi sumber daya untuk beragam situasi.

- Pilih **Orientation** dan tekan simbol **>>** di tengah dialog untuk mengakses qualifier ini.
- Ubah pemilih orientasi Layar ke Landscape dan perhatikan bagaimana nama direktori "values-land" otomatis berubah.

Ini adalah inti qualifier sumber daya: nama direktori memberi tahu Android kapan harus menggunakan file layout spesifik. Dalam hal ini, ini terjadi saat ponsel diputar ke mode lanskap.

7. Klik **OK** untuk membuat file layout baru.
8. salin konstanta integer yang Anda buat ke dalam file sumber daya ini, tetapi ubah nilainya menjadi 2.

Anda sekarang seharusnya memiliki dua file integers.xml individual. Dalam tampilan proyek "Android" di Android Studio, ini seharusnya dikelompokkan ke dalam folder "integers.xml" yang setiap file dalamnya dilabeli dengan qualifier yang Anda pilih ("land" dalam hal ini).

1.2 Modifikasi MainActivity

1. Dalam `onCreate()` di MainActivity, dapatkan integer dari file sumber daya integers.xml:

```
int gridColumnCount = getResources().getInteger(R.integer.grid_column_count);
```

Android Runtime akan memproses penentuan file integers.xml mana yang akan digunakan, bergantung pada status perangkat.

2. Ubah LinearLayoutManager ke GridLayoutManager, dengan meneruskan konteks dan integer yang baru dibuat:

```
mRecyclerView.setLayoutManager(new GridLayoutManager(this, gridColumnCount));
```

3. Jalankan aplikasi dan putar perangkat. Jumlah kolom berubah secara otomatis dengan orientasi perangkat.

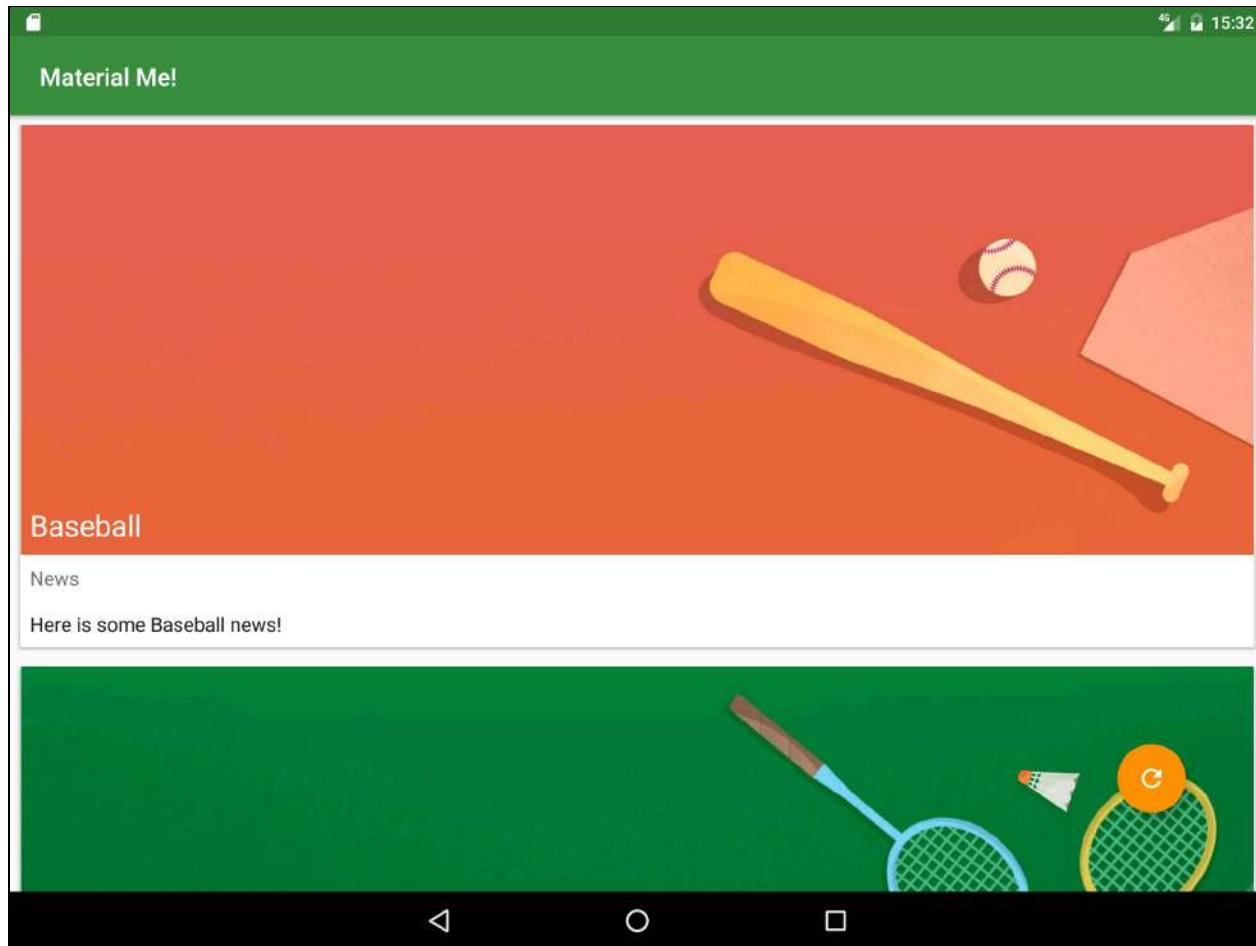
Saat menggunakan aplikasi dalam mode lanskap, Anda akan melihat bahwa fungsionalitas gesek untuk menutup tidak lagi mudah digunakan, karena item sekarang berada dalam grid, bukan dalam daftar. Anda bisa menggunakan variabel `gridColumnCount` untuk menonaktifkan tindakan gesek jika terdapat lebih dari satu kolom:

```
int swipeDirs;
if(gridColumnCount > 1){
    swipeDirs = 0;
} else {
    swipeDirs = ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT;
}
ItemTouchHelper helper = new ItemTouchHelper(new ItemTouchHelper.SimpleCallback
    (ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT | ItemTouchHelper.DOWN
        | ItemTouchHelper.UP, swipeDirs)
```

Tugas 2 : Dukung Tablet

Meskipun Anda telah memodifikasi aplikasi agar terlihat lebih baik dalam mode lanskap, menjalankannya di table yang secara fisik lebih besar menyebabkan semua teks tampak terlalu kecil. Selain itu, saat perangkat berada dalam orientasi lanskap, layar tidak digunakan secara efisien; 3 kolom akan lebih sesuai untuk layar berukuran tablet dalam mode lanskap.

Dalam tugas ini, Anda akan menambahkan qualifier sumber daya tambahan untuk mengubah penampilan aplikasi saat digunakan di tablet.



2.1 Buat Layout Menyesuaikan Tablet

Dalam langkah ini, Anda akan membuat qualifier sumber daya yang berbeda untuk memaksimalkan penggunaan layar untuk perangkat berukuran tablet, dengan meningkatkan jumlah kolom ke 2 untuk orientasi potret dan 3 untuk orientasi lanskap. Qualifier sumber daya yang Anda perlukan bergantung pada persyaratan khusus Anda. Ada beberapa qualifier yang bisa Anda gunakan untuk memilih ketentuan yang benar:

- "smallest width" - Qualifier ini paling sering digunakan untuk memilih tablet. Ini didefinisikan oleh lebar **terkecil** perangkat (apa pun orientasinya), yang menghilangkan ambiguitas saat berurusan dengan "tinggi" dan "lebar" karena beberapa perangkat memang biasanya dipegang dalam mode lanskap, dan lainnya dalam mode potret. Perangkat yang lebar terkecilnya minimal 600dp dianggap sebagai tablet.
- "available width" - Lebar tersedia adalah lebar **efektif** sebuah perangkat, apa pun orientasinya. Lebar yang tersedia berubah bila perangkat diputar, karena tinggi dan lebar efektif perangkat ditukar.
- "available height" - Sama dengan "available width", kecuali ini menggunakan tinggi efektif, sebagai gantinya lebar efektif.

Untuk memulai tugas ini:

1. Buat file integers.xml yang menggunakan qualifier "smallest width" dengan nilai yang disetel ke 600. Android menggunakan file ini kapan pun aplikasi berjalan pada tablet.
2. Salin kode dari file integers.xml dengan qualifier sumber daya lanskap (memiliki jumlah grid 2) dan tempelkan ke file integers.xml yang baru.
3. Buat file integers.xml ketiga yang menyertakan lebar layar terkecil berukuran qualifier 600dp dan qualifier orientasi lanskap. Android menggunakan file ini saat aplikasi berjalan pada tablet dalam mode lanskap.

Catatan: Android akan mencari file sumber daya dengan qualifier sumber daya paling spesifik terlebih dulu, lalu

berpindah ke yang lebih generik. Misalnya, jika nilai didefinisikan di file integers.xml dengan qualifier lebar terkecli dan lanskap, nilai ini akan menggantikan nilai di dalam file integers.xml yang hanya berisi qualifier lanskap. Untuk informasi selengkapnya tentang qualifier sumber daya, kunjungi [Panduan Menyediakan Sumber Daya](#).

4. Ubah variabel grid_column_count ke 3 dalam lanskap, file integers.xml.
5. Buat emulator tablet virtual. Jalankan aplikasi pada emulator tablet serta emulator ponsel dan putar kedua perangkat ke mode lanskap. Dengan file qualifier sumber daya ini, aplikasi menggunakan ruang layar secara jauh lebih efektif.

2.2 Perbarui gaya item daftar tablet

Pada tahap ini, aplikasi Anda mengubah jumlah kolom di GridLayoutManager agar pas dengan orientasi perangkat dan memaksimalkan penggunaan ruang layar. Akan tetapi, semua TextView yang tampak berukuran tepat di layar ponsel sekarang tampak terlalu kecil di layar tablet yang lebih besar. Untuk memperbaiki ini, Anda akan mengekstrak gaya TextAppearance dari file sumber daya layout ke dalam file sumber daya gaya. Anda juga akan membuat file styles.xml tambahan untuk tablet menggunakan qualifier sumber daya.

Catatan: Anda juga bisa membuat file layout alternatif dengan qualifier sumber daya yang sesuai, dan mengubah gaya TextView di dalamnya. Akan tetapi, ini akan memerlukan duplikasi kode lebih banyak, karena sebagian besar informasi layout sama, apa pun perangkat yang Anda gunakan, sehingga Anda hanya akan mengekstrak atribut yang akan berubah.

Buat Gaya

1. Dalam file styles.xml, buat gaya berikut:

| Nama | Induk |
|------------------|-----------------------------------|
| SportsTitle | TextAppearance.AppCompat.Headline |
| SportsDetailText | TextAppearance.AppCompat.Subhead |

Buat file styles.xml untuk tablet

Sekarang Anda akan membuat file tempat Anda mendefinisikan gaya untuk tablet.

1. Buat file sumber daya styles.xml baru yang menggunakan qualifier **Smallest Screen Width** dengan nilai 600.
2. Salin gaya "SportsTitle" dan "SportsDetailText" dari file styles.xml asli ke file styles.xml baru yang berkualifikasi.
3. Ubah induk gaya "SportsTitle" menjadi "TextAppearance.AppCompat.Display1"
4. Gaya `Display1` Android yang telah didefinisikan menggunakan nilai `textColorSecondary` dari tema saat ini (`ThemeOverlay.AppCompat.Dark`) yang dalam hal ini adalah warna abu-abu muda. Warna abu-abu muda tidak tampil dengan baik pada gambar spanduk di aplikasi Anda. Untuk memperbaiki ini, tambahkan atribut `"android:textColor"` ke gaya "SportsTitle" dan setel ke `"?android:textColorPrimary"`.

Catatan: Tanda tanya memberi tahu Android Runtime agar menemukan nilai dalam tema yang diterapkan pada View. Dalam contoh ini, temanya adalah `ThemeOverlay.AppCompat.Dark` yang atribut `textColorPrimary`-nya adalah putih.

5. Ubah induk gaya "SportsDetailText" ke "TextAppearance.AppCompat.Headline".

Perbarui tampilan teks di list_item.xml

1. Kembali ke list_item.xml, ubah atribut gaya TextView "title" ke `"@style/SportsDetailTitle"`
2. Ubah atribut gaya TextView "newsTitle" dan "subTitle" ke `"@style/SportsDetailText"`.
3. Jalankan aplikasi Anda. Setiap item daftar sekarang memiliki ukuran teks lebih besar pada tablet.

2.3 Perbarui gaya detail olahraga tablet

Anda sekarang telah memperbaiki tampilan untuk MainActivity, yang mencantumkan semua CardViews Sports. DetailActivity masih memiliki ukuran font yang sama pada tablet dan ponsel.

1. Buat gaya berikut dalam file styles.xml:

| | |
|------------------------------------------|-----------------------------------|
| styles.xml (sw600dp) | |
| Nama | Induk |
| SportsDetailTitle | TextAppearance.AppCompat.Display3 |
| styles.xml (tidak berkualifikasi) | |
| Nama | Induk |
| SportsDetailTitle | TextAppearance.AppCompat.Headline |

2. Ubah gaya TextView "newsTitleDetail" dan "subTitleDetail" dalam file layout activity_detail.xml menjadi gaya "SportsDetailText" yang telah Anda buat di langkah sebelumnya.
3. Jalankan aplikasi Anda. Semua teks kini lebih besar di tablet, yang sangat meningkatkan pengalaman pengguna aplikasi Anda.

Tugas 3: Melokalkan Aplikasi Anda

"Lokal" mewakili region geografi, politik, atau budaya tertentu di dunia. Qualifier sumber daya bisa digunakan untuk menyediakan sumber daya alternatif berdasarkan lokal pengguna. Sama seperti orientasi dan lebar layar, Android menyediakan kemampuan untuk menyertakan file sumber daya terpisah untuk lokal yang berbeda. Dalam langkah ini, Anda akan memodifikasi file strings.xml agar menjadi lebih internasional.

3.1 Tambahkan file strings.xml

Anda mungkin sudah memperhatikan bahwa informasi olahraga yang berada di aplikasi ini dirancang bagi pengguna di US. Aplikasi menggunakan istilah "soccer" untuk menunjukkan sebuah olahraga yang dikenal sebagai "football" di seluruh bagian dunia yang lain. Untuk membuat aplikasi Anda lebih internasional, Anda bisa menyediakan file strings.xml dengan lokal khusus untuk pengguna US yang menggunakan "soccer", sedangkan semua lokal lain akan menggunakan "football".

1. Buat file sumber daya nilai baru.
2. Panggil file strings.xml dan pilih **Locale** dari daftar qualifier yang tersedia.
3. Di bagian "Language", pilih **en: English**.
4. Di bagian "Specific Region Only:", pilih **US: United States**. Ini akan membuat direktori nilai khusus untuk lokal US, yang bernama "values-en-rUS".
5. Salin keseluruhan file strings.xml generik ke file strings.xml baru dengan lokal khusus.
6. Dalam file strings.xml generik, ubah item "Soccer" di larik sports_titles menjadi "Football", serta ubah item terkait di larik "sports_info".
7. Jalankan aplikasi. Bergantung pada pengaturan bahasa di perangkat Anda, Anda akan melihat "Soccer" atau "Football".

Catatan: Untuk mengubah setelan lokal di perangkat Anda, masuk ke setelan perangkat, lalu pilih **Language & input** dan ubah setelan **Language**. Jika Anda memilih **English (United States)**, aplikasi akan memiliki "Soccer" sebagai string, dan jika tidak akan menampilkan "Football".

Kode solusi

Proyek Android Studio: [MaterialMe-Resource](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan 1: Ternyata beberapa negara selain US menggunakan "soccer" bukannya "football". Cari tahu negara mana saja ini dan tambahkan sumber daya string yang dilokalkan untuk negara tersebut.

Tantangan 2: Gunakan teknik pelokalan yang Anda pelajari di Tugas 3 yang dikombinasikan dengan Google Translate untuk menerjemahkan semua string di aplikasi Anda ke bahasa yang berbeda.

Rangkuman

- GridLayoutManager adalah pengelola layout yang menangani daftar pengguliran 2 dimensi.
- Anda bisa mengubah jumlah kolom dalam GridLayoutManager secara dinamis.
- Android Runtime menggunakan file konfigurasi alternatif untuk beragam lingkungan waktu proses, seperti layout perangkat, dimensi layar, lokal, negara, keyboard, dsb.
- Sumber daya alternatif ditempatkan dalam file yang bernama qualifier sumber daya sebagai bagian dari namanya.
- Format untuk direktori ini adalah `<resources_name>-<qualifier>`.
- File apa pun di direktori "res" Anda bisa dikualifikasi dengan cara ini.
- Beberapa qualifier yang umum adalah:
 - orientation: land, portrait
 - smallest width: sw600dp
 - locale: en-rGB, fr
 - screen density: ldpi, mpdi, xhdpi, xxhdpi, xxxhdpi
 - mobile country code: mcc310 (US), mcc208 (France)
 - dan lebih banyak lagi!

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Menyediakan Sumber Daya untuk Layout Adaptif](#)

Ketahui selengkapnya

Dokumentasi Developer

- [Mendukung Beberapa Layar](#)
- [Menyediakan Sumber Daya](#)

6.1: Menggunakan Espresso untuk menguji UI Anda

Konten:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Siapkan Espresso di proyek Anda](#)
- [Tugas 2: Lakukan pengujian pengalihan aplikasi dan memasukkan teks](#)
- [Tugas 3: Uji tampilan pilihan spinner](#)
- [Tugas 4: Rekam pengujian RecyclerView](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Saat Anda, sebagai seorang developer, menguji interaksi pengguna di dalam aplikasi Anda, tindakan ini membantu memastikan pengguna aplikasi Anda tidak mengalami hasil yang tidak diharapkan atau memiliki pengalaman yang tidak baik saat berinteraksi dengan aplikasi Anda.

Anda bisa menguji antarmuka pengguna untuk aplikasi yang kompleks secara manual dengan menjalankan aplikasi dan mencoba antarmuka pengguna. Tetapi Anda tidak mungkin mencakup semua permutasi interaksi pengguna dan semua fungsionalitas aplikasi. Anda juga harus mengulang pengujian manual ini pada berbagai konfigurasi perangkat yang berbeda di emulator dan pada berbagai perangkat keras yang berbeda.

Jika Anda mengotomatiskan pengujian interaksi UI, Anda membebaskan waktu untuk melakukan pekerjaan lain. Anda bisa menggunakan serangkaian pengujian otomatis untuk melakukan semua interaksi UI secara otomatis yang memudahkan menjalankan pengujian untuk konfigurasi perangkat yang berbeda. Sebaiknya biasakan untuk membuat pengujian antarmuka pengguna (UI) seiring Anda menyusun kode untuk memverifikasi apakah UI aplikasi Anda berfungsi dengan benar.

Espresso adalah framework pengujian untuk Android yang memudahkan penulisan pengujian antarmuka pengguna (UI) yang andal untuk aplikasi Anda. Framework, yang merupakan bagian dari Android Support Repository, menyediakan API untuk menulis pengujian UI untuk menyimulasikan interaksi pengguna di dalam aplikasi—semua dari mengeklik tombol dan menavigasi tampilan sampai memilih pilihan menu dan memasukkan data.

Yang harus sudah Anda KETAHUI

Anda harus sudah bisa:

- Membuat dan menjalankan aplikasi dalam Android Studio.
- Membuat dan mengedit elemen UI menggunakan Layout Editor, memasukkan kode XML secara langsung, dan mengakses elemen UI dari kode Java Anda.
- Menambahkan fungsionalitas onClick ke tombol.
- Membangun dua aplikasi TwoActivities dari pelajaran sebelumnya.
- Membangun aplikasi PhoneNumberSpinner dari pelajaran sebelumnya.
- Membangun aplikasi RecyclerView dari pelajaran sebelumnya.

Yang akan Anda PELAJARI

Dalam praktik ini, Anda akan mempelajari cara:

- Menyiapkan Espresso di proyek aplikasi Anda.
- Menulis pengujian Espresso yang menguji masukan pengguna dan memeriksa kebenaran output.
- Menulis pengujian Espresso untuk menemukan spinner, mengeklik salah satu itemnya, dan memeriksa kebenaran output.
- Menggunakan fungsi Record Espresso Test di Android Studio.

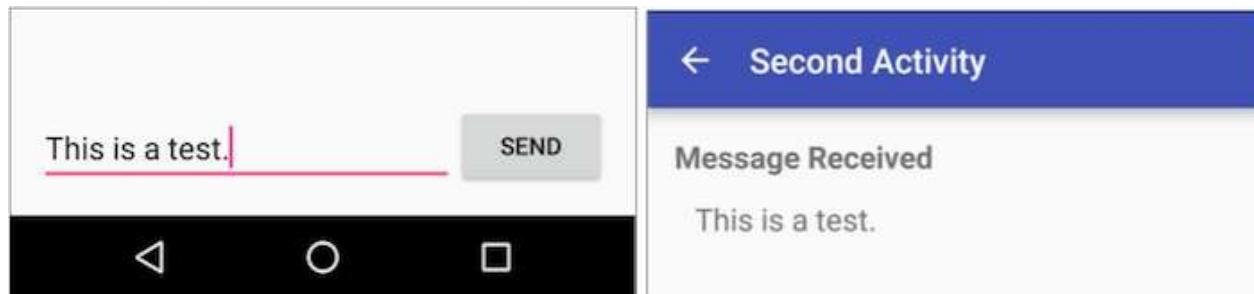
Yang akan Anda LAKUKAN

Dalam praktik ini, Anda akan:

- Memodifikasi proyek untuk membuat pengujian Espresso.
- Menguji masukkan dan keluaran teks aplikasi.
- Menguji klik item spinner dan memeriksa keluarannya.
- Merekam pengujian Espresso pada RecyclerView.

Ringkasan Aplikasi

Anda akan memodifikasi proyek TwoActivities untuk menyiapkan Espresso di proyek untuk pengujian. Anda kemudian akan menguji fungsionalitas aplikasi, yang memungkinkan pengguna memasukkan teks ke dalam bidang teks dan mengeklik tombol **Send**, seperti yang ditampilkan di sebelah kiri gambar bawah dan melihat teks tersebut di aktivitas kedua, seperti yang ditampilkan di sebelah kanan gambar di bawah.



Tip: Untuk pengenalan pengujian Aplikasi, lihat [Uji Aplikasi Anda] (http://d.android.com/tools/testing/testing_android.html).

Proyek Android Studio: [TwoActivities](#)

Tugas 1: Siapkan Espresso di proyek Anda

Untuk menggunakan Espresso, Android Support Repository harus sudah terpasang bersama Android Studio. Anda juga harus mengonfigurasi Espresso di proyek Anda.

Dalam tugas ini, Anda akan memeriksa apakah repositori sudah terpasang. Jika belum, Anda akan memasangnya. Anda kemudian akan mengonfigurasi Espresso di proyek TwoActivities yang telah dibuat sebelumnya.

1.1 Periksa Android Support Repository

1. Mulai Android Studio dan pilih **Tools > Android > SDK Manager**.
2. Klik tab **SDK Tools** dan cari Support Repository.
 - Jika kata "Installed" muncul di kolom Status, Anda sudah siap. Klik **Cancel**.
 - Jika "Not Installed" muncul, atau pembaruan tersedia:
 - i. Klik kotak centang di sebelah Android Support Repository. Ikon unduh seharusnya muncul di samping kotak centang.
 - ii. Klik salah satu dari yang berikut:

- **Apply** untuk mulai memasang repositori dan tetap di SDK Manager untuk membuat perubahan lainnya.
- **OK** untuk memasang repositori dan keluar dari SDK Manager.

1.2 Konfigurasi Espresso untuk proyek

Saat Anda memulai proyek untuk form factor ponsel dan tablet menggunakan **API 15: Android 4.0.3 (Ice Cream Sandwich)** sebagai SDK minimum, Android Studio versi 2.2 atau lebih baru otomatis menyertakan dependensi yang Anda perlukan untuk menggunakan Espresso. Untuk mengeksekusi pengujian, Espresso dan UI Automator menggunakan **JUnit** sebagai framework pengujian. JUnit adalah framework pengujian unit yang paling populer dan banyak digunakan untuk Java. Kelas pengujian Anda menggunakan Espresso atau UI Automator harus dituliskan sebagai kelas pengujian JUnit 4. Jika Anda belum memiliki JUnit, dapatkan di <http://junit.org/junit4/>.

Catatan: Revisi JUnit yang paling baru adalah JUnit 5. Akan tetapi, untuk tujuan penggunaan Espresso atau UI Automator, versi 4.12 disarankan.

Jika Anda telah membuat proyek di versi Android Studio sebelumnya, Anda mungkin harus menambahkan dependensi dan instrumentasi sendiri. Untuk menambahkan dependensi sendiri ikuti langkah-langkah ini:

1. Buka proyek TwoActivities atau jika ingin, buat salinan proyek terlebih dulu lalu buka salinannya. Lihat [Menyalin dan mengubah nama proyek](#) di **Apendiks** untuk mendapatkan petunjuk.
2. Buka file **build.gradle (Module: app)**.

Catatan: *Jangan* buat perubahan pada file build.gradle (Project: yourappname).

3. Periksa apakah yang berikut ini disertakan (bersama dengan dependensi lain) di bagian `dependencies` file **build.gradle (Module: app)** proyek:

```
androidTestCompile
('com.android.support.test.espresso:espresso-core:2.2.2', {
    exclude group: 'com.android.support', module: 'support-annotations'
})
testCompile 'junit:junit:4.12'
```

Catatan: Jika file tidak menyertakan pernyataan dependensi di atas, masukkan ke dalam bagian `dependencies`.

4. Android Studio 2.2 juga menambahkan pernyataan instrumentasi berikut di akhir bagian `defaultConfig` proyek baru:

```
testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
```

Catatan: Jika file tidak menyertakan pernyataan instrumentasi di atas, masukkan di akhir bagian `defaultConfig`.

Instrumentasi adalah serangkaian metode kontrol atau hook, di sistem Android. Hook ini mengontrol komponen Android secara independen terlepas dari daur hidup normal komponen. Hook juga mengontrol cara Android memuat aplikasi. Menggunakan instrumentasi memungkinkan pengujian untuk memanggil metode di aplikasi dan memodifikasi dan memeriksa bidang di aplikasi, terlepas dari daur hidup normal aplikasi.

5. Jika Anda mengubah file **build.gradle (Module: app)**, klik tautan **Sync Now** pada notifikasi tentang file Gradle di sudut kanan atas jendela.

1.3 Matikan animasi di perangkat pengujian Anda

Untuk mengizinkan Android Studio berkomunikasi dengan perangkat Anda, terlebih dahulu Anda harus menonaktifkan USB Debugging di perangkat Anda, seperti yang dijelaskan di bab sebelumnya.

Ponsel dan table Android menampilkan animasi saat berpindah antar aplikasi dan layar. Animasi bersifat atraktif saat menggunakan perangkat, tetapi memperlambat kinerja, dan mungkin menyebabkan hasil yang tidak diinginkan atau menyebabkan pengujian gagal. Jadi sebaiknya nonaktifkan animasi di perangkat fisik Anda. Untuk menonaktifkan animasi pada perangkat pengujian, ketuk ikon Settings di perangkat fisik Anda. Cari **Developer Options**. Sekarang cari bagian **Drawing**. Di bagian ini, nonaktifkan opsi berikut:

- Skala animasi jendela
- Skala animasi transisi
- Skala durasi animator

Tip: Anda juga harus memperhatikan bahwa menginstrumentasikan sistem, misalnya mengeksekusi pengujian unit, bisa mengubah pengaturan waktu fungsi tertentu. Untuk alasan ini, sebaiknya pisahkan pengujian unit dan debug aktual.. Pengujian unit menggunakan Framework Espresso berbasis API dengan hook untuk instrumentasi. Debug menggunakan titik pisah dan metode lain di pernyataan kode aktual di dalam kode aplikasi Anda, seperti yang dijelaskan di pelajaran sebelumnya. </div>

Tugas 2: Lakukan pengujian pengalihan aplikasi dan memasukkan teks

Tulis pengujian Espresso berdasarkan apa yang mungkin dilakukan pengguna saat berinteraksi dengan aplikasi Anda. Pengujian Espresso adalah kelas yang terpisah dari kode aplikasi Anda. Anda bisa membuat pengujian sebanyak yang Anda perlukan, untuk berinteraksi dengan tampilan di UI yang ingin Anda uji.

Pengujian Espresso seperti robot yang harus diberi tahu apa yang harus dilakukan. Espresso harus *menemukan* tampilan yang ingin Anda perintahkan untuk ditemukan di layar dan harus *berinteraksi* dengannya, seperti mengeklik tampilan, memeriksa konten tampilan. Jika gagal melakukan hal ini dengan benar atau jika hasilnya tidak seperti yang Anda harapkan, pengujian akan gagal.

Dengan Espresso, Anda membuat apa yang disebut sebagai skrip tindakan yang akan dilakukan pada setiap tampilan dan memeriksanya terhadap hasil yang diharapkan. Konsep utamanya adalah *mencari* lalu *berinteraksi* dengan elemen UI. Berikut langkah-langkah dasarnya:

1. **Mencocokkan dengan tampilan:** Menemukan tampilan.
2. **Melakukan tindakan:** Melakukan klik atau tindakan lain yang memicu sebuah kejadian dengan tampilan.
3. **Menyatakan dan memverifikasi hasil:** Memeriksa status tampilan untuk memeriksa apakah statusnya mencerminkan status atau perilaku yang diharapkan yang didefinisikan oleh pernyataan.

Hamcrest (anagram dari "matchers") adalah framework yang membantu penulisan pengujian perangkat lunak di Java. Untuk membuat pengujian, Anda harus membuat metode di dalam kelas pengujian yang menggunakan ekspresi Hamcrest.

Tip: Untuk informasi selengkapnya tentang Hamcrest Matcher, lihat [Tutorial Hamcrest](#).

Dengan Espresso, Anda menggunakan tipe ekspresi Hamcrest berikut untuk membantu menemukan tampilan dan berinteraksi dengannya:

- **ViewMatchers:** Ekspresi Hamcrest Matcher dalam kelas `ViewMatchers` yang memungkinkan Anda menemukan tampilan dalam hierarki tampilan saat ini sehingga Anda bisa memeriksa sesuatu atau melakukan tindakan tertentu.
- **ViewActions:** Ekspresi tindakan Hamcrest dalam kelas `ViewActions` yang memungkinkan Anda melakukan tindakan pada tampilan yang ditemukan oleh `ViewMatcher`.
- **ViewAssertions:** Ekspresi pernyataan Hamcrest dalam kelas `ViewAssertions` yang memungkinkan Anda menyatakan atau memeriksa status tampilan yang ditemukan oleh `ViewMatcher`.

Berikut ini proses ketiga ekspresi tersebut bekerja bersama:

1. Menggunakan `ViewMatcher` untuk menemukan tampilan:

```
onView(withId(R.id.my_view))
```

2. Menggunakan `ViewAction` untuk melakukan tindakan:

```
.perform(click())
```

3. Menggunakan `ViewAssertion` untuk memeriksa apakah hasil tindakan cocok dengan pernyataan:

```
.check(matches(isDisplayed()));
```

Berikut ini proses ketiga ekspresi di atas digunakan bersama dalam sebuah pernyataan:

```
onView(withId(R.id.my_view))
    .perform(click())
    .check(matches(isDisplayed()));
```

2.1 Definisikan kelas untuk pengujian dan siapkan aktivitas

Android Studio membuat kelas pengujian Espresso kosong untuk Anda di folder `src/androidTest/java/com.example.package`:

1. Luaskan `com.example.android.twoactivities (androidTest)`, dan buka `ExampleInstrumentedTest`.
2. Untuk membuat pengujian lebih mudah dipahami dan menjelaskan apa yang dilakukan, ganti nama kelas dari `ExampleInstrumentedTest` menjadi berikut:

```
public class ActivityInputOutputTest
```

3. Ubah definisi kelas menjadi berikut:

```
@RunWith(AndroidJUnit4.class)
public class ActivityInputOutputTest {
    @Rule
    public ActivityTestRule mActivityRule = new ActivityTestRule<>(
        MainActivity.class);
}
```

Definisi kelas sekarang menyertakan beberapa anotasi:

- `@RunWith` : Untuk membuat kelas pengujian JUnit 4 yang terinstrumentasi, tambahkan anotasi `@RunWith(AndroidJUnit4.class)` di awal definisi kelas pengujian Anda.
- `@Rule` : Anotasi `@Rule` memungkinkan Anda menambahkan atau mendefinisikan ulang perilaku setiap metode pengujian dengan cara yang bisa dipakai ulang, menggunakan salah satu kelas aturan pengujian yang disediakan oleh Android Testing Support Library, seperti `ActivityTestRule` atau `ServiceTestRule`. Aturan di atas menggunakan objek `ActivityTestRule`, yang menyediakan pengujian fungsional Activity tunggal—dalam hal ini, `MainActivity.class`. Selama durasi pengujian, Anda akan dapat mengubah Activity secara langsung, menggunakan `ViewMatchers`, `ViewActions`, dan `ViewAssertions`.

Pada pernyataan di atas, `ActivityTestRule` dapat berubah menjadi merah terlebih dulu, tetapi kemudian Android Studio menambahkan pernyataan import berikut secara otomatis:

```
import android.support.test.rule.ActivityTestRule;
```

2.2 Uji aktivitas pengalihan:

Aplikasi TwoActivities memiliki dua aktivitas:

- `Main` : Menyertakan tombol `button_main` untuk beralih ke aktivitas `Second` dan tampilan `text_header_reply` yang berfungsi sebagai heading teks untuk aktivitas `Main`.
- `Second` : Menyertakan tombol `button_second` untuk beralih ke aktivitas `Main` dan tampilan `text_header` yang berfungsi sebagai heading teks untuk aktivitas `Second`.

Jika Anda memiliki aplikasi yang mengalihkan aktivitas, Anda harus menguji kemampuannya. Aplikasi TwoActivities menyediakan bidang entri teks dan tombol `Send` (`id button_main`). Mengklik `Send` meluncurkan aktivitas `Second` dengan teks yang dimasukkan yang ditampilkan di tampilan `text_header` aktivitas `Second`.

Tetapi, apa yang terjadi jika tidak ada teks yang dimasukkan? Apakah aktivitas `Second` masih akan muncul?

Kelas `ActivityInputOutputTest` pengujian akan menampilkan bahwa tampilan akan muncul terlepas dari teks dimasukkan atau tidak. Ikuti langkah-langkah berikut untuk menambahkan pengujian Anda ke `ActivityInputOutputTest` :

1. Tambahkan metode `activityLaunch()` ke `ActivityInputOutputTest` untuk menguji apakah tampilan muncul saat meneklik tombol sertakan notasi `@Test` pada baris tepat di atas metode:

```
@Test
public void activityLaunch() { ... }
```

Anotasi `@Test` memberi tahu JUnit bahwa metode `public void` yang menjadi tempat pengaitan bisa dijalankan sebagai kasus pengujian. Sebuah metode pengujian dimulai dengan anotasi `@Test` dan berisi kode untuk dijalankan dan memverifikasi satu fungsi dalam komponen yang ingin Anda uji.

2. Tambahkan ekspresi `ViewMatcher` dan `ViewAction` ke metode `activityLaunch()` untuk menemukan tampilan yang berisi tombol `button_main`, dan menyertakan ekspresi `ViewAction` untuk melakukan klik:

```
onView(withId(R.id.button_main)).perform(click());
```

Metode `onView()` memungkinkan Anda menggunakan argumen `ViewMatcher` untuk menemukan tampilan. Ini akan mencari hierarki tampilan untuk menemukan instance `View` terkait yang memenuhi beberapa kriteria yang disediakan —dalam hal ini, tampilan `button_main`. Ekspresi `.perform(click())` adalah ekspresi `ViewAction` yang melakukan klik pada tampilan.

3. Dalam pernyataan `onView` di atas, `onView`, `withID`, dan `click` mungkin berwarna merah terlebih dulu, tetapi kemudian Android Studio menambahkan pernyataan impor untuk `onView`, `withID`, dan `click`.
4. Tambahkan ekspresi `ViewMatcher` ke metode `activityLaunch()` untuk menemukan tampilan `text_header` (yang berada dalam aktivitas `Second`) dan sebuah ekspresi untuk melakukan pemeriksaan untuk melihat apakah tampilan ditampilkan:

```
onView(withId(R.id.text_header)).check(matches(isDisplayed()));
```

Pernyataan ini menggunakan metode `onView()` untuk menemukan tampilan `text_header` untuk aktivitas `Second` dan memeriksa untuk melihat apakah tampilan ini ditampilkan setelah meneklik tampilan `button_main`.

5. Dalam pernyataan `onView` di atas, metode `check()` mungkin berwarna merah terlebih dulu, tetapi kemudian Android Studio menambahkan pernyataan `import` untuknya.
6. Tambahkan pernyataan yang sama untuk menguji apakah meneklik tombol `button_second` di aktivitas `Second` akan mengalihkan ke aktivitas `Main`:

```
onView(withId(R.id.button_second)).perform(click());
onView(withId(R.id.text_header_reply)).check(matches(isDisplayed()));
```

7. Tinjau metode `activityLaunch()` yang baru Anda buat di kelas `ActivityInputOutputTest`. Akan terlihat seperti ini:

```
@Test
public void activityLaunch() {
    onView(withId(R.id.button_main)).perform(click());
    onView(withId(R.id.text_header)).check(matches(isDisplayed()));
    onView(withId(R.id.button_second)).perform(click());
    onView(withId(R.id.text_header_reply)).check(matches(isDisplayed()));
}
```

8. Untuk menjalankan pengujian, klik kanan (atau Control-klik) **ActivityInputOutputTest** dan pilih **Run ActivityInputOutputTest** dari menu munculan. Anda kemudian bisa memilih untuk menjalankan pengujian pada emulator atau pada perangkat Anda.

Selagi pengujian berjalan, perhatikan bahwa pengujian otomatis memulai aplikasi dan meneklik tombol. Tampilan aktivitas `Second` muncul. Pengujian kemudian meneklik tombol aktivitas `Second` dan tampilan aktivitas `Main` muncul.

Jendela Run (panel bawah Android Studio) menampilkan kemajuan pengujian dan ketika selesai, akan menampilkan "Tests ran to completion". Di kolom kiri, Android Studio menampilkan "All Tests Passed".

2.3 Uji masukan dan keluaran teks

Tuliskan pengujian untuk masukan dan keluaran teks. Aplikasi TwoActivities menggunakan tampilan `editText_main` untuk masukan, tombol `button_main` untuk mengirimkan masukan ke aktivitas `Second` dan tampilan aktivitas `Second` yang menampilkan output dalam bidang dengan id `text_message`.

1. Tambahkan anotasi `@Test` lain dan metode `textInputOutput()` baru ke kelas `ApplicationTest` untuk menguji masukan dan keluaran teks:

```
@Test
public void textInputOutput() {
    onView(withId(R.id.editText_main)).perform(typeText("This is a test."));
    onView(withId(R.id.button_main)).perform(click());
}
```

Metode di atas menggunakan `ViewMatcher` untuk menemukan tampilan yang berisi tampilan `editText_main` dan sebuah `ViewAction` untuk memasukkan teks "This is a test.". Metode kemudian menggunakan `ViewMatcher` lainnya untuk menemukan tampilan yang berisi tombol `button_main` dan `ViewAction` lainnya untuk meneklik tombol.

2. Tambahkan `ViewMatcher` untuk menemukan tampilan `text_message` aktivitas `Second` dan `ViewAssertion` untuk melihat apakah keluaran cocok dengan masukan untuk menguji apakah pesan dikirimkan dengan benar:

```
onView(withId(R.id.text_message)).check(matches(withText("This is a test.")));
```

3. Jalankan pengujian.

Selagi pengujian berjalan, aplikasi akan dimulai dan teks otomatis dimasukkan sebagai input, tombol diklik, dan teks muncul pada layar aktivitas kedua.

Panel bawah Android Studio menampilkan kemajuan pengujian dan jika telah selesai, panel akan menampilkan "Tests ran to completion." Di kolom kiri, Android Studio menampilkan "All Tests Passed". Anda berhasil menguji bidang masukan teks, tombol Send, dan bidang keluaran.

Kode solusi:

Proyek Android Studio: [TwoActivitiesEspressoTest](#)

Lihat `ActivityInputOutputTest.java`.

2.4 Berikan kesalahan untuk menampilkan bahwa pengujian gagal

Berikan kesalahan dalam pengujian untuk melihat bagaimana pengujian yang gagal.

1. Ubah pemeriksaan kecocokan pada tampilan `text_message` dari "This is a test." ke "This is a failing test." :

```
onView(withId(R.id.text_message)).check(matches(withText("This is a failing test.")));
```

2. Jalankan pengujian lagi. Kali ini, Anda akan melihat pesan berwarna merah, "1 test failed", di atas panel bawah dan tanda seru merah di sebelah `textInputOutput` di kolom kiri. Gulir panel bawah ke pesan "Test running started" dan lihat apakah semua hasil setelah tanda seru tersebut berwarna merah. Pernyataan berikutnya setelah "Test running started" adalah:

```
android.support.test.espresso.base.DefaultFailureHandler$AssertionFailedWithCauseError: 'with text: is "This is a failing test."' doesn't match the selected view.
Expected: with text: is "This is a failing test."
Got: "AppCompatTextView{id=2131427417, res-name=text_message ...}
```

Pesan kesalahan fatal lainnya muncul setelah yang di atas, karena efek bergulir dari kegagalan yang menyebabkan kegagalan lain. Anda bisa mengabaikannya dan memperbaiki pengujian itu sendiri.

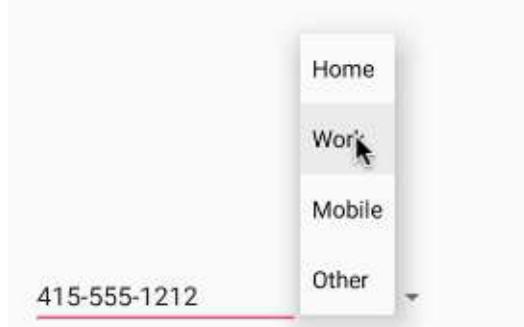
Tugas 3: Uji tampilan pilihan spinner

Metode `onView()` Espresso menemukan tampilan yang bisa Anda uji. Metode ini akan menemukan tampilan dalam hierarki tampilan saat ini. Tetapi, Anda harus berhati-hati—dalam AdapterView seperti spinner, tampilan umumnya diisi dengan tampilan turunan secara dinamis pada waktu proses. Artinya, ada kemungkinan tampilan yang ingin Anda uji mungkin tidak berada dalam hierarki tampilan pada waktu tersebut.

Espresso API menangani masalah ini dengan menyediakan titik masuk `onData()` terpisah, yang dapat memuat item adaptor terlebih dulu dan menampilkannya di fokus sebelum menemukan dan melakukan tindakan pada turunannya yang mana pun.

PhoneNumberSpinner adalah aplikasi dari pelajaran sebelumnya yang menampilkan spinner dengan id `label_spinner`, untuk memilih label nomor telepon (**Home**, **Work**, **Mobile**, dan **Other**). Aplikasi menampilkan pilihan dalam bidang teks, yang digabungkan dengan nomor telepon yang dimasukkan.

Tujuan pengujian ini adalah untuk membuka spinner, membuka setiap item, dan kemudian memverifikasi apakah `TextView` `text_phonelabel` berisi item. Pengujian memperagakan bahwa kode yang mengambil pilihan spinner bekerja dengan benar dan kode yang menampilkan teks item spinner juga bekerja dengan benar. Anda akan menulis pengujian menggunakan sumber daya string dan melakukan iterasi pada item spinner sehingga pengujian akan berfungsi berapa pun item yang berada di spinner atau apa pun teks pada item tersebut, misalnya, teks bisa dalam bahasa yang berbeda.



Proyek Android Studio: [PhoneNumberSpinner](#)

3.1 Buat metode pengujian

1. Buka proyek PhoneNumberSpinner atau jika Anda ingin, buat salinan proyek terlebih dulu, kemudian buka salinannya. Lihat [Menyalin dan mengubah nama proyek](#) di [Apendiks](#) untuk mendapatkan petunjuk.
2. Konfigurasi Espresso di proyek Anda seperti yang dijelaskan sebelumnya.
3. Luaskan `com.example.android.phonenumberspinner (androidTest)` dan buka `ExampleInstrumentedTest`.
4. Ganti nama `ExampleInstrumentedTest` menjadi `SpinnerSelectionTest` di definisi kelas dan tambahkan yang berikut ini:

```
@RunWith(AndroidJUnit4.class)
public class SpinnerSelectionTest {
    @Rule
    public ActivityTestRule<MainActivity> mActivityRule = new ActivityTestRule<>(
        MainActivity.class);
}
```

5. Buat metode `iterateSpinnerItems()` sebagai `public` yang mengembalikan `void`.

3.2 Akses larik yang digunakan untuk item spinner

Sebaiknya uji klik setiap item di spinner berdasarkan jumlah elemen di larik. Bagaimana Anda bisa mengakses larik?

1. Tetapkan larik yang digunakan untuk item spinner ke larik baru untuk digunakan di dalam metode

```
iterateSpinnerItems() :
```

```
public void iterateSpinnerItems() {
    String[] myArray =
        mActivityRule.getActivity().getResources()
            .getStringArray(R.array.labels_array);
}
```

Pada pernyataan di atas, pengujian mengakses larik aplikasi (dengan id `labels_array`) dengan membuat konteks dengan metode `getActivity()` kelas `ActivityTestRule` dan mendapatkan instance sumber daya di paket aplikasi menggunakan `getResources()`.

2. Tetapkan panjang larik ke `size` dan buat loop `for` menggunakan `size` sebagai jumlah maksimum untuk penghitung.

```
int size = myArray.length;
for (int i=0; i<size; i++) {
```

3.3 Temukan item spinner dan klik item ini

1. Tambahkan pernyataan `onView()` di dalam loop `for` untuk menemukan spinner dan klik spinner:

```
// Find the spinner and click on it.
onView(withId(R.id.label_spinner)).perform(click());
```

Pengguna harus mengeklik spinner itu sendiri untuk mengeklik item apa pun di spinner, sehingga pengujian Anda juga harus mengeklik spinner terlebih dulu sebelum mengeklik item.

2. Tuliskan pernyataan `onData()` untuk menemukan dan mengeklik item spinner:

```
// Find the spinner item and click on it.
onData(is(myArray[i])).perform(click());
```

Pernyataan di atas cocok jika objek merupakan item spesifik di spinner, seperti yang ditentukan oleh elemen larik `myArray[i]`.

Jika `onData` muncul berwarna merah, klik kata dan klik ikon bola lampu merah yang muncul di margin kiri. Pilih yang berikut ini dalam menu munculan:

Static import method 'android.support.test.espresso.Espresso.onData'

Jika `is` muncul berwarna merah, klik kata dan klik ikon bola lampu merah yang muncul di margin kiri. Pilih yang berikut ini dalam menu munculan:

Static import method...> Matchers.is (org.hamcrest)

3. Tambahkan dua pernyataan `onView()` lagi ke loop for:

```
// Find the Submit button and click on it.
onView(withId(R.id.button_main)).perform(click());
// Find the text view and check that the spinner item
// is part of the string.
onView(withId(R.id.text_phonelabel))
    .check(matches(withText(containsString(myArray[i]))));
```

Pernyataan pertama menemukan `button_main` dan mengekliknya. Pernyataan kedua memeriksa untuk melihat apakah `text_phonelabel` yang dihasilkan cocok dengan item spinner yang ditentukan oleh `myArray[i]`.

Jika `containsString` muncul berwarna merah, klik kata dan klik ikon bola lampu merah yang muncul di margin kiri. Pilih yang berikut ini dalam menu munculan:

Static import method...> Matchers.containsString (org.hamcrest)

4. Untuk menjalankan pengujian, klik kanan (atau Control-klik) **SpinnerSelectionTest** dan pilih **Run SpinnerSelectionTest** dari menu munculan. Anda kemudian bisa memilih untuk menjalankan pengujian pada emulator atau pada perangkat Anda.

Pengujian menjalankan api, mengeklik spinner, dan "memeriksa" spinner—yaitu mengeklik setiap item spinner dari atas ke bawah, memeriksa apakah item muncul di bidang teks. Tidak masalah berapa banyak item spinner di larik, atau bahasa apa yang digunakan untuk item spinner—pengujian menjalankan semuanya dan memeriksa keluarannya terhadap larik.

Panel bawah Android Studio menampilkan kemajuan pengujian dan jika telah selesai, panel akan menampilkan "Tests ran to completion." Di kolom kiri, Android Studio menampilkan "All Tests Passed".

Kode solusi:

Proyek Android Studio: [PhoneNumberSpinnerEspressoTest](#)

Lihat `SpinnerSelectionTest.java`.

Tugas 4: Rekam pengujian RecyclerView

Anda telah mempelajari cara membuat `RecyclerView` di bab sebelumnya. Seperti `AdapterView` (misalnya `spinner`), sebuah `RecyclerView` secara dinamis mengisikan tampilan anak pada waktu proses. Akan tetapi, `RecyclerView` bukanlah `AdapterView`, sehingga Anda tidak bisa menggunakan `onData()` untuk berinteraksi dengan item daftar seperti yang Anda lakukan di tugas sebelumnya dengan `spinner`. Yang membuat `RecyclerView` rumit dari sudut pandang Espresso adalah `onView()` tidak bisa menemukan tampilan anak jika berada di luar layar.

Anda memiliki dua alat bermanfaat untuk mengatasi kompleksitas ini:

- Sebuah kelas bernama `RecyclerViewActions` yang mengekspos API kecil untuk beroperasi pada sebuah `RecyclerView`.
- Sebuah fitur Android Studio (di versi 2.2 atau lebih baru) yang memungkinkan Anda *merekam* pengujian Espresso. Gunakan aplikasi sebagai pengguna normal. Saat Anda mengeklik UI aplikasi, kode pengujian yang bisa diedit dibuat untuk Anda. Anda juga menambahkan pernyataan untuk memeriksa bila tampilan memiliki nilai tertentu.

Merekam pengujian Espresso, sebagai ganti menulis kode secara manual, memastikan bahwa aplikasi Anda mendapatkan cakupan pengujian UI pada area yang mungkin memakan terlalu banyak waktu atau terlalu sulit untuk ditulis kodennya secara manual.

Kode solusi:

Proyek Android Studio: [RecyclerView](#)

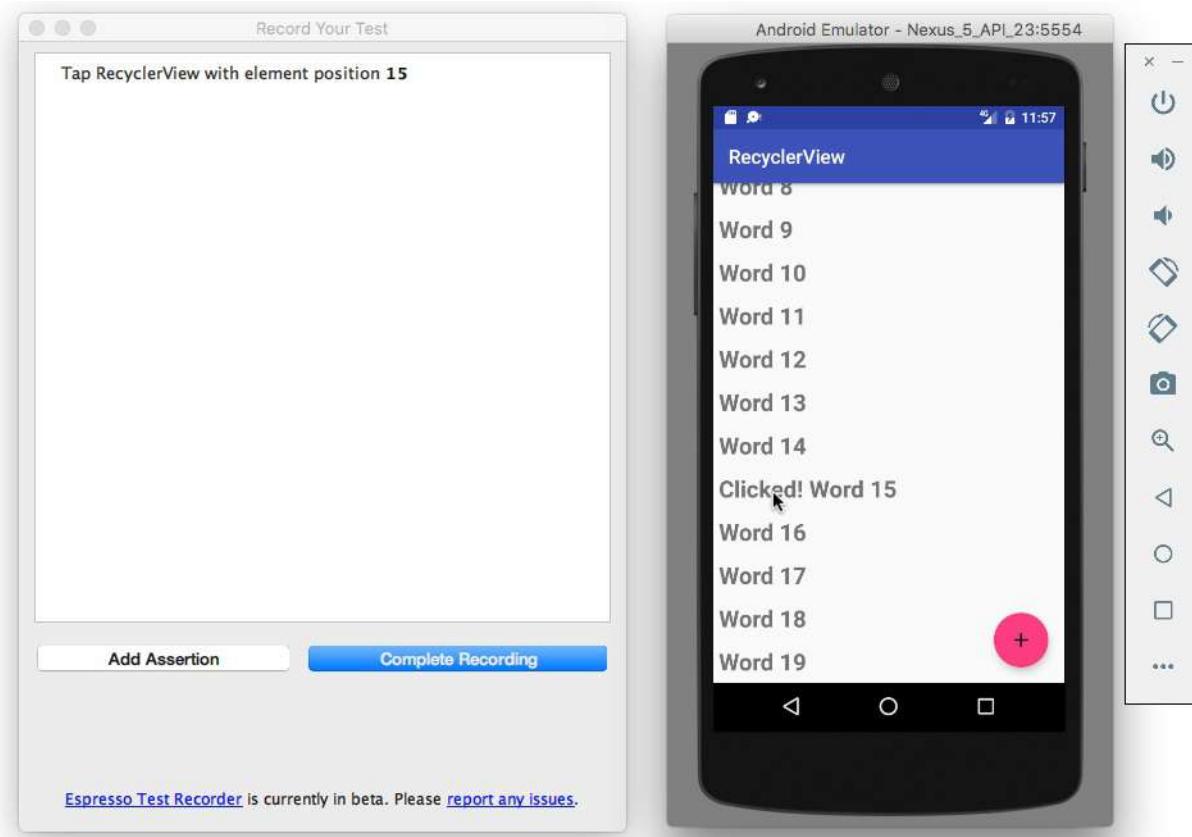
4.1 Buka dan jalankan aplikasi

1. Buka proyek `RecyclerView` atau jika ingin, buat salinan proyek terlebih dulu, lalu buka salinannya. Lihat [Menyalin dan mengubah nama proyek](#) di [Apendiks](#) untuk mendapatkan petunjuk.
2. Konfigurasi Espresso di proyek Anda seperti yang dijelaskan sebelumnya.
3. Jalankan aplikasi untuk memastikan agar berjalan dengan benar. Anda bisa menggunakan emulator atau perangkat Android.

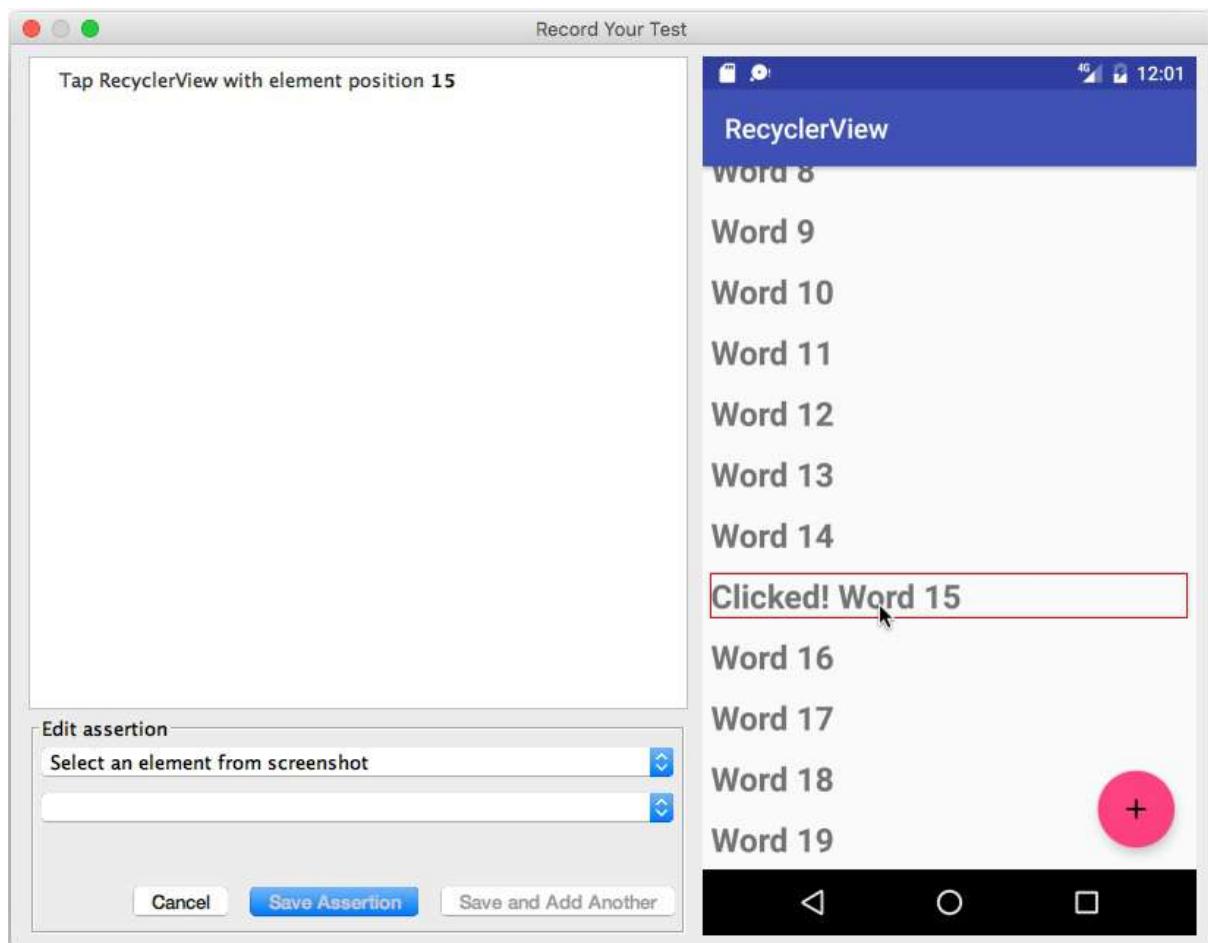
Aplikasi memungkinkan Anda mengulir daftar kata. Saat Anda mengeklik kata, seperti **Word 15**, kata dalam daftar berubah menjadi "Clicked! Word 15".

4.2 Rekam pengujian

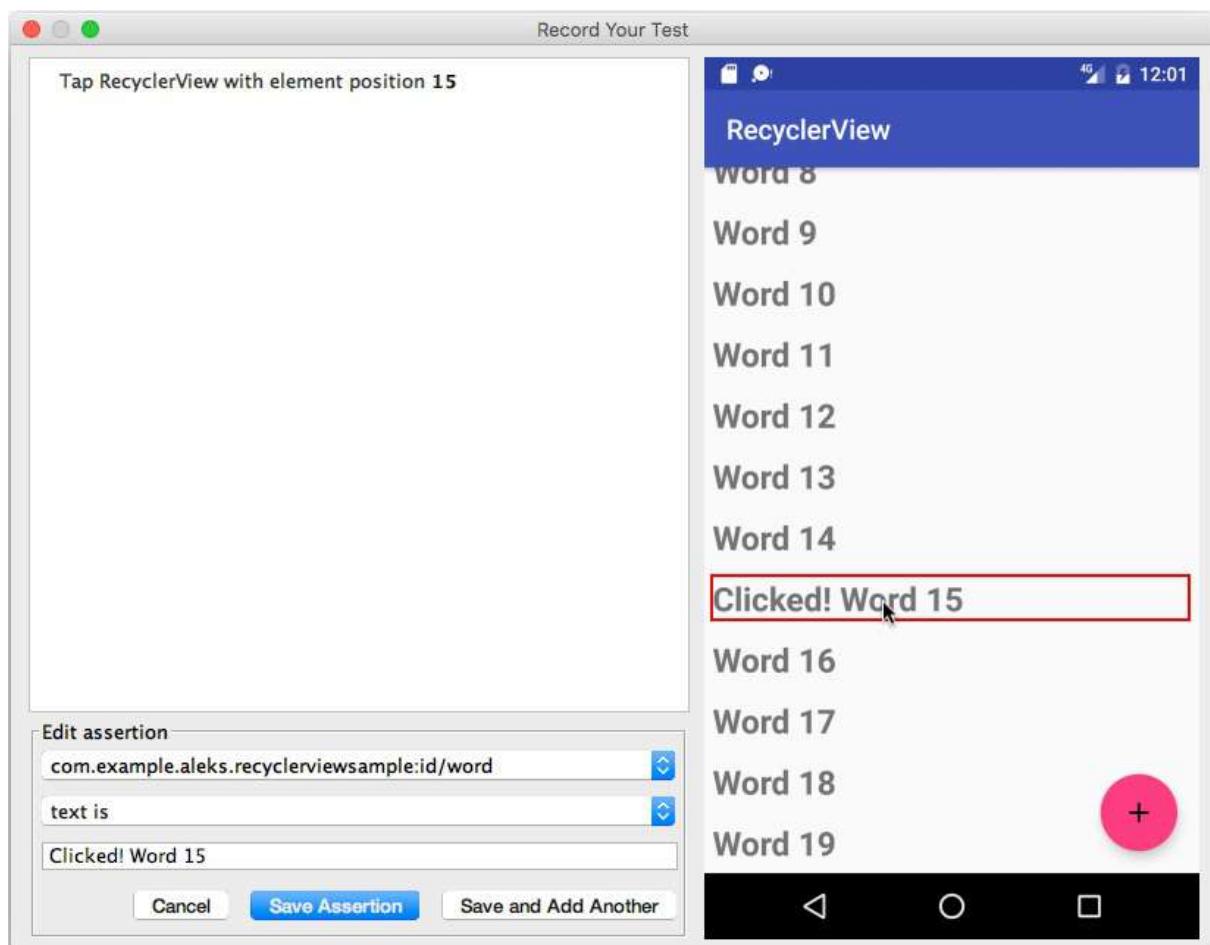
1. Pilih **Run > Record Espresso Test**, pilih target penerapan Anda (emulator atau perangkat) dan klik **OK**.
2. Gulirkan daftar kata di aplikasi pada emulator atau perangkat dan ketuk **Word 15**. Jendela Record Your Test menampilkan tindakan yang direkam ("Ketuk RecyclerView dengan posisi elemen 15").



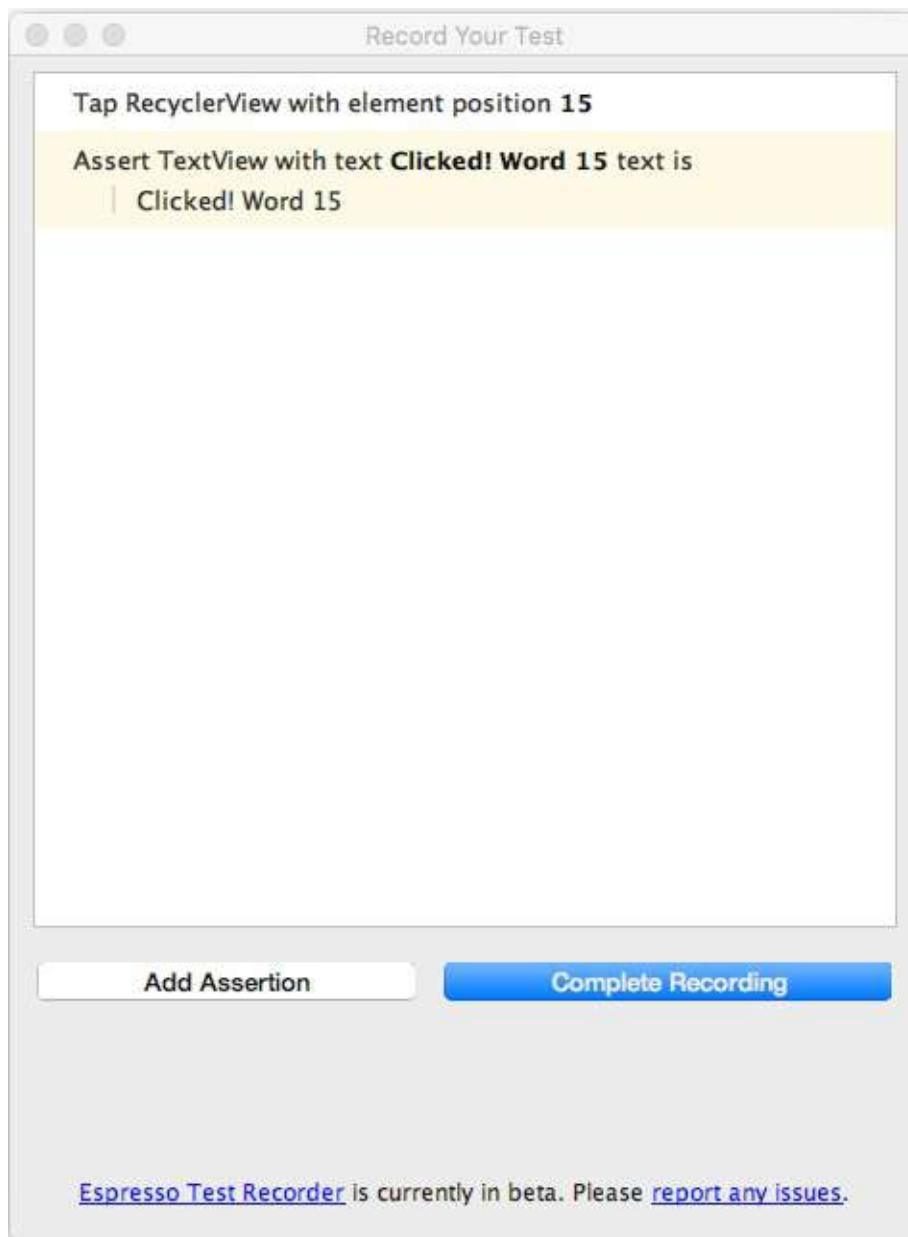
3. Klik **Add Assertion** di jendela Record Your Test. Tangkapan layar UI aplikasi muncul di panel sebelah kanan jendela. Pilih **Clicked! Word 15** di tangkapan layar sebagai elemen UI yang ingin Anda periksa.



4. Pilih **text is** dari menu tarik-turun kedua. Teks yang Anda harap untuk dilihat telah dimasukkan ke bidang bawah menu tarik-turun.



5. Klik **Save Assertion** lalu klik **Complete Recording**.



6. Dalam dialog yang muncul, edit nama pengujian menjadi **RecyclerViewTest** sehingga tujuan pengujian mudah dipahami.
7. Android Studio bisa menampilkan permintaan untuk menambahkan lebih banyak dependensi ke file Gradle Build Anda. Klik **Yes** untuk menambahkan dependensi.
8. Luaskan **com.example.android.recyclerview (androidTest)** untuk melihat pengujian, dan menjalankan pengujian. Pengujian seharusnya berhasil.

Berikut ini adalah pengujian, seperti yang direkam di file RecyclerViewTest.java:

```

@RunWith(AndroidJUnit4.class)
public class RecyclerViewTest {

    @Rule
    public ActivityTestRule<MainActivity> mActivityTestRule =
        new ActivityTestRule<>(MainActivity.class);

    @Test
    public void recyclerViewTest() {
        ViewInteraction recyclerView = onView(
            allOf(withId(R.id.recyclerview), isDisplayed()));
        recyclerView.perform(actionOnItemAtPosition(15, click()));

        ViewInteraction textView = onView(
            allOf(withId(R.id.word), withText("Clicked! Word 15"),
                childAtPosition(
                    childAtPosition(
                       (withId(R.id.recyclerview),
                            11),
                    0),
                isDisplayed())));
        textView.check(matches(withText("Clicked! Word 15")));
    }

    private static Matcher<View> childAtPosition(
        final Matcher<View> parentMatcher, final int position) {

        return new TypeSafeMatcher<View>() {
            @Override
            public void describeTo>Description description) {
                description.appendText("Child at position " + position + " in parent ");
                parentMatcher.describeTo(description);
            }

            @Override
            public boolean matchesSafely(View view) {
                ViewParent parent = view.getParent();
                return parent instanceof ViewGroup && parentMatcher.matches(parent)
                    && view.equals(((ViewGroup) parent).getChildAt(position));
            }
        };
    }
}

```

Pengujian menggunakan objek RecyclerView pada kelas `ViewInteraction`, yang merupakan antarmuka utama untuk melakukan tindakan atau pernyataan pada tampilan, yang menyediakan metode `check()` dan `perform()`. Periksa kode pengujian untuk melihat cara kerjanya:

- **Menjalankan:** Kode berikut menggunakan metode `perform()` dan metode `actionOnItemAtPosition` pada kelas `RecyclerViewActions` untuk menggulir ke posisi (15) dan mengeklik item:

```

ViewInteraction recyclerView = onView(
    allOf(withId(R.id.recyclerview), isDisplayed()));
recyclerView.perform(actionOnItemAtPosition(15, click()));

```

- **Memeriksa apakah cocok dengan pernyataan:** Kode berikut memeriksa apakah item yang diklik cocok dengan pernyataan bahwa nilainya harus `"Clicked! Word 15"`:

```

ViewInteraction textView = onView(
    allOf(withId(R.id.word), withText("Clicked! Word 15"),
        childAtPosition(
            childAtPosition(
                withId(R.id.recyclerview),
                11),
            0),
        isDisplayed()));
textView.check(matches(withText("Clicked! Word 15")));

```

Kode di atas menggunakan metode yang bernama `childAtPosition()`, yang didefinisikan sebagai `Matcher` khusus:

```

private static Matcher<View> childAtPosition(
    final Matcher<View> parentMatcher, final int position) {
    // TypeSafeMatcher() returned
    ...
}

```

- **Mengimplementasikan matcher khusus:** Matcher khusus meluaskan kelas `TypeSaveMatcher` abstrak dan mengharuskan Anda mengimplementasikan yang berikut:

- Metode `matchesSafely()`, ditampilkan di bawah, untuk mendefinisikan cara memeriksa tampilan dalam `RecyclerView`.
- Metode `describeTo()`, ditampilkan di bawah, untuk mendefinisikan cara Espresso menjelaskan matcher keluaran di panel Run di bagian bawah Android Studio jika terjadi kegagalan.

```

...
// TypeSafeMatcher() returned
return new TypeSafeMatcher<View>() {
    @Override
    public void describeTo(Description description) {
        description.appendText("Child at position "
            + position + " in parent ");
        parentMatcher.describeTo(description);
    }

    @Override
    public boolean matchesSafely(View view) {
        ViewParent parent = view.getParent();
        return parent instanceof ViewGroup &&
            parentMatcher.matches(parent)
            && view.equals(((ViewGroup)
            parent).getChildAt(position));
    }
};
}
}

```

Anda bisa merekam beberapa interaksi dengan UI dalam satu sesi rekaman. Anda juga bisa merekam beberapa pengujian dan mengedit pengujian untuk melakukan tindakan lainnya, menggunakan kode yang direkam sebagai cuplikan untuk menyalin, menempel, dan mengedit.

Kode solusi

Proyek Android: [RecyclerViewEspressoTest](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Tuliskan teks Espresso untuk aplikasi Scorekeeper dari pelajaran sebelumnya yang menguji apakah tombol Day Mode muncul setelah mengeklik **Night Mode** dan apakah tombol Night Mode muncul setelah mengeklik **Day Mode**.

Rangkuman

Dalam praktik ini, Anda telah mempelajari cara:

- Menyiapkan Espresso untuk menguji proyek Android Studio:
 - Memeriksa memasang Android Support Repository.
 - Menambahkan instrumentasi dan dependensi ke file **build.gradle (Module: app)**.
 - Menonaktifkan animasi di perangkat pengujian Anda.
 - Mendefinisikan kelas pengujian.
- Menguji untuk melihat apakah sebuah aktivitas diluncurkan:
 - Menggunakan metode `onView()` dengan argumen ViewMatcher untuk menemukan tampilan.
 - Menggunakan ekspresi ViewAction untuk melakukan klik.
 - Menggunakan ekspresi ViewAssertion untuk memeriksa apakah tampilan ditampilkan.
 - Menggunakan ekspresi ViewAssertion untuk melihat apakah keluaran cocok dengan masukan.
- Menguji spinner dan pilihannya:
 - Menggunakan metode `onData()` dengan tampilan yang secara dinamis diisikan oleh adaptor pada waktu proses.
 - Mendapatkan item dari larik aplikasi dengan membuat konteks dengan `getActivity()` dan mendapatkan instance sumber daya menggunakan `getResources()`.
 - Menggunakan pernyataan `onData()` untuk menemukan dan mengeklik setiap item spinner.
 - Menggunakan metode `onView()` dengan ViewAction dan ViewAssertion untuk memeriksa apakah keluaran cocok dengan item spinner yang dipilih.
- Merekam pengujian RecyclerView:
 - Menggunakan kelas `RecyclerViewActions` yang mengekspos metode untuk beroperasi dengan RecyclerView.
 - Merekam pengujian Espresso untuk membuat kode pengujian secara otomatis.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Menguji Antarmuka Pengguna](#)

Ketahui selengkapnya

Dokumentasi Android Studio:

- [Menguji Aplikasi Anda](#)
- [Dasar-Dasar Espresso](#)
- [Rujukan ringkas Espresso](#)

Dokumentasi Developer Android:

- [Praktik Terbaik untuk Pengujian](#)
- [Memulai Pengujian](#)
- [Pengujian UI untuk Aplikasi Tunggal](#)
- [Membangun Pengujian Unit Terinstrumentasi](#)
- [Contoh Lanjutan Espresso](#)
- [Tutorial Hamcrest](#)
- [Hamcrest API dan Kelas Utilitas](#)
- [Test Support API](#)

Pengujian Pustaka Dukungan Android:

- [Dokumentasi Espresso](#)
- [Contoh Espresso](#)

Video

- [Android Testing Support - Android Testing Patterns #1](#) (pendahuluan)
- [Android Testing Support - Android Testing Patterns #2](#) (pencocokan tampilan onView)
- [Android Testing Support - Android Testing Patterns #3](#) (tampilan onData dan adaptor)

Lainnya:

- Blog Pengujian Google: [Android UI Automated Testing](#)
- Atomic Object: "[Espresso – Testing RecyclerViews at Specific Positions](#)"
- Stack Overflow: "[How to assert inside a RecyclerView in Espresso?](#)"
- GitHub: [Contoh Pengujian Android](#)
- Google Codelabs: [Android Testing Codelab](#)

7.1: Membuat AsyncTask

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Tugas 1: Menyiapkan proyek SimpleAsyncTask](#)
- [Tugas 2: Membuat subkelas AsyncTask](#)
- [Tugas 3: Mengimplementasikan Langkah Terakhir](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Thread adalah jalur independen eksekusi dalam program yang berjalan. Saat program Android diluncurkan, sistem Waktu Proses Android membuat thread yang bernama thread "Main". Saat program berjalan, setiap baris kode dieksekusi secara berkelanjutan, baris per baris. Thread utama ini adalah bagaimana aplikasi Anda berinteraksi dengan komponen dari Android UI Toolkit, dan mengapa thread utamanya kadang disebut "thread UI". Namun, kadang-kadang aplikasi perlu melakukan pekerjaan yang intensif sumber daya, seperti mengunduh file, kueri database, memutar media, atau menghitung analitik yang kompleks. Tipe pekerjaan intensif ini dapat memblokir thread UI jika semua kode dieksekusi secara berkelanjutan dalam satu thread. Saat aplikasi melakukan pekerjaan intensif, aplikasi tidak merespons ke pengguna atau menggambar di layar karena aplikasi menunggu pekerjaan selesai. Ini bisa menyebabkan kinerja yang buruk, yang memberikan dampak negatif kepada pengalaman pengguna. Pengguna bisa merasa frustasi dan menghapus aplikasi Android jika kinerja aplikasi tersebut lambat.

Agar pengalaman pengguna (UX) lancar dan merespons gestur pengguna dengan cepat, Android Framework menyediakan kelas helper bernama `AsyncTask` yang memproses kerja dari thread UI. `AsyncTask` adalah kelas Java abstrak yang menyediakan satu cara untuk memindahkan pemrosesan intensif ke thread yang terpisah, sehingga mengizinkan thread UI agar tetap responsif. Karena thread terpisah ini tidak disinkronisasi dengan thread yang memanggil, thread ini disebut thread asinkron. `AsyncTask` juga berisi callback yang mengizinkan Anda menampilkan hasil penghitungan kembali ke thread UI.

Dalam praktik ini, Anda akan mempelajari cara menambahkan tugas latar belakang ke aplikasi Android menggunakan `AsyncTask`.

Yang harus sudah Anda KETAHUI

Anda harus sudah bisa:

- Membuat sebuah Aktivitas.
- Menambahkan `TextView` ke layout untuk aktivitas tersebut.
- Mendapatkan id untuk `TextView` dan menyetel kontennya secara terprogram.
- Menggunakan tampilan Tombol dan fungsionalitas `onClick`.

Yang akan Anda PELAJARI

Dalam praktik ini Anda akan belajar:

- Menambahkan `AsyncTask` ke aplikasi untuk menjalankan tugas di latar belakang, di luar thread UI.
- Mengidentifikasi dan memahami manfaat dan kekurangan menggunakan `AsyncTask` untuk tugas latar belakang.

Yang akan Anda LAKUKAN

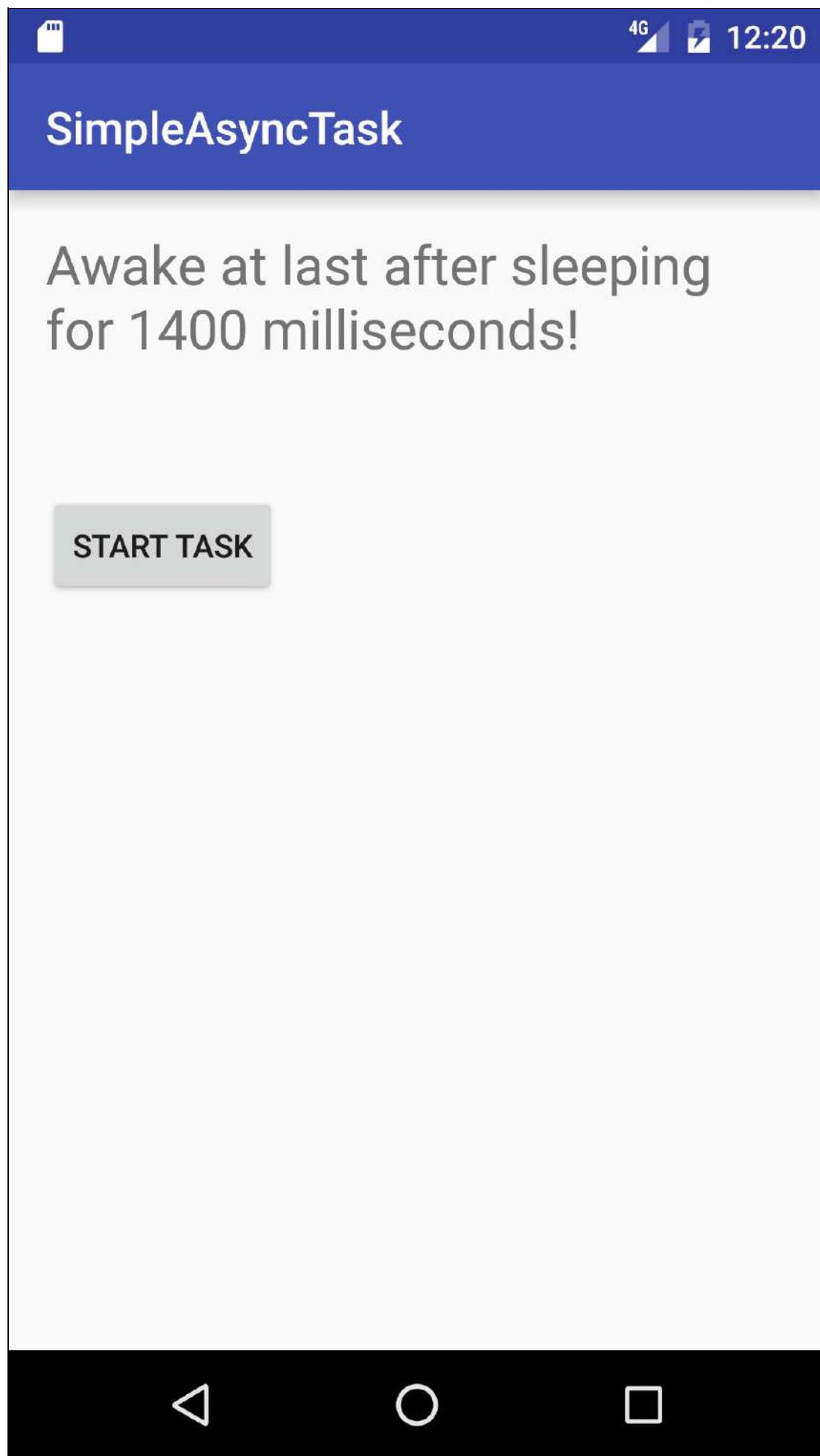
Dalam praktik ini Anda akan:

- Membuat aplikasi sederhana yang mengeksekusi tugas latar belakang menggunakan AsyncTask.
- Menjalankan aplikasi dan melihat apa yang terjadi saat memutar layar.

Ringkasan Aplikasi

Anda akan membangun aplikasi yang memiliki satu TextView dan satu tombol. Saat pengguna meneklik tombol, aplikasi tertidur selama beberapa waktu, dan menampilkan pesan dalam TextView saat aplikasi aktif.

Berikut gambaran aplikasi yang telah selesai:



Tugas 1: Menyiapkan Proyek SimpleAsyncTask

UI SimpleAsyncTask cukup mudah. UI ini berisi tombol yang meluncurkan AsyncTask, dan TextView yang menampilkan status aplikasi.

1.1 Membuat layout

- Buat proyek baru bernama **SimpleAsyncTask** menggunakan template Empty Activity (terima default untuk opsi lainnya).
- Ubah tampilan root `RelativeLayout` ke `LinearLayout`.
- Tambahkan elemen UI penting berikut ke layout untuk MainActivity:

| Tampilan | Atribut | Nilai |
|--------------|-----------------------------------------------------------|-----------------------------------------------------------------------|
| LinearLayout | <code>android:orientation</code> | <code>vertical</code> |
| TextView | <code>android:text</code> <code>android:id</code> | <code>I am ready to start work!</code> <code>@+id/textView1</code> |
| Button | <code>android:text</code> <code>android:onClick</code> | <code>Start Task</code> <code>startTask</code> |

Catatan: Anda bisa menyetel tinggi dan lebar layout setiap tampilan ke ukuran apa pun, selama tampilan pada layar tidak tergantung pada ukuran layar (menggunakan `wrap_content` memastikan bahwa hal ini terjadi).

- Atribut `onClick` untuk tombol akan disorot dengan warna kuning, karena metode `startTask()` belum diimplementasikan di MainActivity. Letakkan kursor pada teks yang disorot, tekan **Alt + Enter (Option + Enter di Mac)** dan pilih **Create 'startTask(View)' dalam 'MainActivity'** untuk membuat stub metode dalam MainActivity.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ready_to_start"
        android:id = "@+id/textView1"
        android:textSize="24sp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/start_task"
        android:id="@+id/button"
        android:layout_marginTop="56dp"
        android:onClick="startTask" />
</LinearLayout>
```

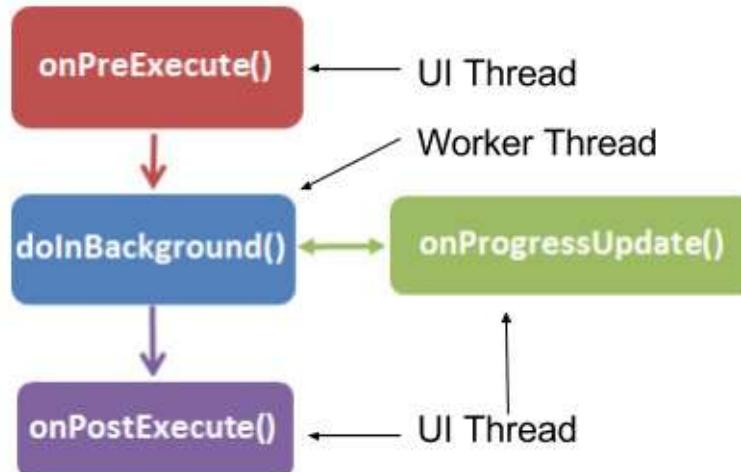
Tugas 2: Membuat subkelas AsyncTask

Karena `AsyncTask` adalah kelas abstrak, Anda perlu menjadikannya subkelas agar dapat menggunakannya. Dalam contoh ini, `AsyncTask` akan mengeksekusi tugas latar belakang sederhana, yaitu tertidur dengan durasi acak. Dalam aplikasi sesungguhnya, tugas latar belakang dapat melakukan semua jenis pekerjaan mulai dari membuat kueri database hingga

menghubungkan ke internet, menghitung gerakan GO berikutnya agar Anda bisa mengalahkan pemenang GO saat ini.

AsyncTask memiliki metode berikut untuk melakukan pekerjaan dari thread utama:

- `onPreExecute()` : Metode ini menjalankan thread UI, dan digunakan untuk menyiapkan tugas Anda (seperti menampilkan bilah kemajuan).
- `doInBackground()` : Di sinilah tempat Anda mengimplementasikan kode untuk mengeksekusi pekerjaan yang akan dilakukan pada thread terpisah.
- `onProgressUpdate()` : Ini akan dipanggil pada thread UI dan digunakan untuk memperbarui kemajuan dalam UI (seperti mengisi bilah kemajuan)
- `onPostExecute()` : Lagi pada thread UI, ini akan digunakan untuk memperbarui hasil ke UI setelah AsyncTask telah



selesai dimuat.

Catatan: Thread latar belakang atau worker adalah thread yang bukan thread UI atau thread utama.

Saat Anda membuat AsyncTask, Anda perlu memberikan informasi tentang pekerjaan yang akan dilakukan, apakah harus atau bagaimana melaporkan kemajuannya, dan dalam bentuk apa untuk mengembalikan hasilnya.

Dalam latihan ini Anda akan menggunakan subkelas AsyncTask untuk mendefinisikan pekerjaan yang akan dijalankan dalam thread yang berbeda dengan thread UI, yang akan menghindari masalah kinerja apa pun.

Saat Anda menggunakan AsyncTask, Anda bisa mengonfigurasinya menggunakan parameter ini:

- Params: Tipe data parameter yang dikirimkan ke tugas saat mengeksekusi `doInBackground()` metode pengganti.
- Progress: Tipe data unit kemajuan yang dipublikasikan menggunakan `onProgressUpdate()` metode pengganti.
- Result: Tipe data hasil yang diberikan oleh `onPostExecute()` metode pengganti.

Contohnya, AsyncTask dengan deklarasi kelas berikut akan menjadikan `String` sebagai parameter dalam `doInBackground()` (untuk digunakan dalam kueri, misalnya), Integer untuk `onProgressUpdate()` (persentase pekerjaan yang sudah selesai), dan Bitmap untuk hasil dalam `onPostExecute()` (hasil kueri):

```
public class MyAsyncTask extends AsyncTask <String, Integer, Bitmap>{}
```

2.1 Menjadikan AsyncTask Subkelas

Dalam implementasi AsyncTask pertama, subkelas AsyncTask akan menjadi sangat sederhana. Subkelas ini tidak memerlukan parameter kueri atau mempublikasikan kemajuannya. Anda hanya perlu menggunakan metode `doInBackground()` and `onPostExecute()`.

1. Buat kelas Java baru bernama SimpleAsyncTask yang memperluas AsyncTask dan yang memerlukan tiga parameter tipe generik:
 - Kosong untuk parameternya karena AsyncTask ini tidak memerlukan masukan apa pun.
 - Kosong untuk tipe kemajuan, karena kemajuan tidak dipublikasikan.
 - String sebagai tipe hasil karena Anda akan memperbarui TextView dengan string saat AsyncTask telah

menyelesaikan eksekusi.

```
public class SimpleAsyncTask extends AsyncTask <Void, Void, String>{}
```

Catatan: Deklarasi kelas akan digarisbawahi merah karena metode `doInBackground()` belum diimplementasikan. AsyncTask akan perlu memperbarui TextView setelah selesai tertidur. Lalu konstruktor perlu menyertakan TextView, agar dapat diperbarui dalam `onPostExecute()`.

2. Definisikan variabel anggota `mTextView`.
3. Implementasikan konstruktor untuk AsyncTask yang memerlukan TextView dan menyetel `mTextView` ke yang diteruskan dalam TextView:

```
public SimpleAsyncTask(TextView tv) {
    mTextView = tv;
}
```

2.2 Mengimplementasikan `doInBackground()`

1. Tambahkan metode `doInBackground()` yang diperlukan. Letakkan kursor pada deklarasi kelas yang disorot, tekan **Alt + Enter** (**Option + Enter** di Mac) dan pilih metode Implement. Pilih `doInBackground()` dan klik **OK**:

```
@Override
protected String doInBackground(Void... voids) {
    return null;
}
```

2. Implementasikan `doInBackground()` ke:

- Buat integer acak antara 0 dan 10
- Kalikan jumlahnya dengan 200
- Buat thread saat ini agar tertidur. (Gunakan `Thread.sleep()`) dalam blok try/catch.
- Kembalikan String "Awake at last after xx milliseconds" (xx adalah jumlah milidetik saat aplikasi tertidur)

```
@Override
protected String doInBackground(Void... voids) {

    // Generate a random number between 0 and 10
    Random r = new Random();
    int n = r.nextInt(11);

    // Make the task take long enough that we have
    // time to rotate the phone while it is running
    int s = n * 200;

    // Sleep for the random amount of time
    try {
        Thread.sleep(s);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    // Return a String result
    return "Awake at last after sleeping for " + s + " milliseconds!";
}
```

2.3 Mengimplementasikan `onPostExecute()`

Saat metode `doInBackground()` selesai, nilai yang dikembalikan secara otomatis diteruskan ke callback `onPostExecute()`.

1. Implementasikan `onPostExecute()` untuk mengambil argumen `String` (yang Anda definisikan dalam parameter ketiga AsyncTask dan yang metode `doInBackground()` kembalikan) dan tampilkan string tersebut dalam TextView:

```
protected void onPostExecute(String result) {
    mTextView.setText(result);
}
```

Catatan: Anda bisa memperbarui UI dalam `onPostExecute()` karena dijalankan pada thread (UI) utama. Anda tidak dapat memanggil `mTextView.setText()` dalam `doInBackground()`, karena metode tersebut dieksekusi pada thread yang terpisah.

Tugas 3: Mengimplementasikan Langkah Terakhir

3.1 Implementasikan metode yang mulai dengan AsyncTask

Aplikasi Anda sekarang memiliki AsyncTask yang melakukan pekerjaan di latar belakang (atau akan melakukannya jika Anda tidak memanggil `sleep()` sebagai pekerjaan yang disimulasi.) Anda sekarang bisa mengimplementasikan metode yang dipanggil saat tombol Start Task diklik, untuk memicu tugas latar belakang.

1. Dalam file MainActivity.java file, tambahkan variabel anggota untuk menyimpan TextView.

```
private TextView mTextView;
```

2. Dalam metode `onCreate()`, inisialisasi `mTextView` ke TextView dalam UI.
3. Tambahkan kode ke metode `startTask()` untuk membuat instance `SimpleAsyncTask`, meneruskan TextView `mTextView` ke konstruktur.
4. Panggil `execute()` pada instance `SimpleAsyncTask` tersebut.

Catatan: Metode `execute()` adalah tempat Anda meneruskan dalam parameter (dipisahkan oleh koma) yang kemudian diteruskan ke `doInBackground()` oleh sistem. Karena AsyncTask tidak memiliki parameter, Anda akan mengosongkannya.

5. Perbarui TextView untuk menampilkan teks "Napping..."

```
public void startTask (View view) {
    // Put a message in the text view
    mTextView.setText("Napping... ");

    // Start the AsyncTask.
    // The AsyncTask has a callback that will update the text view.
    new SimpleAsyncTask(mTextView).execute();
}
```

Kode Solusi untuk MainActivity:

```

package android.example.com.simpleasynctask;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    // The TextView where we will show results
    TextView mTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Initialize mTextView
        mTextView = (TextView) findViewById(R.id.textView1);

    }

    public void startTask (View view) {
        // Put a message in the text view
        mTextView.setText("Napping... ");

        // Start the AsyncTask.
        // The AsyncTask has a callback that will update the text view.
        new SimpleAsyncTask(mTextView).execute();
    }
}

```

3.2 Mengimplementasikan onSaveInstanceState()

1. Jalankan aplikasi dan klik tombol **Start Task**. Berapa lama aplikasi tertidur?
2. Klik tombol **Start Task** lagi, dan saat aplikasi tertidur, putar perangkat. Jika tugas latar belakang selesai sebelum Anda bisa memutar ponsel, coba lagi. Atau, Anda bisa memperbarui kode dan membuatnya tertidur untuk jangka waktu yang lebih lama.

Catatan: Anda akan melihat bahwa ketika perangkat diputar, TextView akan disetel ulang ke konten awalnya dan AsyncTask tidak bisa memperbarui TextView.

Ada beberapa hal yang terjadi di sini:

- Saat Anda memutar perangkat, sistem akan memulai ulang aplikasi, memanggil `onDestroy()` lalu `onCreate()`, yang memulai ulang siklus hidup aktivitas. Karena AsyncTasks tidak lagi terhubung ke siklus hidup aplikasi dan tidak bisa terhubung kembali ke aktivitas.
- AsyncTasks akan terus berjalan hingga selesai di latar belakang, mengonsumsi sumber daya sistem, tetapi tidak pernah menampilkan hasil di UI, yang disetel ulang di `onCreate()`. AsyncTasks tidak akan pernah bisa memperbarui TextView yang diteruskan kepadanya, karena TextView tertentu juga telah dimusnahkan. Akhirnya, sistem akan kehabisan sumber daya dan akan gagal.
- Bahkan tanpa AsyncTask, pemutaran perangkat akan menyetel ulang semua elemen UI ke status default-nya, yang bagi TextView mengimplikasikan string tertentu yang Anda setel dalam file `activity_main.xml`.

Karena alasan ini, AsyncTasks tidak cocok untuk tugas yang mungkin terganggu oleh pemusnahan Aktivitas. Dalam kasus penggunaan di mana hal ini sangat penting, Anda bisa menggunakan tipe kelas berbeda bernama Loader, yang akan Anda implementasikan dalam praktik selanjutnya.

Untuk mencegah TextView disetel ulang ke string awalnya, Anda perlu menyimpan statusnya. Anda sudah belajar cara memelihara status tampilan pada praktik sebelumnya menggunakan kelas `SavedInstanceState`.

Sekarang Anda akan mengimplementasikan `onSaveInstanceState()` untuk mempertahankan konten TextView saat aktivitas secara spontan dimusnahkan.

Catatan: Tidak semua penggunaan AsyncTask mewajibkan Anda untuk menangani status tampilan pada rotasi.

Aplikasi ini menggunakan TextView untuk menampilkan hasil aplikasi, sehingga mempertahankan status akan berguna. Dalam kasus lainnya, seperti saat mengunggah file, Anda mungkin tidak akan memerlukan informasi persisten dalam UI, sehingga mempertahankan status tidaklah penting.

3. Ganti metode `onSaveInstanceState()` dalam MainActivity untuk mempertahankan teks di dalam TextView saat aktivitas dimusnahkan:

```
outState.putString(TEXT_STATE, mTextView.getText().toString());
```

4. Ambil nilai TextView saat aktivitas dipulihkan dalam metode `onCreate()`.

```
// Restore TextView if there is a savedInstanceState
if(savedInstanceState!=null){
    mTextView.setText(savedInstanceState.getString(TEXT_STATE));
}
```

Kode Solusi untuk MainActivity:

```

package android.example.com.simpleasynctask;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;

/**
 * The SimpleAsyncTask app contains a button that launches an AsyncTask
 * which sleeps in the asynchronous thread for a random amount of time.
 */
public class MainActivity extends AppCompatActivity {

    //Key for saving the state of the TextView
    private static final String TEXT_STATE = "currentText";

    // The TextView where we will show results
    private TextView mTextView = null;

    /**
     * Initializes the activity.
     * @param savedInstanceState The current state data
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Initialize mTextView
        mTextView = (TextView) findViewById(R.id.textView1);

        // Restore TextView if there is a savedInstanceState
        if(savedInstanceState!=null){
            mTextView.setText(savedInstanceState.getString(TEXT_STATE));
        }
    }

    /**
     * Handles the onClick for the "Start Task" button. Launches the AsyncTask
     * which performs work off of the UI thread.
     *
     * @param view The view (Button) that was clicked.
     */
    public void startTask (View view) {
        // Put a message in the text view
        mTextView.setText(R.string.mapping);

        // Start the AsyncTask.
        // The AsyncTask has a callback that will update the textView.
        new SimpleAsyncTask(mTextView).execute();
    }

    /**
     * Saves the contents of the TextView to restore on configuration change.
     * @param outState The bundle in which the state of the activity is saved
     * when it is spontaneously destroyed.
     */
    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        // Save the state of the TextView
        outState.putString(TEXT_STATE, mTextView.getText().toString());
    }
}

```

Kode Solusi

Proyek Android Studio: [SimpleAsyncTask](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: AsyncTask menyediakan metode penggantian sangat berguna lainnya: `onProgressUpdate()`, yang mengizinkan Anda untuk memperbarui UI saat AsyncTask berjalan. Gunakan metode ini untuk memperbarui UI dengan waktu tidur saat ini. Periksa [dokumentasi AsyncTask](#) untuk melihat bagaimana `onProgressUpdate()` diimplementasikan dengan tepat. Ingatlah bahwa dalam definisi kelas AsyncTask, Anda akan perlu menentukan tipe data yang akan digunakan dalam metode `onProgressUpdate()`.

Rangkuman

- Hindari pekerjaan intensif sumber daya dalam thread UI yang akan membuat UI Anda lamban atau tidak menentu.
 - Kode apa pun yang tidak melibatkan penggambaran UI atau merespons masukan pengguna harus dipindahkan dari thread UI ke thread berbeda yang terpisah.
- AsyncTask adalah kelas Java abstrak yang memindahkan pemrosesan intensif ke thread yang terpisah.
 - AsyncTask harus dijadikan subkelas agar bisa digunakan.
 - AsyncTask memiliki 4 metode yang berguna: `onPreExecute()`, `doInBackground()`, `onPostExecute()` dan `onProgressUpdate()`.
- `doInBackground()` adalah satu-satunya metode yang dijalankan pada thread worker yang terpisah.
 - Anda tidak boleh memanggil metode UI dalam metode AsyncTask.
 - Metode AsyncTask lainnya berjalan dalam thread UI dan mengizinkan metode memanggil komponen UI.
- Memutar perangkat Android akan memusnahkan dan membuat ulang aktivitas Hal ini bisa memutuskan UI dari thread latar belakang, yang akan terus berjalan.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [AsyncTask dan AsyncTaskLoader](#)

Ketahui selengkapnya

Dokumentasi Developer Android

- [Proses dan thread](#)
- [Memproses Bitmaps di luar thread UI](#) menggunakan AsyncTask
- [AsyncTask](#)

Sumber daya lainnya

- <https://realm.io/news/android-threading-background-tasks/>

Video

- [Dasar-Dasar Kinerja Threading](#) oleh Performance Guru Colt McAnlis. Pelajari lebih lanjut tentang thread utama dan mengapa menjalankan tugas yang berjalan lama di thread utama sangat buruk.
- [Hunting AsyncTask yang Baik](#) oleh Colt McAnlis. Ketahui selengkapnya tentang AsyncTasks

7.2: Menghubungkan ke Internet dengan AsyncTask dan AsyncTaskLoader

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Tugas 1. Menjelajahi Books API](#)
- [Tugas 2. Membuat aplikasi "Who Wrote It?"](#)
- [Tugas 3. Mengimplementasikan praktik terbaik UI](#)
- [Tugas 4. Migrasi AsyncTaskLoader](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Dalam praktik ini Anda akan menggunakan AsyncTask untuk memulai tugas latar belakang yang mendapatkan data dari internet menggunakan REST API sederhana. Anda akan menggunakan [Google API Explorer](#) untuk belajar cara menanyakan Book Search API, mengimplementasikan kueri ini dalam thread worker menggunakan AsyncTask, dan menampilkan hasilnya dalam UI. Lalu Anda akan mengimplementasikan ulang tugas latar belakang yang sama menggunakan AsyncTaskLoader, yang akan lebih efisien dalam memperbarui UI, menangani masalah kinerja, dan meningkatkan keseluruhan UX.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Membuat sebuah aktivitas.
- Menambahkan TextView ke layout untuk aktivitas tersebut.
- Mengimplementasikan fungsionalitas onClick ke tombol dalam layout.
- Mengimplementasikan AsyncTask dan menampilkan hasilnya dalam UI.
- Meneruskan informasi di antara aktivitas sebagai ekstra.

Yang akan Anda PELAJARI

Dalam praktik ini Anda akan belajar:

- Menggunakan Google API Explorer untuk menginvestigasi API Google dan melihat respons JSON terhadap permintaan http.
- Menggunakan Books API sebagai contoh API mengambil data dari internet dan menjaga UI agar cepat dan responsif. Anda tidak akan mempelajari Books API secara detail dalam praktik ini. Aplikasi Anda hanya akan menggunakan fungsi penelusuran book paling sederhana. Untuk mengetahui selengkapnya tentang Books API lihat [dokumentasi referensi Books API](#).
- Parse hasil JSON yang dikembalikan dari kueri API.
- Mengimplementasikan AsyncTaskLoader yang mempertahankan data setelah perubahan konfigurasi.
- Memperbarui UI menggunakan callback Loader.

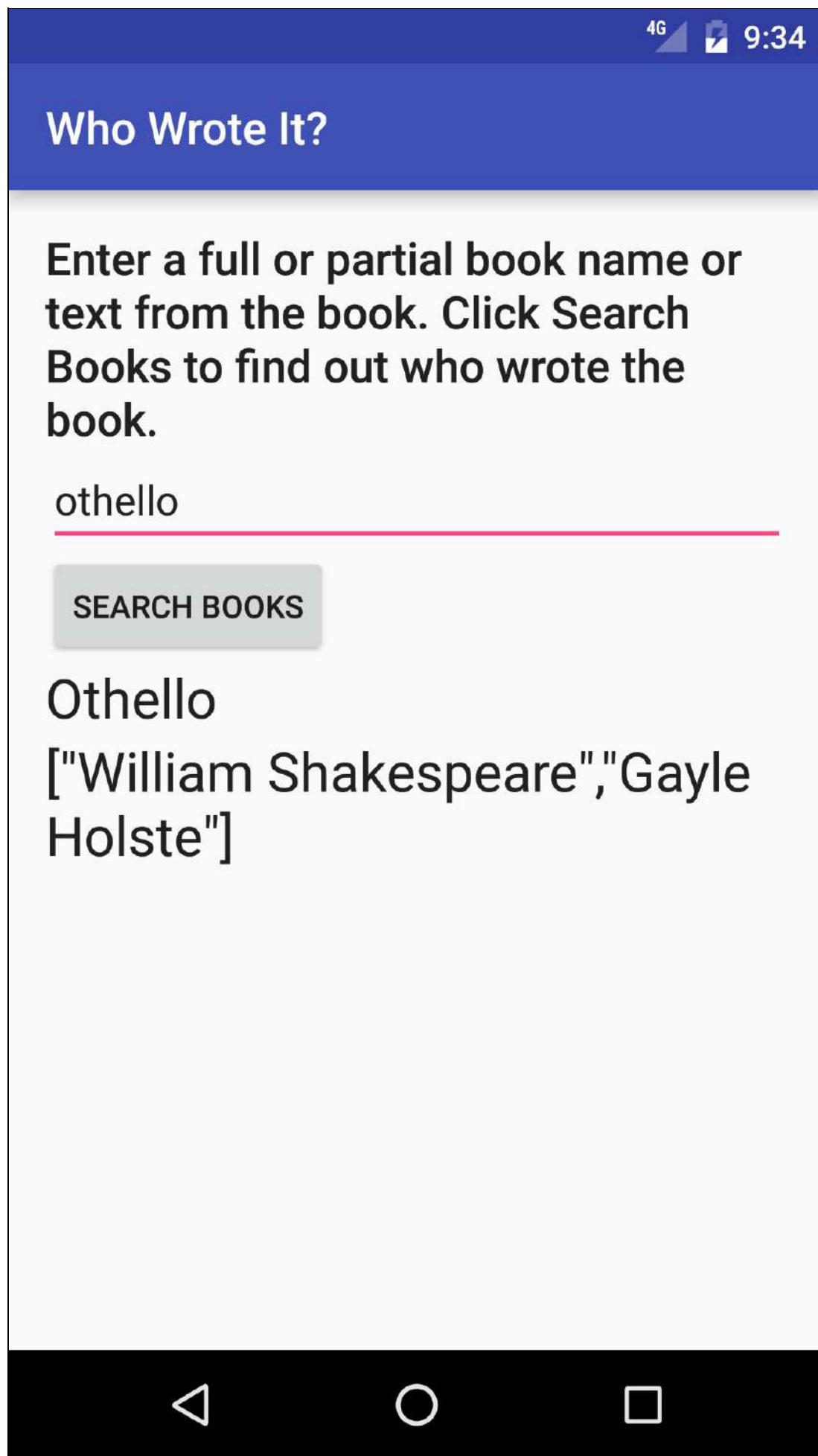
Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Menggunakan Google API Explorer untuk mempelajari tentang fitur penelusuran sederhana API Books.
- Membuat aplikasi "Who Wrote It?" yang menanyakan Books API menggunakan thread worker dan menampilkan hasilnya dalam UI.
- Memodifikasi aplikasi "Who Wrote it?" untuk menggunakan AsyncTaskLoader, sebagai ganti AsyncTask.

Ringkasan Aplikasi

Anda akan membangun aplikasi yang berisi bidang EditText dan Tombol. Pengguna memasukkan nama buku dalam EditText dan meneklik tombol. Tombol mengeksekusi AsyncTask yang menanyakan API Penelusuran Google Book untuk menemukan penulis dan judul buku yang dicari pengguna. Hasilnya diambil dan ditampilkan dalam bidang TextView di bawah tombol. Setelah aplikasi bekerja, Anda akan memodifikasi aplikasi untuk menggunakan [AsyncTaskLoader](#) sebagai ganti kelas [AsyncTask](#).



Tugas 1. Menjelajahi Books API

Dalam praktik ini Anda akan menggunakan API Google Books untuk mencari informasi tentang buku, seperti penulis dan judulnya. API Google Books menyediakan akses pemrograman ke layanan Penelusuran Google Book menggunakan REST API. Ini adalah layanan yang sama yang digunakan di belakang layar saat Anda mengeksekusi penelusuran secara manual di [Google Books](#). Anda bisa menggunakan Google API Explorer dan Penelusuran Google Book pada browser untuk memverifikasi bahwa aplikasi Android Anda mendapatkan hasil yang diharapkan.

1.1 Mengirimkan Permintaan API Books

1. Buka [Google APIs Explorer](#) (bisa ditemukan di <https://developers.google.com/apis-explorer/>).
2. Klik **Books API**.
3. Temukan (**Ctrl-F** atau **Cmd-F**) **books.volumes.list** dan klik nama fungsi tersebut. Anda seharusnya bisa melihat laman web yang mencantumkan berbagai parameter fungsi API Books yang melakukan penelusuran buku.
4. Dalam bidang `q` masukkan nama buku, atau sebagain nama buku. Parameter `q` adalah satu-satunya bidang yang diwajibkan.
5. Gunakan bidang `maxResults` dan `printType` untuk membatasi hasil ke 10 buku yang cocok yang dicetak. Bidang `maxResults` mengambil nilai integer yang membatasi jumlah hasil per kueri. Bidang `printType` mengambil satu dari tiga argumen string: `all`, yang tidak membatasi hasil menurut tipe sama sekali; `books`, yang hanya memberikan hasil buku dalam bentuk cetak; dan `magazines` yang memberikan hasil majalah saja.
6. Pastikan switch "Authorize requests using OAuth 2.0" di bagian atas formulir dinonaktifkan. Klik **Execute without OAuth** di bagian bawah formulir.
7. Gulir ke bawah untuk melihat Permintaan dan Respons.

Bidang `Request` adalah contoh Uniform Resource Identifier (URI). URI adalah string yang memberikan nama atau menemukan sumber daya tertentu. URL adalah tipe URI tertentu untuk mengidentifikasi dan menemukan sumber daya web. Untuk API Books, permintaannya adalah URL yang berisi penelusuran sebagai parameter (mengikuti parameter `q`). Perhatikan bidang kunci API setelah bidang kueri. Untuk alasan keamanan, saat mengakses API publik, Anda biasanya perlu mendapatkan kunci API dan menyertakannya di dalam Permintaan Anda. Namun, API spesifik ini tidak memerlukan kunci, sehingga Anda dapat meninggalkan porsi URI Permintaan dalam aplikasi.

1.2 Menganalisis Respons API Books

Di bagian bawah laman Anda bisa melihat Respons terhadap kueri. Respons menggunakan [format JSON](#), yang merupakan format umum untuk respons kueri API. Dalam laman web API Explorer, kode JSON diformat dengan baik agar dapat dibaca oleh manusia. Dalam aplikasi Anda, respons JSON akan dikembalikan dari layanan API sebagai string tunggal, dan Anda perlu melakukan parsing pada string tersebut untuk mengesektrik informasi yang diperlukan.

1. Dalam bagian Respons, temukan nilai untuk kunci "title". Perhatikan bahwa hasil ini memiliki nilai dan kunci tunggal.
2. Temukan nilai untuk kunci "authors". Perhatikan bahwa kunci ini berisi larik nilai.
3. Dalam praktik ini Anda hanya akan mengembalikan judul dan penulis item pertama.

Tugas 2: Membuat "Who Wrote it?" Aplikasi

Setelah Anda familier dengan metode API Books yang akan digunakan, sekarang waktunya menyiapkan layout aplikasi.

2.1 Membuat proyek dan antarmuka pengguna

1. Buat proyek aplikasi bernama **Who Wrote it?** dengan satu aktivitas, menggunakan Template Empty Activity.
2. Tambahkan elemen UI berikut di dalam file XML, menggunakan LinearLayout vertikal sebagai tampilan root—tampilan yang berisi semua tampilan lain di dalam file XML layout. Pastikan LinearLayout menggunakan `android:orientation="vertical"`:

| Tampilan | Atribut | Nilai |
|----------|-------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| TextView | android:layout_width android:layout_height android:id android:text android:textAppearance | wrap_content wrap_content @+id/instructions @string/instructions @style/TextAppearance.AppCompat.Title |
| EditText | android:layout_width android:layout_height android:id android:inputType android:hint | match_parent wrap_content @+id/bookInput text @string/input_hint |
| Button | android:layout_width android:layout_height android:id android:text android:onClick | wrap_content wrap_content @+id/searchButton @string/button_text searchBooks |
| TextView | android:layout_width android:layout_height android:id android:textAppearance | wrap_content wrap_content @+id/titleText @style/TextAppearance.AppCompat.Headline |
| TextView | android:layout_width android:layout_height android:id android:textAppearance | wrap_content wrap_content @+id/authorText @style/TextAppearance.AppCompat.Headline |

3. Dalam file strings.xml, tambahkan sumber daya string berikut ini:

```
<string name="instructions">Enter a book name, or part of a  
book name, or just some text from a book to find  
the full book title and who wrote the book!</string>  
<string name="button_text">Search Books</string>  
<string name="input_hint">Enter a Book Title</string>
```

4. Buat metode bernama `searchBooks()` dalam MainActivity.java untuk menangani tindakan tombol onClick. Seperti pada semua metode onClick, yang satu ini memerlukan `View` sebagai parameter.

2.2 Menyiapkan Aktivitas Utama

Untuk menanyakan API Books, Anda perlu mendapatkan masukan pengguna dari EditText.

1. Dalam MainActivity.java, buat variabel anggota untuk EditText, TextView penulis dan TextView judul.
2. Inisialisasi variabel ini dalam `onCreate()`.
3. Dalam metode `searchBooks()`, dapatkan teks dari widget EditText dan konversikan ke `String`, menetapkannya ke variabel `string`.

```
String queryString = mBookInput.getText().toString();
```

Catatan: `mBookInput.getText()` mengembalikan jenis data "Editable" yang perlu dikonversi menjadi string.

2.3 Membuat AsyncTask kosong

Sekarang Anda siap untuk terhubung ke internet dan menggunakan Book Search REST API. Konektivitas jaringan terkadang lamban atau mengalami penundaan. Hal ini bisa menyebabkan aplikasi menjadi tidak menentu dan lambat, jadi sebaiknya Anda tidak membuat koneksi jaringan pada thread UI. Jika Anda mencoba koneksi jaringan pada thread UI, Waktu Proses Android mungkin akan mengeluarkan `NetworkOnMainThreadException` untuk memperingatkan Anda bahwa ini bukanlah ide yang baik.

Gunakan AsyncTask untuk membuat koneksi jaringan:

- Buat kelas Java baru bernama `FetchBook` dalam **aplikasi/java** yang diperluas `AsyncTask`. `AsyncTask` memerlukan tiga argumen:
 - Parameter masukan.
 - Indikator kemajuan.
 - Jenis hasil.
 Parameter tipe generik untuk tugas adalah `<String, Void, String>` since the `AsyncTask` takes a `String` as the first parameter (the query), `Void` since there is no progress update, and `String` karena parameter ini mengembalikan string sebagai hasilnya (respons JSON).
- Implementasikan metode yang diperlukan, `doInBackground()`, dengan meletakkan kursor pada teks yang digarisbawahi merah, menekan **Alt + Enter (Opt + Enter di Mac)** dan memilih **Implement methods**. Pilih `doInBackground()` dan klik **OK**. Pastikan parameternya dan kembalikan tipe yang jenisnya benar (Ini memerlukan larik String dan mengembalikan String).
 - Klik menu **Code** dan pilih **Override methods** (atau tekan **Ctrl + O**). Pilih metode `onPostExecute()`. Metode `onPostExecute()` mengambil `String` sebagai parameter dan mengembalikan `void`.
- Untuk menampilkan hasil dalam `TextView`, Anda harus memiliki akses ke `TextView` yang ada di dalam `AsyncTask`. Buat variabel anggota dalam `FetchBook` `AsyncTask` untuk dua `TextView` yang menunjukkan hasilnya, dan inisialisasi keduanya dalam konstruktur. Anda akan menggunakan konstruktor ini dalam `MainActivity` untuk meneruskan `TextView` ke `AsyncTask`.

Kode solusi untuk FetchBook:

```
public class FetchBook extends AsyncTask<String,Void,String>{
    private TextView mTitleText;
    private TextView mAauthorText;

    public FetchBook(TextView mTitleText, TextView mAauthorText) {
        this.mTitleText = mTitleText;
        this_mAauthorText = mAauthorText;
    }

    @Override
    protected String doInBackground(String... params) {
        return null;
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
    }
}
```

2.4 Membuat kelas NetworkUtils dan membangun URI

Dalam langkah ini, Anda akan membuka koneksi internet dan menanyakan API Books. Bagian ini memiliki banyak kode, jadi ingat untuk membuka dokumentasi developer untuk [Menyambungkan ke Jaringan](#) jika Anda merasa buntu. Anda akan menulis kode untuk terhubung ke internet dalam kelas helper bernama `NetworkUtils`.

- Buat kelas Java baru bernama NetworkUtils dengan mengeklik **File > New > Java Class** dan hanya mengisi bidang "Name".

- Buat variabel `LOG_TAG` unik untuk digunakan di semua kelas NetworkUtils untuk membuat catatan log:

```
private static final String LOG_TAG = NetworkUtils.class.getSimpleName();
```

- Buat metode statis baru bernama `getBookInfo()` yang mengambil `String` sebagai parameter (yang akan menjadi istilah penelusuran) dan mengembalikan `String` (respons String JSON dari API yang Anda periksa sebelumnya).

```
static String getBookInfo(String queryString){}
```

- Buat dua variabel lokal berikut dalam `getBookInfo()` yang akan dibutuhkan nanti untuk membantu menyambungkan dan membaca data yang datang.

```
HttpURLConnection urlConnection = null;
BufferedReader reader = null;
```

- Buat variabel lokal lain di akhir `getBookInfo()` untuk memasukkan respons mentah dari kueri dan mengembalikannya:

```
String bookJSONString = null;
return bookJSONString;
```

Jika Anda ingat permintaan dari laman web API Books, Anda akan memperhatikan bahwa semua permintaan dimulai dengan URI yang sama. Untuk menentukan tipe sumber daya, Anda menambahkan parameter kueri ke URI basis. Memisahkan semua parameter kueri ini ke dalam konstanta dan mengombinasikannya menggunakan `Uri.Builder` adalah praktik biasa agar bisa digunakan lagi untuk URI yang berbeda. Kelas Uri memiliki metode yang mudah, `Uri.buildUpon()` yang mengembalikan `Uri.Builder` yang bisa kita gunakan.

Untuk aplikasi ini, Anda akan membatasi jumlah dan jenis hasil yang dikembalikan untuk meningkatkan kecepatan kueri. Untuk membatasi kueri, Anda hanya mencari buku yang dicetak.

- Buat konstanta anggota berikut dalam kelas NetworkUtils:

```
private static final String BOOK_BASE_URL = "https://www.googleapis.com/books/v1/volumes?"; // Base URI for the Books API
private static final String QUERY_PARAM = "q"; // Parameter for the search string
private static final String MAX_RESULTS = "maxResults"; // Parameter that limits search results
private static final String PRINT_TYPE = "printType"; // Parameter to filter by print type
```

- Buat blok try/catch/finally skeleton dalam `getBookInfo()`. Di sinilah Anda akan membuat permintaan HTTP. Kode untuk membangun URI dan mengeluarkan kueri akan masuk ke dalam blok try. Blok catch digunakan untuk menangani masalah apa pun dengan membuat permintaan HTTP dan blok finally untuk menutup koneksi jaringan setelah Anda selesai menerima data JSON dan mengembalikan hasilnya.

```
try {
...
} catch (Exception ex) {
...
} finally {
    return bookJSONString;
}
```

- Bangun URI permintaan dalam blok try:

```
//Build up your query URI, limiting results to 10 items and printed books
Uri builtURI = Uri.parse(BOOK_BASE_URL).buildUpon()
    .appendQueryParameter(QUERY_PARAM, queryString)
    .appendQueryParameter(MAX_RESULTS, "10")
    .appendQueryParameter(PRINT_TYPE, "books")
    .build();
```

9. Konversi URI ke URL:

```
URL requestURL = new URL(builtURI.toString());
```

2.5 Membuat Permintaan

Membuat permintaan API melalui internet adalah hal yang cukup umum. Karena Anda mungkin akan menggunakan fungsionalitas ini lagi, Anda mungkin ingin membuat kelas utilitas dengan fungsionalitas ini atau mengembangkan subkelas yang berguna untuk kemudahan sendiri. Permintaan API ini menggunakan kelas `HttpURLConnection` yang dikombinasikan dengan `InputStream` dan `StringBuffer` untuk mendapatkan respons JSON dari web. Jika pada satu sisi prosesnya gagal dan `InputStream` atau `StringBuffer` kosong, proses akan mengembalikan null yang menandakan bahwa kueri gagal.

1. Dalam blok try metode `getBookInfo()`, buka koneksi URL dan buat permintaan.

```
urlConnection = (HttpURLConnection) requestURL.openConnection();
urlConnection.setRequestMethod("GET");
urlConnection.connect();
```

2. Baca respons menggunakan `InputStream` dan `StringBuffer`, lalu konversikan ke `String`:

```
InputStream inputStream = urlConnection.getInputStream();
StringBuffer buffer = new StringBuffer();
if (inputStream == null) {
    // Nothing to do.
    return null;
}
reader = new BufferedReader(new InputStreamReader(inputStream));
String line;
while ((line = reader.readLine()) != null) {
    /* Since it's JSON, adding a newline isn't necessary (it won't affect
       parsing) but it does make debugging a *lot* easier if you print out the
       completed buffer for debugging. */
    buffer.append(line + "\n");
}
if (buffer.length() == 0) {
    // Stream was empty. No point in parsing.
    return null;
}
bookJSONString = buffer.toString();
```

3. Tutup blok try dan log pengecualiannya dalam blok catch.

```
catch (IOException e) {
    e.printStackTrace();
    return null;
}
```

4. Tutup kedua `urlConnection` dan variabel pembaca dalam blok finally:

```
finally {
    if (urlConnection != null) {
        urlConnection.disconnect();
    }
    if (reader != null) {
        try {
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Catatan: Setiap kali koneksi gagal, kode ini akan mengembalikan null. Ini berarti `onPostExecute()` harus memeriksa parameter masukannya untuk string null dan memungkinkan pengguna mengetahui bahwa koneksi gagal. Strategi

penanganan kesalahan ini sederhana, karena pengguna tidak tahu mengapa koneksi gagal. Solusi yang lebih baik untuk aplikasi produksi adalah menangani setiap poin kesalahan secara berbeda agar pengguna mendapatkan masukan yang tepat.

5. Log nilai variabel `bookJSONString` sebelum mengembalikannya. Sekarang Anda sudah selesai dengan metode `getBookInfo()`.

```
Log.d(LOG_TAG, bookJSONString);
```

6. Dalam metode `AsyncTask doInBackground()`, panggil metode `getBookInfo()`, meneruskan istilah penelusuran yang Anda dapatkan dari argumen `params` yang diteruskan oleh sistem (ini adalah nilai pertama dalam larik `params`). Kembalikan hasil dari metode ini dalam metode `doInBackground()`:

```
return NetworkUtils.getBookInfo(params[0]);
```

7. Sekarang setelah `AsyncTask` disiapkan, Anda perlu meluncurkannya dari `MainActivity` menggunakan metode `execute()`. Tambahkan kode berikut ke metode `searchBooks()` dalam `MainActivity.java` untuk meluncurkan `AsyncTask`:

```
new FetchBook(mTitleText, mAuthorText).execute(mQueryString);
```

8. Jalankan aplikasi Anda. Eksekusi penelusuran. Aplikasi Anda akan crash. Lihat Log untuk memeriksa apa yang menyebabkan kesalahan. Anda seharusnya melihat baris berikut:

```
Caused by: java.lang.SecurityException: Permission denied (missing INTERNET permission?)
```

Kesalahan ini menunjukkan bahwa Anda belum menyertakan izin untuk mengakses internet dalam file `AndroidManifest.xml`. Terhubung ke internet menimbulkan masalah keamanan baru, karena itulah aplikasi Anda tidak memiliki koneksi secara default. Anda harus menambahkan izin secara manual dalam bentuk tag `<uses-permission>` dalam `AndroidManifest.xml`.

2.6 Menambahkan izin internet

1. Buka file `AndroidManifest.xml`.
2. Semua izin aplikasi harus diletakkan dalam file `AndroidManifest.xml` di luar tag `<application>`. Anda harus memastikan untuk mengikuti urutan tempat tag didefinisikan dalam `AndroidManifest.xml`.
3. Tambahkan tag xml berikut di luar tag `<application>`:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

4. Bangun dan jalankan aplikasi Anda lagi. Menjalankan kueri seharusnya menghasilkan string JSON dicetak ke Log.

2.7 Parse string JSON

Sekarang Anda memiliki respons yang benar untuk kueri, Anda harus melakukan parsing kepada hasil untuk mengekstrak informasi yang ingin Anda tampilkan dalam UI. Untungnya, Java memiliki kelas yang sudah ada yang membantu parsing dan menangani data jenis JSON. Proses ini dan pembaruan UI akan terjadi dalam metode `onPostExecute()`.

Ada kemungkinan metode `doInBackground()` tidak mengembalikan string JSON yang diharapkan. Misalnya, try catch mungkin gagal dan mengeluarkan pengecualian, jaringan mungkin sudah habis waktunya atau kesalahan yang tidak dapat ditangani mungkin terjadi. Pada kasus-kasus ini, metode JSON akan gagal melakukan parsing pada data dan akan mengeluarkan pengecualian. Karena itu Anda harus melakukan parsing dalam blok try, dan blok catch harus menangani kasus di mana data yang tidak lengkap dan tidak benar dikembalikan.

Untuk melakukan parsing pada data JSON dan menangani pengecualian yang mungkin, lakukan hal berikut:

1. Dalam `onPostExecute()`, tambahkan blok try/catch di bawah panggilan ke `super`.

2. Gunakan kelas JSON Java bawaan (`JSONObject` dan `JSONArray`) untuk mendapatkan larik JSON item hasil dalam blok try.

```
JSONObject jsonObject = new JSONObject(s);
JSONArray itemsArray = jsonObject.getJSONArray("items");
```

3. Iterasi melalui `itemsArray`, memeriksa judul dan informasi penulis setiap buku. Jika keduanya bukan null, keluar dari loop dan perbarui UI; jika tidak, terus periksa daftarnya. Dengan cara ini, hanya entri dengan judul dan penulis yang akan ditampilkan.

```
//Iterate through the results
for(int i = 0; i<itemsArray.length(); i++){
    JSONObject book = itemsArray.getJSONObject(i); //Get the current item
    String title=null;
    String authors=null;
    JSONObject volumeInfo = book.getJSONObject("volumeInfo");

    try {
        title = volumeInfo.getString("title");
        authors = volumeInfo.getString("authors");
    } catch (Exception e){
        e.printStackTrace();
    }

    //If both a title and author exist, update the TextViews and return
    if (title != null && authors != null){
        mTitleText.setText(title);
        mAuhorText.setText(authors);
        return;
    }
}
```

4. Jika tidak ada hasil yang memenuhi kriteria memiliki penulis dan judul yang valid, setel `TextView` judul untuk membaca "No Results Found", dan hapus `TextView` `authors`.
5. Dalam blok catch, cetak kesalahan ke log, setel `TextView` judul ke "No Results Found", dan hapus `TextView` `authors`.

Kode solusi:

```

//Method for handling the results on the UI thread
@Override
protected void onPostExecute(String s) {
    super.onPostExecute(s);
    try {
        JSONObject jsonObject = new JSONObject(s);
        JSONArray itemsArray = jsonObject.getJSONArray("items");
        for(int i = 0; i<itemsArray.length(); i++){
            JSONObject book = itemsArray.getJSONObject(i);
            String title=null;
            String authors=null;
            JSONObject volumeInfo = book.getJSONObject("volumeInfo");

            try {
                title = volumeInfo.getString("title");
                authors = volumeInfo.getString("authors");
            } catch (Exception e){
                e.printStackTrace();
            }

            if (title != null && authors != null){
                mTitleText.setText(title);
                mAuthorText.setText(authors);
                return;
            }
        }

        mTitleText.setText("No Results Found");
        mAuthorText.setText("");

    } catch (Exception e){
        mTitleText.setText("No Results Found");
        mAuthorText.setText("");
        e.printStackTrace();
    }
}

```

Tugas 3. Mengimplementasikan praktik terbaik UI

Anda sekarang memiliki aplikasi yang berfungsi dan menggunakan API Books untuk mengeksekusi penelusuran buku. Tetapi, ada beberapa hal yang tidak berjalan seperti yang diharapkan:

- Saat pengguna menekan tombol **Search Books**, keyboard tidak muncul, dan tidak ada indikasi bagi pengguna bahwa kueri sebenarnya sedang dieksekusi.
- Jika tidak ada koneksi jaringan, atau bidang penelusuran kosong, aplikasi masih mencoba menanyakan API dan gagal tanpa memperbarui UI dengan benar.
- Jika Anda memutar layar selama kueri, AsyncTask akan terputus koneksinya dari Aktivitas, dan tidak dapat memperbarui UI dengan hasilnya.

Anda akan memperbaiki masalah ini pada bagian selanjutnya.

3.1 Menyembunyikan Keyboard dan Memperbarui TextView

Pengalaman pengguna penelusuran tidak intuitif. Ketika tombol ditekan, keyboard tetap terlihat dan kita tidak akan mengetahui bahwa kueri sedang berjalan. Salah satu dari solusinya adalah dengan secara terprogram menyembunyikan keyboard dan memperbarui salah satu TextView hasil untuk membaca "Loading..." saat kueri sedang dikerjakan. Untuk menggunakan solusi ini, Anda bisa:

1. Menambahkan kode berikut ke metode `searchBooks()` untuk menyembunyikan keyboard saat tombol ditekan:

```
InputMethodManager inputManager = (InputMethodManager)
        getSystemService(Context.INPUT_METHOD_SERVICE);
inputManager.hideSoftInputFromWindowgetCurrentFocus().getWindowToken(),
        InputMethodManager.HIDE_NOT_ALWAYS);
```

2. Menambahkan satu baris kode di bawah panggilan untuk mengeksekusi tugas FetchBook yang mengubah TextView judul untuk membaca "Loading..." dan menghapus TextView penulis.
3. Mengekstrak sumber daya String.

3.2 Mengelola status jaringan dan kasus bidang penelusuran kosong

Kapan pun aplikasi menggunakan jaringan, aplikasi itu perlu menangani kemungkinan koneksi jaringan tidak tersedia. Sebelum mencoba terhubung ke jaringan dalam AsyncTask atau AsyncTaskLoader, aplikasi harus memeriksa status koneksi jaringan.

1. Modifikasi metode `searchBooks()` untuk memeriksa kedua koneksi jaringan dan apakah ada teks dalam bidang penelusuran sebelum mengeksekusi tugas FetchBook.
2. Perbarui UI dalam kasus tidak ada koneksi internet atau tidak ada teks dalam bidang penelusuran. Tampilkan penyebab kesalahan dalam TextView.

Kode solusi:

```
public void searchBooks(View view) {
    String queryString = mBookInput.getText().toString();

    InputMethodManager inputManager = (InputMethodManager)
        getSystemService(Context.INPUT_METHOD_SERVICE);
    inputManager.hideSoftInputFromWindowgetCurrentFocus().getWindowToken(),
        InputMethodManager.HIDE_NOT_ALWAYS);

    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();

    if (networkInfo != null && networkInfo.isConnected() && queryString.length()!=0) {
        new FetchBook(mTitleText, mAuthorText).execute(queryString);
        mAuthorText.setText("");
        mTitleText.setText(R.string.loading);
    }

    else {
        if (queryString.length() == 0) {
            mAuthorText.setText("");
            mTitleText.setText("Please enter a search term");
        } else {
            mAuthorText.setText("");
            mTitleText.setText("Please check your network connection and try again.");
        }
    }
}
```

Tugas 4. Migrasi ke AsyncTaskLoader

Saat menggunakan AsyncTask, AsyncTask tidak bisa memperbarui UI jika perubahan konfigurasi terjadi saat tugas latar belakang sedang berjalan. Untuk mengatasi situasi ini, Android SDK menyediakan serangkaian kelas bernama loader yang didesain secara spesifik untuk memuat data ke dalam UI secara asinkron. Jika Anda menggunakan loader, Anda tidak perlu khawatir loader akan kehilangan kemampuannya untuk memperbarui UI dalam aktivitas yang awalnya membuatnya. Framework Loader tidak bekerja bagi Anda dengan mengasosiasikan ulang loader dengan Aktivitas yang tepat saat perangkat mengubah konfigurasinya. Ini berarti jika Anda memutar perangkat saat tugas sedang berjalan, hasilnya akan ditampilkan dengan benar dalam Aktivitas setelah data dikembalikan.

Dalam tugas ini Anda akan menggunakan loader spesifik bernama [AsyncTaskLoader](#). AsyncTaskLoader adalah subkelas abstrak Loader dan menggunakan AsyncTask untuk memuat data di latar belakang dengan efisien.

Catatan: Saat menggunakan AsyncTask, Anda mengimplementasikan metode `onPostExecute()` dalam `AsyncTask` untuk menampilkan hasilnya di layar. Saat menggunakan AsyncTaskLoader, Anda mendefinisikan metode callback dalam aktivitas untuk menampilkan hasilnya.

Loader menyediakan banyak fungsionalitas tambahan, bukan hanya untuk menjalankan tugas dan menghubungkan kembali ke Aktivitas. Misalnya, Anda bisa melampirkan loader ke sumber data dan membuatnya secara otomatis memperbarui elemen UI saat data dasarnya berubah. Loader juga bisa diprogram untuk melanjutkan memuat jika terganggu.

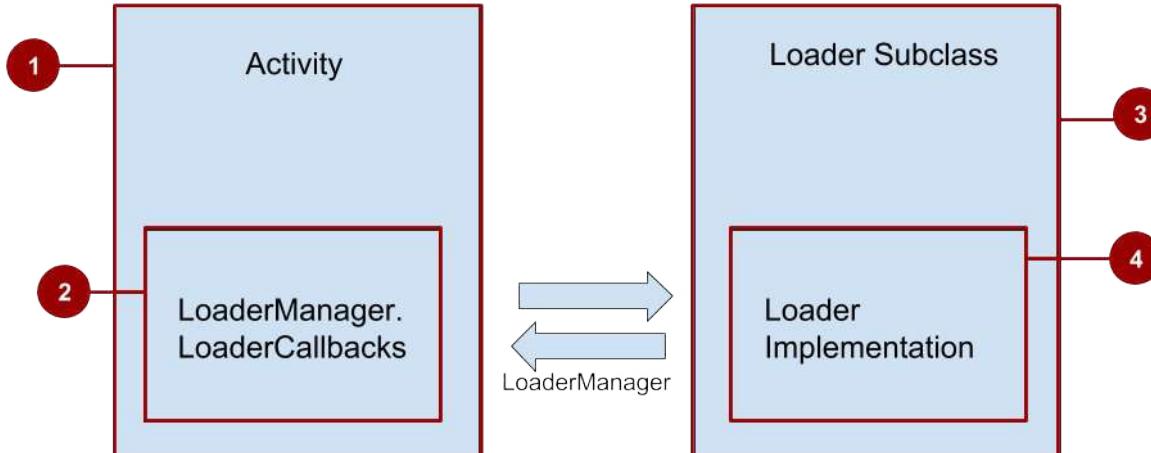
Mengapa harus menggunakan AsyncTask jika AsyncTaskLoader jauh lebih berguna? Jawabannya adalah tergantung situasi. Jika tugas latar belakang mungkin akan selesai sebelum perubahan konfigurasi terjadi, dan tidak perlu memperbarui UI, AsyncTask mungkin akan cukup. Framework Loader sebenarnya menggunakan AsyncTask di belakang layar untuk menjalankan fungsinya.

Prinsipnya, gunakan AsyncTaskLoader sebagai ganti AsyncTask jika pengguna mungkin akan memutar layar saat pekerjaan sedang berlangsung, atau jika UI perlu diperbarui saat pekerjaan selesai.

Dalam latihan ini Anda akan mempelajari cara menggunakan AsyncTaskLoader sebagai ganti AsyncTask untuk menjalankan kueri API Books. Anda akan mempelajari selengkapnya tentang penggunaan loader yang lain pada pelajaran berikutnya.

Mengimplementasikan Loader memerlukan komponen berikut:

- Kelas yang memperluas kelas Loader (dalam kasus ini, AsyncTaskLoader).
- Aktivitas yang mengimplementasikan kelas [LoaderManager.LoaderCallbacks](#).
- Instance [LoaderManager](#).



1. Aktivitas.
2. The `LoaderManager.LoaderCallbacks`.
3. Subkelas Loader.
4. Implementasi Loader.

LoaderManager secara otomatis memindahkan loader melalui siklus hidupnya tergantung status data dan Aktivitas. Misalnya, LoaderManager memanggil `onStartLoading()` saat loader diinisialisasi dan dimusnahkan saat Aktivitas dimusnahkan.

`LoaderManager.LoaderCallbacks` adalah serangkaian metode dalam Aktivitas yang dipanggil oleh LoaderManager saat loader dibuat, ketika data sudah selesai dimuat, dan saat loader disetel ulang. `LoaderCallbacks` bisa mengambil hasil tugas dan meneruskannya kembali ke UI Aktivitas.

Subkelas Loader berisi detail pemuat data, biasanya menggantikan paling tidak `onStartLoading()`. Subkelas ini juga berisi fitur tambahan seperti mengamati sumber data untuk perubahan dan caching data secara lokal.

Subkelas Loader mengimplementasikan metode callback siklus hidup Loader seperti `onStartLoading()`, `onStopLoading()` dan `onReset()`. Subkelas loader juga berisi metode `forceLoad()` yang menginisialisasi pemuat data. Metode ini tidak dipanggil secara otomatis saat loader dimulai karena beberapa penyiapan biasanya diperlukan sebelum pemuat dilakukan. Implementasi paling sederhana akan memanggil `forceLoad()` dalam `onStartLoading()` yang akan menyebabkan pemuat setiap LoaderManager memulai Loader.

4.1 Membuat AsyncTaskLoader

1. Salin [proyek WhoWroteIt](#), untuk mempertahankan hasil dari praktik sebelumnya. Ganti nama proyek yang disalin menjadi **WhoWroteItLoader**.
2. Buat kelas baru dalam direktori Java bernama BookLoader.
3. Buat agar kelas BookLoader memperluas AsyncTaskLoader dengan tipe berparameter ..
4. Pastikan Anda mengimpor loader dari Pustaka Dukungan v4.
5. Implementasikan metode yang diperlukan (`loadInBackground()`). Perhatikan kemiripan antara metode ini dan metode `doInBackground()` awal dengan AsyncTask.
6. Buat konstruktor untuk kelas baru Anda. Dalam Android Studio, kemungkinan deklarasi kelas akan tetap digarisbawahi dengan warna merah karena konstruktor tidak cocok dengan implementasi superkelas. Dengan adanya kursor pada baris deklarasi kelas, tekan **Alt + Enter (Option + Enter di Mac)** dan pilih **Create constructor matching super**. Ini akan membuat konstruktor dengan konteksnya sebagai parameter.

Define `onStartLoading()`

1. Tekan **Ctrl + O** untuk membuka metode Override, dan pilih **onStartLoading**. Metode ini dipanggil oleh sistem saat Anda memulai loader.
2. Loader tidak akan mulai memuat data sampai Anda memanggil metode `forceLoad()`. Di dalam stub metode `onStartLoading()`, panggil `forceLoad()` untuk memulai metode `loadInBackground()` saat Loader dibuat.

Definisikan `loadInBackground()`

1. Buat variabel anggota `mQueryString` yang akan menampung kueri `String`, dan modifikasi konstruktor untuk mengambil `String` sebagai argumen dan menetapkannya ke variabel `mQueryString`.
2. Dalam metode `loadInBackground()`, panggil metode `getBookInfo()` yang meneruskan `mQueryString`, dan kembalikan hasilnya untuk mengunduh informasi dari API Books:

```
@Override
public String loadInBackground() {
    return NetworkUtils.getBookInfo(mQueryString);
}
```

4.2 Memodifikasi MainActivity

Sekarang Anda harus mengimplementasikan [Callback Loader](#) dalam MainActivity untuk menangani hasil dari metode `loadInBackground()` AsyncTaskLoader.

1. Tambahkan implementasi `LoaderManager.LoaderCallbacks` ke deklarasi kelas Aktivitas Utama, yang berparameter dengan tipe `String`:
- ```
public class MainActivity extends AppCompatActivity
 implements LoaderManager.LoaderCallbacks<String>{
```
2. Implementasikan metode yang diperlukan: `onCreateLoader()`, `onLoadFinished()`, `onLoaderReset()`. Letakkan kursor teks pada baris tanda tangan kelas dan masukkan **Alt + Enter (Option + Enter di Mac)**. Pastikan semua metode dipilih.

**Catatan:** Jika impor untuk `Loader` dan `LoaderManager` dalam MainActivity tidak cocok dengan impor untuk `AsyncTaskLoader` untuk kelas `BookLoader`, Anda akan mendapatkan beberapa kesalahan mengetik dalam callback.

Pastikan semua impor berasal dari Pustaka Dukungan Android.

Loader menggunakan kelas `Bundle` untuk meneruskan informasi dari aktivitas memanggil ke LoaderCallbacks. Anda bisa menambahkan data primitif ke bundel dengan metode `putType()` yang tepat.

Untuk memulai loader, Anda memiliki dua opsi:

- `initLoader()` : Metode ini membuat loader baru jika belum ada dan bergerak dalam Bundel argumen. Jika loader ada, Aktivitas memanggil akan dikaitkan kembali dengannya tanpa memperbarui Bundel.
- `restartLoader()` : Metode ini sama dengan `initLoader()` kecuali jika metode ini menemukan loader yang sudah ada, metode ini akan memusnahkan dan membuat ulang loader dengan Bundel yang baru.

Kedua metode didefinisikan dalam LoaderManager, yang mengelola semua instance Loader yang digunakan dalam Aktivitas (atau Fragmen). Setiap Aktivitas memiliki satu instance LoaderManager yang bertanggung jawab terhadap siklus hidup dan Loader yang dikelolanya.

Saat ini, FetchBook AsyncTask dipicu saat pengguna menekan tombol tersebut. Anda perlu memulai loader dengan Bundel baru setiap kali tombol ditekan. Untuk melakukannya, Anda perlu mengedit metode `onClick` untuk tombol itu.

1. Dalam metode `searchBooks()`, yang merupakan metode `onClick` untuk tombol tersebut, ganti panggilan untuk mengeksekusi tugas FetchBook dengan panggilan untuk `restartLoader()`, yang meneruskan string kueri yang Anda dapatkan dari `EditText` dalam Bundel:

```
Bundle queryBundle = new Bundle();
queryBundle.putString("queryString", queryString);
getSupportLoaderManager().restartLoader(0, queryBundle, this);
```

Metode `restartLoader()` mengambil tiga argumen:

- Loader `id` (berguna jika Anda mengimplementasikan lebih dari satu loader dalam aktivitas).
- Argumen `Bundle` (tempat data yang diperlukan oleh loader disimpan).
- Instance `LoaderCallbacks` yang Anda implementasikan dalam aktivitas. Jika ingin loader membawa hasil ke `MainActivity`, tetapkan `this` sebagai argumen ketiga.

2. Periksa metode `Override` dalam kelas `LoaderCallbacks`. Metode ini adalah:

- `onCreateLoader()` : Dipanggil saat Anda membuat instance Loader.
- `onLoadFinished()` : Dipanggil ketika tugas loader sudah selesai. Ini adalah tempat di mana Anda menambahkan kode untuk memperbarui UI dengan hasilnya.
- `onLoaderReset()` : Menghapus sumber daya yang tersisa.

Anda hanya akan mendefinisikan dua metode pertama, karena model data saat ini adalah string sederhana yang tidak membutuhkan perhatian ekstra saat loader disetel ulang.

### Implementasikan `onCreateLoader()`

1. Dalam `onCreateLoader()`, kembalikan instance kelas `BookLoader`, meneruskan `queryString` yang didapatkan dari Bundel argumen:

```
return new BookLoader(this, args.getString("queryString"));
```

### Implementasikan `onLoadFinished()`

1. Perbarui `onLoadFinished()` untuk memproses hasil, yang merupakan respons String JSON mentah dari API Books.
  - i. Salin kode dari `onPostExecute()` dalam kelas `FetchBook` ke to `onLoadFinished()` dalam `MainActivity`, dengan mengecualikan panggilan ke `super.onPostExecute()`.
  - ii. Ganti argumen ke konstruktor `JSONObject` dengan data `String` yang diteruskan.
2. Jalankan aplikasi Anda.

Anda seharusnya memiliki fungsionalitas yang sama seperti sebelumnya, hanya saja sekarang di dalam Loader! Satu hal yang masih tidak berfungsi. Saat perangkat diputar, data `view` hilang. Ini karena saat Aktivitas dibuat (atau dibuat ulang), Aktivitas tidak tahu bahwa ada loader yang sedang berjalan. Metode `initLoader()` dibutuhkan dalam `onCreate()` dari MainActivity untuk menghubungkan ulang loader.

3. Tambahkan kode berikut di `onCreate()` untuk menghubungkan ulang ke Loader jika sudah ada:

```
if(getSupportLoaderManager().getLoader(0)!=null){
 getSupportLoaderManager().initLoader(0,null,this);
}
```

**Catatan:** Jika loader ada, inisialisasikan loader. Anda hanya perlu mengaitkan kembali loader ke Aktivitas jika kueri sudah dieksekusi. Dalam status awal aplikasi, data tidak dimuat sehingga tidak ada yang perlu dipertahankan.

4. Jalankan aplikasi lagi dan putar perangkat. LoaderManager sekarang mempertahankan data di semua konfigurasi perangkat!
5. Hapus kelas FetchBook karena sudah tidak digunakan lagi.

## Kode solusi

Proyek Android Studio: [WhoWroteItLoader](#)

## Tantangan penyusunan kode

**Catatan:** Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

**Tantangan 1:** Jelajahi API spesifik yang Anda gunakan dalam detail yang lebih lengkap dan temukan parameter penelusuran yang membatasi hasil ke buku yang dapat diunduh dalam format epub. Tambahkan parameter ke permintaan Anda dan lihat hasilnya.

**Tantangan 2:** Respons dari API Books berisi hasil sebanyak yang Anda setel dengan parameter `maxResults`, namun dalam implementasi ini Anda sudah mengembalikan hasil Buku valid yang pertama. Modifikasi aplikasi Anda agar data ditampilkan dalam RecyclerView yang memiliki `maxResults` entri.

## Rangkuman

- Tugas yang menghubungkan ke jaringan, atau memerlukan waktu lebih untuk diproses seharusnya tidak dieksekusi pada thread UI.
  - Waktu Proses Android biasanya memiliki default StrictMode yang akan memunculkan pengecualian jika Anda mengupayakan koneksi jaringan atau akses file pada thread UI.
- Google API Explorer adalah alat yang membantu Anda menjelajahi berbagai API Google secara interaktif.
  - API Penelusuran Books adalah serangkaian API RESTful untuk mengakses Google Books secara terprogram.
  - Permintaan API ke Google Books bentuknya adalah URL.
  - Respons ke permintaan API tersebut mengembalikan string JSON.
- Gunakan `getText()` untuk mengambil teks dari tampilan EditText. Ini dapat dikonversikan ke dalam String sederhana menggunakan `toString()`.
- Kelas URI memiliki metode bantu, `Uri.buildUpon()` yang mengembalikan Uri.Builder yang dapat digunakan untuk membuat string URI.
- AsyncTask adalah kelas yang mengizinkan Anda menjalankan tugas di latar belakang, secara asinkron, sebagai ganti pada thread UI.
  - AsyncTask dapat dimulai melalui `execute()`.
  - AsyncTask tidak dapat memperbarui UI jika aktivitas yang dikontrolnya berhenti (seperti perubahan konfigurasi

- pada perangkat).
- AsyncTask harus dijadikan subkelas agar bisa digunakan. Subkelas akan mengganti paling tidak satu metode `doInBackground(Params)`, dan paling sering mengganti yang kedua `onPostExecute(Result)` juga.
  - Setiap kali dieksekusi, AsyncTask melalui 4 langkah:
    1. `onPreExecute()`. Dipanggil pada thread UI sebelum tugas dieksekusi. Langkah ini biasanya digunakan untuk menyiapkan tugas.
    2. `doInBackground(Params)`. Dipanggil pada thread latar belakang segera setelah `onPreExecute()` selesai mengeksekusi. Langkah ini digunakan untuk melakukan penghitungan latar belakang yang bisa memakan waktu lama.
    3. `onProgressUpdate(Progress)`. Dipanggil pada thread UI setelah Anda memanggil `doInBackground` ke `publishProgress(Progress)`.
    4. `onPostExecute(Result)`. Dipanggil pada thread UI setelah penghitungan latar belakang selesai. Hasil penghitungan latar belakang diteruskan ke metode ini sebagai parameter.
  - `AsyncTaskLoader` adalah Loader yang setara dengan AsyncTask. Loader ini menyediakan metode, `loadInBackground()`, yang berjalan pada thread terpisah dan yang hasilnya secara otomatis dibawa ke thread UI (ke callback `LoaderManager onLoadFinished()`).
  - Anda harus mengonfigurasi izin jaringan dalam file manifes Android untuk terhubung ke internet:

```
<uses-permission android:name="android.permission.INTERNET">
```

- Gunakan kelas JSON Java bawaan (`JSONObject` dan `JSONArray`) untuk membuat dan melakukan pada parsing string JSON.
- Loader mengizinkan pemuatan data dalam Aktivitas secara asinkron.
  - Loader dapat digunakan untuk membangun ulang komunikasi ke UI saat sebuah Aktivitas dihentikan sebelum tugas selesai (misalnya karena perangkat diputar).
  - AsyncTaskLoader adalah Loader yang menggunakan kelas helper AsyncTask di belakang layar untuk melakukan pekerjaan di latar belakang, di luar thread utama.
  - Loader dikelola oleh LoaderManager; satu atau beberapa Loader dapat ditetapkan dan dikelola oleh satu LoadManager.
  - LoaderManager mengizinkan Anda untuk menghubungkan Aktivitas yang baru dibuat dengan Loader menggunakan `getSupportLoaderManager().initLoader()`.

## Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Terhubung ke Internet dengan AsyncTask dan AsyncTaskLoader](#)

## Ketahui selengkapnya

[Dokumentasi Developer Android](#)

### Panduan

- [Terhubung ke Jaringan](#)
- [Mengelola Status Jaringan](#)
- [Loader](#)

### Referensi

- [AsyncTask](#)
- [AsyncTaskLoader](#)



## 7.3: Penerima Siaran

### Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Tugas 1. Menyiapkan Proyek PowerReceiver
- Tugas 2. Mengirimkan dan Menerima Siaran Khusus
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Beberapa peristiwa yang bisa terjadi dalam sistem Android mungkin berdampak pada fungsionalitas aplikasi yang dipasang pada perangkat. Misalnya, jika sistem sudah selesai booting, Anda mungkin ingin aplikasi cuaca memperbarui informasinya. Framework Android menanganinya dengan mengirimkan siaran sistem yang berisi Intent yang seharusnya diterima menggunakan [BroadcastReceivers](#). BroadcastReceiver adalah kelas dasar untuk kode yang akan menerima Intent yang dikirimkan oleh `sendBroadcast()`. Ada dua kelas utama siaran yang dapat diterima:

- Siaran normal (yang dikirimkan dengan [Context.sendBroadcast\(\)](#)) sangat asinkron. Semua penerima siaran dijalankan dalam urutan yang tidak ditentukan, sering kali pada saat yang bersamaan. Ini lebih efisien, namun ini berarti penerima tidak dapat menggunakan hasil atau membatalkan API yang disertakan di sini.
- Siaran pesanan (yang dikirim dengan [Context.sendOrderedBroadcast](#)) dikirim ke satu penerima pada satu waktu. Karena setiap penerima mengeksekusi secara bergantian, hal ini dapat menyebarkan hasil ke penerima selanjutnya, atau bisa membatalkan siaran agar tidak diteruskan ke penerima. Perintah yang dijalankan penerima dapat dikontrol dengan atribut [android:priority](#) dari filter intent yang cocok; penerima dengan prioritas yang sama akan dijalankan dalam urutan yang acak.

Bahkan dalam kasus siaran normal, dalam beberapa situasi sistem mungkin kembali menyampaikan siaran ke satu penerima di satu waktu. Secara khusus, untuk penerima yang mungkin memerlukan pembuatan proses, hanya satu yang akan dijalankan pada satu waktu agar sistem tidak kelebihan beban dengan proses yang baru. Namun, dalam situasi ini, semantik yang tidak diperintahkan bertahan: penerima ini tetap tidak bisa mengembalikan hasil atau menghentikan siarannya.

Selain itu, Anda bisa membuat Intent dengan tindakan khusus dan menyiarkannya sendiri dari aplikasi menggunakan metode `sendBroadcast()`. Broadcast akan diterima oleh semua aplikasi dengan BroadcastReceiver yang terdaftar untuk tindakan tersebut. Untuk mengetahui selengkapnya tentang Intent dan penerima Siaran, buka [dokumentasi Intent](#).

Catat bahwa meskipun kelas Intent digunakan untuk mengirim dan menerima siaran, mekanisme siaran Intent sepenuhnya terpisah dari Intent yang digunakan untuk memulai Aktivitas.

Dalam praktik ini, Anda akan membuat aplikasi yang merespons perubahan dalam status pengisian daya perangkat, dan mengirimkan serta menerima Intent Siaran khusus.

### Yang harus sudah Anda KETAHUI

Sebelum praktik ini Anda harus bisa:

- Mengidentifikasi bagian kunci file `AndroidManifest.xml`.
- Membuat Intent Implisit.

### Yang akan Anda PELAJARI

Dalam praktik ini Anda akan belajar:

- Membuat subkelas dan mengimplementasikan BroadcastReceiver.
- Mendaftar intent Siaran sistem.
- Membuat dan mengirim intent Siaran khusus.

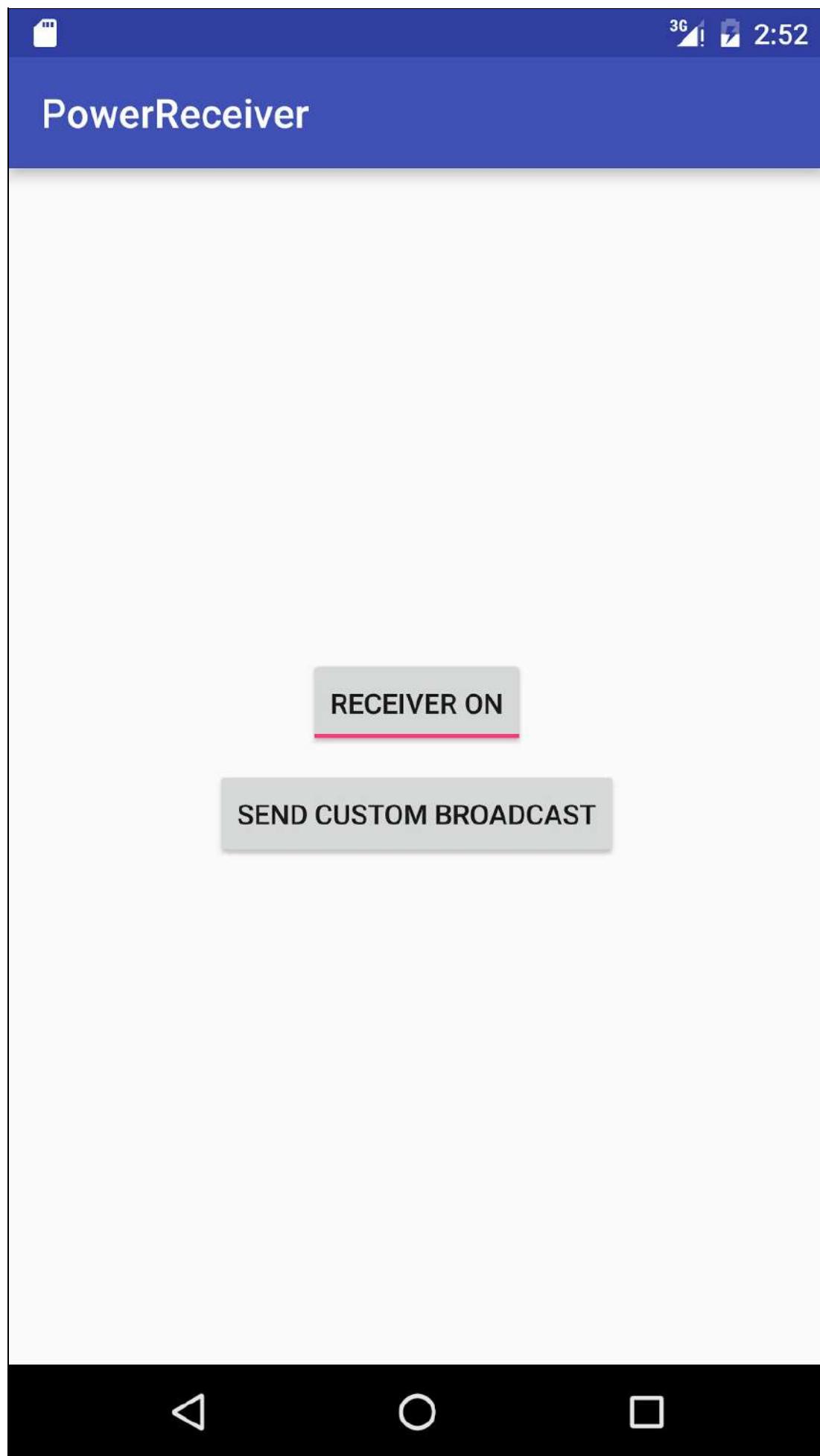
## Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Membuat subkelas BroadcastReceiver untuk menampilkan Toast saat siaran diterima.
- Mendaftarkan penerima untuk mendengarkan siaran sistem.
- Mengirimkan dan menerima intent siaran khusus.

## Ringkasan aplikasi

Aplikasi PowerReceiver akan mendaftarkan BroadcastReceiver yang menampilkan pesan Toast saat perangkat terhubung atau terputus dari aliran listrik. Aplikasi ini juga akan mengirimkan dan menerima Intent Broadcast khusus untuk menampilkan pesan Toast yang berbeda.



# Tugas 1. Menyiapkan Proyek PowerReceiver

## 1.1 Membuat Proyek

1. Membuat proyek baru bernama **PowerReceiver**, menerima opsi default dan menggunakan template Empty.
  2. Membuat Penerima Siaran baru. Pilih nama paket dalam Tampilan Proyek Android dan buka **File > New > Other > Broadcast Receiver**.
  3. Beri nama kelasnya **CustomReceiver** dan pastikan "Exported" dan "Enabled" dicentang.
- Catatan:** Fitur "Exporter" mengizinkan aplikasi untuk merespons ke siaran luar, sementara "Enabled" mengizinkannya dibuat instance oleh sistem.
4. Buka file manifes Android. Ingat bahwa Android Studio secara otomatis membuat tag `<receiver>` dengan opsi terpilih sebagai atribut. BroadcastReceivers juga bisa didaftarkan secara program, tetapi lebih mudah mendefinisikannya dalam manifes.

## 1.2 Mendaftarkan Penerima untuk siaran sistem

Untuk menerima siaran apa pun, pertama-tama Anda perlu menentukan intent siaran mana yang Anda minati. Dalam [dokumentasi Intent](#), di bawah "Standard Broadcast Actions", Anda bisa menemukan beberapa intent siaran umum yang dikirimkan oleh sistem. Dalam aplikasi ini, Anda akan memperhatikan dua siaran tertentu: `ACTION_POWER_CONNECTED` dan `ACTION_POWER_DISCONNECTED`. BroadcastReceivers mendaftarkan siaran seperti Anda mendaftarkan aktivitas untuk Intent implisit: Anda menggunakan filter intent. Anda sudah belajar tentang intent implisit pada praktik sebelumnya.

1. Dalam file **AndroidManifest.xml**, tambahkan kode berikut di antara tag `<receiver>` untuk mendaftarkan Penerima Anda untuk Intent sistem:

```
<intent-filter>
 <action android:name="android.intent.action.ACTION_POWER_CONNECTED"/>
 <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED"/>
</intent-filter>
```

## 1.3 Mengimplementasikan onReceive() dalam BroadcastReceiver

Saat BroadcastReceiver menghalangi siaran yang didaftarkan, Intent dikirimkan ke metode `onReceive()` pengguna, bersama dengan konteks tempat penerima berjalan.

1. Buka file CustomReceiver, dan hapus implementasi default dalam metode `onReceive()`.
2. Dapatkan tindakan dari intent dan simpan dalam variabel `String` bernama `intentAction`:

```
@Override
public void onReceive(Context context, Intent intent) {
 String intentAction = intent.getAction();
}
```

3. Buat pernyataan `switch` dengan string `intentAction`, agar aplikasi dapat menampilkan pesan toast yang berbeda untuk setiap tindakan spesifik yang didaftarkan kepada penerima:

```
switch (intentAction){
 case Intent.ACTION_POWER_CONNECTED:
 break;
 case Intent.ACTION_POWER_DISCONNECTED:
 break;
}
```

4. Inisialisasi variabel `String` bernama `toastMessage` sebelum pernyataan `switch`, dan jadikan nilainya `null` agar bisa disetel tergantung pada tindakan siaran yang Anda terima.
5. Tetapkan `toastMessage` ke "Power connected!" jika tindakannya `ACTION_POWER_CONNECTED`, dan "Power disconnected!"

- jika `ACTION_POWER_DISCONNECTED`. Ekstrak sumber daya string.
- Tampilkan pesan toast untuk durasi pendek setelah pernyataan switch:

```
Toast.makeText(context, toastMessage, Toast.LENGTH_SHORT).show();
```

- Jalankan aplikasi Anda. Setelah dipasang, cabut perangkat. Proses ini mungkin memakan beberapa waktu saat pertama kali dilakukan, tetapi toast pasti ditampilkan setiap kali Anda mencolokkan atau mencabut perangkat.

**Catatan:** Jika menggunakan emulator, Anda dapat mengubah status sambungan daya dengan memilih ikon elips untuk menu tersebut, pilih **Battery** pada bilah kiri, dan ubah menggunakan setelan **Sambungan pengisi daya**.

## 1.4 Membatasi Penerima Siaran

Penerima Siaran selalu aktif, sehingga aplikasi tidak perlu berjalan agar metode `onReceive()` dipanggil.

- Silakan coba: tutup aplikasi, colok dan cabut perangkat.

Pesan toast tetap ditampilkan!

Anda memiliki banyak tanggung jawab, sebagai developer, untuk tidak membuat bingung pengguna dengan notifikasi atau fungsionalitas yang tidak diinginkan setiap kali siaran terjadi. Pada contoh ini, pesan Toast yang muncul setiap kali status daya berubah dapat mengganggu pengguna. Untuk membatasinya, Anda akan menambahkan beberapa kode untuk memastikan penerima siaran hanya aktif saat aplikasi ditampilkan.

Kelas `PackageManager` bertanggung jawab untuk mengaktifkan dan menonaktifkan komponen Android tertentu (seperti layanan, aktivitas, atau penerima siaran). Ini bisa dilakukan menggunakan metode

```
setComponentEnabledSetting() yang memerlukan tiga argumen:
```

- `ComponentName` (identifier untuk komponen yang ingin diaktifkan atau dinonaktifkan).
  - Salah satu dari konstanta kelas `PackageManager` yang mewakili status aktif suatu komponen. Dalam aplikasi ini kita akan menggunakan `PackageManager.COMPONENT_ENABLED_STATE_ENABLED` dan `PackageManager.COMPONENT_ENABLED_STATE_DISABLED`. Lihat referensi `PackageManager` untuk konstanta lainnya.
  - Konstanta bendera opsional yang memerintahkan sistem untuk tidak memusnahkan aplikasi saat mengubah status komponen: `PackageManager.DONT_KILL_APP`.
- Agar penerima siaran hanya aktif ketika aplikasi ditampilkan, aktifkan penerima dalam `onStart()` dan nonaktifkan dalam `onStop()`.

- Buat dua variabel anggota: `PackageManager` dan `ComponentName`.

- Inisialisasi keduanya di `onCreate()`.

Inisialisasi `PackageManager` dengan `getPackageManager()`. Konstruktor untuk `ComponentName` memerlukan konteks aplikasi dan nama kelas komponen:

```
mReceiverComponentName = new ComponentName(this, CustomReceiver.class);
m PackageManager = getPackageManager();
```

- Ganti `onStart()` dan `onStop()`:

```
@Override
protected void onStart() {
 super.onStart();
}
@Override
protected void onStop() {
 super.onStop();
}
```

- Panggil `setComponentEnabledSetting()` di `PackageManager` dalam `onStart()`. Teruskan nama Komponen, konstanta `PackageManager.COMPONENT_ENABLED_STATE_ENABLED`, dan bendera `DONT_KILL_APP`:

```
mPackageManager.setComponentEnabledSetting
(mReceiverComponentName, PackageManager.COMPONENT_ENABLED_STATE_ENABLED,
PackageManager.DONT_KILL_APP);
```

7. Dalam `onStop()`, gunakan PackageManager untuk menonaktifkan CustomReceiver, menggunakan konstanta `PackageManager.COMPONENT_ENABLED_STATE_DISABLED` :

```
mPackageManager.setComponentEnabledSetting
(mReceiverComponentName, PackageManager.COMPONENT_ENABLED_STATE_DISABLED,
PackageManager.DONT_KILL_APP);
```

## Tugas 2. Mengirimkan dan Menerima Siaran Khusus

Selain merespons siaran sistem, aplikasi juga dapat mengirimkan dan menerima Intent Siaran khusus. Intent siaran khusus sama persis dengan sistem, namun di sini Anda harus mendefinisikan tindakan Intent sendiri untuknya (string unik) dan tindakannya dikirimkan menggunakan metode `sendBroadcast()`. Pada tugas ini, Anda akan menambahkan tombol ke aktivitas yang mengirimkan Intent Siaran khusus, yang akan didaftarkan oleh penerima dan ditampilkan dalam pesan Toast.

### 2.1 Mendefinisikan string Tindakan Siaran khusus

Pengirim dan penerima siaran khusus harus menyetujui string tindakan untuk Intent Siaran. Membuat string tindakan unik dengan menambahkan Nama Tindakan di depannya dengan nama paket adalah praktik yang biasa.

- Buat variabel `String` konstan dalam `MainActivity` dan kelas `CustomReceiver` untuk digunakan sebagai Tindakan Intent Siaran (ini adalah string tindakan khusus):

```
private static final String ACTION_CUSTOM_BROADCAST =
"com.example.android.powerreceiver.ACTION_CUSTOM_BROADCAST";
```

### 2.2 Tambahkan Tombol "Send Custom Broadcast"

- Dalam file `activity_main.xml`, tambahkan tampilan Tombol dengan atribut berikut:

| Atribut                            | Nilai                     |
|------------------------------------|---------------------------|
| <code>android:id</code>            | "@+id/sendBroadcast"      |
| <code>android:layout_width</code>  | <code>wrap_content</code> |
| <code>android:layout_height</code> | <code>wrap_content</code> |
| <code>android:text</code>          | "Send Custom Broadcast"   |
| <code>android:layout_margin</code> | "8dp"                     |
| <code>android:onClick</code>       | "sendCustomBroadcast"     |

- Ekstrak sumber daya string.
- Buat stub untuk metode `sendCustomBroadcast()` : Klik pada nama metode `onClick`. **Tekan Alt (Option untuk pengguna Mac) + Enter** dan pilih '`Create 'sendCustomBroadcast(View)`' dalam '`MainActivity`'.

### 2.3 Mengimplementasikan sendCustomBroadcast()

Karena siaran ini ditujukan untuk digunakan hanya oleh aplikasi Anda, gunakan LocalBroadcastManager untuk mengelola siaran dalam aplikasi. LocalBroadcastManager adalah kelas yang mengizinkan Anda untuk mendaftar dan mengirim siaran Intent ke objek lokal dalam aplikasi. Dengan menjaga siaran tetap lokal, data aplikasi tidak akan dibagikan dengan aplikasi Android lainnya, menjadikan informasi lebih aman dan memelihara efisiensi sistem.

1. Dalam metode `sendCustomBroadcast()` di `MainActivity`, buat Intent baru, dengan string tindakan khusus sebagai argumen.

```
Intent customBroadcastIntent = new Intent(ACTION_CUSTOM_BROADCAST);
```

2. Kirimkan siaran menggunakan kelas `LocalBroadcastManager`:

```
LocalBroadcastManager.getInstance(this).sendBroadcast(customBroadcastIntent);
```

## 2.4 Mendaftarkan Siaran Khusus

Untuk siaran sistem, Anda mendaftarkan penerima dalam file `AndroidManifest.xml`. Anda juga bisa mendaftarkan penerima untuk tindakan spesifik secara terprogram. Untuk siaran yang dikirim menggunakan `LocalBroadcastManager`, pendaftaran statis dalam manifes tidak diizinkan.

Jika Anda mendaftarkan penerima siaran secara terprogram, Anda harus berhenti mendaftarkan penerima tersebut ketika tidak diperlukan lagi. Dalam aplikasi, penerima hanya perlu merespons ke siaran khusus saat aplikasi dijalankan agar kita bisa mendaftarkan tindakan tersebut dalam `onCreate()` dan berhenti mendaftarkannya di `onDestroy()`.

1. Buat variabel anggota di `MainActivity` untuk Penerima dan inisialisasi member tersebut:

```
private CustomReceiver mReceiver = new CustomReceiver();
```

2. Dalam `onCreate()`, dapatkan instance `LocalBroadcastManager` dan daftarkan penerima dengan tindakan intent khusus:

```
LocalBroadcastManager.getInstance(this)
 .registerReceiver(mReceiver, new IntentFilter(ACTION_CUSTOM_BROADCAST));
```

3. Ganti metode `onDestroy()` dan berhenti mendaftarkan penerima dari `LocalBroadcastManager`:

```
@Override
protected void onDestroy() {
 LocalBroadcastManager.getInstance(this).unregisterReceiver(mReceiver);
 super.onDestroy();
}
```

## 2.5 Merespons Siaran Khusus

1. Dalam `onReceive()` kelas `CustomReceiver`, tambahkan pernyataan case untuk Tindakan Intent khusus.
2. Modifikasi pesan toast ke "Custom Broadcast Received", ekstrak ke dalam `strings.xml` dan panggil `custom_broadcast_toast` (tekan **Alt + Enter** atau **Option + Enter** di Mac dan pilih **extract string resource**):

```
case ACTION_CUSTOM_BROADCAST:
 toastMessage = context.getString(R.string.custom_broadcast_toast);
 break;
```

**Catatan:** Penerima Siaran yang didaftarkan secara terprogram tidak dipengaruhi oleh proses aktivasi dan penonaktifan yang dilakukan oleh kelas `PackageManager`, yang ditujukan untuk komponen yang terdaftar dalam file Manifes Android. Mengaktifkan atau menonaktifkan penerima seperti ini dilakukan dengan masing-masing mendaftarkan atau berhenti mendaftarkannya. Dalam hal ini, menonaktifkan toggle "Receiver Enabled" akan menghentikan daya yang terhubung atau pesan toast yang terputus, tetapi tidak pesan Toast Intent Siaran Khususnya.

Itu saja! Aplikasi Anda sekarang memberikan intent Siaran khusus dan dapat menerima sistem dan Siaran khusus.

## Kode solusi

Proyek Android Studio: [PowerReceiver](#)

## Tantangan penyusunan kode

**Catatan:** Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

**Tantangan:** Pola umum untuk penerima siaran adalah memulai pembaruan atau tindakan setelah perangkat selesai booting. Implementasikan Penerima Siaran yang akan menampilkan pesan toast setengah jam setelah perangkat selesai booting.

## Rangkuman

- Penerima Siaran adalah salah satu dari komponen dasar aplikasi Android.
- Penerima Siaran dapat menerima Intent yang disiarkan oleh sistem dan aplikasi.
- Mekanisme siaran Intent sepenuhnya terpisah dari Intent yang digunakan untuk memulai Aktivitas.
- Anda perlu membuat subkelas pada kelas BroadcastReceiver dan mengimplementasikan onReceive() untuk memroses Intent yang datang dan terkait dengan siaran.
- Penerima siaran dapat didaftarkan dalam file manifes Android atau secara terprogram.
- Gunakan LocalBroadcastManager untuk mendaftarkan dan mengirimkan Siaran yang bersifat pribadi untuk aplikasi Anda.
- LocalBroadcastManager lebih efisien dan aman dari siaran sistem.
- Untuk siaran yang dikirim menggunakan LocalBroadcastManager, Anda hanya bisa mendaftarkan minat untuk tindakan spesifik secara terprogram.
- Praktik umum untuk membuat nama tindakan Intent yang unik untuk siaran bertujuan untuk menambahkan Nama Tindakan di depan dengan nama paket.

## Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Penerima Siaran](#)

## Ketahui selengkapnya

Dokumentasi Developer Android

### Panduan

- [Intent dan Filter Intent](#)
- [Memanipulasi Penerima Siaran Sesuai Kebutuhan](#)

### Referensi

- [BroadcastReceiver](#)

## 8.1: Notifikasi

### Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Membuat notifikasi dasar](#)
- [Tugas 2. Memperbarui dan membatalkan notifikasi](#)
- [Tugas 3. Menambahkan tindakan notifikasi](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Hingga saat ini, aplikasi yang telah Anda buat menggunakan elemen UI yang hanya terlihat saat aplikasi berjalan. Satu-satunya pengecualian adalah BroadcastReceiver yang Anda implementasikan, yang menunjukkan pesan Toast saat perangkat dihubungkan dan diputuskan dari aliran listrik. Sering kali Anda ingin menunjukkan informasi pengguna bahkan saat aplikasi sedang tidak berjalan. Misalnya, Anda ingin memberi tahu pengguna bahwa ada konten baru atau menginformasikan skor tim favorit mereka. Framework Android menyediakan mekanisme bagi aplikasi untuk memberi tahu pengguna bahkan saat aplikasi tidak ada di latar depan: framework [Notifikasi](#).

Notifikasi adalah pesan yang bisa Anda tampilkan kepada pengguna di luar UI normal aplikasi. Saat Android mengeluarkan notifikasi, notifikasi akan muncul sebagai ikon dalam **area notifikasi perangkat**. Untuk melihat detail spesifik notifikasi, pengguna membuka **panel samping notifikasi**. Baik area notifikasi maupun panel samping notifikasi adalah area-area yang dikontrol sistem yang bisa dilihat pengguna kapan saja.



Dalam praktik ini Anda akan membuat aplikasi yang memicu notifikasi saat tombol ditekan dan memberikan kemampuan untuk memperbarui notifikasi atau membatalkannya.

## Yang harus sudah Anda KETAHUI

Untuk praktik ini Anda harus bisa:

- Mengimplementasikan metode `onclick()` untuk tombol.
- Membuat Intent Implisit.
- Mengirim Intent Siaran Khusus.
- Menggunakan Penerima Siaran.

## Yang akan Anda PELAJARI

Dalam praktik ini Anda akan belajar:

- Membuat Notifikasi menggunakan Builder Notifikasi.
- Menggunakan Intent yang Tertunda untuk merespons tindakan Notifikasi.
- Memperbarui atau membatalkan Notifikasi yang ada.

## Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Mengirimkan notifikasi saat tombol ditekan.
- Memperbarui notifikasi dari tombol dan tindakan yang ada di dalam notifikasi.
- Meluncurkan intent implisit ke laman web dari notifikasi.

## Ringkasan aplikasi

Notify Me! adalah aplikasi yang dapat memicu, memperbarui, dan membatalkan notifikasi. Aplikasi ini juga bereksperimen dengan gaya notifikasi, tindakan, dan prioritas.





# Tugas 1: Membuat notifikasi dasar

## 1.1 Membuat Proyek

- Buat proyek baru bernama "Notify Me!", terima opsi default dan gunakan template kosong.
- Dalam file activity\_main.xml file, ubah elemen rootview ke LinearLayout vertikal dan atribut gravity disetel ke "center".
- Tambahkan tombol dengan atribut berikut ini untuk menggantikan TextView default-nya:

| Atribut               | Nilai          |
|-----------------------|----------------|
| android:id            | "@+id/notify"  |
| android:layout_width  | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:text          | "Notify Me!"   |
| android:layout_margin | "4dp"          |

- Buat stub metode untuk metode `sendNotification()`. Metode seharusnya tidak mengambil argumen dan mengembalikan void:

```
public void sendNotification() {}
```

- Buat variabel anggota untuk Tombol Notifikasi.
- Inisialisasi tombol dalam `onCreate()` dan buat onClickListener untuk tombol tersebut:

```
mNotifyButton = (Button) findViewById(R.id.notify);
mNotifyButton.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 }
});
```

- Panggil `sendNotification()` dari metode onClick.

## 1.2 Membuat notifikasi pertama

Notifikasi dibuat menggunakan kelas [NotificationCompat.Builder](#), yang mengizinkan Anda untuk menyetel konten dan perilaku Notifikasi. Notifikasi harus berisi elemen berikut ini:

- Judul, disetel oleh `setContentTitle()`.
- Teks detail, disetel oleh `setContentText()`.
- Ikon, disetel oleh ikon `setSmallIcon()`.

Notifikasi Android diterapkan oleh [NotificationManager](#). Jika Anda perlu memperbarui atau membatalkan notifikasi di masa yang akan datang, Anda perlu mengaitkan ID notifikasi dengan Notifikasi.

### Membuat Ikon Notifikasi

- Buka **File > New > Image Asset**.
- Dari menu tarik turun **Icon Type**, pilih **Notification Icons**.
- Klik pada ikon di sebelah item **Clip Art** untuk memilih ikon materi yang akan Anda gunakan sebagai ikon untuk notifikasi. Pada contoh ini, Anda bisa menggunakan ikon Android.
- Ganti nama `ic_android` sumber daya dan klik **Next and Finish**. Ini akan membuat beberapa file drawable dengan

resolusi berbeda untuk level API yang berbeda.

- Buat variabel anggota di MainActivity untuk menyimpan Pengelola Notifikasi:

```
private NotificationManager mNotifyManager;
```

- Buat variabel konstanta untuk ID Notifikasi. Karena hanya akan ada satu notifikasi aktif pada satu waktu, kita bisa menggunakan ID yang sama untuk semua notifikasi:

```
private static final int NOTIFICATION_ID = 0;
```

- Buat instance Pengelola Notifikasi dalam onCreate menggunakan `getSystemService()` :

```
mNotifyManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
```

- Bangun dan buat instance Builder Notifikasi dalam metode `sendNotification()` :

```
NotificationCompat.Builder notifyBuilder = new NotificationCompat.Builder(this)
```

**Catatan:** Pastikan kelas NotificationCompat diimpor dari pustaka dukungan v4.

- Setel Judul Notifikasi ke "You've been notified!".
- Setel Teks Notifikasi ke "This is your notification text."
- Setel ikon Notifikasi ke ikon Android yang Anda tambahkan.

```
NotificationCompat.Builder notifyBuilder = new NotificationCompat.Builder(this)
 .setContentTitle("You've been notified!")
 .setContentText("This is your notification text.")
 .setSmallIcon(R.drawable.ic_android);
```

- Panggil `notify()` pada Pengelola Notifikasi di bagian akhir metode `sendNotification()`, meneruskan ID notifikasi dan notifikasi tersebut:

```
Notification myNotification = notifyBuilder.build();
mNotifyManager.notify(NOTIFICATION_ID, myNotification);
```

- Jalankan aplikasi Anda. Tombol "Notify Me!" sekarang mengeluarkan notifikasi (cari ikon pada bilah status), namun kekurangan beberapa fitur penting: tidak ada suara atau getar notifikasi, mengeklik notifikasi tidak melakukan apa pun. Mari tambahkan beberapa fungsionalitas tambahan ke notifikasi.

### 1.3 Menambahkan intent konten

Untuk menyempurnakan notifikasi, Anda perlu menambahkan beberapa fitur yang tersedia melalui kelas `NotificationCompat.Builder`:

- Intent konten, yang diluncurkan saat notifikasi diketuk dan disetel oleh `setContentIntent()`.
- Prioritas, yang menentukan bagaimana sistem menampilkan notifikasi dengan mempertimbangkan notifikasi lain, dan disetel oleh `setPriority()`.
- Opsi default, seperti pesan, getar dan lampu LED (jika tersedia), dan disetel oleh `setDefaults()`.

Mengetuk notifikasi meluncurkan Intent. Intent konten untuk notifikasi sangat serupa dengan Intent yang telah Anda gunakan selama kursus ini. Intent dapat berupa eksplisit intent untuk meluncurkan sebuah aktivitas, intent implisit untuk melakukan tindakan, atau intent siaran untuk memberi tahu sistem tentang sistem atau peristiwa khusus. Perbedaan utama dengan intent dalam notifikasi adalah intent ini harus dibungkus dalam `PendingIntent`, yang mengizinkan notifikasi untuk melakukan tindakan meskipun aplikasi tidak berjalan. `PendingIntent` diberikan ke komponen eksternal (misalnya Pengelola Notifikasi) yang mengizinkan aplikasi eksternal menggunakan izin aplikasi untuk mengeksekusi sepotong kode yang telah didefinisikan sebelumnya. Akibatnya, intent mengotorisasi notifikasi untuk mengirimkan intent atas nama aplikasi.

Untuk contoh ini, intent konten notifikasi (intent yang diluncurkan saat notifikasi ditekan) akan meluncurkan MainActivity aplikasi (jika Anda sudah dalam aplikasi ini tidak akan berpengaruh).

- Buat intent eksplisit dalam metode `sendNotification()` untuk meluncurkan kelas MainActivity:

```
Intent notificationIntent = new Intent(this, MainActivity.class);
```

- Dapatkan PendingIntent menggunakan `getActivity()`, yang meneruskan konstanta ID notifikasi untuk requestCode dan menggunakan bendera FLAG\_UPDATE\_CURRENT:

```
PendingIntent notificationPendingIntent = PendingIntent.getActivity(this,
 NOTIFICATION_ID, notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT);
```

- Tambahkan PendingIntent ke Notifikasi menggunakan `setContentIntent()` dalam NotificationCompat.Builder:

```
.setContentIntent(notificationPendingIntent)
```

- Jalankan aplikasi. Klik tombol Notify Me untuk mengirimkan notifikasi. Hentikan aplikasi. Sekarang lihat notifikasi dan klik. Perhatikan bahwa aplikasi akan terbuka kembali di MainActivity.

## 1.4 Tambahkan prioritas dan default ke notifikasi Anda

Saat pengguna mengeklik tombol "Notify Me!", notifikasi dikeluarkan namun satu-satunya visual yang dilihat pengguna adalah ikon pada bilah notifikasi. Untuk menarik perhatian pengguna, default notifikasi dan prioritas akan disetel dengan benar.

Prioritas adalah nilai integer dari PRIORITY\_MIN (-2) ke PRIORITY\_MAX (2) yang mewakili pentingnya notifikasi kepada pengguna. Notifikasi dengan prioritas yang lebih tinggi akan diurutkan di atas prioritas yang lebih rendah dalam panel samping notifikasi. Notifikasi prioritas HIGH atau MAX akan disampaikan sebagai Notifikasi "Heads - Up", yang ditarik ke bawah di atas layar aktif pengguna.

- Tambahkan baris berikut ke Builder Notifikasi untuk menyetel prioritas notifikasi ke HIGH:

```
.setPriority(NotificationCompat.PRIORITY_HIGH)
```

- Opsi default dalam Builder digunakan untuk menyetel suara, getar, pola warna LED untuk notifikasi Anda (jika perangkat pengguna memiliki indikator LED). Dalam contoh ini, Anda akan menggunakan opsi default dengan menambahkan baris berikut ke Builder Anda:

```
.setDefaults(NotificationCompat.DEFAULT_ALL)
```

- Anda perlu menghentikan aplikasi dan memulainya lagi untuk melihat perubahannya.

**Catatan:** Notifikasi prioritas tinggi tidak akan turun ke bawah di depan layar aktif kecuali prioritas dan default-nya disetel. Prioritas saja tidak bisa.

## Tugas 2. Memperbarui dan membatalkan notifikasi

Setelah mengeluarkan notifikasi, sebaiknya perbarui atau batalkan notifikasi jika informasinya berubah atau menjadi tidak relevan.

Dalam tugas ini, Anda akan mempelajari cara memperbarui dan membatalkan notifikasi.

### 2.1 Menambahkan tombol update dan cancel

- Dalam file layout, buat dua salinan tombol "Notify Me!".
- Ubah atribut teks dalam salinan ke "Update Me!" dan "Cancel Me!".
- Ubah id ke "update" dan "cancel", berturut-turut.
- Tambahkan variabel anggota untuk setiap tombol baru dan inisialisasikan di `onCreate()`.
- Buat dia metode dalam MainActivity yang tidak memerlukan parameter dan mengembalikan kosong:

```
public void updateNotification() {}
public void cancelNotification() {}
```

6. Buat Listener onClick untuk tombol dan panggilan baru `updateNotification()` dalam metode onClick tombol "update" dan `cancelNotification()` dalam metode onClick tombol "cancel".

## 2.2 Mengimplementasikan metode notifikasi pembatalan dan pembaruan

### Batalkan Notifikasi

Membatalkan notifikasi cukup mudah: panggil `cancel()` pada Pengelola Notifikasi, meneruskan ID notifikasi:

```
mNotifyManager.cancel(NOTIFICATION_ID);
```

### Perbarui Notifikasi

Memperbarui notifikasi lebih sulit. Notifikasi Android hadir dengan gaya alternatif yang dapat membantu memadatkan informasi atau mewakilinya dengan lebih efisien. Misalnya, aplikasi Gmail menggunakan notifikasi "InboxStyle" jika ada lebih dari satu pesan tidak terbaca, memadatkan informasi ke dalam notifikasi tunggal.

Dalam contoh ini, Anda akan memperbarui notifikasi untuk menggunakan `BigPictureStyle`, yang mengizinkan Anda untuk menyertakan gambar dalam notifikasi.

1. Unduh [gambar ini](#) untuk digunakan di notifikasi Anda.
2. Letakkan di dalam folder res/drawable.
3. Dalam metode `updateNotification()`, konversi drawable ke bitmap:

```
Bitmap androidImage = BitmapFactory
 .decodeResource(getResources(), R.drawable.mascot_1);
```

4. Salin Intent dan PendingIntent yang dibuat dalam `sendNotification()` ke `updateNotification()`, karena Anda akan menggunakan PendingIntent yang sama sebagai Intent Content.
5. Salin kode `NotificationCompat.Builder` dari `sendNotification()` ke `updateNotification()`, untuk memiliki opsi notifikasi dasar yang sama dalam notifikasi terbaru.
6. Ubah gaya notifikasi dalam `NotificationCompat.Builder` yang sama, setel gambar dan "Judul Konten Besar":

```
.setStyle(new NotificationCompat.BigPictureStyle()
 .bigPicture(androidImage)
 .setBigContentTitle("Notification Updated!"));
```

- Catatan:** `BigPictureStyle` adalah subkelas dari `NotificationCompat.Style` yang menyediakan layout alternatif untuk notifikasi. Lihat dokumentasi untuk subkelas yang terdefinisi lainnya.
7. Ubah prioritas Builder ke default agar Anda tidak mendapatkan notifikasi awal saat diperbarui (notifikasi awal hanya dapat ditampilkan dalam gaya default).

```
.setPriority(NotificationCompat.PRIORITY_DEFAULT)
```

8. Panggil `notify()` pada Pengelola Notifikasi, meneruskan ID notifikasi yang sama seperti sebelumnya.

```
mNotifyManager.notify(NOTIFICATION_ID, notifyBuilder.build());
```

9. Jalankan aplikasi Anda. Setelah mengeklik update, periksa notifikasi lagi. Sekarang aplikasi memiliki gambar dan judul terbaru! Anda bisa mengelikannya kembali ke gaya notifikasi reguler dengan mencubit notifikasi yang diperluas.

## 2.3 Mengalihkan keadaan tombol

Dalam aplikasi ini, pengguna mungkin akan bingung karena status notifikasi tidak dilacak di dalam aktivitas. Misalnya, pengguna mungkin mengetuk "Cancel Me!" saat notifikasi tidak tampil. Anda bisa memperbaikinya dengan mengaktifkan dan menonaktifkan berbagai tombol tergantung status notifikasi. Saat aplikasi pertama kali dijalankan, tombol "Notify Me!" harus menjadi satu-satunya yang diaktifkan karena belum ada notifikasi untuk memperbarui atau membatkalkannya. Setelah notifikasi dikirimkan, tombol cancel dan update harus diaktifkan, dan tombol notifikasi harus dinonaktifkan karena notifikasi sudah disampaikan. Setelah notifikasi diperbarui, tombol update dan notify harus dinonaktifkan, sehingga hanya tombol cancel yang diaktifkan. Akhirnya, jika notifikasi dibatalkan, tombolnya harus kembali ke kondisi awal dan tombol notify menjadi satu-satunya tombol yang aktif.

Berikut kode toggle status aktif untuk setiap metode:

#### **onCreate():**

```
mNotifyButton.setEnabled(true);
mUpdateButton.setEnabled(false);
mCancelButton.setEnabled(false);
```

#### **sendNotification():**

```
mNotifyButton.setEnabled(false);
mUpdateButton.setEnabled(true);
mCancelButton.setEnabled(true);
```

#### **updateNotification():**

```
mNotifyButton.setEnabled(false);
mUpdateButton.setEnabled(false);
mCancelButton.setEnabled(true);
```

#### **cancelNotification():**

```
mNotifyButton.setEnabled(true);
mUpdateButton.setEnabled(false);
mCancelButton.setEnabled(false);
```

## **Tugas 3. Menambahkan tindakan notifikasi**

Kadang, notifikasi memerlukan interaksi segera: mematikan alarm sementara, membalas pesan teks, dan sebagainya. Saat jenis notifikasi ini terjadi, pengguna mungkin mengetuk notifikasi untuk merespons ke peristiwa. Android lalu memuat Aktivitas yang tepat dalam aplikasi untuk direspon pengguna. Untuk menghindari membuka aplikasi, framework notifikasi memungkinkan Anda menyematkan tindakan notifikasi langsung di dalam notifikasi itu sendiri. Ini akan mengizinkan pengguna untuk menindak notifikasi tanpa membuka aplikasi.

Komponen yang diperlukan untuk tindakan adalah:

- Ikon, untuk diletakkan dalam notifikasi.
- String label, diletakkan di sebelah ikon.
- PendingIntent, dikirimkan saat tindakan notifikasi diklik.

Dalam contoh berikut, Anda akan menambahkan dua tindakan ke notifikasi. Pertama, Anda akan menambahkan tindakan "Learn More" dengan intent implisit yang meluncurkan laman web, lalu tindakan "Update" dengan intent siaran yang memperbarui notifikasi tanpa membuka aplikasi.

### **3.1 Mengimplementasikan tindakan "Learn More"**

Sebagai contoh pertama tindakan notifikasi, Anda akan mengimplementasikan satu yang meluncurkan intent implisit untuk membuka situs web.

1. Buat variabel String anggota yang berisi URL ke panduan Desain Material untuk notifikasi:  
<https://developer.android.com/design/patterns/notifications.html>.
2. Buat Intent implisit yang membuka URL tersimpan dalam metode `sendNotification()` sebelum membuat notifikasi.
3. Buat PendingIntent dari intent implisit, menggunakan bendera FLAG\_ONE\_SHOT agar PendingIntent tidak bisa digunakan kembali:

```
Intent learnMoreIntent = new Intent(Intent.ACTION_VIEW, Uri
 .parse(NOTIFICATION_GUIDE_URL));
PendingIntent learnMorePendingIntent = PendingIntent.getActivity
 (this, NOTIFICATION_ID, learnMoreIntent, PendingIntent.FLAG_ONE_SHOT);
```



4. Tambahkan ikon ini menggunakan Image Asset Studio, dan sebut ikon `ic_learn_more`:
5. Tambahkan kode baris berikut ke builder dalam `sendNotification()` dan `updateNotification()` untuk menambahkan tindakan ke notifikasi asli dan terbaru:

```
.addAction(R.drawable.ic_learn_more, "Learn More", learnMorePendingIntent);
```

6. Jalankan aplikasi Anda. Notifikasi sekarang akan memiliki ikon yang dapat diklik dan membawa Anda ke web!

## 3.2 Mengimplementasikan tindakan "Update"

Anda sudah melihat tindakan notifikasi menggunakan PendingIntent untuk merespons interaksi pengguna. Pada langkah terakhir, Anda menambahkan tindakan yang menggunakan PendingIntent yang dibuat menggunakan metode `getActivity()`. Anda juga bisa membuat PendingIntent yang memberikan intent siaran dengan memanggil `getBroadcast()` pada kelas PendingIntent. Intent Siaran sangat berguna dalam notifikasi, karena penerima siaran dapat mendaftarkan minatnya dalam intent dan memberikan respons yang sesuai, semuanya tanpa meluncurkan aktivitas yang spesifik.

Sekarang Anda akan mengimplementasikan Penerima Siaran yang akan memanggil metode `updateNotification()` saat tindakan "Update" dalam notifikasi ditekan. Menambahkan fungsionalitas ke notifikasi yang sudah ada dalam aplikasi adalah pola umum, agar pengguna tidak perlu membuka aplikasi apa pun untuk melakukan tindakan tersebut.

1. Buat subkelas BroadcastReceiver sebagai kelas bagian dalam pada MainActivity dan ganti metode `onReceive()`. Jangan lupa untuk menyertakan konstruktur kosong:

```
public class NotificationReceiver extends BroadcastReceiver {

 public NotificationReceiver() {
 }

 @Override
 public void onReceive(Context context, Intent intent) {
 }
}
```

2. Dalam metode `onReceive()`, panggil `updateNotification()`.
3. Buat variabel anggota kontan dalam MainActivity untuk mewakili tindakan pembaruan untuk BroadcastIntent. Pastikan variabel dimulai dengan nama paket untuk memastikan keunikannya:

```
private static final String ACTION_UPDATE_NOTIFICATION =
 "com.example.android.notifyme.ACTION_UPDATE_NOTIFICATION";
```

4. Buat variabel anggota untuk penerima dan inisialisasikan menggunakan konstruktur default.
5. Dalam metode `onCreate()`, daftarkan Penerima Siaran untuk menerima intent ACTION\_UPDATE\_NOTIFICATION:

```
registerReceiver(mReceiver, new IntentFilter(ACTION_UPDATE_NOTIFICATION));
```

6. Ganti metode `onDestroy()` Aktivitas untuk berhenti mendaftarkan penerima:

```
@Override
protected void onDestroy() {
 unregisterReceiver(mReceiver);
 super.onDestroy();
}
```

**Catatan:** Dalam contoh ini, Anda mendaftarkan Penerima Siaran secara terprogram karena penerima didefinisikan sebagai kelas dalam. Saat penerima didefinisikan dengan cara ini, penerima tidak bisa didaftarkan dalam Manifest Android karena bersifat dinamis dan mungkin berubah selama masa hidup aplikasi.

Tampaknya siaran yang dikirimkan oleh notifikasi hanya memperhatikan aplikasi Anda dan harus dikirim dengan LocalBroadcastManager. Tetapi, penggunaan PendingIntent mendeklasikan tanggung jawab pengiriman notifikasi ke Android Framework. Karena waktu proses Android menangani siaran, LocalBroadcastManager tidak dapat digunakan.

#### Buat Tindakan Pembaruan

1. Buat Intent siaran dalam metode `sendNotification()` menggunakan tindakan pembaruan khusus.
2. Dapatkan PendingIntent menggunakan `getBroadcast()` :

```
Intent updateIntent = new Intent(ACTION_UPDATE_NOTIFICATION);
PendingIntent updatePendingIntent = PendingIntent.getBroadcast
 (this, NOTIFICATION_ID, updateIntent, PendingIntent.FLAG_ONE_SHOT);
```



3. Buat ikon ini menggunakan Image Asset Studio, beri nama `ic_update`.
4. Tambahkan tindakan ke builder dalam metode `sendNotification()`, beri judul "Update":

```
.addAction(R.drawable.ic_update, "Update", updatePendingIntent)
```

5. Jalankan aplikasi Anda. Sekarang Anda bisa memperbarui notifikasi tanpa membuka aplikasi!

## Kode solusi

Proyek Android Studio: [NotifyMe](#)

## Tantangan penyusunan kode

**Catatan:** Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

**Tantangan:** Mengaktifkan dan menonaktifkan berbagai tombol adalah cara yang umum untuk memastikan pengguna tidak melakukan tindakan apa pun yang tidak didukung dalam status aplikasi saat ini (misalnya menonaktifkan tombol "Sync" saat tidak ada jaringan). Namun pada aplikasi ini, ada satu kasus di mana status tombol tidak cocok dengan status aplikasi: saat pengguna menghilangkan notifikasi dengan menggeseknya atau menghapus semua panel samping notifikasi. Dalam hal ini, aplikasi tidak bisa mengetahui bahwa notifikasi dihapus dan bahwa status tombol harus diubah.

Buat intent siaran lain yang akan memberi tahu aplikasi bahwa pengguna telah menghilangkan notifikasi dan ubah status tombol.

Petunjuk: Periksa kelas `NotificationCompat.Builder` untuk metode yang mengirimkan Intent saat notifikasi telah dihilangkan oleh pengguna.

## Rangkuman

- Notifikasi adalah pesan yang bisa Anda tampilkan kepada pengguna di luar UI normal aplikasi.
- Notifikasi menyediakan cara bagi aplikasi untuk berinteraksi dengan pengguna bahkan saat aplikasi sedang tidak berjalan.
- Saat Android mengeluarkan notifikasi, notifikasi akan muncul sebagai ikon dalam area notifikasi perangkat.
- UI dan tindakan untuk notifikasi ditentukan menggunakan `NotificationCompat.Builder`.
- Untuk membuat notifikasi gunakan `NotificationCompat.Builder.build()`.
- Untuk mengeluarkan notifikasi, teruskan objek Notifikasi ke sistem waktu proses Android dengan `NotificationManager.notify()`.
- Untuk memperbarui atau membatalkan notifikasi, Anda perlu mengaitkan ID notifikasi dengan Notifikasi.
- Intent dapat berupa bagian dari notifikasi (Eksplisit, Implisit, atau Siaran).
- Intent dalam notifikasi harus "dibungkus" dalam `PendingIntent`, yang sebenarnya bukanlah Intent. `PendingIntent` adalah implementasi pola dekorator.
- Komponen notifikasi yang diperlukan adalah: ikon kecil (`setSmallIcon()`), judul (`setContentTitle()`) dan beberapa teks detail (`setContentText()`).
- Beberapa komponen opsional notifikasi adalah: intent, gaya diperluas, prioritas, dsb. Lihat `NotificationCompat.Builder` untuk informasi selengkapnya.

## Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Notifikasi](#)

## Ketahui selengkapnya

### Panduan

- [Notifikasi](#)
- [Panduan Desain Notifikasi](#)

### Referensi

- [NotificationCompat.Builder](#)
- [NotificationCompat.Style](#)

## 8.2: Alarm Manager

### Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Tugas 1. Menyiapkan Proyek dan Tampilan Parking Alarm
- Tugas 2. Menyiapkan Notifikasi
- Tugas 3. Membuat Alarm Berulang
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Dalam praktik sebelumnya, Anda telah mempelajari cara membuat aplikasi merespons interaksi pengguna dengan menekan tombol atau mengetuk notifikasi. Anda juga belajar cara membuat aplikasi merespons peristiwa sistem menggunakan BroadcastReceivers. Namun bagaimana jika aplikasi perlu mengambil tindakan pada waktu yang spesifik, seperti pada kasus notifikasi kalender? Dalam kasus ini, Anda perlu menggunakan [AlarmManager](#), kelas yang mengizinkan Anda meluncurkan dan mengulangi [PendingIntent](#) pada waktu dan interval yang spesifik.

Dalam praktik ini Anda akan membuat timer yang akan mengingatkan Anda untuk berdiri jika sudah duduk terlalu lama.

### Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Mengimplementasikan listener onCheckChanged untuk tombol toggle.
- Menyiapkan intent siaran khusus.
- Menggunakan penerima siaran.
- Mengirimkan notifikasi.

### Yang akan Anda PELAJARI

Anda akan belajar:

- Menjadwalkan alarm berulang dengan AlarmManager.
- Memeriksa apakah Alarm sudah disiapkan.
- Membatalkan alarm berulang.

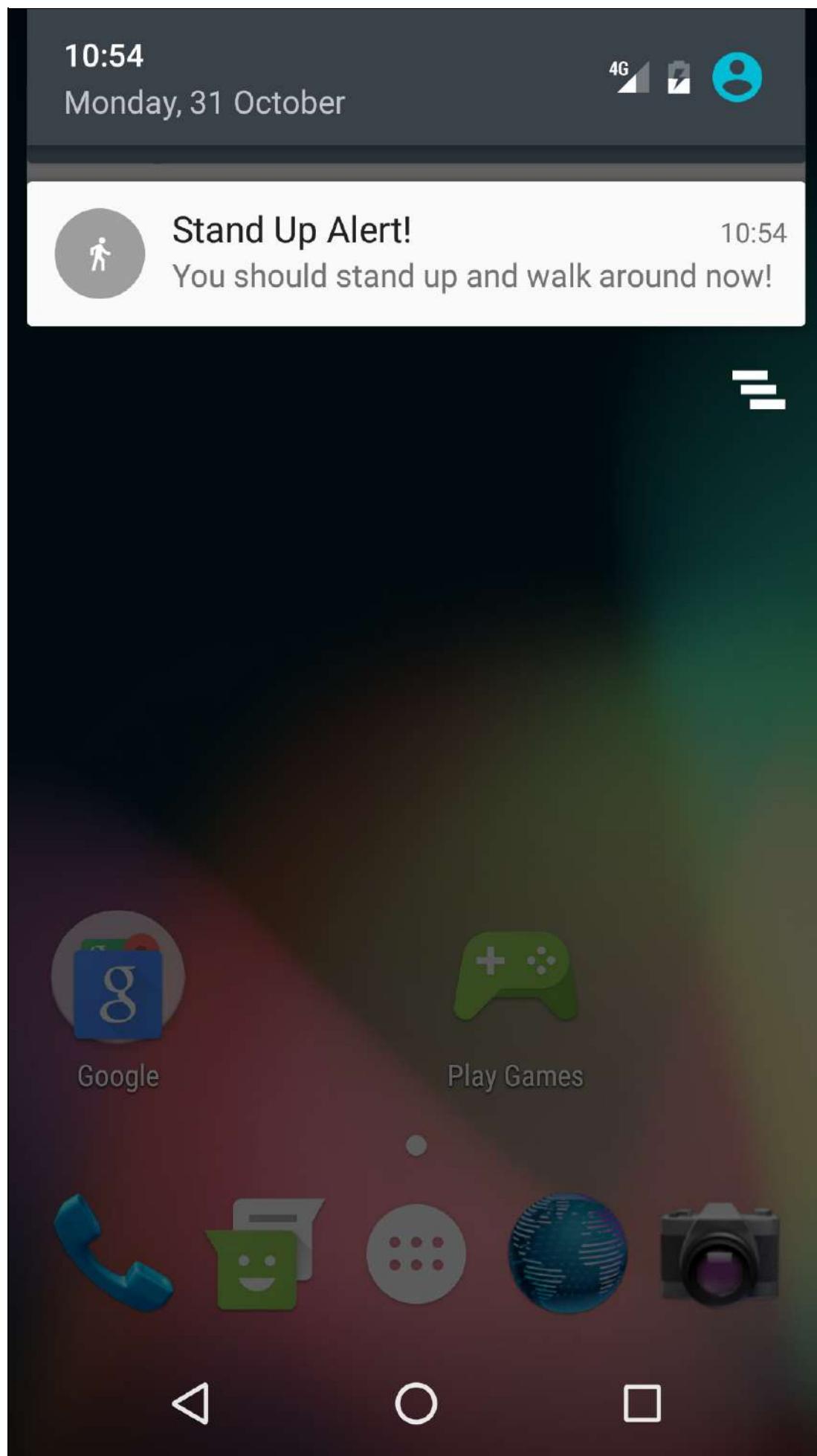
### Yang akan Anda LAKUKAN

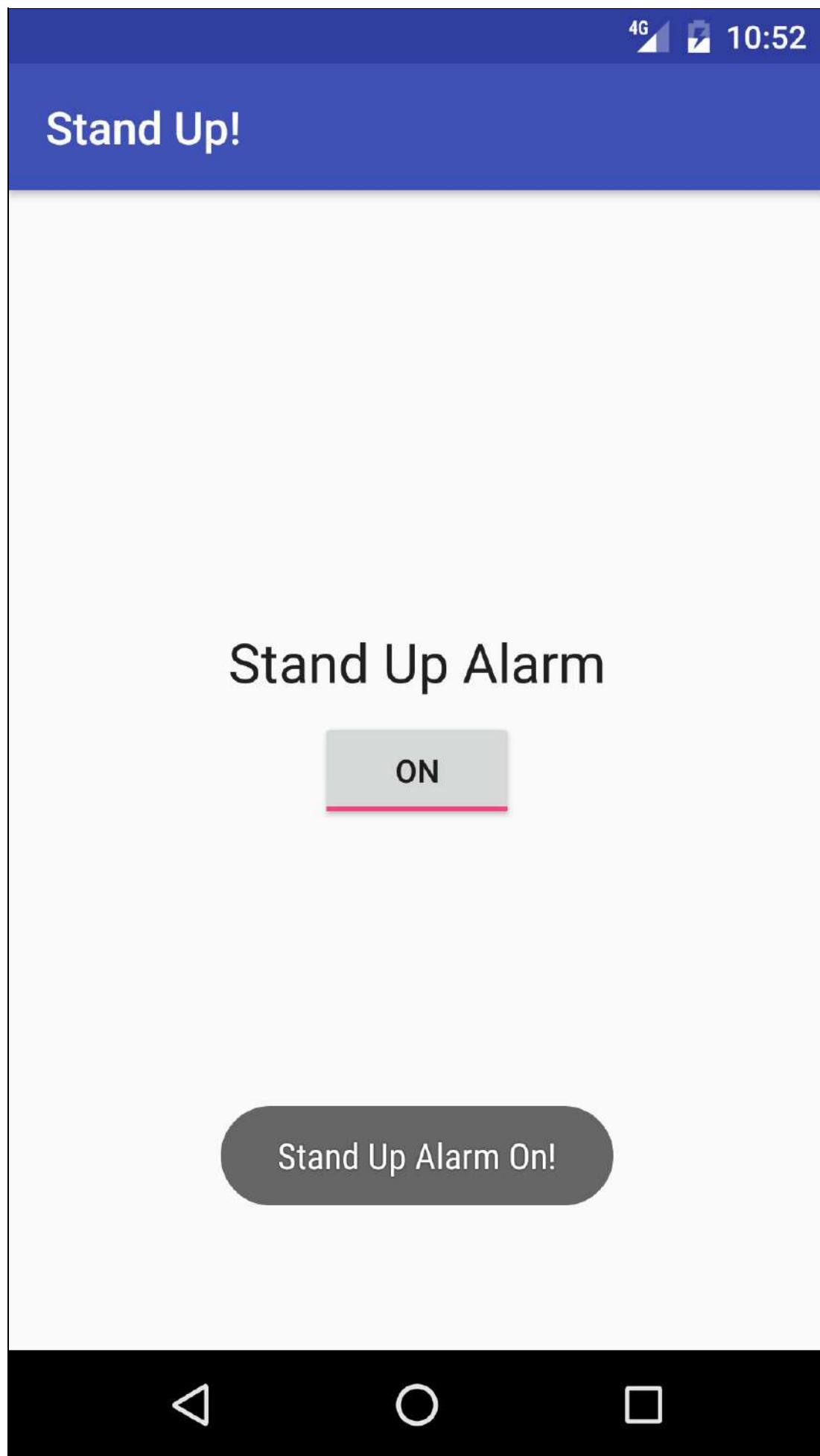
Dalam praktik ini Anda akan:

- Menyetel alarm berulang untuk memberi tahu Anda setiap lima belas menit.
- Menggunakan ToggleButton untuk menyetel dan melacak alarm.
- Menggunakan pesan Toast untuk memberitahukan pengguna saat Alarm diaktifkan atau dinonaktifkan.

### Ringkasan Aplikasi

Stand Up! adalah aplikasi yang membantu Anda tetap sehat dengan mengingatkan Anda untuk berdiri dan berjalan-jalan setiap sekitar lima belas menit. Aplikasi ini menggunakan notifikasi untuk memberi tahu Anda ketika lima belas menit sudah berlalu. Aplikasi menyertakan tombol alih yang dapat mengaktifkan dan menonaktifkan alarm.





# Tugas 1. Menyiapkan Stand Up! Proyek dan Tampilan

## 1.1 Membuat Stand Up! Layout proyek

1. Membuat proyek baru bernama "Stand Up!", menerima opsi default dan menggunakan template aktivitas kosong.
2. Hapus TextView default dan tambahkan elemen berikut:

| TextView     | Atribut                         | Nilai                                    |
|--------------|---------------------------------|------------------------------------------|
|              | android:layout_width            | "wrap_content"                           |
|              | android:layout_height           | "wrap_content"                           |
|              | android:layout_above            | "@+id/alarmToggle"                       |
|              | android:layout_centerHorizontal | "true"                                   |
|              | android:layout_margin           | "8dp"                                    |
|              | android:text                    | "Stand Up Alarm"                         |
|              | android:textAppearance          | @style/TextAppearance.AppCompat.Headline |
| ToggleButton | Atribut                         | Nilai                                    |
|              | android:id                      | "@+id/alarmToggle"                       |
|              | android:layout_width            | "wrap_content"                           |
|              | android:layout_height           | "wrap_content"                           |
|              | android:layout_centerHorizontal | "true"                                   |
|              | android:layout_centerVertical   | "true"                                   |

## 1.2 Menyiapkan metode setOnCheckedChangeListener()

Aplikasi Stand Up! memiliki tombol alih yang digunakan untuk menyetel dan membatalkan alarm, sekaligus tampak mewakili status alarm saat ini. Untuk menyetel alarm saat tombol alih diaktifkan, Anda perlu menggunakan metode `onCheckedChangeListener()` :

1. Dalam metode `MainActivity onCreate()`, temukan Alarm Toggle menurut id.
2. Panggil `setOnCheckedChangeListener()` pada instance tombol alih, dan mulai mengetik "`new OnCheckedChangeListener`". Android Studio akan menyelesaikan metode untuk Anda secara otomatis, termasuk metode penggantian `onCheckedChanged()` yang diperlukan. Metode ini memiliki dua parameter: CompoundButton yang diklik (dalam kasus ini tombol Toggle Alarm), dan boolean yang mewakili status Tombol Toggle saat ini (yakni, apakah tombol alih saat ini diaktifkan atau dinonaktifkan).

```
alarmToggle.setOnCheckedChangeListener(
 new CompoundButton.OnCheckedChangeListener() {

 @Override
 public void onCheckedChanged(CompoundButton compoundButton,
 boolean isChecked) {
 }
 });
}
```

3. Akan sangat berguna jika pengguna menerima masukan selain tombol alih yang diaktifkan dan dinonaktifkan untuk menunjukkan bahwa alarm memang disetel (Anda belum mengimplementasikan alarm, ini akan dilakukan pada bagian berikutnya). Setel blok if/else menggunakan parameter boolean dalam metode `onCheckedChanged()` yang mengirimkan pesan toast untuk memberi tahu pengguna apakah alarm diaktifkan atau dinonaktifkan. Jangan lupa untuk mengekstrak sumber daya string.

```
String toastMessage;
if(isChecked){
 //Set the toast message for the "on" case
 toastMessage = getString(R.string.alarm_on_toast);
} else {
 //Set the toast message for the "off" case
 toastMessage = getString(R.string.alarm_off_toast);
}

//Show a toast to say the alarm is turned on or off
Toast.makeText(MainActivity.this, toastMessage, Toast.LENGTH_SHORT)
 .show();
```

## Tugas 2. Menyiapkan Notifikasi

Langkah berikutnya adalah membuat notifikasi yang akan mengingatkan pengguna untuk berdiri setiap lima belas menit. Untuk sekarang, notifikasi akan dikirimkan segera setelah tombol alih disetel.

### 2.1 Membuat notifikasi

Pada langkah ini Anda akan membuat metode `deliverNotification()` yang akan mengeposkan pengingat untuk berdiri dan berjalan-jalan.

1. Buat variabel anggota dalam MainActivity bernama `mNotificationManager` dari tipe `NotificationManager`.
2. Inisialisasikan dalam `onCreate()` dengan memanggil `getSystemService()` :

```
mNotificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
```

3. Buat metode MainActivity bernama `deliverNotification()` yang mengambil Context sebagai argumen dan tidak mengembalikan apa pun.

```
private void deliverNotification(Context context) {}
```

4. Buat konstanta anggota dalam MainActivity bernama `NOTIFICATION_ID` dan setel ke 0. Aplikasi hanya akan mendapatkan satu notifikasi pada satu waktu, jadi Anda akan menggunakan id notifikasi yang sama untuk semua notifikasi.

**Catatan:** ID Notifikasi digunakan untuk membedakan notifikasi dalam aplikasi. Pengelola Notifikasi hanya bisa membatalkan notifikasi yang dikirimkan dari aplikasi agar Anda bisa menggunakan ID yang sama dalam aplikasi yang berbeda.

#### Intent Konten Notifikasi

1. Buat Intent dalam `onCreate()` yang akan Anda gunakan untuk Intent Konten notifikasi:

```
Intent contentIntent = new Intent(context, MainActivity.class);
```

2. Buat PendingIntent dari Intent konten tepat di bawah definisi contentIntent menggunakan metode `getActivity()`, meneruskan ID notifikasi dan menggunakan `FLAG_UPDATE_CURRENT` flag:

```
PendingIntent contentPendingIntent = PendingIntent.getActivity
 (context, NOTIFICATION_ID, contentIntent, PendingIntent.FLAG_UPDATE_CURRENT);
```

**Catatan: bendera PendingIntent** memberi tahu sistem cara menangani situasi saat banyak instance PendingIntent yang sama dibuat (maksudnya berisi intent yang sama). Bendera `FLAG_UPDATE_CURRENT` memerintahkan sistem untuk menggunakan Intent lama tetapi mengganti data ekstra. Karena Anda tidak memiliki ekstra dalam intent ini, Anda akan menggunakan kembali PendingIntent yang sama berulang-ulang.

### Judul dan Teks Notifikasi

1. Buat sumber daya string dalam file strings.xml bernama notification\_title. Setel agar setara dengan "Stand Up Alert".
2. Buat sumber daya string dalam file strings.xml bernama notification\_text. Setel agar setara dengan "You should stand up and walk around now!".

### Ikon Notifikasi

1. Tambahkan asset gambar untuk digunakan sebagai ikon notifikasi (gunakan [Image Asset Studio](#)). Pilih ikon apa pun



yang Anda rasa cocok untuk alarm ini:

### Buat notifikasi

1. Gunakan NotificationCompat.Builder untuk membuat notifikasi dalam metode `deliverNotification()` menggunakan judul, teks, ikon, dan intent konten notifikasi di atas.

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(context)
 .setSmallIcon(R.drawable.ic_stand_up)
 .setContentTitle(context.getString(R.string.notification_title))
 .setContentText(context.getString(R.string.notification_text))
 .setContentIntent(contentPendingIntent)
```

2. Setel prioritas Notifikasi ke `PRIORITY_HIGH`:

```
.setPriority(NotificationCompat.PRIORITY_HIGH)
```

3. Tambahkan opsi ke builder untuk menyetel AutoCancel ke true, dan opsi lainnya untuk menggunakan lampu, suara dan pola getar default:

```
.setAutoCancel(true)
.setDefaults(NotificationCompat.DEFAULT_ALL);
```

### Kirim notifikasi

1. Gunakan NotificationManager untuk mengirimkan notifikasi:

```
mNotificationManager.notify(NOTIFICATION_ID, builder.build());
```

2. Panggil `deliverNotification()` saat tombol alih alarm diaktifkan, meneruskan konteks aktivitas:

3. Panggil `cancelAll()` pada NotificationManager jika tombol alih dinonaktifkan untuk menghapus notifikasi.

```

if(isChecked){
 deliverNotification(MainActivity.this);
 //Set the toast message for the "on" case
 toastMessage = getString(R.string.alarm_on_toast);
} else {
 //Cancel notification if the alarm is turned off
 mNotificationManager.cancelAll();

 //Set the toast message for the "off" case
 toastMessage = getString(R.string.alarm_off_toast);
}

```

- Jalankan aplikasi dan periksa apakah notifikasi dikirimkan dengan semua opsi yang diinginkan.

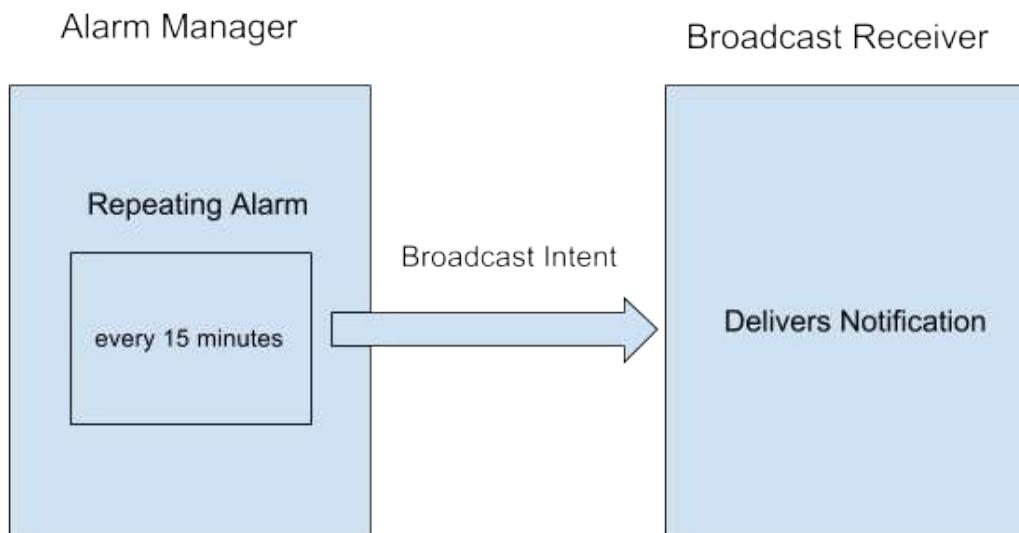
Pada tahap ini tidak ada alarm sama sekali: notifikasi segera dikirimkan saat tombol alih alarm diaktifkan. Dalam bagian ini, Anda akan mengimplementasikan AlarmManager untuk menjadwalkan dan mengirimkan notifikasi setiap 15 menit sekali.

## Tugas 3. Membuat Alarm Berulang

Sekarang karena aplikasi sudah dapat mengirimkan notifikasi, waktunya untuk mengimplementasikan komponen utama aplikasi: [AlarmManager](#). Ini adalah kelas yang akan bertanggung jawab untuk mengirimkan peggat secara berkala agar Anda berdiri. AlarmManager memiliki banyak jenis alarm bawaan, baik alarm satu waktu maupun alarm periodik, alarm eksak atau tidak eksak. Untuk mengetahui selengkapnya tentang jenis-jenis alarm, lihat [panduan ini](#).

AlarmManager, seperti notifikasi, menggunakan [PendingIntent](#) sehingga dikirim dengan opsi yang spesifik. Karena hal ini, AlarmManager dapat mengirimkan Intent saat aplikasi tidak berjalan lagi. Dalam aplikasi ini, PendingIntent akan mengirimkan siaran Intent dengan tindakan "Notify".

Intent siaran akan diterima oleh penerima siaran yang mengambil tindakan yang tepat (mengirimkan notifikasi).



[AlarmManager](#) dapat memicu peristiwa sekali waktu atau berulang yang terjadi saat perangkat tertidur atau aplikasi tidak berjalan. Peristiwa dapat dijadwalkan dengan pilihan [currentTimeMillis\(\)](#) saat menggunakan versi real time (RTC) atau [elapsedRealtime\(\)](#) saat menggunakan versi waktu yang sudah berlalu (ELAPSED\_REALTIME), dan mengirimkan PendingIntent saat terjadi. Untuk informasi selengkapnya tentang jam berbeda yang tersedia dan informasi tentang cara mengontrol waktu peristiwa, lihat [Referensi Developer SystemClock](#).

### 3.1 Menyiapkan intent tertunda siaran

AlarmManager bertanggung jawab untuk mengirimkan PendingIntent pada interval waktu yang telah ditentukan. PendingIntent ini akan mengirimkan intent siaran yang memberi tahu aplikasi kapan waktunya untuk memperbarui waktu yang tersisa dalam notifikasi.

- Buat konstanta string sebagai variabel anggota dalam MainActivity untuk digunakan sebagai tindakan intent siaran yang akan mengirimkan notifikasi:

```
private static final String ACTION_NOTIFY =
 "com.example.android.standup.ACTION_NOTIFY";
```

**Catatan:** gunakan nama paket yang berkualitas untuk string Intent, untuk memastikan Siaran unik, dan tidak dapat digunakan secara tidak sengaja oleh aplikasi lain dengan tindakan yang serupa.

- Buat Intent bernama notifyIntent dalam `onCreate()` dengan string khusus sebagai tindakannya:

```
Intent notifyIntent = new Intent(ACTION_NOTIFY);
```

- Buat PendingIntent notifikasi menggunakan konteks, variabel NOTIFICATION\_ID, intent notifikasi baru, dan bendera PendingIntent UPDATE\_CURRENT:

```
PendingIntent notifyPendingIntent = PendingIntent.getBroadcast
 (this, NOTIFICATION_ID, notifyIntent, PendingIntent.FLAG_UPDATE_CURRENT);
```

## 3.2 Menyetel alarm berulang

Anda sekarang akan menggunakan AlarmManager untuk mengirimkan Intent siaran setiap 15 menit. Untuk tugas ini, jenis alarm yang tepat adalah alarm tidak eksak berulang yang menggunakan waktu yang sudah lewat dan akan mengaktifkan perangkat jika tertidur. Jam real time tidak relevan di sini, karena kita ingin mengirimkan notifikasi setiap lima belas menit.

- Inisialisasi AlarmManager dalam `onCreate()` dengan memanggil `getSystemService()`:

```
AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
```

- Dalam metode `onCheckedChanged()`, panggil `setInexactRepeating()` pada instance alarm manager saat pengguna meneklik "ON" pada Alarm (Parameter keduanya adalah `true`). Anda akan menggunakan alarm `setInexactRepeating()` karena lebih efisien sumber daya untuk menggunakan penentuan waktu yang tidak tepat (sistem dapat membundel alarm dari beberapa aplikasi yang berbeda) dan alarm tidak perlu pas berulang setiap 15 menit. Metode `setInexactRepeating()` mengambil 4 argumen:
- Jenis alarm. Pada kasus ini Anda menggunakan tipe waktu yang sudah lewat sejak boot, karena hanya waktu relatif yang penting. Anda juga akan mengaktifkan perangkat jika tertidur, sehingga jenis alarmnya adalah `ELAPSED_REALTIME_WAKEUP`.
- Waktu pemicu dalam milidetik. Untuk ini, gunakan waktu yang sudah lewat, plus 15 menit. Untuk mendapatkan waktu yang sudah lewat, Anda bisa memanggil `SystemClock.elapsedRealtime()`. Lalu Anda bisa menggunakan konstanta AlarmManager untuk menambahkan 15 menit ke waktu yang sudah lewat: `AlarmManager.INTERVAL_FIFTEEN_MINUTES`.
- Interval waktu dalam milidetik. Anda ingin agar notifikasi diposting setiap 15 menit. Anda bisa menggunakan konstanta `AlarmManager.INTERVAL_FIFTEEN_MINUTES` lagi.
- PendingIntent yang akan dikirim. Anda membuat PendingIntent pada tugas sebelumnya.

```
long triggerTime = SystemClock.elapsedRealtime()
 + AlarmManager.INTERVAL_FIFTEEN_MINUTES;

long repeatInterval = AlarmManager.INTERVAL_FIFTEEN_MINUTES;

//If the Toggle is turned on, set the repeating alarm with a 15 minute interval
alarmManager.setInexactRepeating(AlarmManager.ELAPSED_REALTIME_WAKEUP,
 triggerTime, repeatInterval, notifyPendingIntent);
```

**Catatan:** Karena Anda mengakses instance AlarmManager dan `notifyPendingIntent` dari kelas bagian dalam yang

- anonim, Android Studio dapat menjadikan instance ini final. Jika tidak, Anda harus menjadikannya final sendiri.
7. Hapus panggilan ke `deliverNotification()` dalam metode `onCheckedChanged()`.
  8. Jika tombol alih alarm dinonaktifkan (dengan mengeklik toggle dalam status ON), batalkan alarm dengan memanggil `cancel()` pada `AlarmManager`, yang meneruskan intent tertunda yang digunakan untuk membuat alarm.
- ```
alarmManager.cancel(notifyPendingIntent);
```

Pertahankan panggilan ke `cancelAll()` di `NotificationManager`, karena menonaktifkan tombol alih seharusnya tetap menghapus notifikasi yang ada.

`AlarmManager` sekarang akan mulai mengirimkan Intent Siaran mulai dari lima belas menit sejak Alarm disetel, dan setiap lima belas menit setelahnya. Aplikasi harus bisa merespons intent ini dengan mengirimkan notifikasi. Pada langkah selanjutnya Anda akan membuat subkelas `Penerima Siaran` untuk menerima intent siaran dan mengirimkan notifikasi.

3.3 Membuat Penerima Siaran

`Penerima Siaran` bertanggung jawab untuk menerima intent siaran dari `AlarmManager` dan bereaksi dengan tepat.

1. Dalam Android Studio, klik **File > New > Other > Broadcast Receiver**.
2. Masukkan `AlarmReceiver` sebagai namanya, pastikan kotak centang `Exported` sudah tidak dicentang (untuk memastikan aplikasi lain tidak dapat memanggil `Penerima Siaran` ini). Anda juga bisa mengubah setelan ini dalam `AndroidManifest` dengan menyetel atribut `android:exported` ke `false`. Android Studio akan membuat subkelas `BroadcastReceiver` dengan metode yang diperlukan (`onReceive()`), dan menambahkan penerima ke `AndroidManifest`. Anda mungkin perlu menambahkan Filter Intent ke tag `<receiver>` dalam `AndroidManifest` untuk memilih Intent Siaran masuk yang tepat.
3. Dalam `Manifests Android`, buat `<intent-filter>` tag pembuka dan penutup di antara tag `<receiver>` buat sebuah item `<action>` dalam filter maksud dengan `android:name` disetel ke string tindakan `ACTION_NOTIFY` khusus yang Anda buat:

```
<intent-filter>
    <action android:name="com.example.android.standup.ACTION_NOTIFY" />
</intent-filter>
```

4. Salin dan tempel metode `deliverNotification()` ke metode `onReceive()` dalam `BroadcastReceiver` dan panggil dari `onReceive()`. Pengelola notifikasi dan id notifikasi belum diinisialisasi dalam kelas `BroadcastReceiver` jadi akan disorot dengan warna merah.
5. Salin variabel `NOTIFICATION_ID` dari `MainActivity` ke dalam kelas `BroadcastReceiver`.
6. Inisialisasi `NotificationManager` di awal metode `onReceive()`. Anda harus memanggil `getSystemService()` dari yang diteruskan dalam Konteks:

```
NotificationManager notificationManager = (NotificationManager)
    context.getSystemService(Context.NOTIFICATION_SERVICE);
```

7. Hapus baris yang memunculkan `UnsupportedOperationException`.
8. Jalankan aplikasi Anda. Jika tidak ingin menunggu lima belas menit untuk melihat notifikasi, Anda bisa mengubah waktu pemicu ke `SystemClock.elapsedRealtime()` untuk segera melihat notifikasi. Anda juga bisa mengubah interval ke waktu yang lebih singkat untuk memastikan bahwa alarm berulang berfungsi.

Sekarang Anda memiliki aplikasi yang dapat menjadwalkan dan melakukan operasi berulang, bahkan jika aplikasi tidak lagi berjalan. Lanjutkan, keluar dari aplikasi, notifikasi akan tetap dikirimkan. Ada satu komponen terakhir yang akan memastikan pengalaman pengguna yang layak: jika aplikasi ditutup, tombol alih akan disetel ulang ke status nonaktif, bahkan jika alarm sudah disetel. Untuk memperbaikinya, Anda perlu memeriksa status alarm setiap kali aplikasi diluncurkan.

3.5 Memeriksa Status Alarm

Untuk melacak status alarm, Anda memerlukan variabel boolean yang `true` jika alarm sudah ada, dan `false` jika belum. Untuk menyetel boolean ini, Anda bisa memanggil `PendingIntent.getBroadcast()` dengan bendera `PendingIntent.FLAG_NO_CREATE`. Dalam hal ini, `PendingIntent` dikembalikan jika sudah ada, atau kelas mengembalikan null. Memeriksa apakah alarm sudah disetel sangatlah berguna.

Catatan: Saat Anda membuat `PendingIntent`, sistem menggunakan metode `Intent.filterEquals()` untuk menentukan apakah `PendingIntent` dengan `Intent` yang sama sudah ada. Ini artinya untuk memiliki `PendingIntent` yang berbeda, `Intent` berisi harus berbeda dalam salah satu tindakan, data, tipe, kelas, atau kategori. Ekstra intent tidak termasuk dalam perbandingan. Bendera `PendingIntent` menentukan apa yang terjadi saat `PendingIntent` yang `Intent`-nya cocok dengan yang Anda coba sudah ada. Dalam kasus bendera `NO_CREATE`, ini akan mengembalikan null kecuali `PendingIntent` dengan `Intent` yang cocok sudah ada.

1. Buat boolean yang benar jika `PendingIntent` tidak null, dan sebaliknya salah, menggunakan strategi ini. Gunakan boolean untuk menyetel status `ToggleButton` dengan benar saat aplikasi dimulai. Kode ini harus datang sebelum `PendingIntent` dibuat, atau kode akan selalu mengembalikan benar:

```
boolean alarmUp = (PendingIntent.getBroadcast(this, NOTIFICATION_ID, notifyIntent,
    PendingIntent.FLAG_NO_CREATE) != null);
```

2. Setel keadaan tombol alih segera setelah Anda mendefinisikan boolean `alarmUp`:

```
alarmToggle.setChecked(alarmUp);
```

Ini akan memastikan tombol alih selalu diaktifkan jika alarm disetel, dan dinonaktifkan jika tidak. Itu saja. Sekarang Anda memiliki alarm terjadwal yang akan mengingatkan Anda untuk berdiri setiap lima belas menit.

3. Jalankan aplikasi Anda. Aktifkan alarm. Keluar dari aplikasi. Buka aplikasi lagi. Tombol alarm akan menunjukkan bahwa alarm aktif.

Kode solusi

**Proyek Android Studio: [StandUp](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Kelas `AlarmManager` juga menangani jam alarm untuk fungsi yang biasa, misalnya yang membangunkan Anda di pagi hari. Pada perangkat yang menjalankan API 21+, Anda bisa mendapatkan informasi tentang jam alarm jenis ini berikutnya dengan memanggil `getNextAlarmClock()` pada pengelola alarm.

Tambahkan tombol ke aplikasi yang menampilkan waktu jam alarm berikutnya, yang telah disetel pengguna dalam pesan `Toast`.

Rangkuman

- `AlarmManager` mengizinkan Anda untuk menjadwalkan tugas berdasarkan jam real time atau waktu yang sudah lewat sejak booting.
- `AlarmManager` menyediakan berbagai jenis alarm, yang periodik dan sekali waktu, dengan opsi untuk mengaktifkan perangkat jika tertidur.
- `AlarmManager` ditujukan untuk situasi saat penentuan waktu yang tepat sangatlah penting (seperti acara kalender). Jika tidak, pertimbangkan framework [Penjadwalan Pekerjaan](#) untuk penentuan waktu dan penjadwalan yang lebih efisien sumber daya.
- Gunakan versi penentuan waktu yang tidak tepat dari `AlarmManager` kapan pun memungkinkan untuk meminimalkan beban yang disebabkan oleh banyaknya perangkat pengguna atau aplikasi yang melakukan tugas pada waktu yang

- bersamaan.
- AlarmManager menggunakan PendingIntent untuk melakukan operasi, jadi Anda bisa menjadwalkan siaran, layanan, dan aktivitas menggunakan PendingIntent yang tepat.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Menjadwalkan Alarm](#)

Ketahui selengkapnya

Dokumentasi Developer Android

Panduan

- [Menjadwalkan Alarm Berulang](#)

Referensi

- [AlarmManager](#)
- [SystemClock](#)

Sumber daya web lainnya

- [Entri Blog tentang memilih jenis alarm yang benar](#)

8.3: Job Scheduler

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Tugas 1. Mengimplementasikan Layanan Pekerjaan
- Tugas 2. Mengimplementasikan kondisi pekerjaan
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Anda sudah melihat bahwa Anda bisa memicu acara berdasarkan jam real-time atau waktu yang sudah lewat sejak booting menggunakan kelas `AlarmManager`. Namun sebagian besar tugas tidak memerlukan waktu tepat, tetapi harus dijadwalkan berdasarkan kombinasi sistem dan persyaratan pengguna. Misalnya, aplikasi berita mungkin memperbarui berita di pagi hari, tetapi dapat menunggu sampai perangkat diisi daya dan dihubungkan ke wifi untuk memperbarui berita, untuk mempertahankan data pengguna dan sumber daya sistem.

Kelas `JobScheduler` ditujukan untuk penjadwalan seperti ini; kelas ini mengizinkan Anda menyetel kondisi, atau parameter menjalankan tugas. Berdasarkan kondisi ini, `JobScheduler` menghitung waktu terbaik untuk menjadwalkan eksekusi pekerjaan. Beberapa contoh parameter ini adalah: persistensi pekerjaan di semua boot ulang, interval yang harus dijalankan pekerjaan, terlepas dari apakah perangkat dicolokkan ke arus listrik atau tidak, atau apakah perangkat tersebut sedang idle atau tidak.

Tugas yang akan dijalankan diimplementasikan sebagai subkelas `JobService` dan dieksekusi berdasarkan batasan yang telah ditentukan.

`JobScheduler` hanya tersedia pada perangkat yang menjalankan API 21+ dan saat ini tidak tersedia dalam pustaka dukungan. Untuk kompatibilitas mundur, gunakan `GcmNetworkManager` (akan segera menjadi `FirebaseJobDispatcher`).

Dalam praktik ini Anda akan membuat aplikasi yang menjadwalkan notifikasi untuk diposkan saat parameter disetel oleh pengguna terpenuhi, dan persyaratan sistemnya dipenuhi.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Mengirimkan notifikasi.
- Mendapatkan nilai integer dari tampilan Spinner.
- Menggunakan tampilan Switch untuk masukan pengguna.
- Membuat `PendingIntent`.

Yang akan Anda PELAJARI

Anda akan belajar:

- Mengimplementasikan `JobService`.
- Membuat objek `JobInfo` dengan batasan spesifik.
- Menjadwalkan `JobService` berdasarkan objek `JobInfo`.

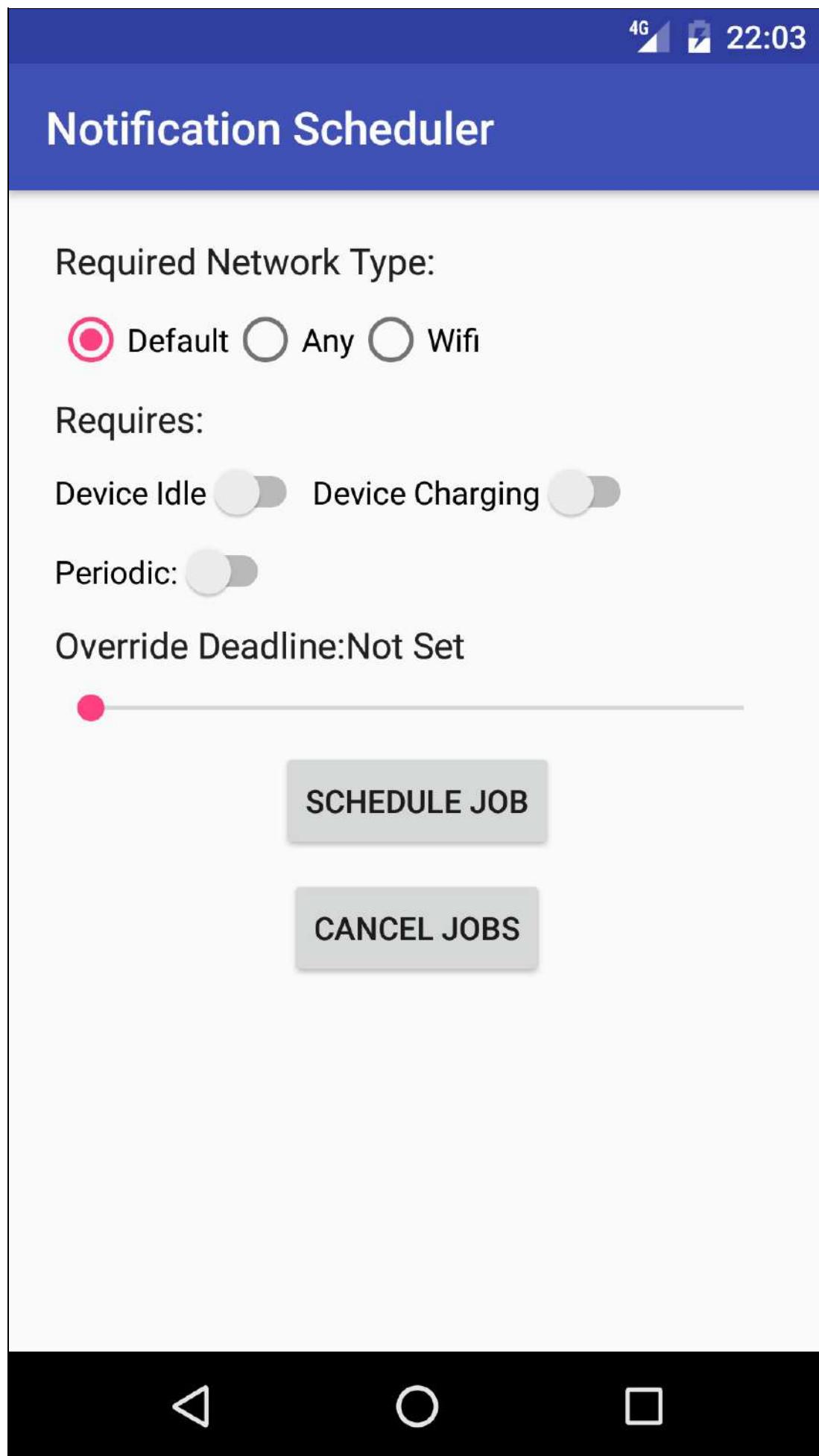
Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Mengimplementasikan JobService yang mengirimkan notifikasi sederhana untuk memberi tahu pengguna bahwa pekerjaan sedang berjalan.
- Mendapatkan masukan pengguna untuk mengonfigurasi batasan (seperti menunggu sampai perangkat diisi daya) di JobService yang Anda jadwalkan.
- Menjadwalkan pekerjaan menggunakan JobScheduler.

Ringkasan Aplikasi

Dalam praktik ini Anda akan membuat aplikasi bernama "Notification Scheduler". Aplikasi akan mendemonstrasikan framework JobScheduler dengan mengizinkan pengguna untuk memilih batasan dan menjadwalkan pekerjaan. Saat pekerjaan dieksekusi, notifikasi akan diposkan (dalam aplikasi ini, notifikasi secara efektif adalah "pekerjaan" Anda).



Untuk menggunakan JobScheduler, Anda memerlukan dua bagian tambahan: `JobService` dan `JobInfo`. Objek `JobInfo` berisi serangkaian kondisi yang akan memicu pekerjaan untuk berjalan. `JobService` adalah implementasi pekerjaan yang akan dijalankan di bawah kondisi tersebut.

Tugas 1. Mengimplementasikan JobService

Pertama, Anda harus membuat layanan yang akan dijalankan pada waktu yang ditetapkan oleh kondisi. `JobService` secara otomatis dieksekusi oleh sistem, dan bagian yang perlu Anda implementasikan adalah:

callback `onStartJob()`

- dipanggil saat sistem menentukan bahwa tugas harus dijalankan. Anda mengimplementasikan pekerjaan untuk dikerjakan dalam metode ini.
Catatan: `onStartJob()` dieksekusi pada thread utama, sehingga tugas yang berjalan lama harus diturunkan ke thread yang lain. Dalam hal ini, Anda hanya perlu mengeposkan notifikasi, yang bisa dilakukan dengan aman pada thread utama.
- mengembalikan boolean yang menunjukkan apakah pekerjaan perlu diteruskan pada thread yang terpisah. Jika benar, pekerjaan diturunkan ke thread yang berbeda dan aplikasi harus memanggil `jobFinished()` secara eksplisit dalam thread untuk menunjukkan bahwa pekerjaan sudah selesai. Jika nilai pengembaliamnya adalah salah, framework tahu bahwa pekerjaan sudah diselesaikan di akhir `onStartJob()` dan secara otomatis akan memanggil `jobFinished()` atas nama Anda.

callback `onStopJob()`

- dipanggil jika kondisi tidak terpenuhi, yang berarti pekerjaan harus dihentikan.
- mengembalikan boolean yang menentukan apa yang harus dilakukan jika pekerjaan tidak selesai. Jika nilai pengembaliamnya true, pekerjaan akan dijadwalkan ulang, atau akan dibatalkan.

1.1 Membuat Proyek dan NotificationJobService

Verifikasi bahwa SDK minimum yang Anda gunakan adalah API 21. Sebelum API 21, JobScheduler tidak berfungsi, karena kekurangan beberapa API yang diperlukan.

1. Gunakan template kosong, dan buat proyek baru bernama "Notification Scheduler".
2. Buat kelas baru bernama `NotificationJobService` yang memperluas `JobService`.
3. Tambahkan metode yang diperlukan: `onStartJob()` dan `onStopJob()`.
4. Dalam file `AndroidManifest.xml`, daftarkan `JobService` dengan izin berikut dalam tag `<application>`:

```
<service
    android:name=".NotificationJobService"
    android:permission="android.permission.BIND_JOB_SERVICE"/>
```

1.2 Mengimplementasikan `onStartJob()`

1. Tambahkan ikon notifikasi untuk notifikasi "Job Running".
2. Dalam `onStartJob()`, buat `PendingIntent` untuk meluncurkan `MainActivity` aplikasi untuk digunakan sebagai intent konten untuk notifikasi Anda.
3. Dalam `onStartJob()`, konstruksi dan kirimkan notifikasi dengan atribut berikut:

| Atribut | Judul |
|----------------|----------------------------------|
| Content Title | "Job Service" |
| Content Text | "Your Job is running!" |
| Content Intent | contentPendingIntent |
| Small Icon | R.drawable.ic_job_running |
| Priority | NotificationCompat.PRIORITY_HIGH |
| Defaults | NotificationCompat.DEFAULT_ALL |
| AutoCancel | true |

4. Pastikan `onStartJob()` mengembalikan false, karena semua pekerjaan diselesaikan dalam callback itu.
5. Buat `onStopJob()` mengembalikan true, agar pekerjaan dijadwalkan ulang jika gagal.

```

@Override
public boolean onStartJob(JobParameters jobParameters) {
    //Set up the notification content intent to launch the app when clicked
    PendingIntent contentPendingIntent = PendingIntent.getActivity
        (this, 0, new Intent(this, MainActivity.class),
        PendingIntent.FLAG_UPDATE_CURRENT);

    NotificationManager manager =
        (NotificationManager) getSystemService(NOTIFICATION_SERVICE);

    NotificationCompat.Builder builder = new NotificationCompat.Builder(this)
        .setContentTitle(getString(R.string.job_service))
        .setContentText(getString(R.string.job_running))
        .setContentIntent(contentPendingIntent)
        .setSmallIcon(R.drawable.ic_job_running)
        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setDefaults(NotificationCompat.DEFAULT_ALL)
        .setAutoCancel(true);

    manager.notify(0, builder.build());

    return false;
}

```

Tugas 2. Mengimplementasikan kondisi kerja

Sekarang karena Anda telah memiliki JobService, ini saatnya mengidentifikasi kriteria untuk menjalankan pekerjaan. Untuk ini, gunakan komponen JobInfo. Anda akan membuat serangkaian kondisi berparameter untuk menjalankan pekerjaan menggunakan berbagai tipe konektivitas jaringan dan status perangkat.

Pertama-tama, Anda akan membuat grup tombol radio untuk menentukan jenis jaringan yang diperlukan untuk pekerjaan ini.

2.1 Mengimplementasikan batasan jaringan

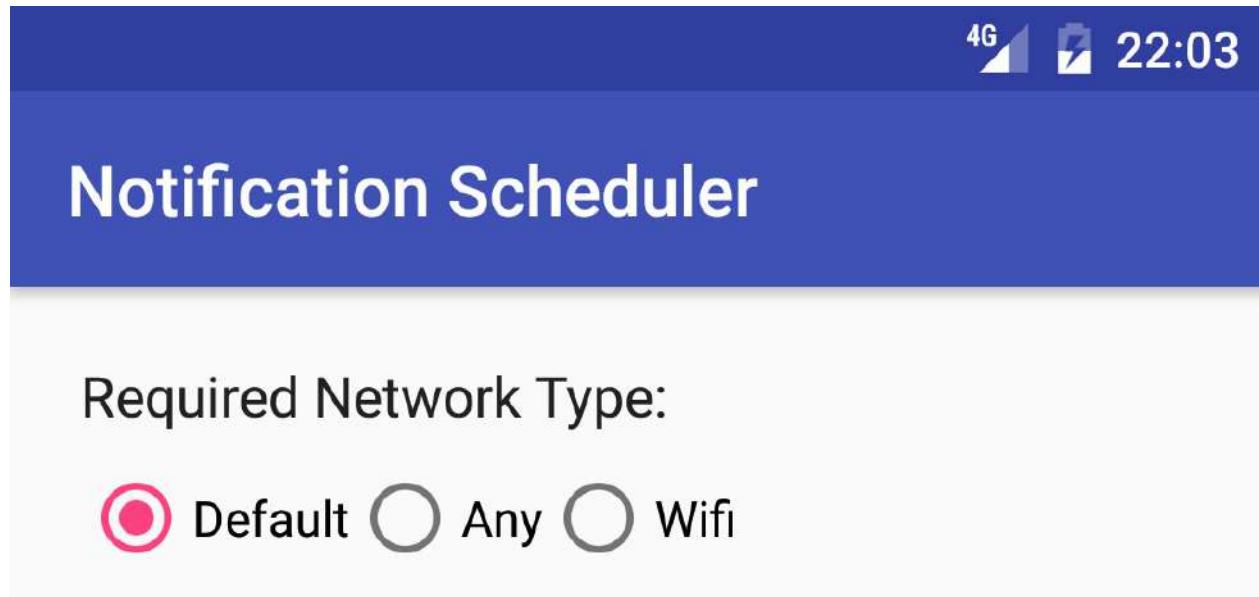
Salah satu dari kondisi yang memungkinkan untuk menjalankan pekerjaan adalah status konektivitas jaringan. Anda bisa membatasi JobService untuk dieksekusi hanya ketika kondisi jaringan tertentu dipenuhi. Opsinya adalah:

- **NETWORK_TYPE_NONE**: pekerjaan akan berjalan dengan atau tanpa koneksi jaringan. Ini adalah nilai default-nya.
- **NETWORK_TYPE_ANY**: pekerjaan akan berjalan selama jaringan (seluler, wifi) tersedia.
- **NETWORK_TYPE_UNMETERED**: pekerjaan akan berjalan selama perangkat terhubung ke wifi yang tidak

menggunakan HotSpot.

Buat layout untuk aplikasi

Buat layout untuk aplikasi untuk menunjukkan tombol yang bisa dipakai pengguna untuk memilih kriteria jaringan.



- Dalam file activity_main.xml, ubah elemen rootview ke LinearLayout ***vertical***.
- Ubah TextView agar memiliki atribut berikut:

| Atribut | Nilai |
|------------------------|-------------------------------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:text | "Network Type Required: " |
| android:textAppearance | "@style/TextAppearance.AppCompat.Subhead" |
| android:layout_margin | "4dp" |

- Tambahkan elemen kontainer RadioGroup di bawah TextView dengan atribut berikut:

| Atribut | Nilai |
|-----------------------|-----------------------|
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:orientation | "horizontal" |
| android:id | "@+id/networkOptions" |
| android:layout_margin | "4dp" |

Catatan: Menggunakan radio memastikan bahwa hanya satu dari anaknya yang dapat pilih pada satu waktu. Untuk informasi selengkapnya tentang Tombol Radio, lihat [panduan ini](#).

- Tambahkan tiga RadioButton sebagai anak ke RadioGroup dengan tinggi dan lebar layout yang disetel ke "wrap_content" dan atribut berikut ini:

| | |
|----------------------|--------------------|
| RadioButton 1 | |
| android:text | "None" |
| android:id | "@+id/noNetwork" |
| android:checked | true |
| RadioButton 2 | |
| android:text | "Any" |
| android:id | "@+id/anyNetwork" |
| RadioButton 3 | |
| android:text | "Wifi" |
| android:id | "@+id/wifiNetwork" |

5. Tambahkan dua tombol di bawah grup tombol radio dengan tinggi dan lebar yang disetel ke "wrap_content" dengan atribut berikut ini:

| | |
|------------------------|---------------------|
| Button 1 | |
| android:text | "Schedule Job" |
| android:onClick | "scheduleJob" |
| android:layout_gravity | "center_horizontal" |
| android:layout_margin | "4dp" |
| Button 2 | |
| android:text | "Cancel Jobs" |
| android:onClick | "cancelJobs" |
| android:layout_gravity | "center_horizontal" |
| android:layout_margin | "4dp" |

6. Tambahkan stub metode untuk kedua metode `onClick()` di MainActivity.

Dapatkan opsi jaringan pilihan

- Dalam `scheduleJob()`, temukan RadioGroup menurut id dan simpan dalam variabel instance bernama `networkOptions`.
- Dapatkan id jaringan pilihan dan simpan dalam variabel integer:

```
int selectedNetworkID = networkOptions.getCheckedRadioButtonId();
```

- Buat variabel integer opsi jaringan pilihan dan setel agar setara dengan opsi jaringan default (jaringan tidak diperlukan):

```
int selectedNetworkOption = JobInfo.NETWORK_TYPE_NONE;
```

4. Buat pernyataan switch dengan id jaringan pilihan, dan tambahkan kasus untuk setiap id yang mungkin:

```
switch(selectedNetworkID){
    case R.id.noNetwork:
        break;
    case R.id.anyNetwork:
        break;
    case R.id.wifiNetwork:
        break;
}
```

5. Tetapkan opsi jaringan pilihan ke konstanta jaringan JobInfo yang sesuai, tergantung kasusnya:

```
switch(selectedNetworkID){
    case R.id.noNetwork:
        selectedNetworkOption = JobInfo.NETWORK_TYPE_NONE;
        break;
    case R.id.anyNetwork:
        selectedNetworkOption = JobInfo.NETWORK_TYPE_ANY;
        break;
    case R.id.wifiNetwork:
        selectedNetworkOption = JobInfo.NETWORK_TYPE_UNMETERED;
        break;
}
```

Buat JobScheduler dan objek JobInfo

1. Dalam MainActivity, buat variabel anggota untuk JobScheduler, dan inisialisasikan di `scheduleJob()` menggunakan `getSystemService()`:

```
mScheduler = (JobScheduler) getSystemService(JOB_SCHEDULER_SERVICE);
```

2. Buat konstanta anggota untuk JOB_ID, dan setel agar setara dengan 0.
 3. Buat objek `JobInfo.Builder` dalam `scheduleJob()`. Konstruktor untuk kelas `JobInfo.Builder` memerlukan dua parameter:
 - JOB_ID.
 - ComponentName untuk JobService yang Anda buat. ComponentName digunakan untuk mengidentifikasi JobService dengan objek JobInfo.

```
ComponentName serviceName = new ComponentName(getApplicationContext(),
NotificationJobService.class.getName());
JobInfo.Builder builder = new JobInfo.Builder(JOB_ID, serviceName)
```

4. Panggil `setRequiredNetworkType()` pada objek `JobInfo.Builder`, meneruskan opsi jaringan pilihan:

```
.setRequiredNetworkType(selectedNetworkOption);
```

5. Panggil `schedule()` pada objek `JobScheduler`, meneruskan objek `JobInfo` dengan metode `build()`:

```
JobInfo myJobInfo = builder.build();
mScheduler.schedule(myJobInfo);
```

6. Tampilkan pesan Toast, beri tahu pengguna bahwa pekerjaan dijadwalkan.
 7. Dalam metode `cancelJobs()`, periksa apakah objek `JobScheduler` adalah null, dan jika tidak, panggil `cancelAll()` pada metode untuk menghapus semua pekerjaan yang tertunda, setel ulang `JobScheduler` ke null, dan tampilkan pesan Toast untuk memberi tahu pengguna bahwa pekerjaan dibatalkan:

```

if (mScheduler!=null){
    mScheduler.cancelAll();
    mScheduler = null;
    Toast.makeText(this, "Jobs Canceled", Toast.LENGTH_SHORT).show();
}

```

- Jalankan aplikasi. Anda sekarang bisa menyetel tugas yang memiliki keterbatasan jaringan dan melihat berapa lama waktu yang dibutuhkan untuk dieksekusi. Dalam hal ini, tugasnya adalah mengirimkan notifikasi. Untuk menghilangkan notifikasi, gesek layar atau ketuk untuk membuka notifikasi.

Anda mungkin melihat bahwa jika Anda tidak mengubah batasan jaringan ke "Any" atau "Wifi", aplikasi akan crash dengan pengecualian berikut ini:

```

java.lang.IllegalArgumentException:
    You're trying to build a job with no constraints, this is not allowed.

```

Ini karena kondisi "No Network Required" adalah default-nya dan sebenarnya tidak dihitung sebagai batasan. JobScheduler membutuhkan paling tidak satu batasan untuk menjadwalkan JobService dengan benar. Dalam bagian ini Anda akan membuat ketentuan yaitu `true` saat paling tidak satu batasan disetel, dan `false` jika tidak. Lalu Anda perlu menjadwalkan tugas jika `true`, dan tampilkan Toast untuk memberi tahu pengguna untuk menyetel batasan jika tidak.

2.2 Memeriksa batasan

JobScheduler memerlukan paling tidak satu batasan untuk disetel. Dalam tugas ini Anda akan membuat boolean yang akan melacak apakah persyaratan ini sudah dipenuhi, jadi Anda bisa memberi tahu pengguna untuk menyetel paling tidak satu batasan jika mereka belum melakukannya. Saat membuat opsi tambahan pada langkah-langkah berikutnya, Anda perlu memodifikasi boolean ini agar selalu `true` jika paling tidak salah satu batasan disetel dan `false` jika tidak.

- Buat variabel boolean bernama constraintSet yang true jika opsi jaringan pilihannya bukan default

`JobInfo.NETWORK_TYPE_NONE`:

```

boolean constraintSet = selectedNetworkOption != JobInfo.NETWORK_TYPE_NONE;

```

- Buat blok if/else menggunakan boolean `constraintSet`.
- Pindahkan kode yang menjadwalkan tugas dan menampilkan pesan Toast ke dalam blok `if`.
- Jika `constraintSet` adalah `false`, tampilkan pesan Toast ke pengguna untuk menyetel paling tidak satu batasan. Jangan lupa untuk mengekstrak sumber daya string:

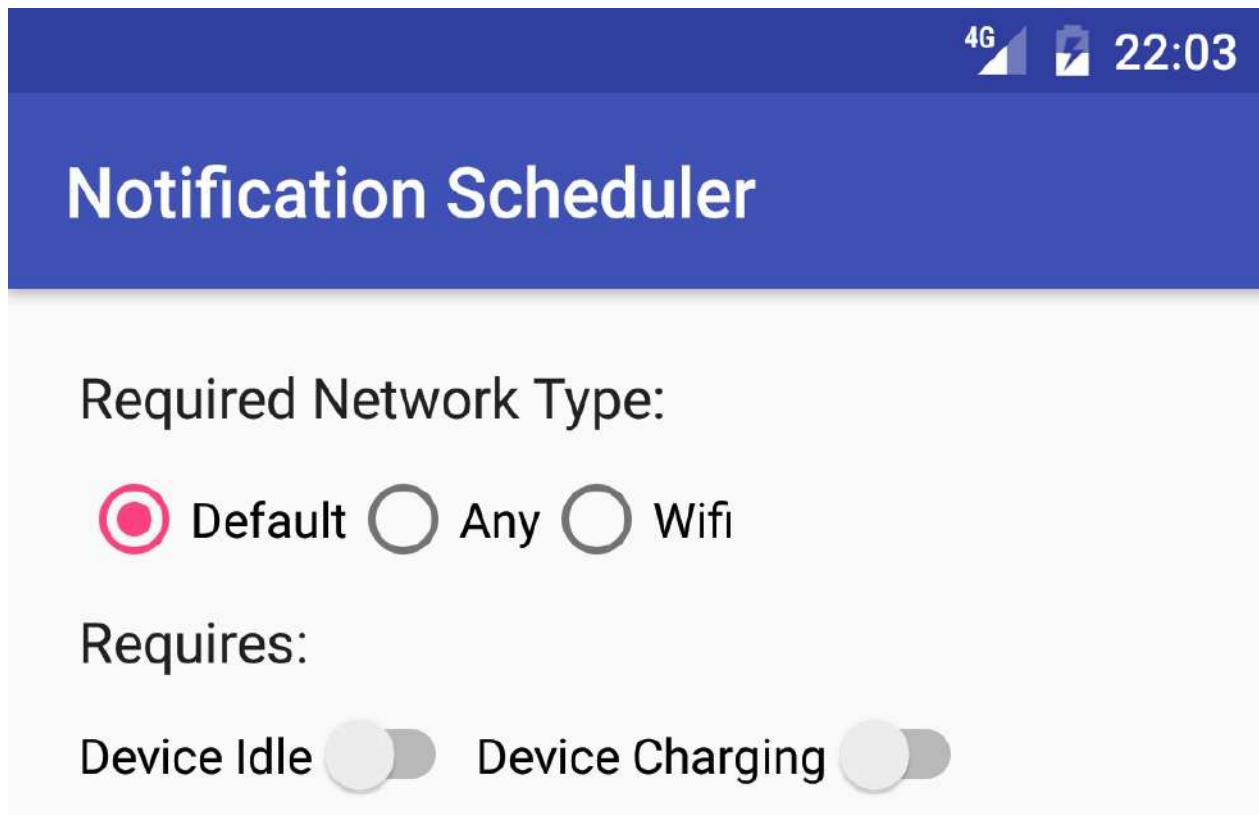
```

if(constraintSet) {
    //Schedule the job and notify the user
    JobInfo myJobInfo = builder.build();
    mScheduler.schedule(myJobInfo);
    Toast.makeText(this, R.string.job_scheduled, Toast.LENGTH_SHORT).show();
} else {
    Toast.makeText(this, R.string.no_constraint_toast, Toast.LENGTH_SHORT).show();
}

```

2.3 Mengimplementasikan batasan Device Idle dan Device Charging

JobScheduler menyertakan kemampuan untuk menunggu hingga perangkat diisi dayanya atau dalam status idle (layar dinonaktifkan dan CPU telah tertidur) untuk mengeksekusi JobService. Anda sekarang akan menambahkan switch ke aplikasi untuk mengganti batasan ini pada JobService. **Menambahkan elemen UI untuk batasan baru**



1. Dalam file activity_main.xml, salin TextView label jenis jaringan dan salin di bawah RadioGroup.
2. Ubah atribut `android:text` ke "Requires:".
3. Di bawah textView ini, sisipkan LinearLayout horizontal dengan margin 4dp.
4. Buat dua tampilan Switch sebagai anak bagi LinearLayout horizontal dengan tinggi dan lebar yang disetel ke "wrap_content" dan atribut berikut ini:

| Switch 1 | |
|-----------------|-----------------------|
| android:text | "Device Idle" |
| android:id | "@+id/idleSwitch" |
| Switch 2 | |
| android:text | "Device Charging" |
| android:id | "@+id/chargingSwitch" |

Tambahkan kode untuk batasan baru

1. Dalam MainActivity, buat variabel anggota, `mDeviceIdle` dan `mDeviceCharging`, untuk switch dan inisialisasikan switch di `onCreate()`.
2. Dalam metode `scheduleJob()`, tambahkan panggilan berikut untuk menyetel batasan pada JobScheduler berdasarkan pilihan pengguna dalam switch:

```
builder.setRequiresDeviceIdle(mDeviceIdle.isChecked());
builder.setRequiresCharging(mDeviceCharging.isChecked());
```

3. Perbarui kode yang menyetel `constraintSet` untuk mempertimbangkan batasan baru ini:

```
boolean constraintSet = (selectedNetworkOption != JobInfo.NETWORK_TYPE_NONE)
    || mDeviceChargingSwitch.isChecked() || mDeviceIdleSwitch.isChecked();
```

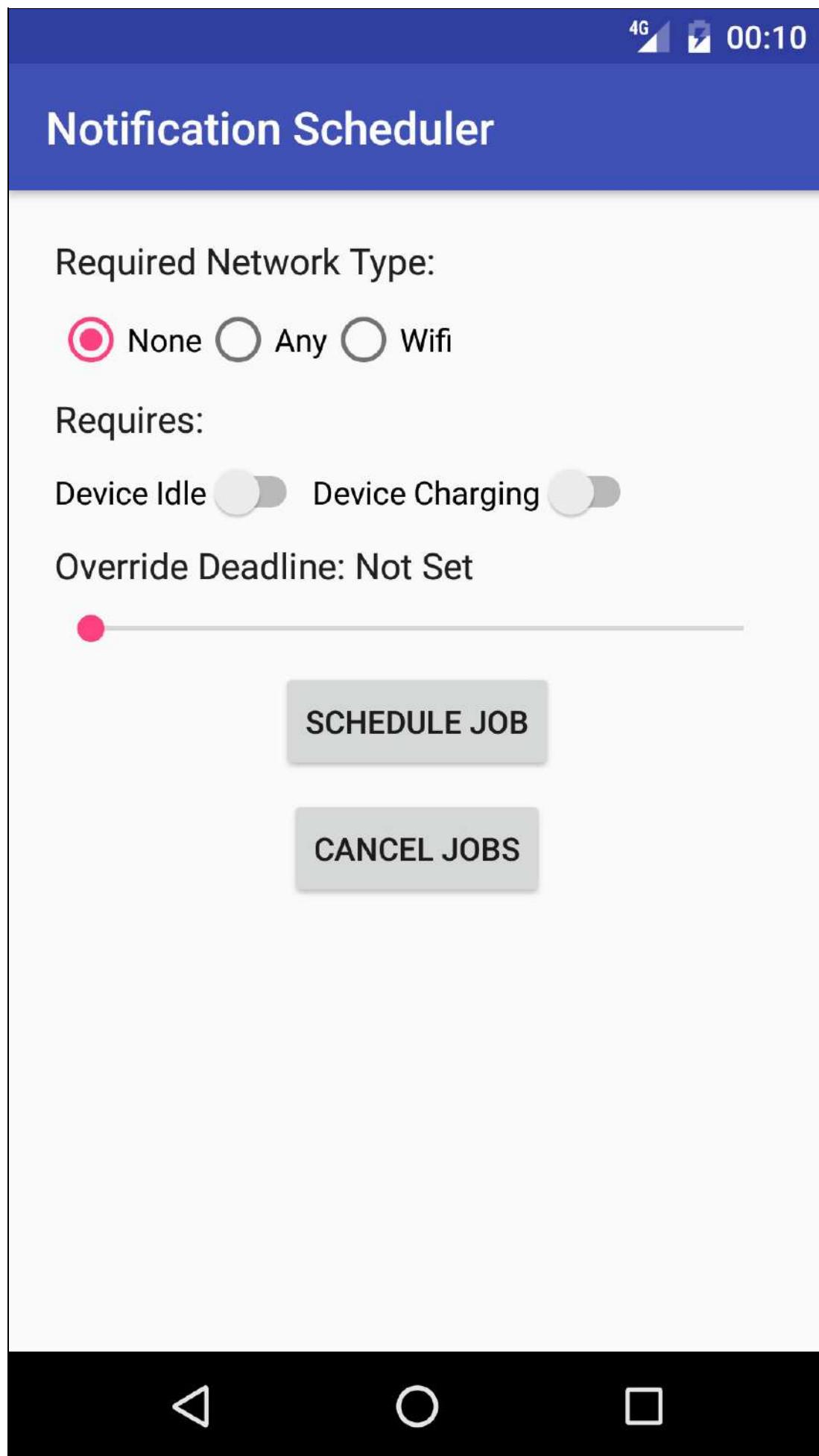
4. Jalankan aplikasi, sekarang dengan batasan tambahan. Coba kombinasi perbedaan switch untuk melihat saat notifikasi dikirimkan (yang menandakan bahwa pekerjaan dijalankan). Anda bisa menguji batasan status pengisian daya dalam emulator dengan membuka menu (ikon elips di sebelah perangkat yang diemulasi), buka panel **Battery** dan alihkan menu tarik turun Battery Status. Tidak ada cara untuk menyetel emulator dalam mode Idle secara manual saat praktik ini ditulis.

Menunggu sampai perangkat berada dalam mode idle dan dicolokkan adalah pola umum untuk tugas yang menguras baterai seperti mengunduh atau mengunggah file besar.

2.4 Mengimplementasikan batasan Batas Waktu Pengganti

Sampai tahap ini, tidak ada cara untuk dengan tepat mengetahui kapan framework akan mengeksekusi tugas. Sistem memperhitungkan manajemen sumber daya efektif yang mungkin menunda tugas Anda tergantung pada status perangkat, dan tidak menjamin bahwa tugas akan berjalan tepat waktu. Misalnya, aplikasi berita baru mungkin ingin mengunduh berita terbaru hanya ketika wifi tersedia dan perangkat dicolokkan dan diisi dayanya; tetapi pengguna bisa tidak sengaja lupa mengaktifkan wifi atau mengisi daya perangkat. Jika Anda tidak menambahkan parameter waktu ke pekerjaan terjadwal, pengguna tersebut akan kecewa jika mereka membaca berita kemarin saat terbangun di pagi hari. Untuk alasan ini, API JobScheduler menyertakan kemampuan untuk menetapkan batas waktu keras yang akan mengganti batasan sebelumnya.

Tambahkan UI baru untuk menyetel batas waktu menjalankan tugas



Dalam langkah ini Anda akan menggunakan komponen UI baru, [SeekBar](#), untuk mengizinkan pengguna menyetel batas waktu antara 0 hingga 100 detik untuk mengeksekusi tugas.

Pengguna menyetel nilai dengan menyeret SeekBar.

- Buat LinearLayout horizontal di bawah LinearLayout yang ada dengan switch-nya, yang akan berisi label untuk SeekBar.
- SeekBar akan memiliki dua label: label statis seperti label untuk RadioGroup tombol, dan label dinamis yang akan diperbarui dengan nilai dari SeekBar. Tambahkan dua TextView ke LinearLayout dengan atribut berikut ini:

| | |
|------------------------|-------------------------------------------|
| TextView 1 | |
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:text | "Override Deadline: " |
| android:id | "@+id/seekBarLabel" |
| android:textAppearance | "@style/TextAppearance.AppCompat.Subhead" |
| TextView 2 | |
| android:layout_width | "wrap_content" |
| android:layout_height | "wrap_content" |
| android:text | "Not Set" |
| android:id | "@+id/seekBarProgress" |
| android:textAppearance | "@style/TextAppearance.AppCompat.Subhead" |

- Tambahkan tampilan SeekBar di bawah LinearLayout dengan atribut berikut ini:

| Atribut | Nilai |
|-----------------------|----------------|
| android:layout_width | "match_parent" |
| android:layout_height | "wrap_content" |
| android:id | "@+id/seekBar" |
| android:layout_margin | "4dp" |

Tulis kode untuk menambahkan batas waktu

- Dalam MainActivity, buat variabel anggota untuk SeekBar dan inisialisasikan di `onCreate()`:

```
mSeekBar = (SeekBar) findViewById(R.id.seekBar);
```

- Buat variabel akhir untuk kedua TextView (yang akan diakses dari kelas bagian dalam) dan inisialisasikan di `onCreate()`:

```
final TextView label = (TextView) findViewById(R.id.seekBarLabel);
final TextView seekBarProgress = (TextView) findViewById(R.id.seekBarProgress);
```

3. Dalam `onCreate()`, panggil `setOnSeekBarChangeListener()` di `SeekBar`, yang meneruskan `OnSeekBarChangeListener` baru (Android Studio harus membuat metode yang diperlukan):

```
mSeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {}

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {}

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {}
});
```

4. Argumen kedua `onProgressChanged()` adalah nilai saat ini dari `SeekBar`. Dalam callback `onProgressChanged()`, periksa apakah nilai integer lebih besar dari 0 (berarti nilai telah ditetapkan oleh pengguna), dan jika iya, setel label kemajuan `SeekBar` ke nilai integer, yang diikuti dengan "s" untuk menunjukkan detik:

```
if (i > 0){
    mSeekBarProgress.setText(String.valueOf(i) + " s");
}
```

5. Jika tidak, setel `TextView` agar terlihat sebagai "Not Set":

```
else {
    mSeekBarProgress.setText("Not Set");
}
```

6. Batas waktu pengganti seharusnya hanya disetel jika nilai integer `SeekBar` lebih besar dari 0. Dalam metode `scheduleJob()`, buat integer untuk menyimpan kemajuan `SeekBar` dan variabel boolean yang true jika `SeekBar` memiliki nilai integer yang lebih besar dari 0:

```
int seekBarInteger = mSeekBar.getProgress();
boolean seekBarSet = seekBarInteger > 0;
```

7. Jika boolean ini true, panggil `setOverrideDeadline()` pada `JobInfo.Builder`, yang meneruskan nilai integer dari `SeekBar` dikalikan 1000 (parameter dalam milidetik, Anda menginginkan agar pengguna menyetel batas waktu dalam detik):

```
if (seekBarSet) {
    builder.setOverrideDeadline(seekBarInteger * 1000);
}
```

8. Modifikasi boolean `constraintSet` untuk menyertakan nilai `seekBarSet` sebagai kemungkinan batasan:

```
boolean constraintSet = selectedNetworkOption != JobInfo.NETWORK_TYPE_NONE
    || mDeviceChargingSwitch.isChecked() || mDeviceIdleSwitch.isChecked()
    || seekBarSet;
```

9. Jalankan aplikasi. Pengguna sekarang bisa menyetel batas waktu tegas dalam detik saat `JobService` harus dijalankan!

2.5 Mengimplementasikan batasan Periodik

`JobScheduler` juga mengizinkan Anda menjadwalkan tugas berulang, seperti `AlarmManager`. Opsi ini memiliki beberapa kekurangan:

- Tugas tidak dijamin berjalan dalam periode yang ditentukan (kondisi lain mungkin tidak terpenuhi, atau sumber daya sistem tidak cukup).
- Menggunakan batasan ini mencegah Anda juga menyetel batas waktu pengganti atau latensi () minimum, karena opsi ini tidak masuk akal untuk tugas yang berulang. Lihat dokumentasi `JobInfo.Builder` untuk informasi selengkapnya.

Tambahkan Periodic Switch ke layout

Anda akan menambahkan `Switch` untuk mengizinkan pengguna untuk berpindah-pindah antara menjalankan tugas sekali atau secara berulang-ulang pada interval periodik.

1. Dalam `activity_main.xml`, tambahkan tampilan `Switch` di antara dua `LinearLayout` horizontal. Gunakan atribut-atribut berikut:

| Atribut | Nilai |
|------------------------------------|-----------------------|
| <code>android:layout_width</code> | "wrap_content" |
| <code>android:layout_height</code> | "wrap_content" |
| <code>android:text</code> | "Periodic" |
| <code>android:id</code> | "@+id/periodicSwitch" |
| <code>android:layout_margin</code> | "4dp" |

2. Buat variabel anggota untuk `switch` dan inisialisasikan di `onCreate()`:

```
mPeriodicSwitch = (Switch) findViewById(R.id.periodicSwitch);
```

Buat kode untuk menggunakan Periodic Switch

Deadline pengganti dan batasan periodik biasanya saling eksklusif. Anda akan menggunakan `switch` tersebut untuk berpindah antara fungsionalitas dan label `SeekBar` untuk mewakili batas waktu pengganti atau interval periodik.

1. Panggil `setOnCheckedChangeListener()` pada `switch` periodik, yang meneruskan `OnCheckedChangeListener` baru.
2. Jika dicentang, setel label ke "Periodic Interval: ", jika tidak, ke "Override Deadline":

```
mPeriodicSwitch.setOnCheckedChangeListener(
    new CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton compoundButton, boolean isChecked) {
            if (isChecked){
                label.setText(R.string.periodic_interval);
            } else {
                label.setText(R.string.override_deadline);
            }
        }
    });
});
```

Yang tersisa sekarang adalah mengimplementasikan logika dalam metode `scheduleJob()` untuk menyetel batasan pada objek `JobInfo` dengan benar.

Jika opsi periodiknya adalah **on**:

- Jika `SeekBar` memiliki nilai **non-zero**, setel batasan dengan memanggil `setPeriodic()` pada objek `JobInfo.Builder`.
- Jika `SeekBar` memiliki **nilai 0**, tampilkan pesan `Toast` yang meminta pengguna untuk menyetel interval periodik dengan `SeekBar`.

Jika opsi periodiknya adalah **off**:

- Jika `SeekBar` memiliki **nilai selain nol**, pengguna telah menyetel batas waktu pengganti. Aplikasikan batas waktu pengganti menggunakan opsi `setOverrideDeadline()`.
- Jika `SeekBar` memiliki **nilai 0**, pengguna belum menentukan batas waktu pengganti atau tugas periodik, jadi jangan tambahkan apa pun ke objek `JobInfo.Builder`.
- Ganti kode yang menyetel batas waktu pengganti ke `JobInfo.Builder` dalam `scheduleJob()` dengan kode berikut untuk mengimplementasikan logika ini:

```

if (mPeriodicSwitch.isChecked()){
    if (seekBarSet){
        builder.setPeriodic(seekBarInteger * 1000);
    } else {
        Toast.makeText(MainActivity.this,
                "Please set a periodic interval", Toast.LENGTH_SHORT).show();
    }
} else {
    if (seekBarSet){
        builder.setOverrideDeadline(seekBarInteger * 1000);
    }
}

```

Kode solusi

Projek Android Studio: [NotificationScheduler](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Tantangan: Sampai saat ini, tugas Anda dijadwalkan oleh JobService yang difokuskan pada pengiriman notifikasi. Namun, pada sebagian besar waktu, JobScheduler digunakan untuk tugas latar belakang yang lebih sulit, seperti memperbarui cuaca atau menyinkronkan dengan database. Karena tugas latar belakang sifatnya bisa lebih kompleks, baik dari sudut pandang program dan fungsionalitas, pekerjaan memberi tahu framework saat tugas selesai adalah tugas developer. Untungnya, developer bisa melakukannya dengan memanggil `jobFinished()`.

1. Implementasikan JobService yang memulai AsyncTask saat batasan yang diberikan terpenuhi. AsyncTask seharusnya tertidur selama 5 detik. Ini akan mewajibkan Anda untuk memanggil `jobFinished()` saat tugas selesai. Jika batasan tidak lagi terpenuhi saat thread tertidur, tampilkan pesan Toast yang mengatakan bahwa pekerjaan gagal dan jadwalkan ulang pekerjaan.

Rangkuman

- JobScheduler menyediakan framework fleksibel untuk dengan cerdas menyelesaikan layanan latar belakang.
- JobScheduler hanya tersedia pada perangkat yang menjalankan API 21+
- Untuk menggunakan JobScheduler, Anda memerlukan dua bagian: JobService dan JobInfo.
- JobInfo adalah serangkaian kondisi yang akan memicu pekerjaan berjalan.
- JobService mengimplementasikan pekerjaan agar berjalan di bawah kondisi yang ditentukan oleh JobInfo.
- Anda hanya perlu mengimplementasikan metode callback `onStartJob()` dan `onStopJob()` dalam JobService.
- Implementasi pekerjaan terjadi (atau dimulai) di `onStartJob()`.
- `onStartJob()` mengembalikan boolean yang menandakan apakah layanan perlu memproses pekerjaan dalam thread yang berbeda.
- Jika `onStartJob()` mengembalikan true, Anda harus secara eksplisit memanggil `jobFinished()`. Jika `onStartJob()` mengembalikan salah, waktu proses akan memanggil `jobFinished()` mewakili Anda.
- JobService diproses pada thread utama, jadi hindari penghitungan panjang untuk I/O.
- JobScheduler adalah kelas pengelola yang bertanggung jawab untuk menjadwalkan tugas batch task. JobScheduler bersama-sama untuk memaksimalkan efisiensi sumber daya sistem, yang berarti Anda tidak memiliki kontrol pasti tentang kapan tugas dieksekusi.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Mentransfer Data dengan Efisien](#)

Ketahui selengkapnya

Dokumentasi Developer Android

Referensi

- [JobScheduler](#)
- [JobInfo](#)
- [JobInfo.Builder](#)
- [JobService](#)
- [JobParameters](#)

9.1: Preferensi Bersama

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1. Menjelajahi HelloSharedPrefs](#)
- [Tugas 2. Menyimpan dan memulihkan data ke preferensi bersama](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Preferensi bersama memungkinkan Anda membaca dan menulis data primitif dalam jumlah kecil sebagai pasangan kunci/nilai ke file pada storage perangkat. Kelas SharedPreference menyediakan API untuk mendapatkan handle ke file preferensi dan untuk membaca, menulis, dan mengelola data ini. File preferensi bersama ini sendiri dikelola oleh framework Android, dan bisa diakses oleh (dibagikan dengan) semua komponen aplikasi Anda. Akan tetapi, data tidak dibagikan dengan atau bisa diakses other aplikasi lain.

Data yang Anda simpan ke preferensi bersama berbeda dengan data di status aktivitas yang disimpan yang telah Anda pelajari di bab sebelumnya. Data di keadaan instance aktivitas dipertahankan di instance aktivitas di sesi pengguna yang sama. Preferensi bersama tetap ada di semua sesi pengguna, meskipun aplikasi Anda ditutup dan dimulai ulang atau perangkat dimulai ulang.

Gunakan preferensi bersama hanya jika Anda perlu menyimpan data dalam jumlah kecil sebagai pasangan kunci/nilai. Untuk mengelola sejumlah besar data aplikasi yang persisten, gunakan metode lain seperti database SQL yang akan Anda pelajari di bab berikutnya.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah memahami:

- Membuat, membangun, dan menjalankan aplikasi di Android Studio.
- Merancang layout dengan tombol dan tampilan teks.
- Menggunakan gaya dan tema.
- Menyimpan dan memulihkan keadaan instance aktivitas.

Yang akan Anda PELAJARI

Anda akan belajar:

- Mengenali apa yang dimaksud dengan preferensi bersama.
- Membuat file preferensi bersama untuk aplikasi Anda.
- Menyimpan data di preferensi bersama dan membaca preferensi tersebut kembali.
- Menghapus data di preferensi bersama.

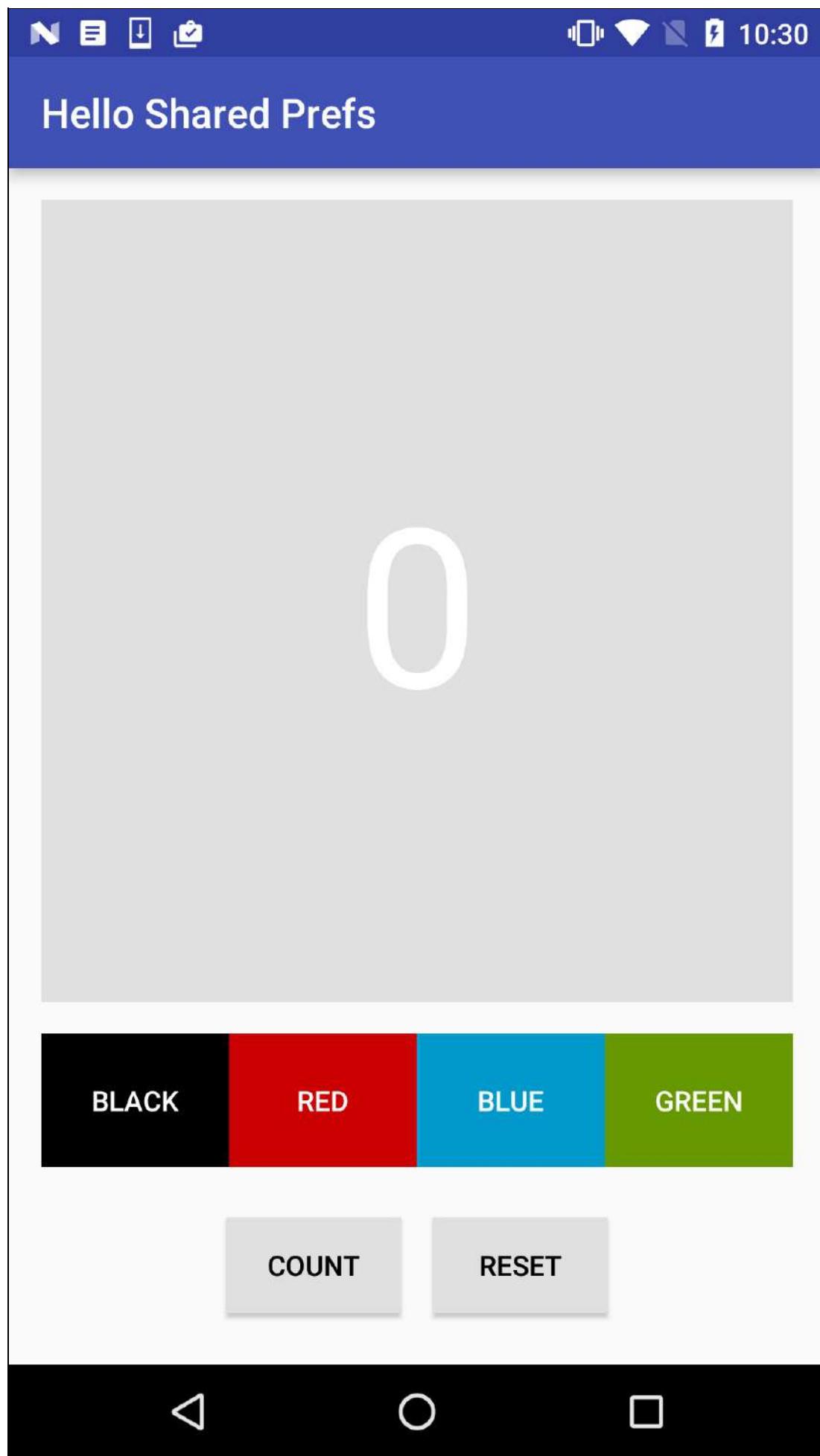
Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Menambahkan kemampuan untuk menyimpan, mengambil, dan menyetel ulang preferensi bersama ke aplikasi.

Ringkasan Aplikasi

Aplikasi HelloSharedPrefs adalah variasi lain dari aplikasi HelloToast yang telah Anda buat di Pelajaran 1. Aplikasi ini menyeret tombol untuk menambah angka, mengubah warna latar belakang, dan menyetel ulang angka dan warna ke defaultnya. Aplikasi juga menggunakan tema dan gaya untuk mendefinisikan tombol.



Anda akan memulai dengan aplikasi awal di praktik ini dan menambahkan preferensi bersama ke kode aktivitas utama. Anda juga akan menambahkan tombol setel ulang yang akan menyetel hitungan dan warna latar belakang ke default dan menghapus file preferensi.

Tugas 1. Menjelajahi HelloSharedPrefs

Proyek aplikasi awal yang lengkap untuk praktik ini tersedia di [HelloSharedPrefs-Start](#). Dalam tugas ini Anda akan memuat proyek ke Android Studio dan menjelajahi beberapa fitur utama aplikasinya.

1.1 Buka dan Jalankan Proyek HelloSharedPrefs

1. Unduh aplikasi [HelloSharedPrefs-Start](#) dan buka zip file.
2. Buka aplikasi di Android Studio.
3. Bangun dan jalankan proyek di Android Studio. Coba hal-hal berikut ini:
 - Klik tombol Count untuk menambah angka di tampilan teks utama.
 - Klik tombol warna apa pun untuk mengubah warna latar belakang tampilan teks utama.
 - Putar perangkat dan perhatikan bahwa warna latar belakang dan hitungan dipertahankan.
 - Klik tombol Reset untuk menyetel warna dan hitungan kembali ke defaultnya.
4. Paksa aplikasi keluar menggunakan salah satu metode berikut:
 - Di Android Studio, pilih Run > Stop 'app' atau klik ikon Berhenti  di toolbar.
 - Di perangkat, klik tombol Recents (tombol kota di sudut kanan bawah). Gesek kartu agar aplikasi HelloSharedPrefs keluar atau klik X di sudut kanan. Jika Anda keluar dari aplikasi dengan cara ini, tunggu beberapa detik sebelum Anda memulainya lagi sehingga sistem bisa melakukan pembersihan.
5. Jalankan kembali aplikasi.

Aplikasi dimulai ulang dengan penampilan default -- hitungan kembali ke 0 dan warna latar belakang abu-abu.

1.2 Jelajahi kode Aktivitas

1. Buka MainActivity (java/com.example.android.simplecalc/MainActivity).
2. Periksa kodennya dan catat yang berikut ini:
 - Hitungan (mCount) didefinisikan oleh sebuah integer. Metode handler klik countUp() menambah nilai ini dan memperbarui TextView utama.
 - Warna (mColor) juga merupakan integer yang awalnya didefinisikan sebagai abu-abu di file sumber daya colors.xml sebagai default_background.
 - Metode handler klik changeBackground() mendapatkan warna latar belakang tombol yang diklik, lalu menyetel warna latar belakang TextView utama.
 - Kedua integer hitungan dan warna disimpan ke bundel keadaan instance di onSaveInstanceState(), dan dipulihkan di onCreate(). Kunci bundel untuk hitungan dan warna didefinisikan oleh variabel pribadi (COUNT_KEY) dan (COLOR_KEY).

Tugas 2. Menyimpan dan memulihkan data ke file preferensi bersama

Dalam tugas ini, Anda akan menyimpan keadaan aplikasi ke file preferensi bersama dan membaca data kembali saat aplikasi dimulai ulang. Karena keadaan data yang Anda simpan di preferensi bersama (hitungan dan warna saat ini) adalah data yang **sama** yang Anda pertahankan di keadaan instance, Anda tidak harus melukannya dua kali -- Anda bisa mengganti keadaan instance sekaligus dengan keadaan preferensi bersama.

2.1 Inisialisasi preferensi

1. Tambahkan variabel anggota ke kelas MainActivity untuk menampung nama file preferensi bersama dan sebuah referensi ke objek SharedPreferences.

```
private SharedPreferences mPreferences;
private String sharedPrefFile = "com.example.android.hellosharedprefs";
```

Anda bisa menamai file preferensi bersama dengan nama apa pun yang Anda ingin, tetapi umumnya namanya sama dengan nama paket aplikasi Anda.

2. Di metode onCreate(), inisialisasi preferensi bersama. Pastikan Anda menyisipkan kode ini sebelum pernyataan `if`:

```
mPreferences = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);
```

Metode getSharedPreferences() membuka file di nama file yang diberikan (sharedPrefFile) dengan mode MODE_PRIVATE.

Catatan: Versi lama Android memiliki mode lain yang memungkinkan Anda membuat file preferensi bersama yang bisa dibaca dan ditulis oleh umum. Mode ini tidak digunakan lagi di API 17, dan sekarang **sangat tidak dianjurkan** karena alasan keamanan. Jika Anda perlu berbagi data dengan aplikasi lain, gunakan layanan atau penyedia konten.

Kode Solusi (Aktivitas Utama - sebagian)

```
public class MainActivity extends AppCompatActivity {
    private int mCount = 0;
    private TextView mShowCount;
    private int mCurrentColor;

    private SharedPreferences mPreferences;
    private String sharedPrefFile = "com.example.android.hellosharedprefs";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mShowCount = (TextView) findViewById(R.id.textview);
        mCurrentColor = ContextCompat.getColor(this, R.color.default_background);
        mPreferences = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);

        // ...
    }
}
```

2.2 Simpan preferensi di onPause()

Menyimpan preferensi sangat mirip dengan menyimpan keadaan instance -- kedua operasi ini menyisihkan data ke objek Bundel sebagai pasangan kunci/nilai. Untuk preferensi bersama, Anda menyimpan data di callback daur hidup onPause() dan Anda perlu objek editor bersama ([SharedPreferences.Editor](#)) untuk menulis ke objek preferensi bersama.

1. Klik baris terakhir kelas MainActivity, tepat sebelum tanda kurung penutup.
2. Pilih Code > Generate, lalu pilih Override Methods.
3. Ketikkan "onPause", pilih tanda tangan metode untuk metode onPause() dan klik OK.

Metode onPause() kerangka ditambahkan ke titik penyisipan.

4. Dapatkan sebuah editor untuk objek SharedPreferences:

```
SharedPreferences.Editor preferencesEditor = mPreferences.edit();
```

Editor preferensi bersama diperlukan untuk menulis ke objek preferensi bersama. Tambahkan baris ini ke onPause() setelah panggilan ke super.onPause().

5. Gunakan metode putInt() untuk menempatkan integer mCount dan mCurrentColor ke dalam preferensi bersama dengan kunci yang sesuai:

```
preferencesEditor.putInt(COUNT_KEY, mCount);
preferencesEditor.putInt(COLOR_KEY, mCurrentColor);
```

Kelas SharedPreferences.Editor menyertakan beberapa metode put untuk berbagai tipe data, termasuk putInt() dan putString().

6. Panggil apply() untuk menyimpan preferensi:

```
preferencesEditor.apply();
```

Metode apply() menyimpan preferensi secara asinkron, di luar thread UI. Editor preferensi bersama juga memiliki metode commit() untuk menyimpan preferensi secara sinkron. Metode commit() tidak disarankan karena bisa memblokir operasi lain.

7. Hapus seluruh metode onSaveInstanceState(). Karena keadaan instance aktivitas berisi data yang sama dengan preferensi bersama, Anda bisa mengganti keadaan instance sekaligus.

Kode Solusi (MainActivity - metode onPause())

```
@Override
protected void onPause(){
    super.onPause();

    SharedPreferences.Editor preferencesEditor = mPreferences.edit();
    preferencesEditor.putInt(COUNT_KEY, mCount);
    preferencesEditor.putInt(COLOR_KEY, mColor);
    preferencesEditor.apply();
}
```

2.3 Pulihkan preferensi di onCreate()

Seperti pada keadaan instance, aplikasi Anda membaca preferensi bersama apa pun yang disimpan di metode onCreate(). Sekali lagi, karena preferensi bersama berisi data yang sama dengan keadaan instance, kita juga bisa mengganti keadaan dengan preferensi di sini. Setiap kali onCreate() dipanggil -- saat aplikasi dimulai, saat konfigurasi berubah -- preferensi bersama digunakan untuk memulihkan keadaan tampilan.

1. Cari bagian dari metode onCreate() yang menguji apakah argumen savedInstanceState adalah null dan mengembalikan keadaan instance:

```
if (savedInstanceState != null) {
    mCount = savedInstanceState.getInt(COUNT_KEY);
    if (mCount != 0) {
        mShowCountTextView.setText(String.format("%s", mCount));
    }
    mColor = savedInstanceState.getInt(COLOR_KEY);
    mShowCountTextView.setBackgroundColor(mColor);
}
```

2. Hapus seluruh blok tersebut.
3. Di metode onCreate(), di tempat yang sama dengan kode simpan keadaan instance, dapatkan hitungan dari preferensi dengan kunci COUNT_KEY dan tetapkan ke variabel mCount.

```
mCount = mPreferences.getInt(COUNT_KEY, 0);
```

Ketika Anda membaca data dari preferensi, Anda tidak perlu mendapatkan editor preferensi bersama. Gunakan metode get apa pun pada objek preferensi bersama untuk mengambil data preferensi.

Perhatikan bahwa metode getInt() mengambil dua argumen: satu untuk kunci dan yang lain untuk nilai default jika kunci tidak bisa ditemukan. Dalam hal ini, nilai default adalah 0, yang sama dengan nilai awal mCount.

4. Perbarui nilai tampilan teks utama dengan hitungan baru.

```
mShowCountTextView.setText(String.format("%s", mCount));
```

5. Dapatkan warna dari preferensi dengan kunci COLOR_KEY dan tetapkan ke variabel mColor.

```
mColor = mPreferences.getInt(COLOR_KEY, mColor);
```

Seperti sebelumnya, argumen kedua ke getInt() adalah nilai default yang akan dipakai dalam hal kunci tidak ada di preferensi bersama. Dalam hal ini, Anda bisa cukup menggunakan nilai mColor, yang baru diinisialisasi ke latar belakang default nanti di metode.

6. Perbarui warna latar belakang tampilan teks utama.

```
mShowCountTextView.setBackgroundColor(mColor);
```

7. Jalankan aplikasi. Klik tombol Count dan ubah warna latar belakang untuk memperbarui keadaan instance dan preferensi.
8. Putar perangkat atau emulator untuk memverifikasi bahwa hitungan dan warna disimpan di perubahan konfigurasi.
9. Paksa aplikasi keluar menggunakan salah satu metode berikut:
 - o Di Android Studio, pilih Run > Stop 'app'.
 - o Di perangkat, klik tombol Recents (tombol kota di sudut kanan bawah). Gesek kartu agar aplikasi HelloSharedPrefs keluar atau klik X di sudut kanan.
10. Jalankan kembali aplikasi. Aplikasi dimulai ulang dan memuat preferensi, dengan mempertahankan keadaan.

Kode Sousi (Main Activity - onCreate())

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Initialize views, color, preferences
    mShowCountTextView = (TextView) findViewById(R.id.count_textview);
    mColor = ContextCompat.getColor(this, R.color.default_background);
    mPreferences = getSharedPreferences(mSharedPrefFile, MODE_PRIVATE);

    // Restore preferences
    mCount = mPreferences.getInt(COUNT_KEY, 0);
    mShowCountTextView.setText(String.format("%s", mCount));
    mColor = mPreferences.getInt(COLOR_KEY, mColor);
    mShowCountTextView.setBackgroundColor(mColor);
}
```

2.4 Setel ulang preferensi di handler klik reset()

Tombol Reset di aplikasi awal menyetel ulang hitungan dan warna untuk aktivitas ke nilai defaultnya. Karena preferensi menampung keadaan aktivitas, penting juga untuk menghapus preferensi di waktu yang sama.

1. Di metode handler klik reset(), setelah warna dan hitungan disetel ulang, dapat editor untuk objek SharedPreferences:

```
SharedPreferences.Editor preferencesEditor = mPreferences.edit();
```

2. Hapus semua preferensi bersama:

```
preferencesEditor.clear();
```

3. Terapkan perubahan:

```
preferencesEditor.apply();
```

Kode Solusi (metode reset()):

```
public void reset(View view) {
    // Reset count
    mCount = 0;
    mShowCountTextView.setText(String.format("%s", mCount));

    // Reset color
    mColor = ContextCompat.getColor(this, R.color.default_background);
    mShowCountTextView.setBackgroundColor(mColor);

    // Clear preferences
    SharedPreferences.Editor preferencesEditor = mPreferences.edit();
    preferencesEditor.clear();
    preferencesEditor.apply();

}
```

Kode solusi

Proyek Android Studio: [HelloSharedPrefs](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode bersifat opsional dan bukan merupakan prasyarat untuk materi di bab berikutnya.

Tantangan: Ubah aplikasi HelloSharedPrefs sehingga, sebagai ganti otomatis menyimpan keadaan ke file preferensi, menambahkan aktivitas kedua untuk mengubah, menyetel ulang, dan menyimpan preferensi tersebut. Tambahkan sebuah tombol ke aplikasi yang bernama Settings untuk meluncurkan aktivitas tersebut. Sertakan tombol toggle dan spinner untuk memodifikasi preferensi dan tombol Save dan Reset untuk menyimpan dan membersihkan preferensi.

Rangkuman

- Kelas [SharedPreferences](#) memungkinkan aplikasi untuk menyimpan sejumlah kecil data primitif sebagai [pasangan kunci-nilai](#).
- Preferensi bersama bersifat persisten di semua sesi pengguna yang berbeda di aplikasi yang sama.
- Untuk menulis ke preferensi bersama, dapatkan objek [SharedPreferences.Editor](#).
- Gunakan berbagai metode `put*` di objek [SharedPreferences.Editor](#), seperti `putInt()` atau `putString()`, untuk menempatkan data ke dalam preferensi bersama dengan kunci dan nilai.
- Gunakan berbagai metode `get*` di objek [SharedPreferences](#), seperti `getInt()` atau `getString()`, untuk mendapatkan data dari preferensi bersama dengan sebuah kunci.
- Gunakan metode `clear()` di objek [SharedPreferences.Editor](#) untuk membuang semua data yang disimpan di preferensi.
- Gunakan metode `apply()` di objek [SharedPreferences.Editor](#) untuk menyimpan perubahan ke file preferensi.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Preferensi Bersama](#)

Ketahui selengkapnya

- [Menyimpan Data](#) (Panduan Android)
- [Opsi Storage](#) (Panduan Android)
- [Menyimpan Rangkaian Nilai-Kunci](#) (Pelatihan Android)
- [SharedPreferences](#) (Referensi API Android)
- [SharedPreferences.Editor](#) (Referensi API Android)
- [Cara menggunakan SharedPreferences di Android untuk menyimpan, mengambil, dan mengedit nilai](#) (Stack Overflow)
- [onSavedInstanceState vs. SharedPreferences](#)(Stack Overflow)

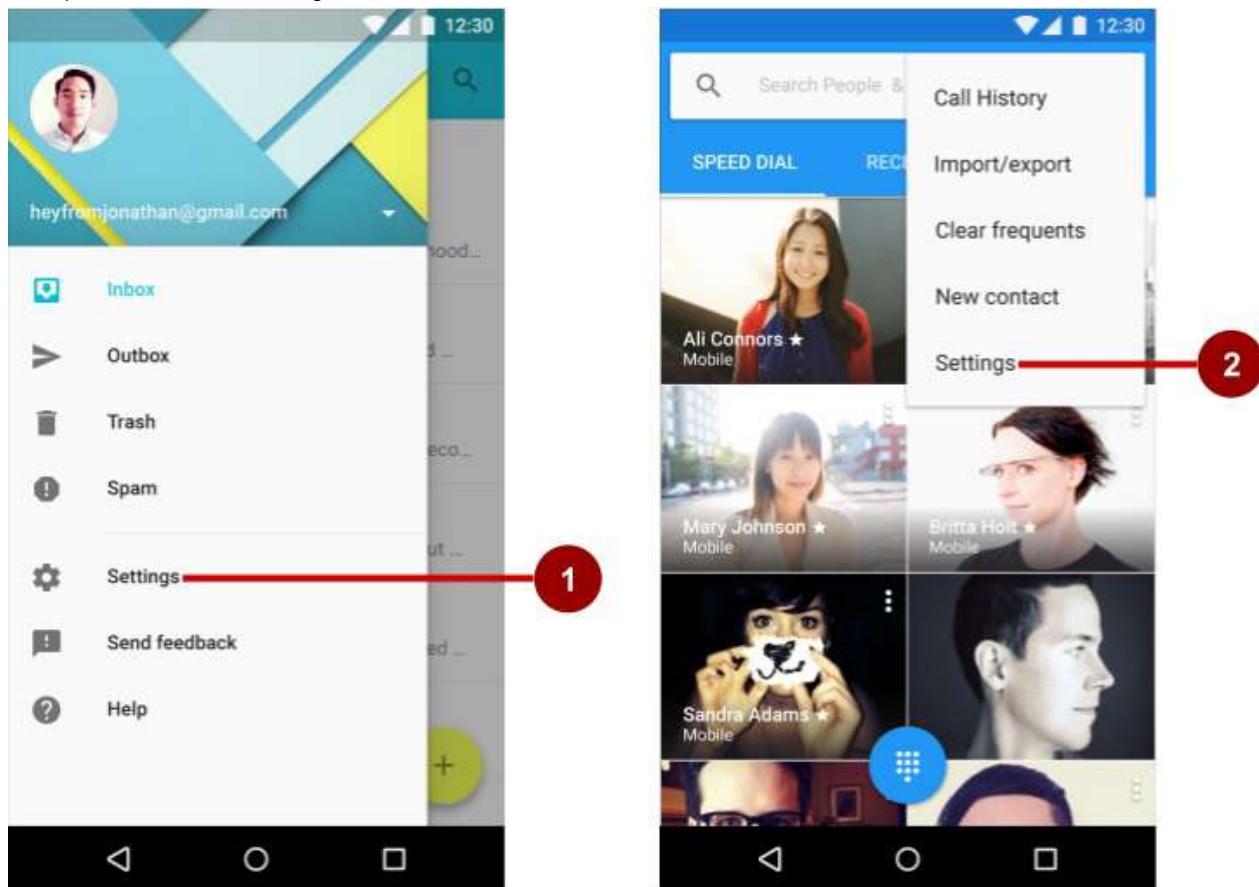
9.2: Menambahkan Setelan ke Aplikasi

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1: Menambahkan setelan pengalihan ke aplikasi
- Tugas 2: Menggunakan template Settings Activity
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Aplikasi sering kali menyertakan setelan yang memungkinkan pengguna memodifikasi fitur dan perilaku aplikasi. Misalnya, beberapa aplikasi memungkinkan pengguna menyetel lokasi rumahnya, unit default untuk pengukuran, opsi makan malam, dan setelan lain yang berlaku pada seluruh aplikasi. Setelan biasanya tidak sering diakses, karena sekali pengguna mengubah setelan, seperti lokasi rumah, pengguna jarang perlu kembali dan mengubahnya lagi.

Pengguna mengharapkan masuk ke pengaturan aplikasi dengan mengetuk **Settings** di navigasi samping, seperti panel samping navigasi seperti yang ditampilkan di sebelah kiri gambar di bawah, atau di menu opsi di bilah aplikasi, yang ditampilkan di sebelah kanan gambar di bawah.



Dalam gambar di atas:

1. **Settings** di navigasi samping (panel samping navigasi)
2. **Settings*** di menu opsi bilah aplikasi

Dalam praktik ini Anda akan menambahkan aktivitas setelan ke aplikasi. Pengguna akan bisa masuk ke setelan aplikasi dengan mengetuk **Settings**, yang akan ditempatkan di menu opsi di bilah aplikasi.

Yang harus sudah Anda KETAHUI

Dari praktik sebelumnya, Anda harus sudah bisa:

- Menambahkan aktivitas ke aplikasi.
- Merancang layout dengan tombol dan tampilan teks.
- Mengekstrak sumber daya string dan mengedit nilai string dan larik string.
- Membuat menu opsi di bilah aplikasi.
- Menambahkan dan mengedit item menu di menu opsi.
- Menambahkan handler kejadian untuk klik item menu.
- Mengedit file AndroidManifest.xml untuk menambahkan navigasi naik untuk aktivitas kedua.
- Membaca preferensi dari `sharedPreferences`.

Yang akan Anda PELAJARI

Anda akan belajar:

- Menambahkan aktivitas dan memahami penggunaan fragmen untuk mengelola setelan.
- Membuat file sumber daya XML setelan dengan atributnya.
- Membuat navigasi ke aktivitas setelan.
- Menyetel nilai default setelan.
- Membaca nilai setelan yang diubah oleh pengguna.
- Menyesuaikan template Settings Activity untuk Anda gunakan sendiri.

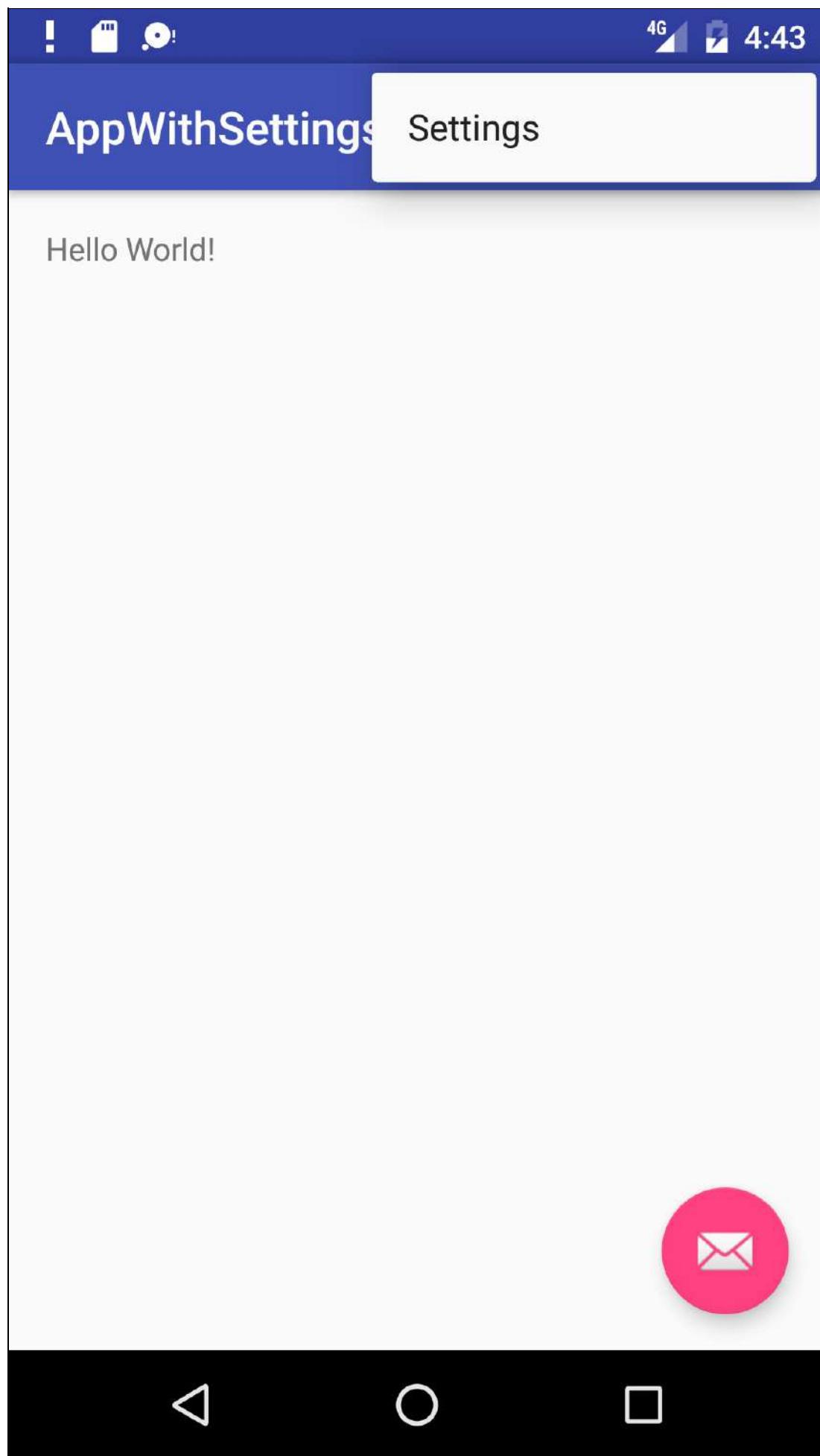
Yang akan Anda LAKUKAN

Dalam praktik ini Anda akan:

- Membuat aplikasi yang menyertakan *Settings** di menu opsi.
- Menambahkan "Opsi Setelan" sebagai tombol alih.
- Menambahkan kode untuk menyetel nilai default untuk setelan dan mengakses nilai setelan setelah diubah.
- Menggunakan dan menyesuaikan template Settings Activity Android Studio.

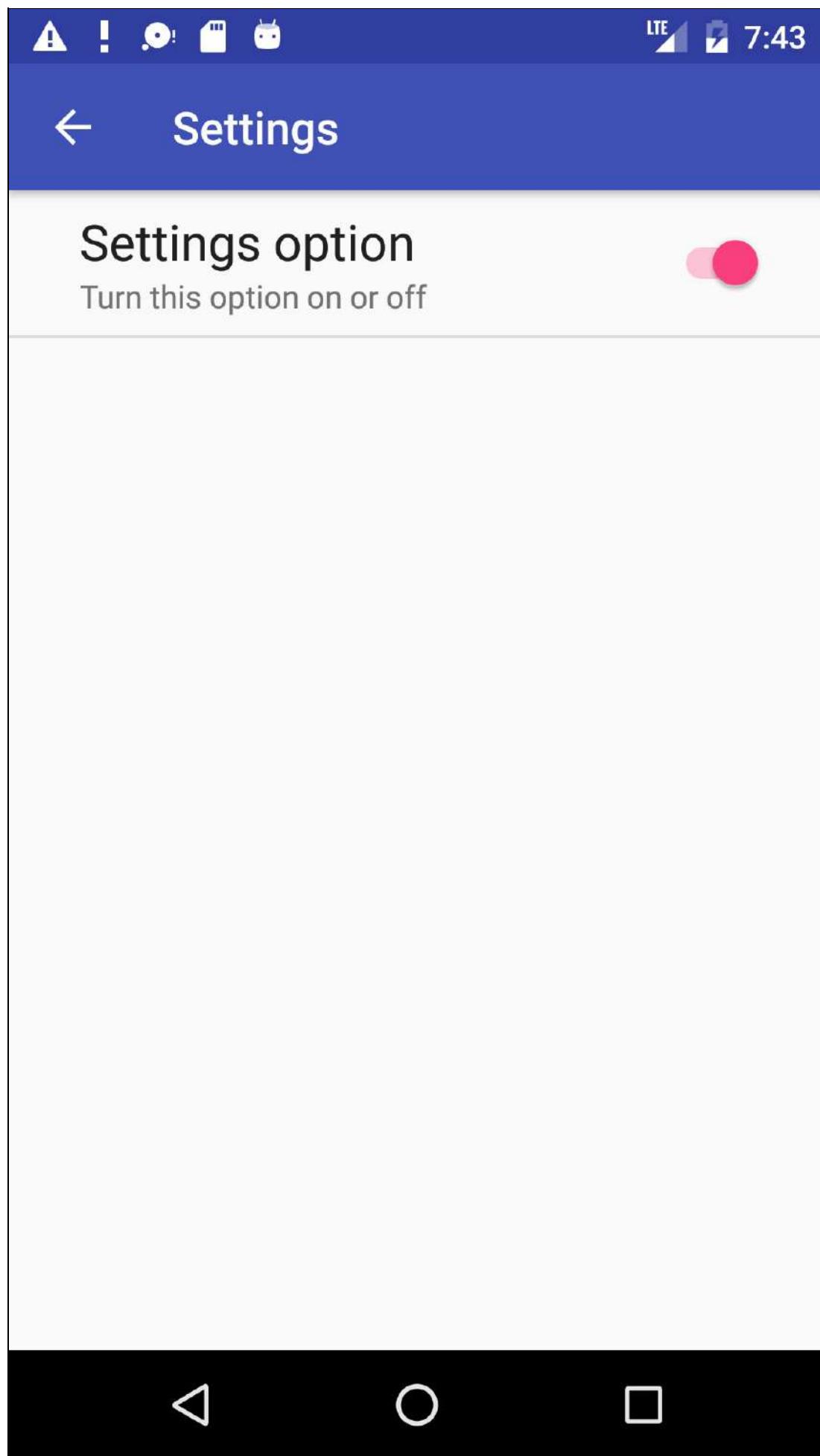
Ringkasan aplikasi

Android Studio menyediakan pintasan untuk menyiapkan menu opsi dengan **Settings**. Jika Anda memulai proyek Android Studio untuk ponsel cerdas atau tablet menggunakan template Basic Activity, aplikasi baru menyertakan **Settings** seperti yang ditampilkan di bawah ini:



Template juga menyertakan tombol aksi mengambang di sudut kanan bawah layar dengan ikon amplop. Anda bisa mengabaikan tombol ini untuk praktik ini, karena Anda tidak akan menggunakaninya.

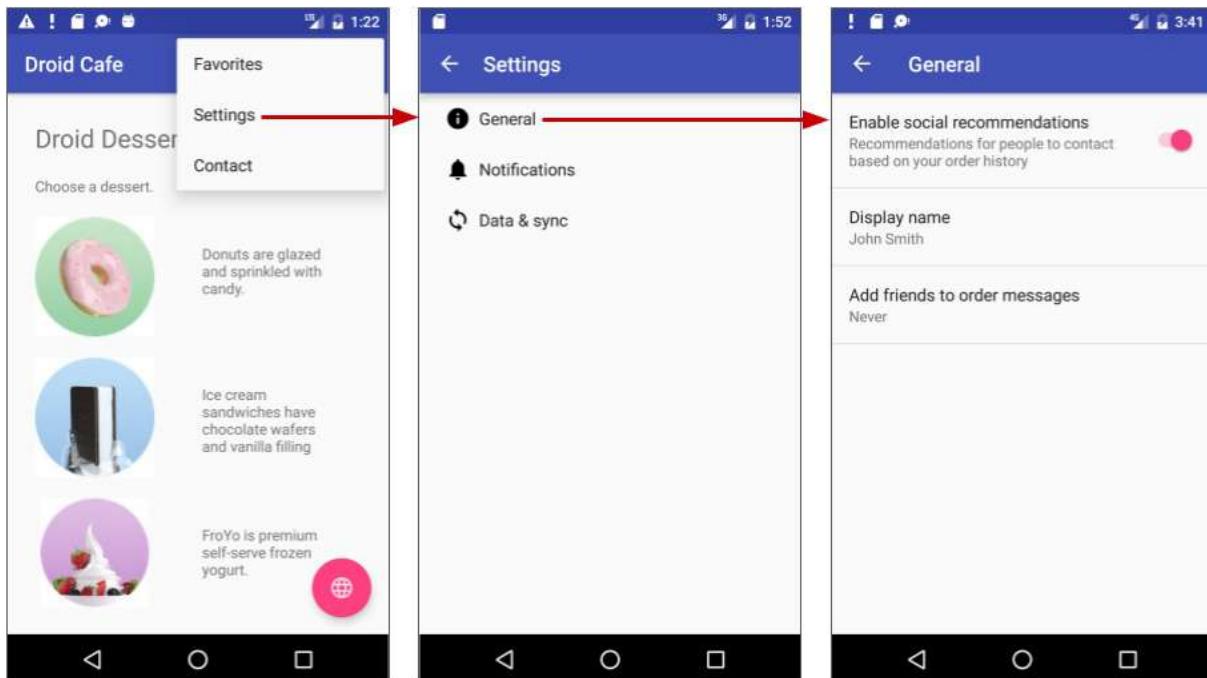
Anda akan memulai dengan membuat aplikasi yang bernama AppWithSettings menggunakan template Basic Activity, dan menambahkan aktivitas setelan yang menyediakan satu setelan tombol alih yang bisa diaktifkan atau dinonaktifkan pengguna:



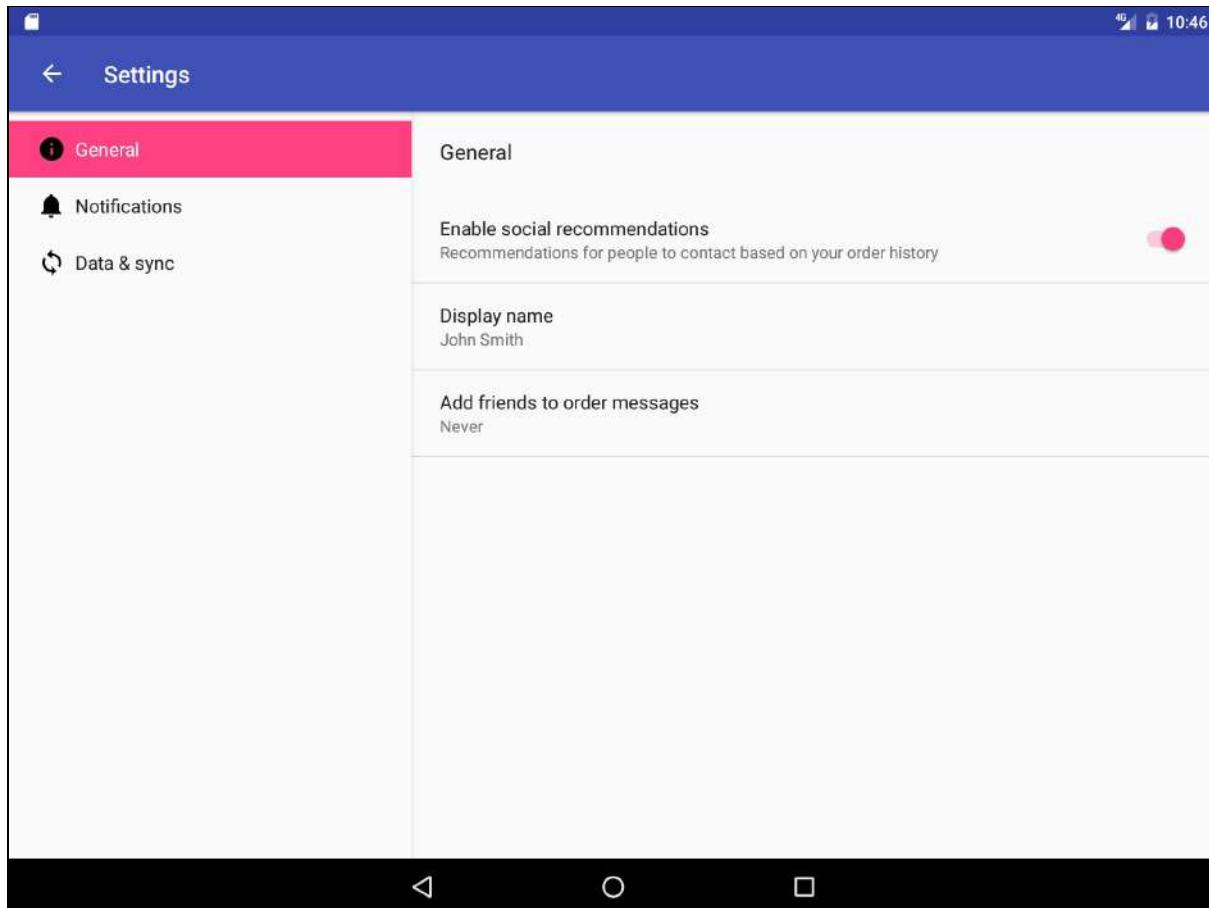
Anda akan menambahkan kode untuk membaca setelan dan melakukan tindakan berdasarkan nilainya. Agar sederhana, tindakan akan menampilkan pesan toast dengan nilai setelan.

Dalam tugas kedua, Anda akan menambahkan template Settings Activity standar yang disediakan oleh Android Studio ke aplikasi DroidCafe yang telah Anda buat di pelajaran sebelumnya. Template Settings Activity sudah diisi terlebih dulu dengan setelan yang bisa Anda sesuaikan untuk aplikasi, dan menyesuaikan layout yang berbeda untuk ponsel cerdas dan tablet:

- *Ponsel cerdas:* Layar Settings utama dengan tautan header untuk setiap grup setelan, seperti General untuk setelan umum, seperti yang ditampilkan di bawah ini.



- **Tablet:** Layout layar detail/master dengan tautan header untuk setiap grup di sebelah kiri (master), dan grup setelan di sebelah kanan (detail), seperti yang ditampilkan dalam gambar di bawah ini.



Yang perlu Anda lakukan untuk menyesuaikan template adalah mengubah header, judul setelan, deskripsi setelan, dan nilai untuk setelan dan menulis kode yang biasa Anda tulis untuk menggunakan nilai setelan.

Aplikasi Droid Cafe dibuat di pelajaran sebelumnya dari template Basic Activity, yang menyediakan menu opsi di bilah aplikasi untuk menempatkan opsi **Settings**. Anda akan menyesuaikan template Settings Activity yang disediakan dengan mengubah satu judul, keterangan, nilai, dan nilai default setelan. Anda akan menambahkan kode untuk membaca nilai setelan setelah pengguna mengubahnya dan menampilkan nilai tersebut.

Tugas 1: Menambahkan setelan pengalihan ke aplikasi

Dalam tugas ini, Anda akan:

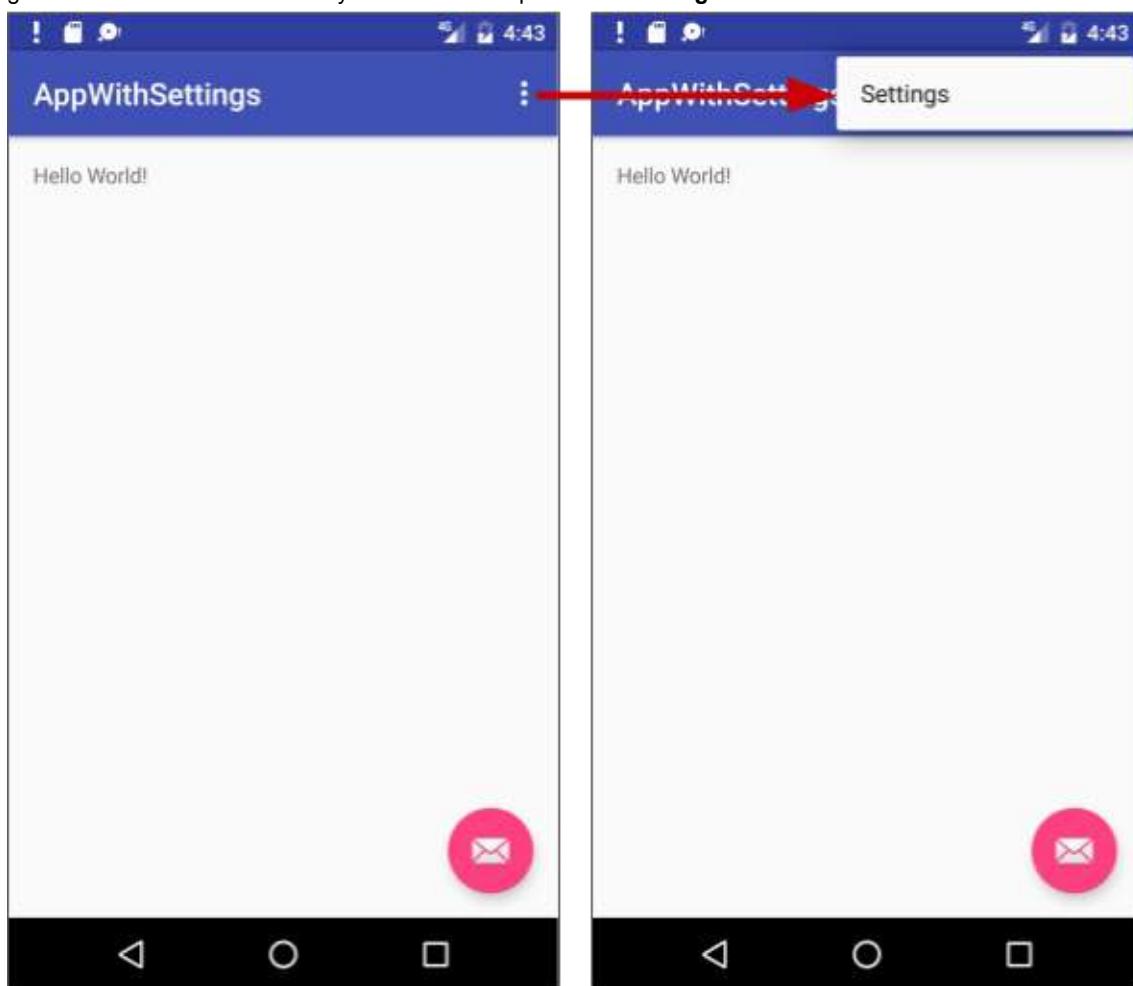
- Membuat proyek baru berdasarkan template Basic Activity (yang menyediakan menu opsi).
- Menambahkan setelan tombol alih dengan atribut di file XML preferensi.
- Menambahkan aktivitas untuk setelan dan fragmen untuk setelan khusus. akan menggunakan versi `PreferenceFragmentCompat` `PreferenceFragment` untuk mempertahankan kompatibilitas dengan `AppCompatActivity` .
Anda juga akan menambahkan pustaka `android.support:preference-v7`.
- Menghubungkan item **Settings** di menu opsi ke aktivitas setelan.

1.1. Membuat proyek dan menambahkan direktori xml dan file sumber daya

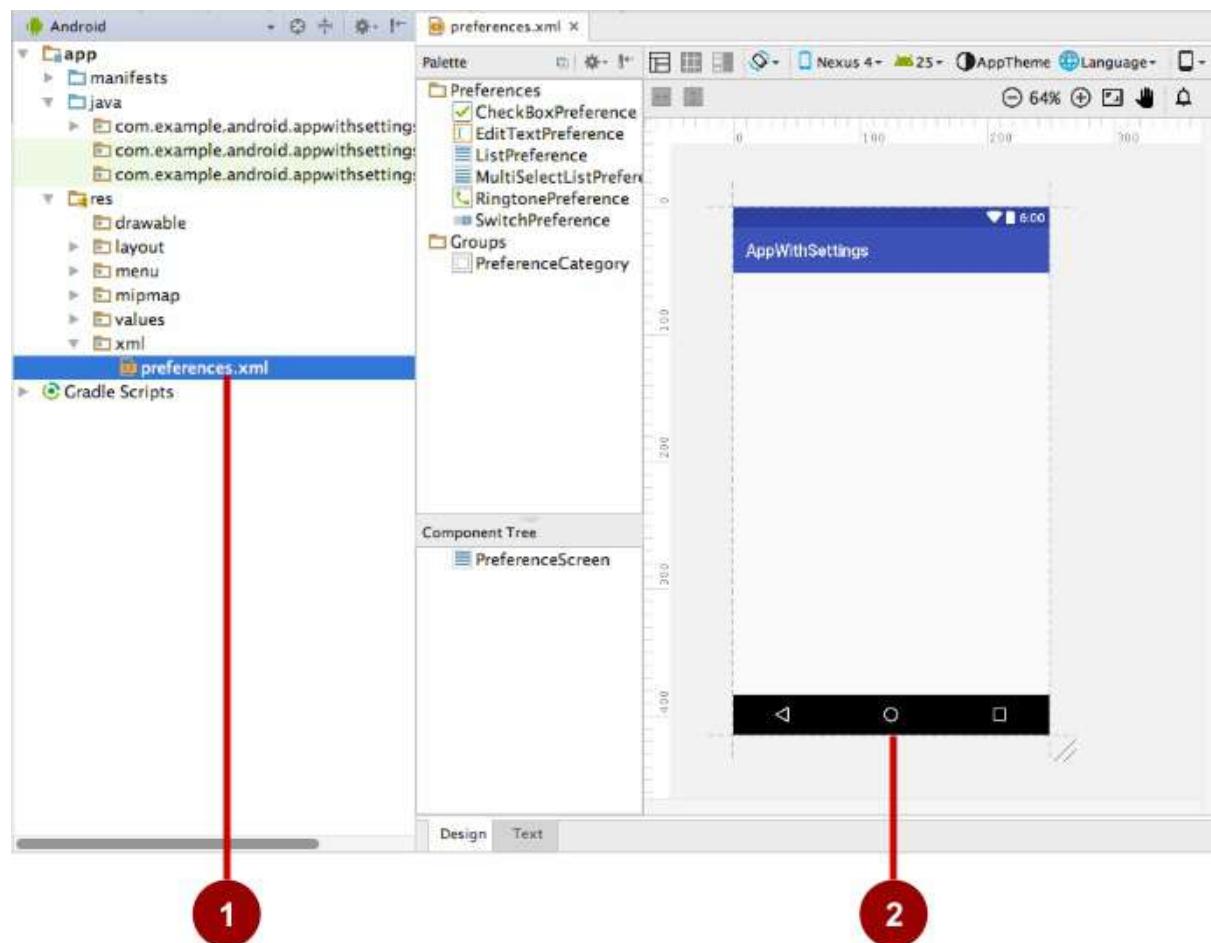
1. Di Android Studio, buatlah proyek baru dengan parameter berikut:

| Atribut | Nilai |
|------------------------------|------------------------------------------------|
| Application Name | AppWithSettings |
| Company Name | android.example.com (atau domain Anda sendiri) |
| Phone and Tablet Minimum SDK | API15: Android 4.0.3 IceCreamSandwich |
| Use a Fragment? | Jangan dicentang |
| Template | Basic Activity |

2. Jalankan aplikasi dan ketuk ikon luapan di bilah aplikasi untuk melihat menu opsi, seperti yang ditampilkan dalam gambar di bawah ini. Satu-satunya item di menu opsi adalah **Settings**.



3. Buat direktori sumber daya baru untuk menampung file XML yang berisi setelan:
- Pilih direktori **res** di tampilan Project: Android, dan pilih **File > New > Android Resource Directory**. Dialog New Resource Directory akan muncul.
 - Di menu tarik-turun tipe Resource, pilih **xml**. Nama Direktori otomatis berubah menjadi **xml**.
 - Klik **OK**.
4. Direktori **xml** muncul di tampilan Project: Android di dalam direktori **res**. Pilih direktori **xml** dan pilih **File > New > XML resource file** (atau klik kanan direktori **xml** dan pilih **New > XML resource file**).
5. Masukkan nama file XML, **preferences**, di bidang nama File dan klik **OK**. File preferences.xml muncul di dalam direktori **xml** dan editor layout muncul, seperti yang ditampilkan dalam gambar di bawah ini.

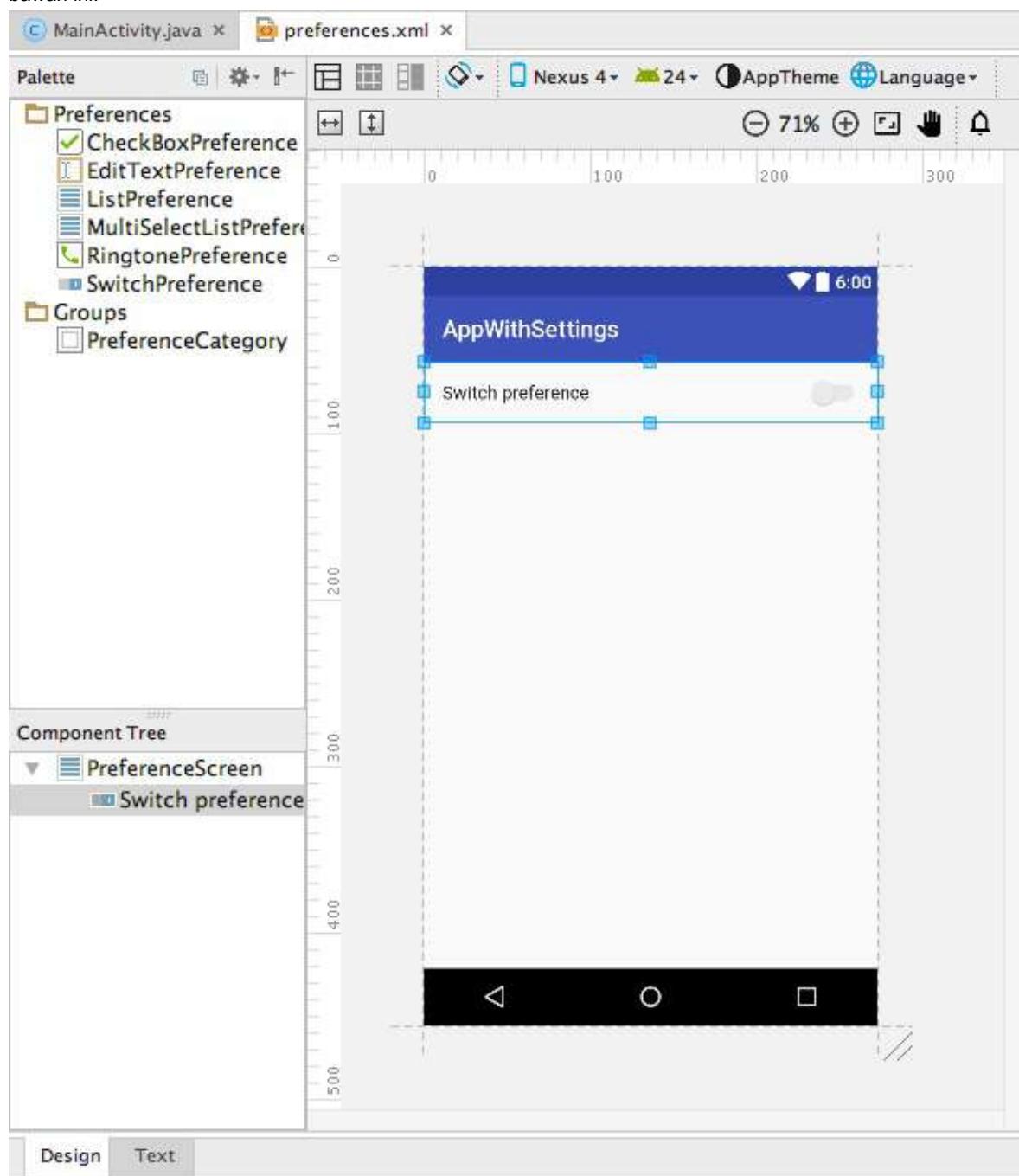


Dalam gambar di atas:

1. File preferences.xml di dalam direktori xml.
2. Editor layout menampilkan konten preferences.xml.

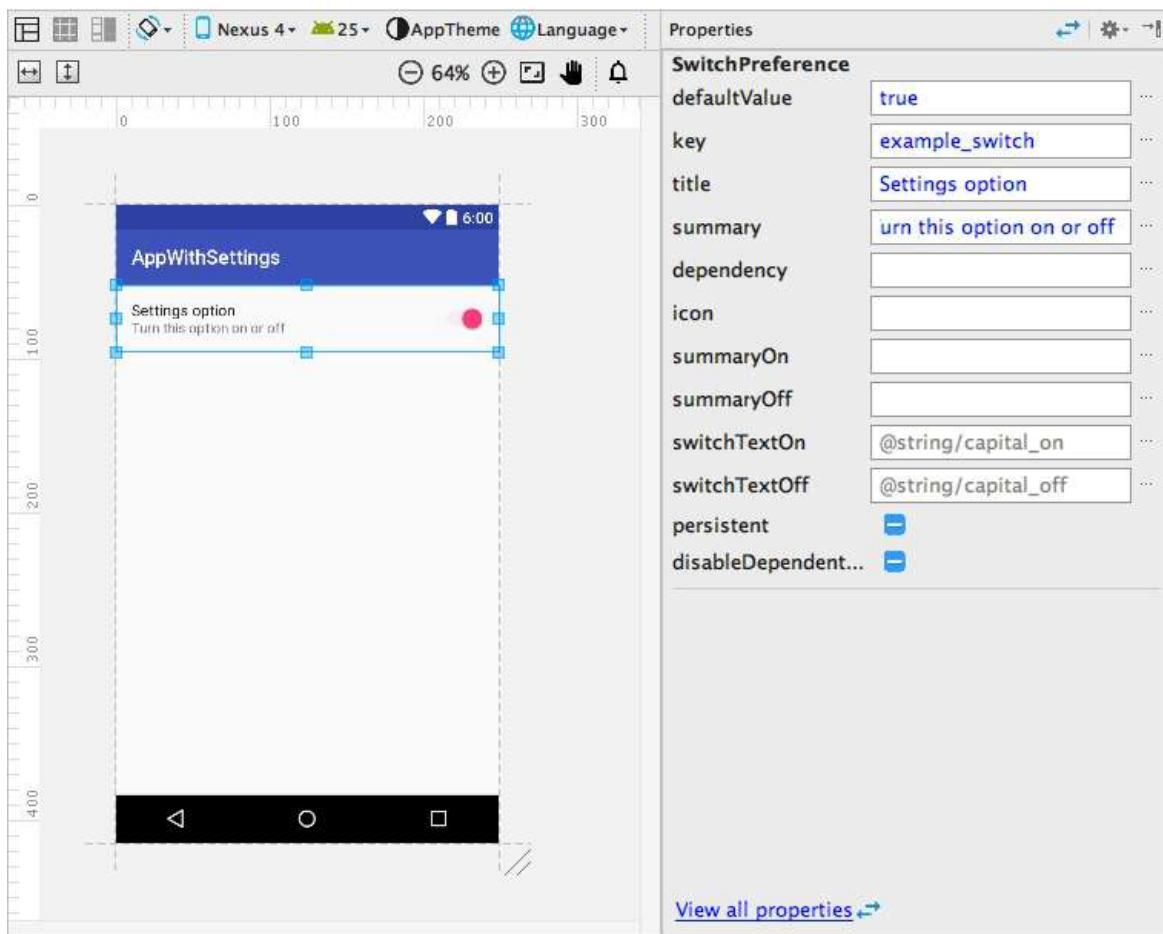
1.2 Tambahkan preferensi XML dan atribut untuk setelan.

1. Seret **SwitchPreference** dari panel Palette di sebelah kiri ke atas layout, seperti yang ditampilkan dalam gambar di bawah ini.



2. Ubah nilai di panel Properties di sebelah kanan editor layout seperti berikut (lihat gambar di bawah):

- i. **defaultValue: true**
- ii. **key:example_switch**
- iii. **title: Settings option**
- iv. **summary: Turn this option on or off**



3. Klik tab Text di bawah editor layout untuk mengedit kode XML:

```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <SwitchPreference
        android:defaultValue="true"
        android:title="Settings option"
        android:key="example_switch"
        android:summary="Turn this option on or off" />
</PreferenceScreen>
```

Nilai Properties yang Anda masukkan merepresentasikan atribut XML:

- `android:defaultValue` : Nilai default setelan saat aplikasi dimulai pertama kali.
- `android:title` : Judul setelan. Untuk `SwitchPreference`, judul muncul di sebelah kiri tombol alih.
- `android:key` : Kunci yang digunakan untuk menyimpan nilai setelan. Setiap setelan memiliki pasangan nilai-kunci yang sesuai yang digunakan sistem untuk menyimpan setelan dalam file `SharedPreferences` default untuk setelan aplikasi Anda.
- `android:summary` : Rangkuman teks muncul di bawah setelan.

4. Ekstrak sumber daya string untuk nilai atribut `android:title` dan `android:summary` menjadi `@string/switch_title` dan `@string/switch_summary`.

5. Ubah `<SwitchPreference` dalam kode ke `<android.support.v7.preference.SwitchPreferenceCompat`:

```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <android.support.v7.preference.SwitchPreferenceCompat
        ...
    </PreferenceScreen>
```

Untuk menggunakan versi `PreferenceFragmentCompat` dari `PreferenceFragment`, Anda juga harus menggunakan versi `android.support.v7` dari `SwitchPreference` (`switchPreferenceCompat`).

Baris `SwitchPreferenceCompat` di atas mungkin menampilkan ikon bola lampu kuning dengan peringatan, tetapi Anda bisa mengabaikannya.

6. Bukan file `styles.xml` dan tambahkan deklarasi `preferenceTheme` berikut ke `AppTheme` :

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    ...
    <item name="preferenceTheme">@style/PreferenceThemeOverlay</item>
</style>
```

Untuk menggunakan versi `PreferenceFragmentCompat` dari `PreferenceFragment`, Anda juga harus mendeklarasikan `preferenceTheme` dengan gaya `PreferenceThemeOverlay` ke tema aplikasi.

7. Buka file `build.gradle` (Module: app), dan tambahkan yang berikut ini ke bagian `dependencies` :

```
dependencies {
    ...
    compile 'com.android.support:preference-v7:25.0.1'
}
```

Hal di atas menambahkan pustaka `android.support:preference-v7` untuk menggunakan versi `PreferenceFragmentCompat` dari `PreferenceFragment`.

1.3 Tambahkan aktivitas untuk setelan dan fragmen untuk setelan yang spesifik

1. Untuk membuat aktivitas Settings yang menyediakan UI untuk setelan, tambahkan Empty Activity ke aplikasi:
 - Pilih **app** di bagian atas Project: Android.
 - Pilih **New > Activity > Empty Activity**.
 - Beri nama aktivitas **SettingsActivity**.
 - Hapus centang opsi untuk menghasilkan file layout (Anda tidak memerlukannya).
 - Biarkan opsi Backwards Compatibility (AppCompat) dicentang.
 - Nama paket harus sudah disetel ke **com.example.android.projectname**, dan Target Source Set harus disetel ke **main**. Jika tidak, buat pilihan ini di menu tarik-turun.
 - Klik **Finish**.

Hasilnya adalah definisi kelas berikut di `SettingsActivity`:

```
public class SettingsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

2. Tambahkan fragmen kosong untuk grup setelan yang sama (*tanpa* layout, metode bawaan, atau callback antarmuka) ke aplikasi, untuk menukarannya ke dalam layar aktivitas Settings jika diperlukan:

- Pilih **app** di bagian atas tampilan Project: Android lagi.
- Pilih **New > Fragment > Fragment (Blank)**.
- Beri nama fragmen **SettingsFragment**.
- Hapus centang opsi untuk menghasilkan file layout (Anda tidak memerlukannya).
- Hapus centang opsi untuk menyertakan metode bawaan fragmen.
- Hapus centang opsi untuk menyertakan callback antarmuka.
- Klik **Finish**.

Hasilnya adalah definisi kelas berikut di `SettingsFragment`:

```

public class SettingsFragment extends Fragment {

    public SettingsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater,
                           ViewGroup container, Bundle savedInstanceState) {
        TextView textView = new TextView(getActivity());
        textView.setText(R.string.hello_blank_fragment);
        return textView;
    }
}

```

3. Ubah definisi kelas `SettingsFragment` untuk memperluas `PreferenceFragmentCompat` :

```

public class SettingsFragment extends PreferenceFragmentCompat {
    ...
}

```

Anda menggunakan subkelas `Fragment` yang khusus untuk menampilkan daftar setelan. Praktik terbaiknya adalah menggunakan `Aktivitas` rutin yang meng-host `PreferenceFragment` yang menampilkan setelan aplikasi. `Fragmen` seperti `PreferenceFragment` menyediakan arsitektur yang lebih fleksibel untuk aplikasi Anda dibandingkan dengan hanya menggunakan aktivitas. Fragmen mirip dengan bagian modular sebuah aktivitas—fragmen memiliki daur hidupnya sendiri dan menerima kejadian masukannya sendiri, dan bisa menambahkan atau membuang fragmen saat aktivitas sedang berjalan.

Gunakan versi `PreferenceFragmentCompat` dari `PreferenceFragment` dengan aktivitas yang memperluas `AppCompatActivity`. Untuk memperluas fragmen, Anda mungkin harus menambahkan pernyataan import berikut:

```
import android.support.v7.preference.PreferenceFragmentCompat;
```

4. Ganti seluruh metode `onCreateView()` di fragmen dengan metode `onCreate()` ini:

```

@Override
public void onCreatePreferences(Bundle savedInstanceState,
                           String rootKey) {
}

```

Alasan mengapa Anda mengganti `onCreateView()` dengan `onCreatePreferences()` di `SettingsFragment` adalah Anda akan menambahkan fragmen ini ke `SettingsActivity` yang sudah ada untuk menampilkan preferensi, bukan menampilkan layar fragmen yang terpisah. Menambahkannya ke aktivitas yang sudah ada memudahkan Anda menambahkan atau membuang fragmen saat aktivitas sedang berjalan. Fragmen preferensi di-root di `PreferenceScreen` menggunakan `rootKey`.

Anda bisa dengan aman membuang constructor kosong dari fragmen, karena fragmen tidak ditampilkan sendiri:

```

public SettingsFragment() {
    // Required empty public constructor
}

```

5. Di akhir metode `onCreatePreferences()` di `SettingsFragment`, Anda perlu mengaitkan sumber daya setelan `preferences.xml` yang baru saja Anda buat dengan fragmen ini. Tambahkan sebuah panggilan ke `setPreferencesFromResource` dengan meneruskan id file XML (`R.xml.preferences`) dan `rootKey` untuk mengidentifikasi root preferensi di `PreferenceScreen`:

```
setPreferencesFromResource(R.xml.preferences, rootKey);
```

Metode `onCreatePreferences()` sekarang harus terlihat seperti ini:

```
@Override
public void onCreatePreferences(Bundle savedInstanceState,
                               String rootKey) {
    setPreferencesFromResource(R.xml.preferences, rootKey);
}
```

6. Tambahkan kode berikut ke akhir metode `onCreate()` `SettingsActivity` sehingga fragmen ditampilkan sebagai konten utama:

```
getSupportFragmentManager().beginTransaction()
    .replace(android.R.id.content, new SettingsFragment())
    .commit();
```

Kode di atas adalah pola umum yang digunakan untuk menambahkan fragmen ke sebuah aktivitas sehingga fragmen muncul sebagai konten utama aktivitas. Anda menggunakan:

- o `getFragmentManager()` jika kelas memperluas `Activity` dan fragmen memperluas `PreferenceFragment`.
- o `getSupportFragmentManager()` jika kelas memperluas `AppCompatActivity` dan fragmen memperluas `PreferenceFragmentCompat`.

Seluruh metode `onCreate()` di `SettingsActivity` sekarang harus terlihat seperti berikut:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getSupportFragmentManager().beginTransaction()
        .replace(android.R.id.content, new SettingsFragment())
        .commit();
}
```

1.4 Hubungkan item menu Settings ke aktivitas setelan

Gunakan `intent` untuk meluncurkan `SettingsActivity` dari `MainActivity`.

1. Temukan blok `if` di metode `onOptionsItemSelected()` di `MainActivity` yang menangani ketukan pada **Settings** di menu opsi:

```
if (id == R.id.action_settings) {
    return true;
}
```

2. Tambahkan `intent` ke blok `if` untuk meluncurkan `SettingsActivity`:

```
if (id == R.id.action_settings) {
    Intent intent = new Intent(this, SettingsActivity.class);
    startActivity(intent);
    return true;
}
```

3. Tambahkan navigasi tombol **Up** ke `SettingsActivity` dengan mengedit deklarasinya di file **AndroidManifest.xml** untuk mendefinisikan induk aktivitas sebagai `MainActivity`.

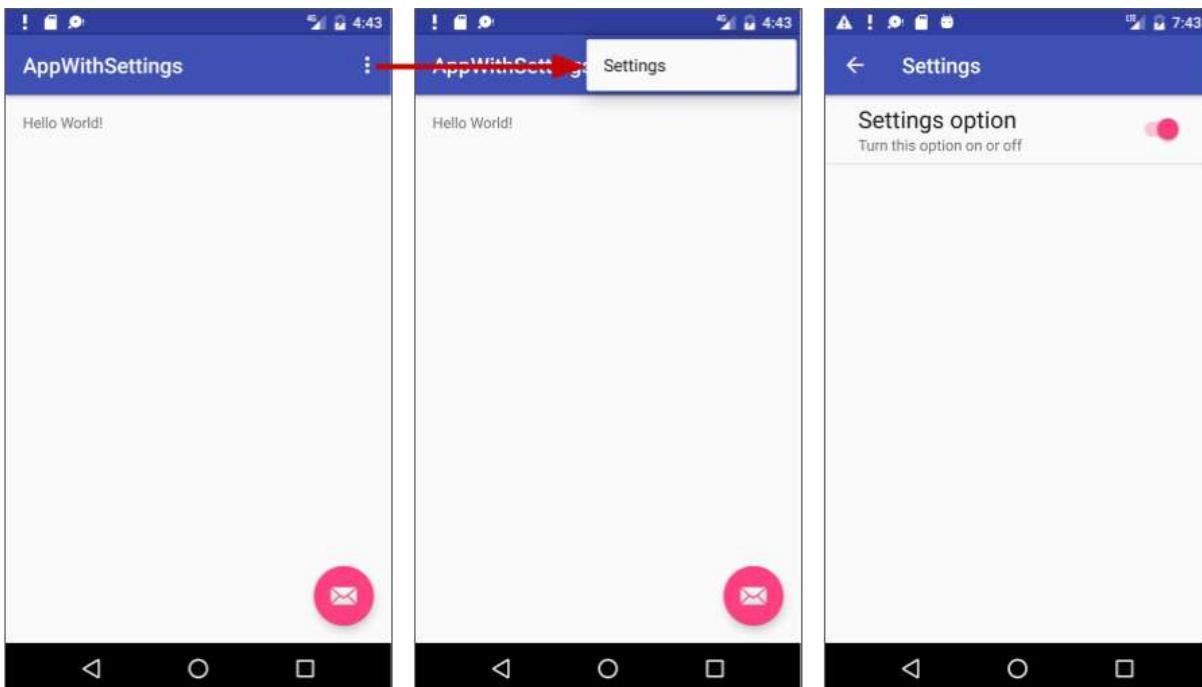
- i. Temukan deklarasi `SettingsActivity` di `AndroidManifest.xml`:

```
<activity android:name=".SettingsActivity"></activity>
```

- ii. Ubah deklarasi menjadi yang berikut ini:

```
<activity android:name=".SettingsActivity"
    android:label="Settings"
    android:parentActivityName=".MainActivity">
<meta-data
    android:name="android.support.PARENT_ACTIVITY"
    android:value=".MainActivity"/>
</activity>
```

4. Jalankan aplikasi. Ketuk ikon luapan untuk menu opsi (seperti yang ditampilkan di sebelah kiri gambar di bawah) dan ketuk **Settings** untuk melihat aktivitas setelan (seperti yang ditampilkan di tengah gambar di bawah). Ketuk tombol **Naik** di bilah aplikasi pada aktivitas setelan, yang ditampilkan di sebelah kanan gambar di bawah, untuk kembali ke aktivitas utama.



1.5 Simpan nilai default di preferensi bersama

Meskipun nilai default untuk setelan tombol alih telah disetel di atribut `android:defaultValue` (di [Langkah 1.2](#) tugas ini), aplikasi harus menyimpan nilai default di file SharedPreferences untuk setiap setelan saat pengguna pertama kali membuka aplikasi. Ikuti langkah-langkah berikut untuk menyetel nilai default untuk tombol alih:

1. Di `MainActivity`, tambahkan yang berikut ke akhir metode `onCreate()` yang sudah ada:

```
protected void onCreate(Bundle savedInstanceState) {
    ...
    PreferenceManager.setDefaultValues(this, R.xml.preferences, false);
}
```

Kode di atas memastikan setelan diinisialisasi dengan benar dengan nilai defaultnya. Metode `setDefaultValues()` membutuhkan tiga argumen:

2. **Konteks** aplikasi, seperti `this`.
3. ID sumber daya (`preferences`) untuk file sumber daya XML dengan satu atau beberapa setelan.
4. Boolean yang menunjukkan apakah nilai default harus disetel lebih dari satu kali. Bila `false`, sistem akan mengatur nilai default hanya jika metode ini belum pernah dipanggil sebelumnya. Selama Anda menyetel argumen ketiga ini ke `false`, Anda bisa dengan aman memanggil metode ini setiap kali aktivitas utama dimulai tanpa mengganti nilai setelan pengguna yang disimpan. Akan tetapi, jika Anda mengaturnya ke `true`, metode akan mengganti nilai sebelumnya dengan default.

1.6 Bacalah nilai setelan yang diubah dari preferensi bersama

Saat aplikasi dimulai, metode `onCreate()` `MainActivity` bisa membaca nilai setelan yang telah berubah, dan menggunakan nilai yang telah berubah, bukannya nilai default.

Setiap setelan dikenali menggunakan pasangan kunci-nilai. Sistem Android menggunakan pasangan kunci-nilai ini saat menyimpan atau mengambil setelan dari file `SharedPreferences` untuk aplikasi Anda. Bila pengguna mengubah setelan, sistem akan memperbarui nilai yang bersangkutan dalam file `SharedPreferences`. Untuk menggunakan nilai setelan, aplikasi bisa menggunakan kunci untuk mendapatkan setelan dari file `SharedPreferences`.

Ikuti langkah-langkah berikut untuk menambahkan kode tersebut:

1. Sebelum menambahkan kode untuk membaca nilai setelan, buat variabel string statis di `SettingsActivity` untuk menampung kunci untuk nilai:

```
public class SettingsActivity extends AppCompatActivity {
    public static final String
        KEY_PREF_EXAMPLE_SWITCH = "example_switch";
    ...
}
```

2. Di metode `onCreate()` di `MainActivity`, tambahkan yang berikut ini ke akhir metode:

```
protected void onCreate(Bundle savedInstanceState) {
    ...
    SharedPreferences sharedPref =
        PreferenceManager.getDefaultSharedPreferences(this);
    Boolean switchPref = sharedPref.getBoolean
        (SettingsActivity.KEY_PREF_EXAMPLE_SWITCH, false);
}
```

Cuplikan kode di atas menggunakan

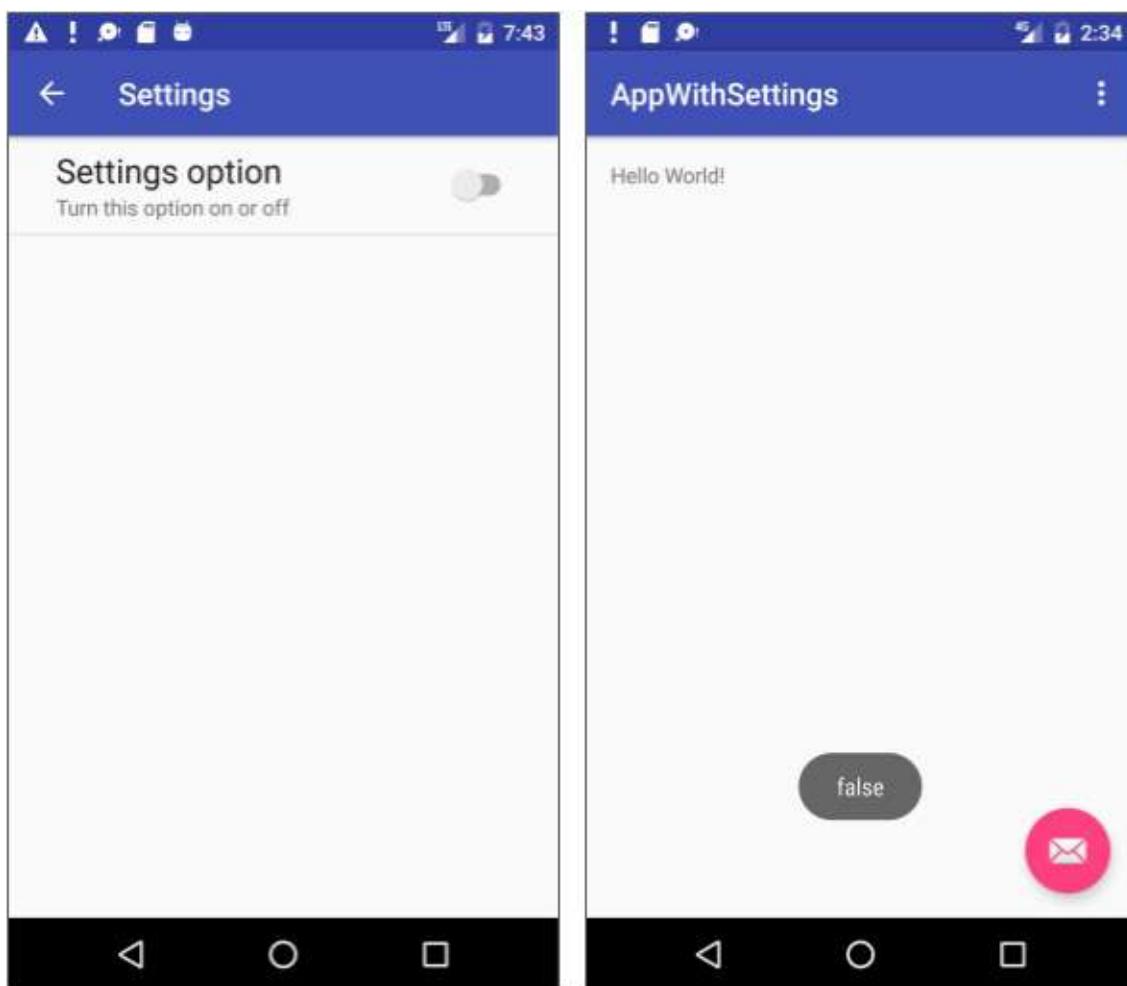
- o `PreferenceManager.getDefaultSharedPreferences(this)` untuk mendapatkan setelan sebagai objek `SharedPreferences` (`sharedPref`).
- o `getBoolean()` untuk mendapatkan nilai Boolean setelan yang menggunakan kunci (`KEY_PREF_EXAMPLE_SWITCH` yang didefinisikan di `SettingsActivity`) dan menetapkannya ke `switchPref`. Jika tidak ada nilai untuk kunci, metode `getBoolean()` akan menyetel nilai setelan (`switchPref`) ke `false`. Untuk nilai lain seperti string, integer, dan bilangan titik mengambang, Anda bisa menggunakan metode `getString()`, `getInt()`, atau `getFloat()`.

3. Tambahkan metode `Toast.makeText()` ke `onCreate()` yang menampilkan nilai setelan `switchPref` di sebuah toast:

```
Toast.makeText(this, switchPref.toString(), Toast.LENGTH_SHORT).show();
```

4. Jalankan aplikasi, lalu ikuti langkah-langkah ini:

- i. Ketuk **Settings** untuk melihat aktivitas setelan.
- ii. Ketuk setelan untuk mengubah tombol alih dari on ke off, seperti yang ditampilkan di sebelah kiri gambar di bawah.
- iii. Ketuk tombol **Naik** di aktivitas setelan untuk kembali ke aktivitas utama. Pesan toast akan muncul di aktivitas utama dengan nilai setelan, seperti yang ditampilkan di sebelah kanan gambar di bawah.
- iv. Ulangi langkah-langkah ini untuk melihat perubahan pesan toast ketika Anda mengubah setelan.



Kapan pun MainActivity dimulai atau dimulai ulang, metode `onCreate()` harus membaca nilai setelan untuk menggunakannya di aplikasi. Metode `Toast.makeText()` akan diganti dengan metode yang menginisialisasi setelan.

Anda sekarang memiliki aktivitas setelan di aplikasi yang berfungsi.

Kode solusi:

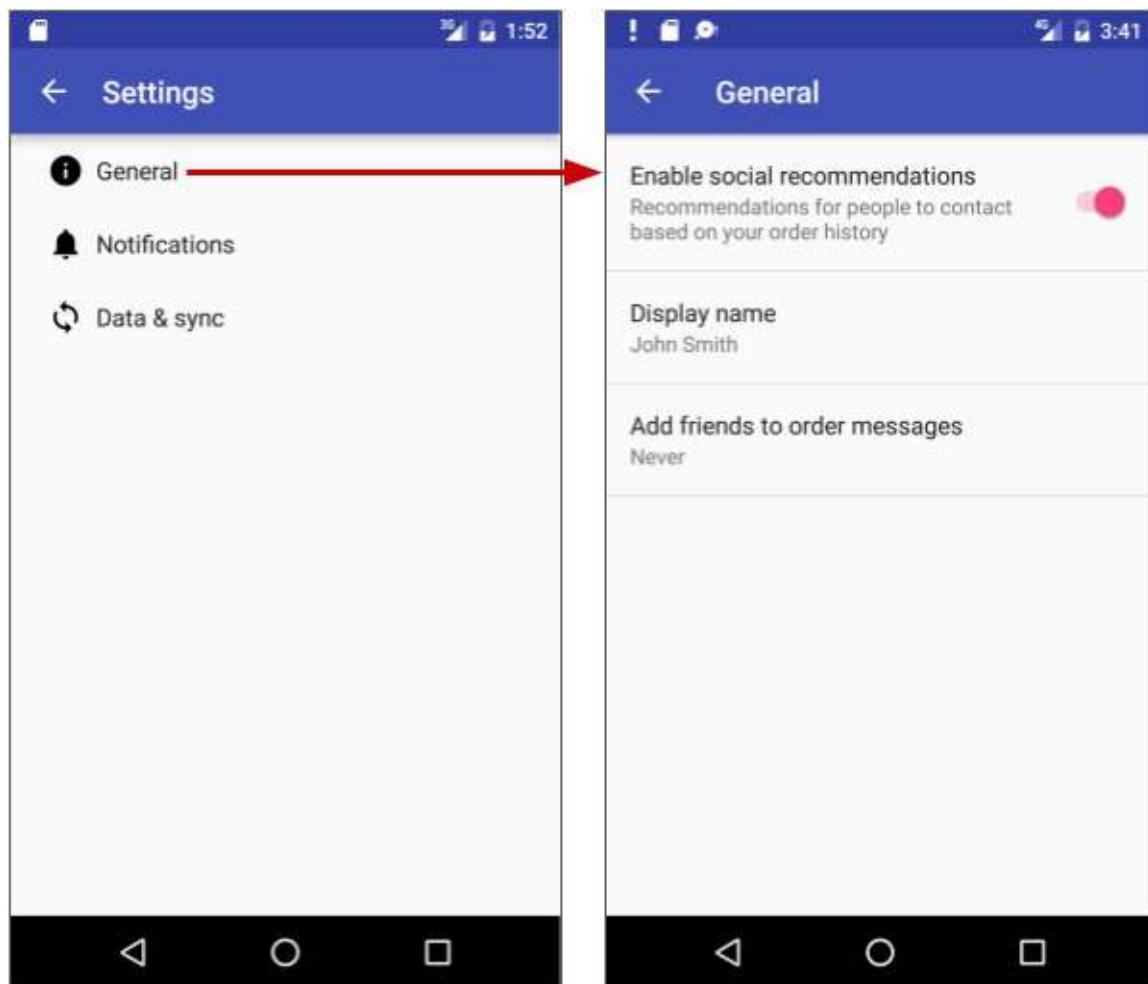
Proyek Android Studio: [AppWithSettings](#)

Tugas 2: Menggunakan template Settings Activity

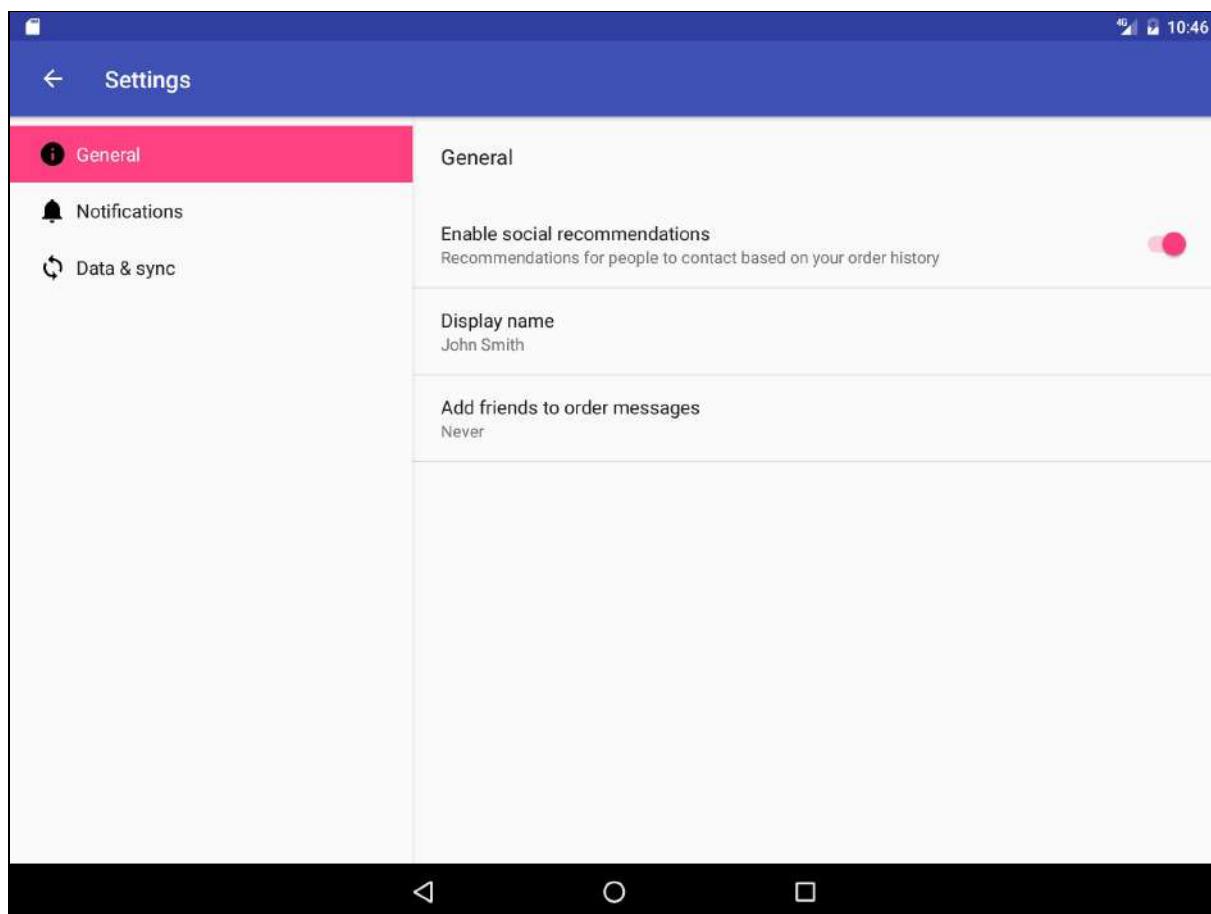
Jika Anda perlu membangun beberapa sublayar setelan dan ingin memanfaatkan layar berukuran tablet, serta mempertahankan kompatibilitas dengan versi Android untuk tablet yang lebih lama, Android Studio menyediakan pintasan: template Settings Activity.

Di tugas sebelumnya, Anda telah belajar cara menggunakan aktivitas setelan kosong dan fragmen kosong untuk menambahkan setelan ke aplikasi. Tugas 2 sekarang akan menampilkan cara menggunakan template Settings Activity yang disediakan di Android Studio untuk:

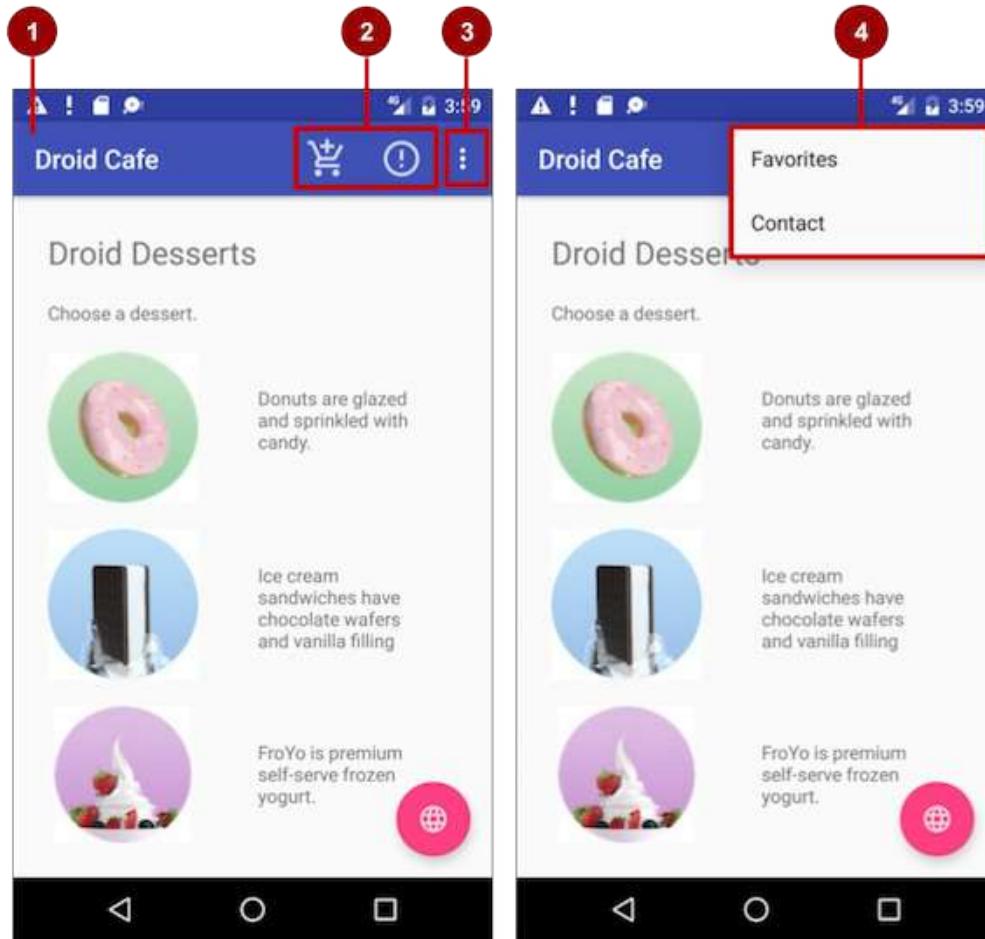
- Membagi beberapa setelan ke dalam grup.
- Menyesuaikan setelan dan nilainya.
- Menampilkan layar Settings utama dengan tautan header untuk setiap grup setelan, seperti General untuk setelan umum, seperti yang ditampilkan dalam gambar di bawah ini.



- Menampilkan layout layar detail/master dengan tautan header untuk setiap grup di sebelah kiri (master), dan grup setelan di sebelah kanan (detail), seperti yang ditampilkan dalam gambar di bawah ini.



Di praktik sebelumnya, Anda telah membuat aplikasi yang bernama Droid Cafe menggunakan template Basic Activity, yang menyediakan menu opsi di bilah aplikasi seperti yang ditampilkan di bawah ini.



Pada gambar di atas:

1. Bilah aplikasi.
2. Ikon tindakan menu opsi.
3. Tombol luapan.
4. Menu luapan opsi.

Proyek Android Studio: Untuk memulai proyek dari posisi terakhir yang Anda tinggalkan di praktik sebelumnya, unduh proyek Android Studio [DroidCafe](#).

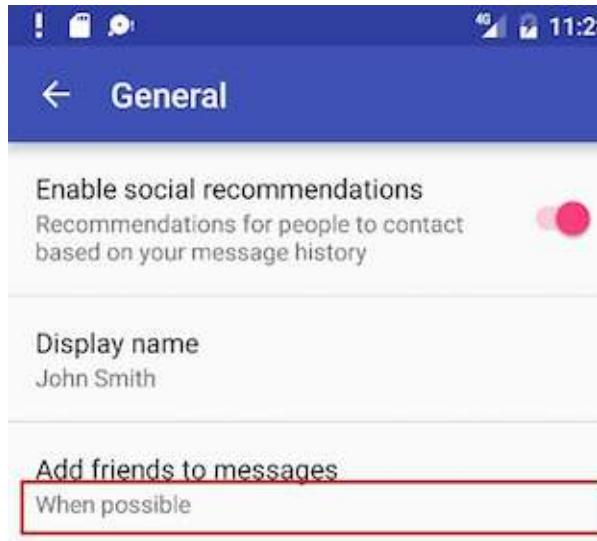
2.1 Menjelajahi template Settings Activity

Untuk menyertakan template Settings Activity di proyek aplikasi Anda di Android Studio, ikuti langkah-langkah ini:

1. Salin folder proyek **DroidCafe**, ubah namanya menjadi **DroidCafeWithSettings** dan optimalkan kodenya. (Lihat [Apendiks](#) untuk petunjuk menyalin proyek.) Jalankan aplikasi untuk memastikan berjalan lancar.
2. Pilih **app** di bagian atas tampilan Project: Android dan pilih **New > Activity > Settings Activity**.
3. Dalam dialog yang muncul, terima Activity Name (**SettingsActivity** adalah nama yang disarankan) dan Title (**Settings**).
4. Klik tiga titik di akhir bidang Hierarchical Parent dan pilih **MainActivity** sebagai aktivitas induk, sehingga tombol **Naik** di Settings Activity mengembalikan pengguna ke MainActivity. Memilih aktivitas induk otomatis memperbarui file **AndroidManifest.xml** untuk mendukung navigasi tombol **Naik**.
5. Klik **Finish**.

Template Settings Activity tidak hanya menyediakan layout untuk layar berukuran ponsel dan tablet, tetapi juga menyediakan fungsi untuk mendengarkan perubahan setelan dan mengubah setelan untuk merefleksikan perubahan setelan. Misalnya, jika Anda mengubah setelan "Add friends to messages" (opsinya adalah **Always**, **When possible**, atau

Never), opsi yang Anda pilih muncul di rangkuman di bawah setelan:



Secara umum, Anda tidak perlu mengubah kode template Settings Activity untuk menyesuaikan aktivitas untuk setelan yang Anda inginkan di aplikasi. Anda bisa menyesuaikan judul, rangkuman, nilai yang memungkinkan, nilai default setelan tanpa mengubah kode template, dan bahkan menambahkan setelan lain ke grup yang disediakan.

Anda menggunakan kode Settings Activity apa adanya. Agar berfungsi untuk aplikasi Anda, tambahkan kode ke Main Activity untuk menyetel nilai setelan default, dan untuk *membaca* dan *menggunakan* nilai setelan, seperti yang ditampilkan nanti di tugas ini.

Template Settings Activity membuat yang berikut ini untuk Anda:

- File XML di direktori **res > xml**, tempat Anda bisa menambah atau menyesuaikan setelan yang Anda inginkan.
 - **pref_data_sync.xml**: Layout PreferenceScreen untuk setelan "Data & sync".
 - **pref_general.xml**: Layout PreferenceScreen untuk setelan "General".
 - **pref_headers.xml**: Layout header untuk layar tampilan Setelan.
 - **pref_notification.xml**: Layout PreferenceScreen untuk setelan "Notifications".

Layout XML di atas menggunakan beragam subkelas dari kelas [Preference](#), bukan objek [View](#), dan subkelas langsung menyediakan container untuk layout yang melibatkan beberapa setelan. Misalnya [PreferenceScreen](#) merepresentasikan [Preference](#) tingkat atas, yaitu root hierarki Preference. File di atas menggunakan [PreferenceScreen](#) di bagian atas setiap layar setelan. Subkelas [Preference](#) untuk setelan lainnya menyediakan UI yang sesuai bagi pengguna untuk mengubah setelan. Misalnya:

- [CheckBoxPreference](#): Kotak centang untuk setelan yang diaktifkan atau dinonaktifkan.
- [ListPreference](#): Dialog yang berisi daftar tombol radio.
- [SwitchPreference](#): Opsi dua keadaan yang bisa dialihkan (misalnya on/off atau true/false).
- [EditTextPreference](#): Dialog dengan sebuah widget [EditText](#).
- [RingtonePreference](#): Sebuah dialog dengan nada dering di perangkat.

Tip: Anda bisa mengedit file XML untuk mengubah setelan default ke setelan yang Anda perlukan untuk aplikasi.

- Sumber daya string di file `strings.xml` di direktori **res > values** yang bisa Anda sesuaikan untuk setelan yang Anda inginkan.

Semua yang digunakan di Settings Activity, seperti judul untuk setelan, lirik string untuk daftar, dan keterangan untuk setelan, didefinisikan sebagai sumber daya string di akhir file ini. String ini ditandai oleh komentar, seperti `<!-- strings related to Settings -->` and `<!-- Example General settings -->`.

Tip: Anda bisa mengedit string ini untuk menyesuaikan setelan yang Anda perlukan untuk aplikasi Anda.

- `SettingsActivity` di direktori **java > com.example.android.projectname**, yang bisa Anda gunakan apa adanya.

Ini aktivitas yang menampilkan setelan. `SettingsActivity` memperluas `AppCompatPreferenceActivity` untuk mempertahankan kompatibilitas dengan versi Android yang lebih lama.

- `AppCompatPreferenceActivity` di direktori `java > com.example.android.projectname` directory, yang bisa Anda gunakan apa adanya.

Aktivitas ini adalah kelas helper yang digunakan oleh `SettingsActivity` untuk mempertahankan kompatibilitas mundur dengan versi Android lama.

2.2 Tambahkan item menu Settings dan hubungkan ke aktivitas

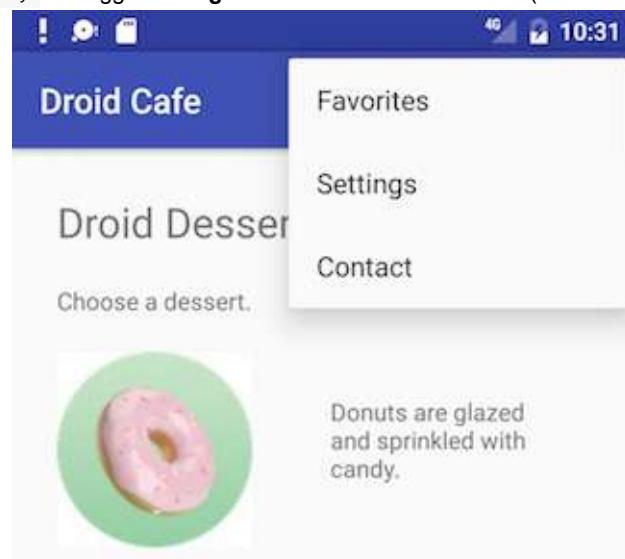
Seperti yang sudah Anda pelajari di praktik sebelumnya, Anda bisa mengedit file `menu_main.xml` di direktori `res > menu` untuk menu opsi untuk menambahkan atau membuang item menu.

1. Editi file `menu_main.xml` untuk menambahkan item menu lainnya yang bernama `Settings` dengan id sumber daya baru `action_settings` :

```
<item
    android:id="@+id/action_settings"
    android:orderInCategory="50"
    android:title="Settings"
    app:showAsAction="never" />
```

Tetapkan "never" untuk atribut `app:showAsAction` sehingga **Settings** hanya muncul di menu opsi luapan, dan tidak di bilah aplikasi itu sendiri, karena tidak akan sering digunakan.

Tetapkan "50" untuk atribut `android:orderInCategory` sehingga **Settings** muncul di bawah **Favorites** (setel ke



"40") tetapi di atas **Contact** (setel di atas "100").

2. Ekstrak sumber daya string untuk "Settings" di atribut `android:title` ke nama sumber daya `settings`.
3. Di `MainActivity`, temukan blok `switch-case` di metode `onOptionsItemSelected()` yang menangani ketukan pada item di menu opsi:

```

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_order:
            displayToast(getString(R.string.action_order_message));
            return true;
        case R.id.action_status:
            displayToast(getString(R.string.action_status_message));
            return true;
        case R.id.action_favorites:
            displayToast(getString(R.string.action_favorites_message));
            return true;
        case R.id.action_contact:
            displayToast(getString(R.string.action_contact_message));
            return true;
    }

    return super.onOptionsItemSelected(item);
}

```

4. Gunakan `Intent` untuk meluncurkan `SettingsActivity` dari `MainActivity`. Tambahkan `Intent` di akhir blok `switch case` :

```

...
case R.id.action_settings:
    Intent intent = new Intent(this, SettingsActivity.class);
    startActivity(intent);
    return true;
}

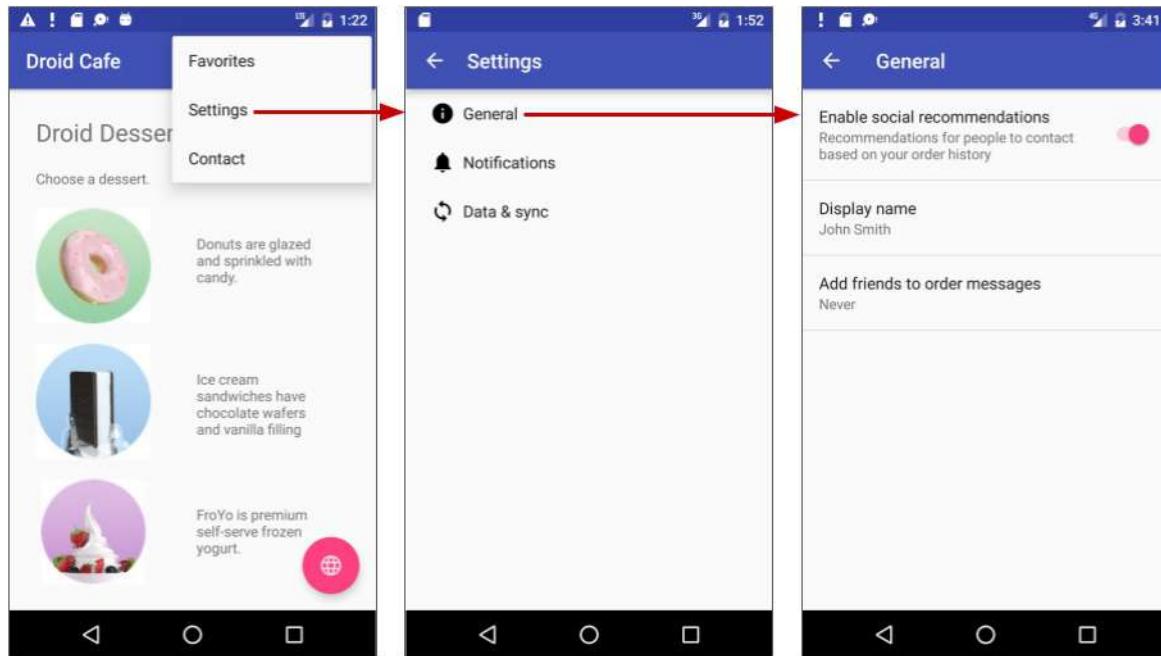
return super.onOptionsItemSelected(item);
}

```

5. Jalankan aplikasi menggunakan ponsel cerdas atau emulator ponsel cerdas sehingga Anda bisa melihat bagaimana template `Settings Activity` menangani ukuran layar ponsel cerdas, dan ikuti langkah-langkah ini:

- i. Ketuk ikon luapan untuk menu opsi dan ketuk **Settings** untuk melihat aktivitas setelan, seperti yang ditampilkan di sebelah kiri gambar di bawah.
- ii. Ketuk setiap header setelan (**General**, **Notifications**, dan **Data & sync**), seperti yang ditampilkan di bagian tengah gambar di bawah, untuk melihat grup setelan di setiap layar anak di layar anak `Settings`, seperti yang ditampilkan di sebelah kanan gambar di bawah.

- iii. Ketuk tombol **Naik** di aktivitas setelan untuk kembali ke aktivitas utama.



2.3 Sesuaikan setelan yang disediakan oleh template

Untuk menyesuaikan setelan yang disediakan oleh template Settings Activity, edit sumber daya string dan laris string di file strings.xml dan atribut layout untuk setiap setelan di file di direktori **xml**. Pada langkah ini, Anda akan mengubah setelan "Data & sync".

1. Buka file **strings.xml** di direktori **res > values** dan konten ke komentar `<!-- Example settings for Data & Sync -->`:

```
<!-- Example settings for Data & Sync -->
<string name="pref_header_data_sync">Data & sync</string>

<string name="pref_title_sync_frequency">Sync frequency</string>
<string-array name="pref_sync_frequency_titles">
    <item>15 minutes</item>
    <item>30 minutes</item>
    <item>1 hour</item>
    <item>3 hours</item>
    <item>6 hours</item>
    <item>Never</item>
</string-array>
<string-array name="pref_sync_frequency_values">
    <item>15</item>
    <item>30</item>
    <item>60</item>
    <item>180</item>
    <item>360</item>
    <item>-1</item>
</string-array>
...
...
```

2. Edit sumber daya string `pref_header_data_sync` yang disetel ke `Data & sinkron` (`&` adalah kode HTML untuk ampersand). Ubah nilai ke **Account** (tanpa tanda petik).
3. Optimalkan nama sumber daya dengan mengikuti langkah-langkah ini (aplikasi masih akan berjalan tanpa mengoptimalkan nama, tetapi pengoptimalan akan membuat kode lebih mudah dipahami):
 - i. Kontrol-klik (atau klik kanan) nama sumber daya `pref_header_data_sync` dan pilih **Refactor > Rename**.
 - ii. Ubah nama menjadi `pref_header_account`, klik opsi untuk menelusuri di komentar dan string, dan klik **Refactor**.
4. Edit sumber daya string `pref_title_sync_frequency` (yang disetel ke `Sync frequency`) menjadi **Market**.
5. **Refactor > Rename** nama sumber daya menjadi `pref_title_account` seperti yang sudah Anda lakukan sebelumnya.

6. **Refactor > Rename** sumber daya larik string `pref_sync_frequency_titles` menjadi `pref_market_titles`.
7. Ubah setiap nilai di larik string `pref_market_titles` (`15 minutes` , `30 minutes` , `1 hour` , dsb.) menjadi judul pasar, seperti **United States**, **Canada**, dsb., bukan frekuensi:

```
<string-array name="pref_market_titles">
    <item>United States</item>
    <item>Canada</item>
    <item>United Kingdom</item>
    <item>India</item>
    <item>Japan</item>
    <item>Other</item>
</string-array>
```

8. **Refactor > Rename** sumber daya larik string `pref_sync_frequency_values` menjadi `pref_market_values`.
9. Ubah setiap nilai di larik string `pref_market_values` (`15` , `30` , `60` , dsb.) menjadi nilai pasar—yaitu singkatan, seperti **US**, **CA**, dsb.:

```
<string-array name="pref_market_values">
    <item>US</item>
    <item>CA</item>
    <item>UK</item>
    <item>IN</item>
    <item>JA</item>
    <item>-1</item>
</string-array>
```

10. Gulir ke bawah ke sumber daya string `pref_title_system_sync_settings` dan edit sumber daya (yang disetel ke `System sync settings`) menjadi **Account settings**.
11. **Refactor > Rename** sumber daya larik string `pref_title_system_sync_settings` menjadi `pref_title_account_settings`.
12. Buka file `pref_data_sync.xml`. `ListPreference` di layout ini mendefinisikan setelan yang baru saja Anda ubah. Perhatikan bahwa sumber daya string untuk atribut `android:entries` , `android:entryValues` , dan `android:title` sekarang diubah menjadi nilai yang Anda berikan di langkah sebelumnya:

```
<ListPreference
    android:defaultValue="180"
    android:entries="@array/pref_market_titles"
    android:entryValues="@array/pref_market_values"
    android:key="sync_frequency"
    android:negativeButtonText="@null"
    android:positiveButtonText="@null"
    android:title="@string/pref_title_account" />
```

13. Ubah atribut `android:defaultValue` :

```
    android:defaultValue="US"
```

Karena kunci untuk preferensi setelan ini (`"sync_frequency"`) di-hardcode di mana saja di kode Java, jangan ubah atribut `android:key` —terus gunakan `"sync_frequency"` sebagai kunci untuk setelan di contoh ini. Jika Anda menyesuaikan setelan secara detail di aplikasi dunia nyata, Anda akan menghabiskan waktu mengubah kunci yang di-hardcode di seluruh kode.

Catatan: Mengapa tidak menggunakan sumber daya string untuk kunci? Karena sumber daya string bisa dilokalkan ke bahasa yang berbeda menggunakan file XML multibahasa dan kunci bisa tidak sengaja diterjemahkan bersama dengan dengan string lain, yang bisa menyebabkan aplikasi crash.

2.4 Tambahkan kode untuk menyetel nilai default untuk setelan

Temukan metode `onCreate()` di `MainActivity`, dan tambahkan pernyataan `PreferenceManager.setDefaultValues` berikut di akhir metode:

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    PreferenceManager.setDefaultValues(this, R.xml.pref_general, false);
    PreferenceManager.setDefaultValues(this, R.xml.pref_notification, false);
    PreferenceManager.setDefaultValues(this, R.xml.pref_data_sync, false);
}

```

Nilai default sudah ditetapkan di file XML dengan atribut `android:defaultValue`, tetapi pernyataan di atas memastikan bahwa file Preferensi Bersama diinisialisasi dengan benar dengan nilai default. Metode `setDefaultvalues()` membutuhkan tiga argumen:

- Konteks aplikasi, seperti `this`.
- ID sumber daya untuk file XML layout setelan yang menyertakan nilai default yang disetel oleh atribut `android:defaultValue`.
- Boolean yang menunjukkan apakah nilai default harus disetel lebih dari satu kali. Bila `false`, sistem akan mengatur nilai default hanya jika metode ini belum pernah dipanggil sebelumnya. Selama Anda menyetel argumen ketiga ini ke `false`, Anda bisa dengan aman memanggil metode ini setiap kali aktivitas Anda dimulai tanpa mengesampingkan nilai setelan yang disimpan pengguna dengan menyetel ulang preferensi tersebut ke nilai default. Akan tetapi, jika Anda mengurnya ke `true`, metode akan mengganti nilai sebelumnya dengan default.

2.5 Tambahkan kode untuk membaca nilai untuk setelan

1. Tambahkan kode berikut di akhir metode `onCreate()` `MainActivity`. Anda bisa langsung menambahkannya setelah kode yang Anda tambahkan di langkah sebelumnya untuk menyetel default untuk setelan:

```

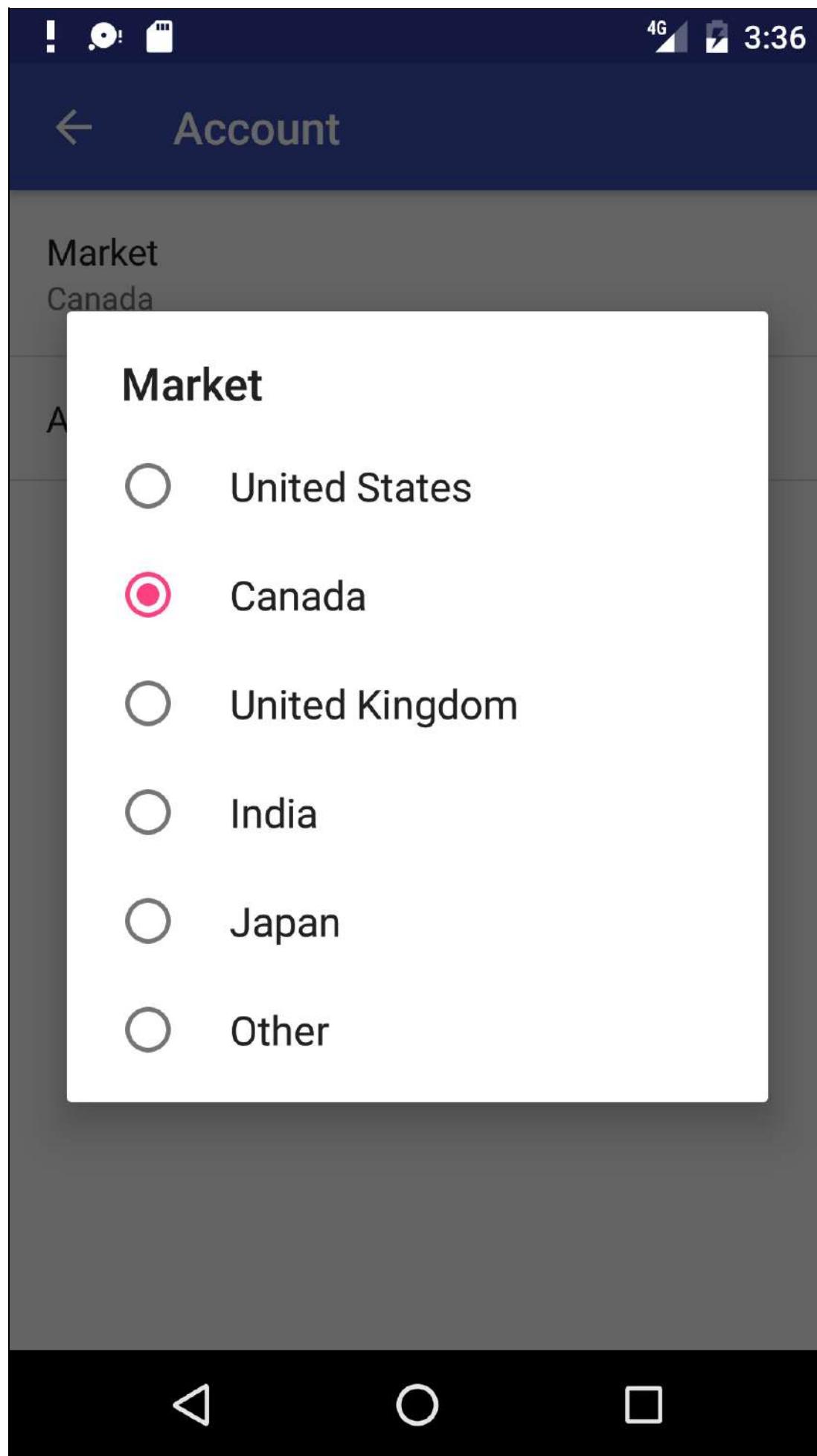
...
SharedPreferences sharedPref =
    PreferenceManager.getDefaultSharedPreferences(this);
String marketPref = sharedPref.getString("sync_frequency", "-1");
Toast.makeText(this, marketPref, Toast.LENGTH_SHORT).show();
}

```

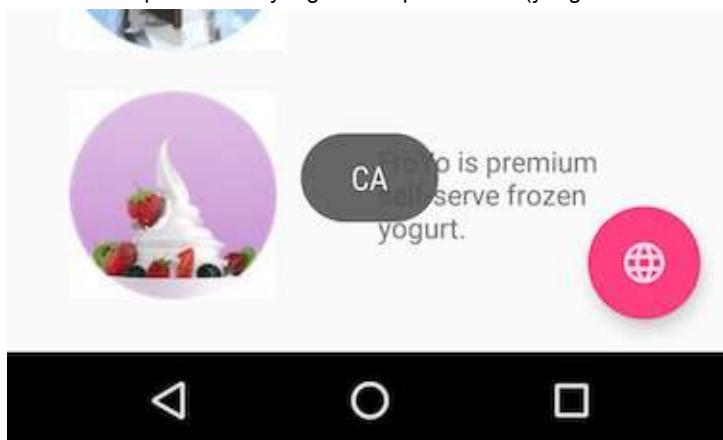
Seperti yang telah Anda pelajari di tugas sebelumnya, Anda menggunakan

`PreferenceManager.getDefaultSharedPreferences(this)` untuk mendapatkan setelan sebagai objek `SharedPreferences` (`marketPref`). Anda kemudian menggunakan `getString()` untuk mendapatkan nilai string dari setelan yang menggunakan kunci (`sync_frequency`) dan menetapkannya ke `marketPref`. Jika tidak ada nilai untuk kunci, metode `getString()` menyetel nilai setelan `marketPref` ke `-1`, yang merupakan nilai `other` di larik `pref_market_values`.

2. Jalankan aplikasi lagi menggunakan ponsel cerdas atau emulator ponsel cerdas. Saat layar utama aplikasi muncul pertama kali, Anda akan melihat pesan toast di bawah layar. Pertama kali menjalankan aplikasi, Anda akan melihat `"-1"` yang ditampilkan di toast karena Anda belum mengubah setelan.
3. Ketuk **Settings** di menu opsi dan ketuk **Account** di layar Settings. Pilih *Canada** di "Market" seperti yang ditampilkan di bawah ini:

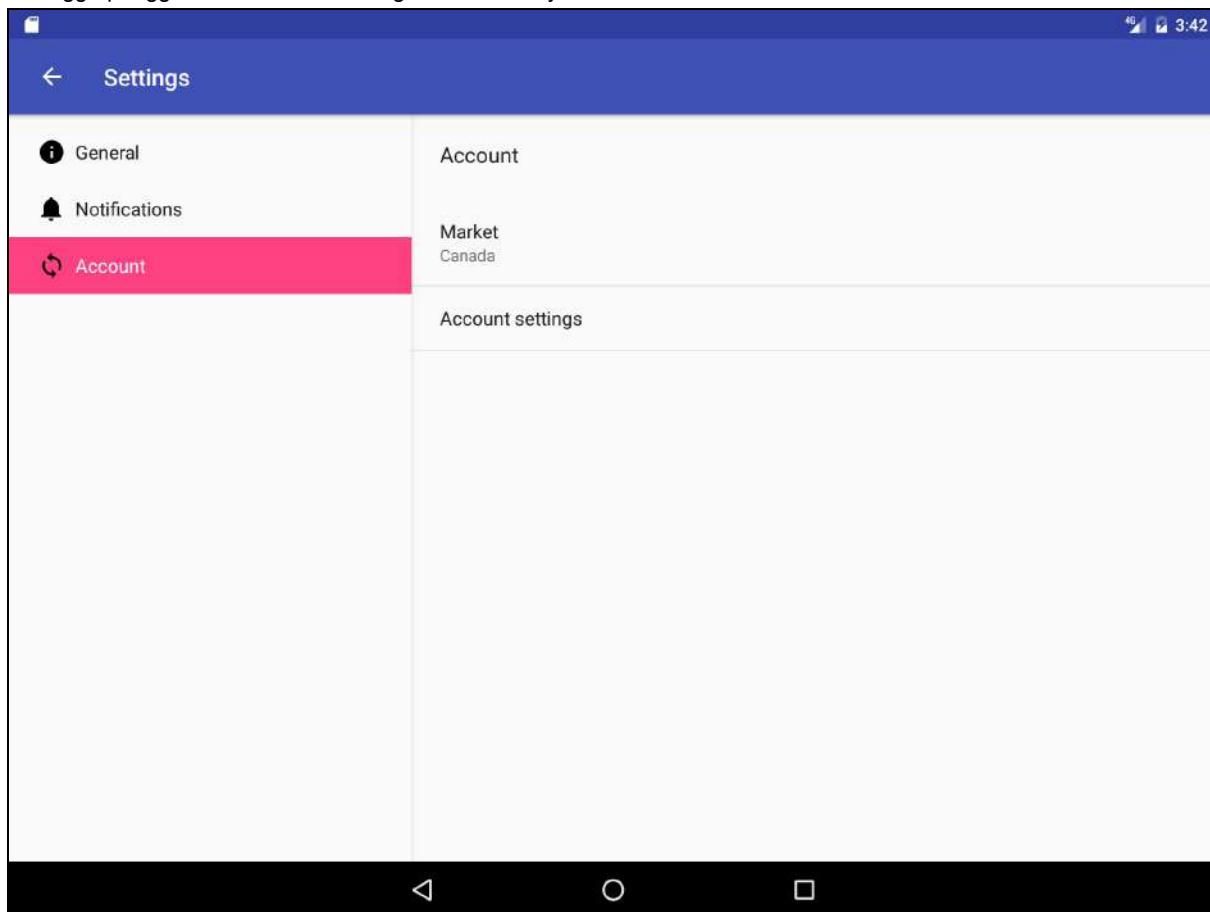


4. Ketuk tombol **Naik** di bilah aplikasi untuk kembali ke layar Settings, dan ketuk lagi untuk kembali ke layar utama. Anda akan melihat pesan toast yang menampilkan "CA" (yang berarti Canada):



Anda telah berhasil mengintegrasikan Settings Activity dengan aplikasi Droid Cafe.

5. Sekarang jalankan aplikasi di tablet atau emulator tablet. Karena secara fisik tablet memiliki layar yang lebih besar, waktu proses Android memanfaatkan ruang ekstra ini. Di tablet, setelan dan detailnya ditampilkan di layar yang sama, sehingga pengguna lebih mudah mengelola setelannya.



Kode solusi

Proyek Android Studio: [DroidCafeWithSettings](#) (Termasuk tantangan penyusunan kode #1).

Proyek Android Studio: [DroidCafeWithSettingsChallenge](#) (Termasuk tantangan penyusunan kode #2).

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode bersifat opsional dan bukan merupakan prasyarat untuk materi di bab berikutnya.

Tantangan 1: Tambahkan kode ke DroidCafeWithSettings yang membaca nilai tombol alih "Enable Social recommendations" layar anak General dari Settings, dan menampilkan nilainya beserta setelan "Market" di pesan toast yang sama layar utama.

Petunjuk: Gunakan variabel `Boolean` dengan `shared.Pref.getBoolean` dan kunci "`example_switch`".

Tantangan 2: Aplikasi DroidCafeWithSettings menampilkan setelan pada layar yang berukuran tablet dengan benar, tetapi tombol **Up** di bilah aplikasi tidak mengembalikan pengguna ke MainActivity seperti di layar berukuran ponsel cerdas. Ini karena metode `onOptionsItemSelected()` di setiap fragmen di SettingsActivity. Ini menggunakan yang berikut ini untuk memulai ulang SettingsActivity saat pengguna mengetuk tombol **Naik**:

```
startActivity(new Intent(getActivity(), SettingsActivity.class));
```

Yang di atas adalah tindakan yang sesuai di layar ponsel cerdas, tempat header Settings (**General**, **Notifications**, dan **Account**) muncul di layar yang terpisah. Setelah mengubah setelan, Anda menginginkan ketukan pengguna pada tombol **Naik** untuk mengembalikan pengguna ke header Settings.

Akan tetapi, di tablet, header selalu tampak di panel kiri (sedangkan setelan di panel kanan). Hasilnya, mengetuk tombol **Naik** tidak membawa pengguna ke MainActivity.

Temukan cara untuk membuat tombol **Naik** berfungsi dengan benar di SettingsActivity pada layar berukuran tablet.

Petunjuk: Ada beberapa cara untuk memperbaiki masalah ini. Pertimbangkan yang berikut:

- Anda bisa menggunakan beberapa file `dimens.xml` di aplikasi Anda untuk mengakomodasi ukuran layar yang berbeda. Saat aplikasi berjalan di perangkat tertentu, file `dimens.xml` yang sesuai dipilih berdasarkan qualifier untuk file `dimens.xml`. Misalnya, aplikasi sudah memiliki file `dimens.xml (w820dp)` di direktori `res > values`, menggunakan qualifier `(w820dp)` untuk menetapkan perangkat dengan lebar layar 820dp atau lebih besar. Anda bisa menambahkan file `dimens.xml` lain dengan qualifier `Large` untuk menetapkan perangkat apa pun dengan layar lebar, seperti tablet. Aplikasi juga menyertakan file `dimens.xml` di direktori `res > values` untuk semua perangkat lain, seperti ponsel cerdas.
- Anda bisa menambahkan sumber daya `bool` berikut ini antara file `<resources>` and `</resources>` tags in the `dimens.xml (large)` yang otomatis dipilih untuk tablet:

```
<resources>
    <bool name="isTablet">true</bool>
</resources>
```

- Anda bisa menambahkan sumber daya `bool` berikut ke file `dimens.xml`, yang dipilih saat aplikasi berjalan pada perangkat apa pun yang *tidak* besar:

```
<bool name="isTablet">false</bool>
```

- Sekarang Anda bisa menambahkan blok `if-else` ke metode `onOptionsItemSelected()` di setiap fragmen di SettingsActivity yang memeriksa apakah `isTablet` true. Jika benar, kode Anda bisa mengalihkan tindakan tombol **Naik** ke MainActivity.

Rangkuman

Dalam praktik ini Anda telah belajar:

- Menambahkan setelan tombol alih (`SwitchPreference`) dengan atribut di file XML preferensi, dan menyetel atributnya:
 - `android:defaultValue` : Nilai default setelan.
 - `android:title` : Judul setelan.

- `android:key` : Kunci setelan.
- `android:summary` : Rangkuman setelan.
- Menambahkan aktivitas setelan ke setelan tampilan, dan menambahkan fragmen yang memperluas `PreferenceFragment` untuk setiap setelan tertentu.
 - Menggunakan `getFragmentManager()` untuk menambahkan fragmen ke aktivitas setelan.
 - Menggunakan `addPreferencesFromResource()` di setiap fragmen untuk memuat file XML preferensi yang sesuai untuk fragmen tersebut.
- Menggunakan `intent` untuk menghubungkan item **Settings** di menu opsi ke aktivitas setelan.
- Menyetel nilai default untuk setelan menggunakan `PreferenceManager.setDefaultValues()`.
- Membaca nilai setelan dari `SharedPreferences` menggunakan `PreferenceManager.getDefaultSharedPreferences()`, dan memperoleh setiap nilai setelan menggunakan `.getString` , `.getBoolean` , etc.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Setelan Aplikasi](#)

Ketahui selengkapnya

- Dokumentasi Android Studio:
 - [Panduan Pengguna Android Studio](#)
- Panduan Android API, bagian "Kembangkan":
 - [Setelan \(pengkodean\)](#)
 - [Kelas Preference](#)
 - [PreferenceFragment](#)
 - [Fragmen](#)
 - [SharedPreferences](#)
 - [Menyimpan Rangkaian Nilai-Kunci](#)
 - [Mendukung Ukuran Layar Berbeda](#)
- Spesifikasi Desain Material:
 - [Setelan \(desain\)](#)
- Lainnya:
 - Stack Overflow: [How does one get dimens.xml into Android Studio?](#)
 - Stack Overflow: [Determine if the device is a smartphone or tablet?](#)

11.1A: Mengimplementasikan Penyedia Konten Minimalis

Daftar Isi:

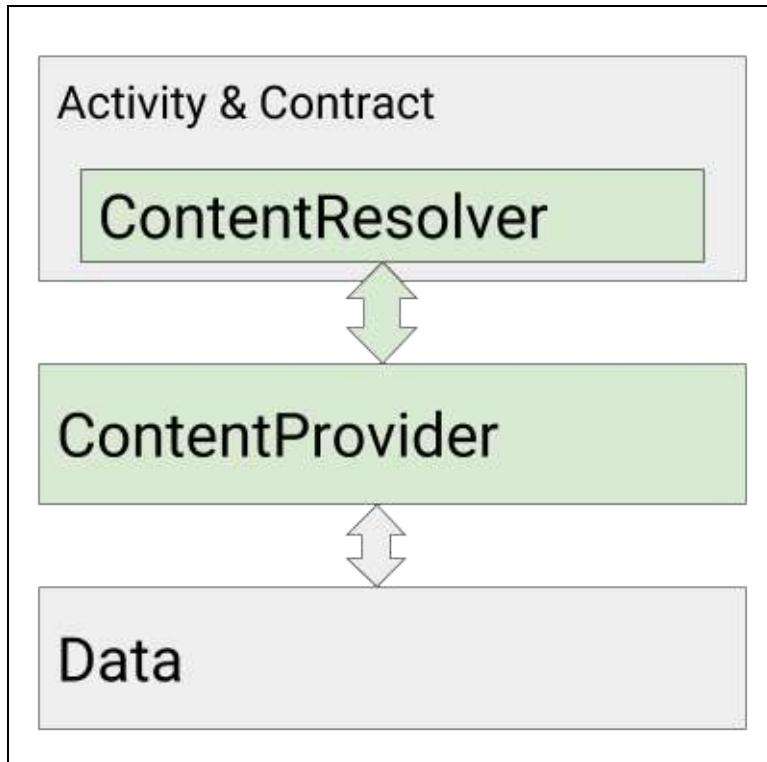
- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Membuat projek MinimalistContentProvider](#)
- [Tugas 2: Membuat kelas Contract, sebuah skema URI, dan data tiruan](#)
- [Tugas 3: Mengimplementasikan kelas MiniContentProvider](#)
- [Tugas 4: Menggunakan ContentResolver untuk mendapatkan data](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Penyedia konten adalah komponen yang secara aman mengelola akses ke repositori data bersama. Penyedia konten menyediakan antarmuka yang konsisten bagi aplikasi untuk mengakses data bersama. Aplikasi tidak mengakses penyedia secara langsung, tetapi menggunakan objek content resolver yang menyediakan antarmuka ke dan mengelola koneksi dengan penyedia konten.

Penyedia konten berguna karena:

- Aplikasi tidak bisa membagikan data di Android—kecuali melalui penyedia konten.
 - Penyedia konten mengizinkan beberapa aplikasi untuk mengakses, menggunakan, dan memodifikasi sumber data tunggal dengan aman. Contoh: Kontak, skor game, kamus pemeriksa ejaan.
 - Anda bisa menetapkan tingkat kontrol akses (izin) untuk penyedia konten Anda.
 - Anda bisa menyimpan data secara terpisah dari aplikasi. Dengan memiliki penyedia konten, Anda bisa mengubah bagaimana data disimpan tanpa perlu mengubah antarmuka pengguna. Misalnya, Anda bisa membangun prototipe menggunakan data tiruan, lalu menggunakan database SQL untuk aplikasi yang nyata. Anda bahkan bisa menyimpan sebagian data Anda di awan dan sebagian data secara lokal, dan antarmuka pengguna tetap sama untuk aplikasi Anda.
 - Arsitektur ini memisahkan data dari antarmuka pengguna, sehingga tim development bisa bekerja secara independen pada aplikasi yang berhadapan dengan klien dan komponen sisi-server aplikasi Anda. Untuk aplikasi yang kompleks dan besar, antarmuka pengguna dan layanan data bisa dikembangkan oleh tim yang berbeda. Mereka bahkan bisa merupakan aplikasi yang terpisah. Bahkan aplikasi dengan penyedia konten tidak harus memiliki antarmuka pengguna.
 - Anda bisa menggunakan CursorLoader dan kelas lain yang diharapkan untuk berinteraksi dengan penyedia konten.
- Catatan:** Jika aplikasi Anda tidak berbagi data dengan aplikasi lain, aplikasi Anda tidak memerlukan penyedia konten. Akan tetapi, karena penyedia konten secara jelas memisahkan implementasi backend Anda dari antarmuka pengguna, penyedia konten juga bisa berguna untuk merencanakan aplikasi yang lebih kompleks.

Diagram berikut merangkum bagian-bagian arsitektur penyedia konten.



Data: Aplikasi yang membuat penyedia konten memiliki data dan menentukan izin apa yang dimiliki aplikasi lain untuk bekerja dengan data.

Data sering kali disimpan dalam database SQLite, tetapi tidak wajib. Umumnya, data disediakan untuk penyedia konten sebagai tabel, mirip seperti tabel database, setiap baris mewakili satu entri dan setiap kolom mewakili atribut untuk entri tersebut. Misalnya, setiap baris dalam database kontak berisi satu entri dan entri tersebut bisa memiliki kolom untuk alamat email dan nomor telepon.

ContentProvider: Penyedia konten menyediakan antarmuka publik yang aman ke data, sehingga aplikasi lain bisa mengakses data dengan izin yang sesuai.

ContentResolver: Digunakan oleh Aktivitas untuk mengirimkan kueri ke penyedia konten. Content resolver mengembalikan data sebagai objek Cursor yang kemudian bisa digunakan, misalnya, oleh sebuah adaptor, untuk menampilkan data.

Kelas Contract (tidak diperlihatkan): Contract adalah kelas publik yang mengekspos informasi penting tentang penyedia konten ke aplikasi lain. Ini biasanya menyertakan URI untuk mengakses data, konstanta penting, dan struktur data yang akan dikembalikan.

Aplikasi mengirimkan permintaan ke penyedia konten menggunakan Uniform Resource Identifier atau URI konten. URI konten untuk penyedia konten memiliki bentuk umum berikut:

`scheme://authority/path-to-data/dataset-name`

- skema (untuk URI konten, ini selalu content://)
- authority (merepresentasikan domain, dan untuk penyedia materi biasanya berakhiran .provider)
- path (merepresentasikan jalur ke data)
- ID (secara unik mengidentifikasi rangkaian data untuk ditelusuri; misalnya nama file atau nama tabel)

URI berikut bisa digunakan untuk meminta semua entri di tabel "words":

```
content://com.android.example.wordcontentprovider.provider/words
```

Merancang skema URI adalah topik tersendiri dan tidak dicakup dalam kursus ini.

Content Resolver: Objek ContentResolver menyediakan metode query(), insert(), update(), dan delete() untuk mengakses data dari penyedia konten dan mengelola semua interaksi dengan penyedia konten untuk Anda. Dalam kebanyakan situasi, Anda cukup menggunakan content resolver default yang disediakan oleh sistem Android.

Dalam praktik ini, Anda akan membangun penyedia konten dasar dari awal. Anda akan membuat dan memproses data tiruan sehingga Anda bisa berfokus pada memahami arsitektur penyedia konten. Dengan demikian, antarmuka pengguna untuk menampilkan data adalah minimal. Dalam praktik selanjutnya, Anda akan menambahkan penyedia konten ke aplikasi WordList, menggunakan aplikasi minimalis ini sebagai template Anda.

Yang harus sudah Anda KETAHUI

Untuk praktik ini, Anda harus sudah memahami cara:

- Membuat, membangun, dan menjalankan aplikasi interaktif di Android Studio.
- Menampilkan data di sebuah RecyclerView menggunakan adaptor.
- Mengabstraksi dan melingkupi data dengan model data.
- Membuat, mengelola, dan berinteraksi dengan database SQLite menggunakan SQLiteOpenHelper.

Yang akan Anda PELAJARI

Anda akan mempelajari:

- Arsitektur dan anatomi penyedia konten.
- Apa yang perlu Anda lakukan untuk membangun penyedia konten minimal yang bisa Anda gunakan sebagai template untuk membuat penyedia konten lainnya.

Yang akan Anda LAKUKAN

- Anda akan membangun aplikasi mandiri untuk mempelajari cara kerja pembangunan penyedia konten.

Ringkasan Aplikasi

- Aplikasi ini menghasilkan data tiruan dan menyimpannya dalam daftar tautan yang bernama "words".
- Aplikasi meminta data melalui content resolver dan menampilkannya. UI terdiri dari satu aktivitas dengan TextView dan dua Tombol. Tombol "List all words" akan menampilkan semua kata dan tombol "List first word" menampilkan kata pertama di TextView.
- Penyedia konten mengabstraksi dan mengelola interaksi antara sumber data dan antarmuka pengguna.
- Contract mendefinisikan URI dan konstanta publik.



Catatan: Versi SDK minimum adalah API15: Android 4.0.3 IceCreamSandwich dan SDK *target* adalah versi Android saat ini (versi 23 saat buku ini ditulis).

Tugas 1. Membuat proyek MinimalistContentProvider

1.1. Buat proyek di dalam batasan yang diberikan

Buat aplikasi dengan satu aktivitas yang hanya menampilkan satu tampilan teks dan dua tombol. Satu tombol menampilkan kata pertama di data (daftar) dan tombol lainnya akan menampilkan semua kata. Kedua tombol memanggil `onClickDisplayEntries()` saat diklik. Untuk saat ini, metode ini akan menggunakan pernyataan switch hanya untuk menampilkan pernyataan bahwa tombol tertentu diklik. Gunakan tabel di bawah sebagai panduan untuk menyetel proyek Anda.

| | |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama aplikasi | MinimalistContentProvider |
| Satu Aktivitas | Template Empty Activity Name: MainActivity <pre>private static final String TAG = MainActivity.class.getSimpleName(); public void onClickDisplayEntries (View view){Log.d (TAG, "Yay, I was clicked!");}</pre> |
| TextView | <code>@+id/textview</code> <code>android:text="response"</code> |
| Button | <code>@+id/button_display_all</code> <code>android:text="List all words"</code> <code>android:onClick="onClickDisplayEntries"</code> |
| Button | <code>@+id/button_display_first</code> <code>android:text="List first word"</code> <code>android:onClick="onClickDisplayEntries"</code> |

1.2. Selesaikan persiapan dasar

Selesaikan persiapan dasar antarmuka pengguna:

1. Di `MainActivity`, buat variabel anggota untuk tampilan teks dan lakukan inisialisasi di `onCreate()`.
2. Di `onClickDisplayEntries()`, gunakan pernyataan switch untuk memeriksa tombol mana yang ditekan. Gunakan view id untuk mengenali tombol. Cetak pernyataan log untuk setiap kasus.
3. Di `onClickDisplayEntries()`, di bagian akhir, tambahkan beberapa teks ke `TextView`.
4. Seperti biasanya, ekstrak sumber daya string.
5. Jalankan aplikasi.

`MainActivity` seharusnya sama dengan solusi ini.

Solusi:

```

package android.example.com.minimalistcontentprovider;

[... imports]

public class MainActivity extends AppCompatActivity {

    private static final String TAG = MainActivity.class.getSimpleName();

    TextView mTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mTextView = (TextView) findViewById(R.id.textview);
    }

    public void onClickDisplayEntries(View view) {
        Log.d(TAG, "Yay, I was clicked!");

        switch (view.getId()) {
            case R.id.button_display_all:
                Log.d(TAG, "Yay, " + R.id.button_display_all + " was clicked!");
                break;
            case R.id.button_display_first:
                Log.d(TAG, "Yay, " + R.id.button_display_first + " was clicked!");
                break;
            default:
                Log.d(TAG, "Error. This should never happen.");
        }
        mTextView.append("Thus we go! \n");
    }
}

```

Tugas 2: Membuat kelas Contract, URI, dan data tiruan

Contract berisi informasi tentang tentang data yang diperlukan aplikasi untuk membangun kueri.

- Contract adalah kelas publik yang menyertakan informasi penting untuk aplikasi lain yang ingin terhubung dengan penyedia konten ini dan mengakses data Anda.
- URI menampilkan cara membangun URI untuk mengakses data. Skema URI berperilaku sebagai API untuk mengakses data. Ini mirip dengan merancang panggilan REST untuk CRUD. Aplikasi lain akan menggunakan URI Konten ini.

2.1. Buat kelas Contract

1. Buat **kelas Java Contract** publik yang baru dengan tanda tangan berikut. Ini harus final.

```
public final class Contract {}
```

2. Untuk mencegah seseorang tidak sengaja membuat instance kelas Contract, berikan constructor pribadi kosong.

```
private Contract() {}
```

2.2 Buat URI

URI konten untuk penyedia konten memiliki bentuk umum berikut:

```
scheme://authority/path/id
```

- **scheme** selalu content:// untuk URI konten.

- **authority** (merepresentasikan domain, dan untuk penyedia konten biasanya berakhiran `.provider`)
- **path** adalah jalur menuju data
- **id** secara unik mengidentifikasi rangkaian data untuk ditelusuri

URI berikut bisa digunakan untuk meminta semua entri di tabel "words":

```
content://com.android.example.wordcontentprovider.provider/words
```

URI untuk mengakses penyedia konten didefinisikan di Contract sehingga hanya tersedia bagi aplikasi yang ingin melakukan kueri pada penyedia konten ini. Biasanya, ini dilakukan dengan mendefinisikan konstanta untuk AUTHORITY, CONTENT_PATH, dan CONTENT_URI

1. Di kelas Contract, buat sebuah konstanta untuk AUTHORITY. Untuk menjadikan Authority unik, gunakan nama paket yang ditambah dengan "provider." `public static final String AUTHORITY = "com.android.example.minimalistcontentprovider.provider";`
2. Buat konstanta untuk CONTENT_PATH. Jalur konten mengidentifikasi data. Anda harus menggunakan sesuatu yang deskriptif, misalnya, nama tabel atau file atau jenis data, misalnya "words".
`public static final String CONTENT_PATH = "words";`
3. Buat konstanta untuk CONTENT_URI. Ini adalah URI yang bergaya content:// yang menunjuk ke satu rangkaian data. Jika Anda memiliki beberapa "data container" di backend, Anda harus membuat URI konten untuk masing-masing data container tersebut.

[URI](<https://developer.android.com/reference/android/net/Uri.html>) adalah kelas helper untuk membangun dan memanipulasi URI. Karena ini tidak pernah berubah, string untuk semua instance kelas Contract, Anda bisa melakukan inisialisasi secara statis. `public static final Uri CONTENT_URI = Uri.parse("content://" + AUTHORITY + "/" + CONTENT_PATH);`

4. Buat konstanta bantu untuk ALL_ITEMS. Ini adalah nama kumpulan data yang akan Anda gunakan saat mengambil semua kata. Nilainya -2 karena ini adalah nilai terendah pertama yang tidak dikembalikan oleh panggilan metode.
`static final int ALL_ITEMS = -2;`
5. Buat konstanta bantu untuk WORD_ID. Ini adalah id yang akan Anda gunakan saat mengambil satu kata. `static final String WORD_ID = "id";`

2.3. Tambahkan Tipe MIME

Penyedia konten menyediakan konten dan Anda perlu menetapkan tipe konten seperti apa yang disediakannya. Aplikasi perlu mengetahui struktur dan format data yang dikembalikan, sehingga bisa menanganinya dengan benar.

Tip MIME adalah tipe/subtipe bentuk, seperti `text/html` untuk laman HTML. Untuk penyedia konten, Anda perlu mendefinisikan tipe MIME khusus vendor untuk jenis data yang dikembalikan oleh penyedia konten Anda. Tipe tipe MIME Android khusus vendor adalah selalu:

- `vnd.android.cursor.item` untuk satu item data (sebuah catatan)
- `vnd.android.cursor.dir` untuk kumpulan s data (beberapa catatan).

Subtipe bisa berupa apa pun, tetapi sebaiknya buat yang informatif. Misalnya:

- vnd—tipe MIME vendor
- com.example—domain
- provider—untuk penyedia konten
- words—nama tabel

Baca [Mengimplementasikan Tipe MIME Penyedia Konten](#) untuk detailnya.

1. Deklarasikan tipe MIME untuk satu item data.

```
static final String SINGLE_RECORD_MIME_TYPE = "vnd.android.cursor.item/vnd.com.example.provider.words";
```

2. Deklarasikan tipe MIME untuk beberapa catatan.

```
static final String MULTIPLE_RECORD_MIME_TYPE = "vnd.android.cursor.dir/vnd.com.example.provider.words";
```

2.4 Buat data tiruan

Penyedia konten selalu menyajikan hasil sebagai [Cursor](#) dalam format tabel yang mirip dengan database SQL. Ini terpisah dari bagaimana data sebenarnya disimpan. Aplikasi ini menggunakan larik string kata.

Dalam strings.xml, tambahkan satu daftar kata yang pendek:

```
<string-array name="words">
    <item>Android</item>
    <item>Activity</item>
    <item>ContentProvider</item>
    <item>ContentResolver</item>
</string-array>
```

Tugas 3: Mengimplementasikan kelas MiniContentProvider

3.1. Buat kelas MiniContentProvider

- Buat **kelas Java** MiniContentProvider yang memperluas ContentProvider. (Untuk praktik ini, **jangan** gunakan opsi menu Create Class > Other > Content Provider.)
- Implementasikan metode (**Code > Implement methods**).
- Tambahkan tag log.
- Tambahkan variabel anggota untuk data tiruan.

```
public String[] mData;
```

- Di onCreate(), inisialisasikan `mData` dari larik kata dan kembalikan true.

```
@Override
public boolean onCreate() {
    Context context = getContext();
    mData = context.getResources().getStringArray(R.array.words);
    return true;
}
```

- Tambahkan pesan log ke metode insert, delete, dan update. Anda tidak akan mengimplementasikan metode ini untuk praktik ini.

```
Log.e(TAG, "Not implemented: update uri: " + uri.toString());
```

3.2. Terbitkan penyedia konten dengan menambahkannya ke manifest Android.

Untuk mengetahui penyedia konten, aplikasi Anda dan aplikasi lain perlu mengetahui bahwa penyedia konten ada. Tambahkan deklarasi untuk penyedia konten ke manifes Android di dalam sebuah .

Deklarasi berisi nama penyedia konten dan otoritas (identifier uniknya).

- Dalam AndroidManifest, di dalam tag aplikasi, setelah tag penutup aktivitas, tambahkan:

```
<provider
    android:name=".MiniContentProvider"
    android:authorities="com.android.example.minimalistcontentprovider.provider" />
```

- Jalankan kode Anda untuk memastikan dikompilasi dengan baik.

3.3. Persiapkan pencocokan URI

ContentProvider perlu merespons permintaan data dari aplikasi menggunakan sejumlah URI yang berbeda. Untuk mengambil tindakan yang sesuai bergantung pada URI permintaan tertentu, penyedia konten harus menganalisis URI untuk melihat apakah cocok. [UriMatcher](#) adalah kelas helper yang bisa Anda gunakan untuk memproses skema URI yang diterima untuk penyedia konten yang diberikan.

Langkah-langkah dasar untuk menggunakan UriMatcher:

- Buat instance UriMatcher
- Tambahkan setiap URI yang dikenali penyedia konten Anda ke UriMatcher.
- Tetapkan konstanta numerik ke setiap URI. Memiliki konstanta numerik untuk setiap URI memudahkan Anda saat Anda memproses URI yang masuk karena Anda bisa menggunakan pernyataan switch/case pada nilai numerik untuk memproses URI.

Buat perubahan berikut di kelas MiniContentProvider.

1. Dalam kelas MiniContentProvider, buat variabel statis pribadi untuk UriMatcher.

Argumen untuk constructor menentukan nilai untuk dikembalikan jika tidak ada kecocokan. Sebagai praktik terbaik, gunakan UriMatcher.NO_MATCH.

```
private static UriMatcher sUriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
```

2. Buat metode Anda sendiri untuk menginisialisasi URI matcher.

```
private void initializeUriMatching(){}
```

3. Panggil initializeUriMatching di onCreate() kelas MiniContentProvider.

4. Dalam metode initializeUriMatching(), tambahkan URI yang diterima penyedia konten Anda ke matcher dan tetapkan sebagai kode integer. Ini adalah URI berdasarkan otoritas dan jalur konten yang ditetapkan di Contract.

Simbol # mencocokkan string karakter numerik dengan panjang berapa saja. Di aplikasi ini, ini mengacu pada indeks kata di larik string. Di aplikasi produksi, ini bisa jadi id entri dalam database. Tetapkan URI ini nilai numerik 1.

```
sUriMatcher.addURI(Contract.AUTHORITY, Contract.CONTENT_PATH + "/#", 1);
```

5. URI kedua adalah yang Anda tentukan di Contract untuk mengembalikan semua item. Tetapkan nilai numerik 0.

```
sUriMatcher.addURI(Contract.AUTHORITY, Contract.CONTENT_PATH, 0);
```

Perhatikan bahwa aplikasi Anda lebih kompleks dan menggunakan lebih banyak URI, menggunakan konstanta bernama untuk kode, seperti yang ditampilkan di [dokumentasi UriMatcher](#).

Solusi:

```
private void initializeUriMatching(){
    sUriMatcher.addURI(Contract.AUTHORITY, Contract.CONTENT_PATH + "/#", 1);
    sUriMatcher.addURI(Contract.AUTHORITY, Contract.CONTENT_PATH, 0);
}
```

3.4 Implementasikan metode getType()

Metode getType() penyedia konten mengembalikan tipe MIME untuk setiap URI yang ditetapkan.

Kecuali Anda melakukan sesuatu yang khusus di kode Anda, implementasi metode ini akan sangat mirip untuk penyedia konten apa pun. Ini melakukan yang berikut:

1. Mencocokkan URI.
2. Mengaktifkan kode yang dikembalikan.
3. Mengembalikan tipe MIME yang sesuai.

Pelajari selengkapnya di [dokumentasi UriMatcher](#).

Solusi:

```
public String getType(Uri uri) {
    switch (sUriMatcher.match(uri)) {
        case 0:
            return Contract.MULTIPLE_RECORD_MIME_TYPE;
        case 1:
            return Contract.SINGLE_RECORD_MIME_TYPE;
        default:
            // Alternatively, throw an exception.
            return null;
    }
}
```

3.5 Implementasikan metode query()

Tujuan metode query() adalah untuk mencocokkan URI, mengonversinya ke mekanisme akses data internal Anda (misalnya, kueri SQLite), mengeksekusi kode akses data internal, dan mengembalikan hasil di objek [Cursor](#).

Metode query()

Metode query memiliki tanda tangan berikut:

```
public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder){}
```

Argumen ke metode ini merepresentasikan bagian kueri SQL. Bahkan jika Anda menggunakan jenis mekanisme storage data lain, Anda masih harus menerima kueri dengan gaya ini dan menangani argumen dengan benar. (Di tugas berikutnya, Anda akan membangun kueri di MainActivity untuk melihat bagaimana argumen digunakan.) Metode mengembalikan Cursor dengan jenis apa pun.

| | |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| uri | URI lengkap. Ini tidak boleh null. |
| projection | Menunjukkan kolom/atribut mana untuk diakses. |
| selection | Menunjukkan baris/catatan objek mana untuk diakses. |
| selectionArgs | Parameter yang melekat pada argumen pilihan sebelumnya. Untuk alasan keamanan, argumen diproses terpisah. |
| sortOrder | Apakah akan mengurutkan, dan jika demikian, apakah naik, turun, atau berdasarkan. Jika ini null, pengurutan default atau tanpa pengurutan akan diterapkan. |

Analisis metode query()

- Identifikasi langkah-langkah pemrosesan berikut di kode metode kueri() yang ditampilkan di bawah di bagian solusi.

Pemrosesan kueri selalu terdiri dari langkah-langkah ini:

- Mencocokkan URI.
- Mengaktifkan kode yang dikembalikan.
- Memproses argumen dan membangun kueri yang sesuai untuk backed.
- Dapatkan data dan (jika perlu) masukkan ke dalam Cursor.

- v. Kembalikan kursor.
2. Identifikasi bagian kode yang perlu diubah di aplikasi dunia nyata.

Implementasi kueri untuk aplikasi dasar ini mengambil beberapa pintaan.

- Penanganan kesalahan minimal.
 - Karena aplikasi menggunakan data tiruan, Cursor bisa diisi secara langsung.
 - Karena skema URI sederhana, metode ini cukup singkat.
3. Identifikasi setidaknya satu keputusan desain yang memudahkan pemahaman dan pengelolaan kode.
- Menganalisis kueri dan mengeksekusinya untuk mengisi kursor dipisahkan ke dalam dua metode.
 - Kode berisi lebih banyak kode daripada kode yang bisa dieksekusi.
4. Tambahkan kode ke aplikasi Anda.

Catatan: Anda akan mendapatkan kesalahan untuk metode populateCursor() dan akan menanganinya langkah berikutnya.

Kode Solusi dengan Anotasi untuk metode query() di MiniContentProvider.java

```

@Nullable
@Override
public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder) {
    int id = -1;
    switch (sUriMatcher.match(uri)) {
        case 0:
            // Matches URI to get all of the entries.
            id = Contract.ALL_ITEMS;
            // Look at the remaining arguments
            // to see whether there are constraints.
            // In this example, we only support getting
            // a specific entry by id. Not full search.
            // For a real-life app, you need error-catching code;
            // here we assume that the
            // value we need is actually in selectionArgs and valid.
            if (selection != null){
                id = parseInt(selectionArgs[0]);
            }
            break;

        case 1:
            // The URI ends in a numeric value, which represents an id.
            // Parse the URI to extract the value of the last,
            // numeric part of the path,
            // and set the id to that value.
            id = parseInt(uri.getLastPathSegment());
            // With a database, you would then use this value and
            // the path to build a query.
            break;

        case UriMatcher.NO_MATCH:
            // You should do some error handling here.
            Log.d(TAG, "NO MATCH FOR THIS URI IN SCHEME.");
            id = -1;
            break;
        default:
            // You should do some error handling here.
            Log.d(TAG, "INVALID URI - URI NOT RECOGNIZED.");
            id = -1;
    }
    Log.d(TAG, "query: " + id);
    return populateCursor(id);
}

```

3.6. Implementasikan metode populateCursor()

Setelah metode query() telah mengidentifikasi URI, metode akan memanggil populateCursor() Anda dengan segmen jalur terakhir, yang merupakan id (indeks) kata yang akan diambil. Metode populateCursor() memisahkan pencocokan kueri dari mendapatkan data dan membuat kursor hasil. Ini adalah praktik terbaik seperti di aplikasi nyata, metode query() bisa menjadi sangat besar.

Metode query harus mengembalikan tipe Cursor, sehingga metode populateCursor() harus membuat, mengisi, dan mengembalikan kursor.

- Jika data Anda disimpan di database SQLite, mengeksekusi kueri akan mengembalikan sebuah Cursor.
- Jika Anda tidak menggunakan metode storage data yang mengembalikan kursor, seperti file atau data tiruan, Anda bisa menggunakan [MatrixCursor](#) untuk menampung data yang akan dikembalikan. MatrixCursor adalah kursor untuk penggunaan umum ke dalam larik objek yang terus bertambah sesuai kebutuhan. Untuk membuat MatrixCursor, Anda harus mengisinya dengan larik string nama kolom.

Metode populateCursor() melakukan yang berikut:

1. Menerima id yang diekstrak dari URI.
2. Membuat MatrixCursor untuk menyimpan data yang diterima (karena data tiruan yang diterima bukan kursor).
3. Membuat dan mengeksekusi kueri. Untuk aplikasi ini, ini akan mengambil string di indeks `id` dari larik string. Dalam aplikasi yang lebih realistik, ini bisa mengeksekusi kueri ke database.
4. Menambahkan hasil ke kursor.
5. Mengembalikan kursor.

```
private Cursor populateCursor(int id) {
    MatrixCursor cursor = new MatrixCursor(new String[] { Contract.CONTENT_PATH });
    // If there is a valid query, execute it and add the result to the cursor.
    if (id == Contract.ALL_ITEMS) {
        for (int i = 0; i < mData.length; i++) {
            String word = mData[i];
            cursor.addRow(new Object[]{word});
        }
    } else if (id >= 0) {
        // Execute the query to get the requested word.
        String word = mData[id];
        // Add the result to the cursor.
        cursor.addRow(new Object[]{word});
    }
    return cursor;
}
```

Tugas 4: Menggunakan ContentResolver untuk mendapatkan data

Setelah mempersiapkan penyedia konten, metode onClickDisplayEntries() di MainActivity bisa diluaskan untuk melakukan kueri dan menampilkan data ke UI. Ini memerlukan langkah-langkah berikut:

1. Buat kueri bergaya SQL, bergantung pada tombol mana yang ditekan.
2. Gunakan content resolver untuk berinteraksi dengan penyedia konten untuk mengeksekusi kueri dan mengembalikan sebuah Cursor.
3. Proses hasilnya di Cursor.

4.1. Dapatkan content resolver

Content resolver berinteraksi dengan penyedia konten atas nama Anda.

Content resolver mengharapkan URI Content yang di-parse bersama dengan parameter kueri yang membantu mengambil data.

Anda tidak harus membuat content resolver sendiri. Anda bisa menggunakan yang disediakan di konteks aplikasi Anda oleh framework Android dengan memanggil `getContentResolver()`.

1. Di MainActivity, hapus semua kode dari dalam onClickDisplayEntries().
2. Tambahkan kode ini ke onClickDisplayEntries() di MainActivity.

```
Cursor cursor = getContentResolver().query(Uri.parse(queryUri), projection, selectionClause, selectionArgs, sortOrder);
```

Perhatikan: argumen ke getContentResolver.query() sama dengan parameter ContentProvider.query().

Selanjutnya, Anda harus mendefinisikan argumen ke getContentResolver.query().

4.2. Definisikan argumen kueri

Agar getContentResolver.query() bisa berfungsi, Anda harus mendeklarasikan dan menetapkan nilai ke semua argumennya.

1. **URI:** Deklarasikan ContentURI yang mengidentifikasi penyedia konten dan tabel. Dapatkan informasi untuk URI yang benar dari contract.

```
String queryUri = Contract.CONTENT_URI.toString();
```

2. **Projection:** Sebuah larik string dengan nama kolom untuk dikembalikan. Menyetel ini ke null akan mengembalikan semua kolom. Jika hanya ada satu kolom, seperti di contoh ini, menyetel ini secara eksplisit adalah opsional, tetapi bisa berguna untuk tujuan dokumentasi.

```
// Only get words. String[] projection = new String[] {Contract.CONTENT_PATH};
```
3. **selectionClause:** Klaus argumen untuk kriteria pilihan, yaitu, baris mana yang akan dikembalikan. Diformat sebagai klaus SQL WHERE (mengecualikan kata kunci "WHERE"). Meneruskan null akan mengembalikan semua baris untuk tampilan URI yang diberikan. Karena ini akan berbeda sesuai dengan tombol mana yang ditekan, deklarasikan ini sekarang dan setel nanti.

```
String selectionClause;
```

4. **selectionArgs:** Nilai argumen untuk kriteria pilihan. Jika Anda menyertakan ? di String selection, ? akan digantikan oleh nilai dari selectionArgs, dengan urutan munculnya.

PENTING: Praktik terbaik keamanan adalah selalu memisahkan selection dan selectionArgs.

```
String selectionArgs[];
```

5. **sortOrder:** Urutan untuk mengurutkan hasil. Diformat sebagai klaus SQL ORDER BY (mengecualikan kata kunci ORDER BY). Biasanya ASC atau DESC; null meminta tata urutan default, yang bisa jadi tidak berurutan.

```
// For this example, accept the order returned by the response.
String sortOrder = null;
```

4.3. Putuskan kriteria selection

Nilai selectionClause dan selectionArgs bergantung pada tombol mana yang ditekan di UI.

- Untuk menampilkan semua kata, setel kedua argumen ke null.
 - Untuk mendapatkan kata pertama, kueri kata dengan ID 0. (Ini mengasumsikan bahwa ID kata dimulai dari 0 dan dibuat sesuai urutan. Anda mengetahui ini, karena informasinya diperlihatkan di contract. Untuk penyedia konten yang berbeda, sebaiknya ketahui id-nya dan telusuri dengan cara yang berbeda.)
1. Ganti blok switch yang sudah ada dengan kode berikut ini di onClickDisplayEntries, sebelum Anda mendapatkan content resolver.

```

switch (view.getId()) {
    case R.id.button_display_all:
        selectionClause = null;
        selectionArgs = null;
        break;
    case R.id.button_display_first:
        selectionClause = Contract.WORD_ID + " = ?";
        selectionArgs = new String[] {"0"};
        break;
    default:
        selectionClause = null;
        selectionArgs = null;
}

```

4.4. Proses Cursor

Setelah mendapatkan content resolver, Anda harus memproses hasil dari Cursor.

- Jika ada data, tampilkan di TextView.
- Jika tidak ada data, laporan kesalahan.

1. Periksa kode berikut dan pastikan Anda memahami semuanya.

```

if (cursor != null) {
    if (cursor.getCount() > 0) {
        cursor.moveToFirst();
        int columnIndex = cursor.getColumnIndex(projection[0]);
        do {
            String word = cursor.getString(columnIndex);
            mTextView.append(word + "\n");
        } while (cursor.moveToNext());
    } else {
        Log.d(TAG, "onClickDisplayEntries " + "No data returned.");
        mTextView.append("No data returned." + "\n");
    }
    cursor.close();
} else {
    Log.d(TAG, "onClickDisplayEntries " + "Cursor is null.");
    mTextView.append("Cursor is null." + "\n");
}

```

2. Sisipkan kode ini di akhir onClickDisplayEntry().
3. Jalankan aplikasi Anda.
4. Klik tombol untuk melihat data yang diambil di TextView.

Kode solusi

Proyek Android Studio: [MinimalistContentProvider](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

Mengimplementasikan metode yang tidak ada

Tantangan Penyusunan Kode 1: Implementasikan metode insert, delete, dan update untuk aplikasi MinimalistContentProvider. Berikan cara kepada pengguna untuk menyisipkan, menghapus, dan memperbarui data.

Petunjuk: Jika Anda tidak ingin membangun antarmuka pengguna, buat tombol untuk setiap tindakan dan mengaitkan data yang disisipkan, diperbarui, dan dihapus. Latihan ini ditujukan untuk bekerja dengan penyedia konten, bukan antarmuka pengguna.

Mengapa: Anda akan mengimplementasikan penyedia konten yang sepenuhnya berfungsi dengan UI di praktik berikutnya, ketika Anda menambahkan penyedia konten ke aplikasi WordListSQL.

Menambahkan Pengujian Unit untuk penyedia konten

Tantangan Penyusunan Kode 2: Setelah Anda mengimplementasikan penyedia konten, Anda tidak memiliki cara untuk mengetahui apakah kode bekerja atau tidak. Dalam contoh ini, Anda membangun front-end dan dengan memperhatikan cara kerjanya, Anda mengasumsikan aplikasi bekerja dengan benar. Dalam aplikasi dunia nyata, ini saja tidak cukup dan Anda mungkin tidak memiliki akses ke front-end. Cara yang sesuai untuk menentukan apakah setiap metode berfungsi sesuai harapan, tuliskan pengujian unit untuk MiniContentProvider.

Rangkuman

Pada bab ini, Anda telah mempelajari

- Penyedia konten adalah abstraksi data tingkat tinggi yang mengelola akses ke repositori bersama
- Penyedia konten utamanya ditujukan untuk digunakan oleh aplikasi selain aplikasi Anda sendiri.
- Penyedia konten (sisi-server) diakses oleh Content Resolver (sisi-aplikasi)
- Contract adalah kelas publik yang mengekspos informasi penting tentang penyedia konten.
- Contract bisa berguna di luar penyedia konten.
- Penyedia konten perlu mendefinisikan serangkaian URI konten sehingga aplikasi bisa mengakses data melalui penyedia konten Anda.
- URI konten terdiri dari beberapa komponen: "content://", sebuah otoritas konten yang unik (biasanya berupa nama paket yang sepenuhnya berkualifikasi) dan content-path.
- Gunakan content resolver untuk meminta data dari penyedia konten dan menampilkannya kepada pengguna.
- Jika aplikasi Anda tidak berbagi data dengan aplikasi lain, aplikasi Anda tidak memerlukan penyedia konten.
- Penyedia konten harus mengimplementasikan metode `getType()` yang mengembalikan tipe MIME untuk setiap tipe konten.
- Penyedia konten perlu "diterbitkan" di manifes Android menggunakan elemen `provider`.
- Penyedia konten perlu memeriksa URI masuk untuk menentukan kecocokan pola URI untuk mengakses data apa pun.
- Anda harus menambahkan pola URI target ke penyedia konten Anda. Kelas UriMatcher adalah kelas yang berguna untuk tujuan ini.
- Inti penyedia konten diimplementasikan di metode `query()`-nya.
- Tanda tangan metode dari metode `query()` dalam content resolver (peminta data) cocok dengan tanda tangan metode dari metode `query()` dalam penyedia konten (sumber data).
- Metode `query()` mengembalikan objek cursor bergaya database dengan tidak mempertimbangkan apakah data bersifat relasional atau tidak.

[Konsep terkait]

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Penyedia Konten](#)

Ketahui selengkapnya

Dokumentasi Developer

- [Uniform Resource Identifier atau URI](#)

- [Tipe MIME](#)
- [MatrixCursor dan Cursor](#)
- [Penyedia Konten](#)

11.1B: Menambahkan penyedia konten ke database Anda.

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1. Mengunduh dan menjalankan kode dasar](#)
- [Tugas 2: Menambahkan kelas Contract ke WordListSQLInteractive](#)
- [Tugas 3: Membuat Penyedia Konten](#)
- [\[Tugas 4. Mengimplementasikan Metode Penyedia Konten\]](#)
- [Tantangan penyusunan kode](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Penyedia konten di aplikasi nyata lebih kompleks dari versi dasar yang Anda bangun di praktik sebelumnya.

Di dunia nyata:

- Backend adalah database, sistem file, atau opsi storage persisten lainnya.
- Front-end menampilkan data pada UI yang nyaman dilihat dan memungkinkan pengguna mengubah data.

Anda akan jarang membuat aplikasi dari awal. Sering kali, Anda akan melakukan debug, mengoptimalkan, atau memperluas aplikasi yang sudah ada.

Dalam praktik ini, Anda akan memeriksa aplikasi WordListSQL dan memperluasnya untuk menggunakan penyedia konten sebagai layer antara database SQL dan RecyclerView.

Anda bisa melakukannya dengan dua cara.

- Mengoptimalkan dan memperluas aplikasi WordListSQL. Ini akan mencakup mengubah arsitektur aplikasi dan mengoptimalkan kode.
- Mulai dari awal dan gunakan kembali kode dari WordListSQL dan MinimalistContentProvider.

Praktik ini akan memperagakan cara mengoptimalkan aplikasi WordListSQL yang sudah ada karena ini yang kemungkinan besar Anda temui di pekerjaan.

Yang harus sudah Anda KETAHUI

Untuk praktik ini, Anda harus sudah mengenal cara:

- Menampilkan data di RecyclerView.
- Memulai dan mengembalikan dari Aktivitas.
- Membuat, mengubah, dan berinteraksi dengan database SQLite menggunakan SQLiteOpenHelper.
- Memahami arsitektur penyedia konten minimal yang sudah Anda bangun di praktik sebelumnya.

Yang akan Anda PELAJARI

Anda akan mempelajari cara:

- Membuat penyedia konten yang sepenuhnya dikembangkan untuk aplikasi yang sudah ada.
- Mengoptimalkan aplikasi untuk mengakomodasi penyedia konten.

Yang akan Anda LAKUKAN

Praktik ini memerlukan persiapan yang lebih mirip dengan development aplikasi di dunia nyata.

Mulai dengan aplikasi WordListSQLInteractive yang sudah dibuat di praktik sebelumnya, yang menampilkan kata dari database SQLite di RecyclerView, dan pengguna bisa membuat, mengedit, dan menghapus kata.

Anda akan memperluas dan memodifikasi aplikasi ini:

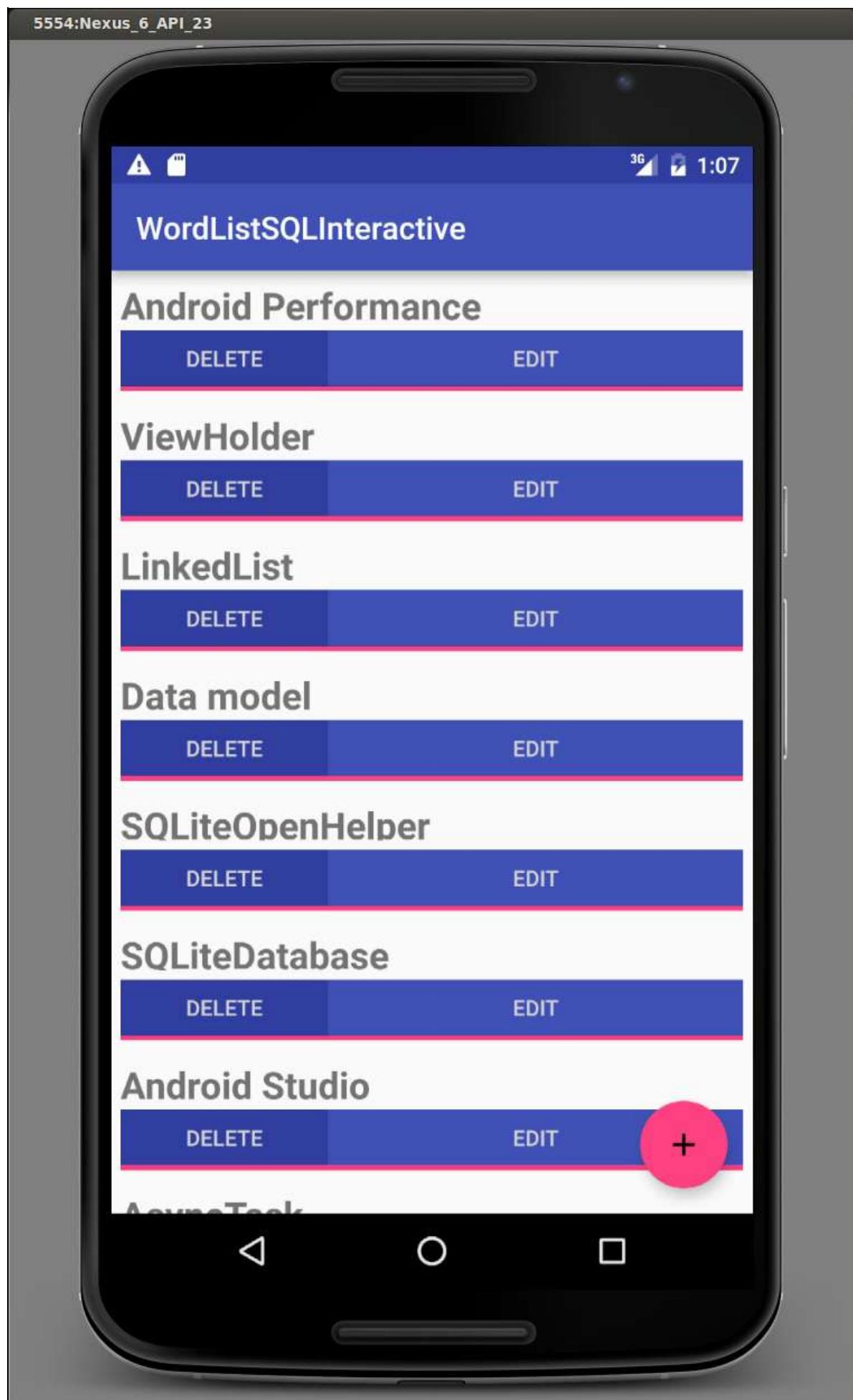
- Mengimplementasikan kelas Contract untuk mengekspos antarmuka aplikasi anda ke aplikasi lain.
- Mengimplementasikan ContentProvider dan melakukan kueri menggunakan ContentResolver.
- Mengoptimalkan kelas MainActivity, WordListAdapter, dan WordListOpenHelper untuk bekerja dengan penyedia konten.

Ringkasan Aplikasi

Aplikasi WordListSQLWithContentProvider akan memiliki fitur berikut:

- Penyedia konten yang bisa menyisipkan, menghapus, memperbarui, dan melakukan kueri pada database.
- Sebuah Contract dan rangkaian izin yang memungkinkan aplikasi lain untuk mengakses penyedia konten ini.
- Resolver konten yang berinteraksi dengan penyedia konten untuk menyisipkan, menghapus, memperbarui, dan melakukan kueri data.
- Antarmuka pengguna dan fungsionalitas yang tidak berubah.

Aplikasi Anda akan terlihat sama seperti di akhir praktik storage data.



Ringkasan komponen aplikasi

Diagram berikut menampilkan ringkasan komponen aplikasi yang menggunakan SQLiteDatabase dengan penyedia konten. Satu-satunya perbedaan dari aplikasi penyedia konten minimal adalah penyedia konten mengambil data dari database melalui helper yang terbuka.

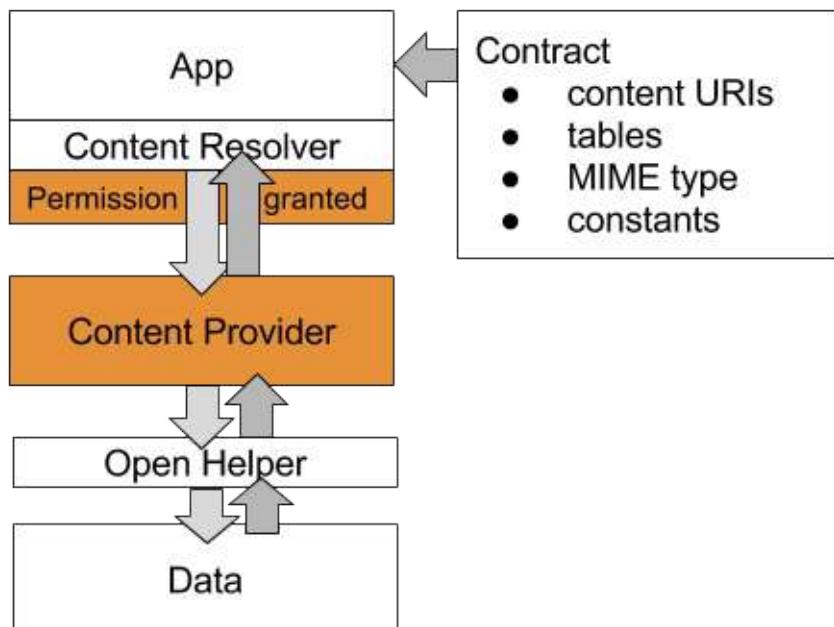
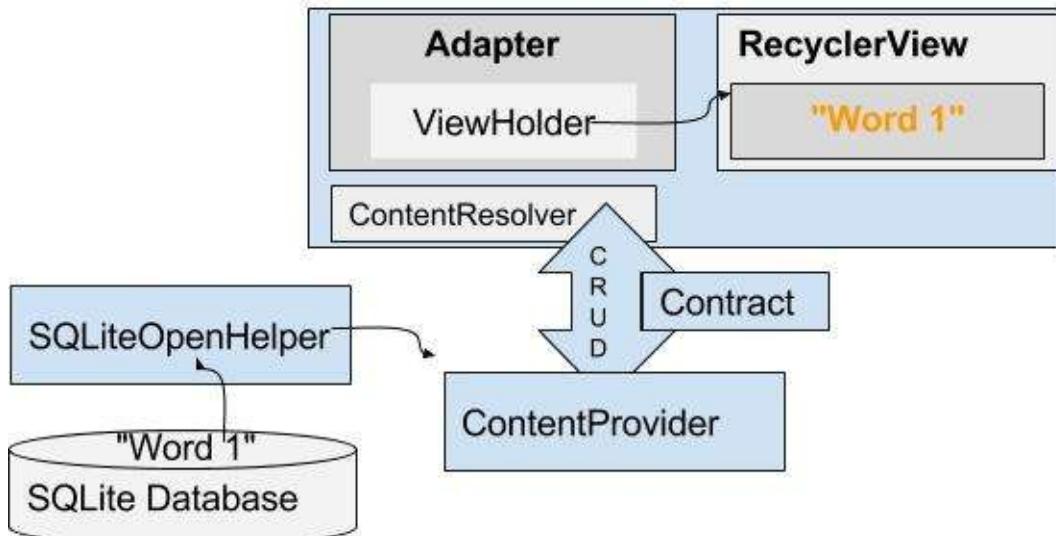


Diagram berikut menampilkan arsitektur aplikasi WordListSQLInteractive dengan penyedia konten yang ditambahkan; ini adalah aplikasi WordListSQLWithContentProvider yang akan Anda bangun di praktik ini.



Lihat bab konsep untuk penjelasan yang detail mengenai semua komponen dan cara interaksinya.

Ringkasan perubahan

Berikut adalah ringkasan perubahan yang akan Anda buat pada WordListInteractive untuk menambahkan penyedia konten.

- Kelas Baru: **Contract**, **ContentProvider**, **ContentResolver**
- Kelas yang berubah: **WordListOpenHelper**, **MainActivity**, **WordListAdapter**
- Kelas yang harus tidak berubah: **WordItem**, **MyButtonOnClickListener**, **ViewHolder**

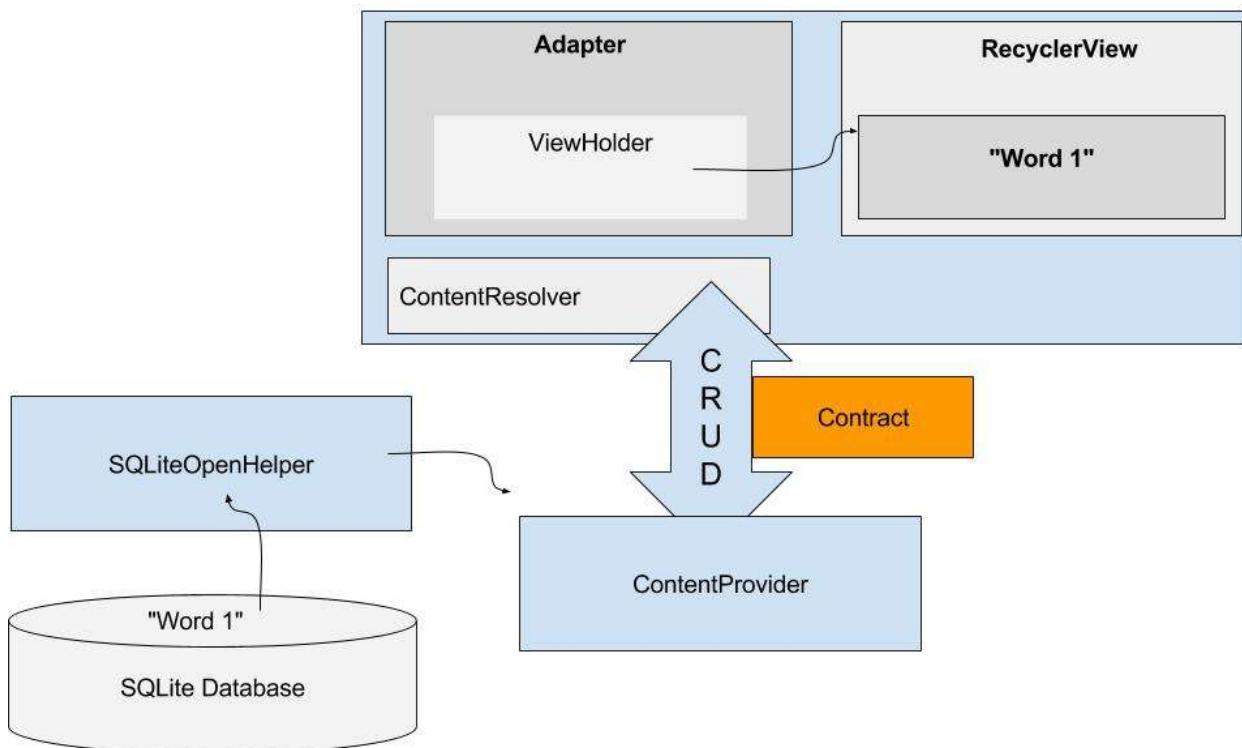
Tugas 1. Mengunduh dan menjalankan kode dasar

Praktik ini melanjutkan aplikasi WordListSQLInteractive dan MinimalistContentProvider yang telah Anda buat sebelumnya. Anda akan memperluas salin WordListSQLInteractive. Anda bisa memulai dari kode Anda sendiri atau mengunduh aplikasi.

- [WordListSQLInteractive](#)
- [MinimalistContentProvider](#)
- Buat salinan WordListSQLInteractive dan muat ke Android Studio.
- Ubah nama paket menjadi `wordlistsqlwithcontentprovider`.

Tugas 2. Menambahkan kelas Contract ke WordListSQLInteractive

Mulai dengan membuat kelas Contract yang mendefinisikan konstanta database publik, konstanta URI, dan tipe MIME. Anda akan menggunakan konstanta ini di semua kelas lainnya.



2.1 Tambahkan kelas Contract

1. Pelajari [dokumentasi Mendefinisikan Skema dan Kontrak](#).
2. Tambahkan kelas final publik ke proyek Anda dan beri nama Contract.

Kelas Contract ini berisi semua informasi yang diperlukan aplikasi lain untuk menggunakan penyedia konten aplikasi Anda. Anda bisa memberi nama apa pun pada kelas ini, tetapi umumnya disebut sebagai "Contract".

```
public final class Contract {}
```

3. Agar kelas Contract tidak dibuat instance-nya, tambahkan constructor pribadi yang kosong.

Ini adalah pola standar untuk kelas yang digunakan untuk menampung informasi meta dan konstanta untuk aplikasi.

```
private Contract() {}
```

2.2 Pindahkan konstanta database ke dalam Contract

Pindahkan konstanta untuk database sehingga aplikasi lain perlu mengetahui di luar WordListOpenHelper ke dalam Contract dan membuatnya publik.

1. Pindahkan DATABASE_NAME dan buat menjadi publik.

```
public static final String DATABASE_NAME = "wordlist";
```

Buat kelas dalam abstrak statis untuk setiap tabel dengan nama kolom. Kelas dalam yang baru ini biasanya mengimplementasikan antarmuka `BaseColumns`. Dengan mengimplementasikan antarmuka `BaseColumns`, kelas Anda bisa mewarisi bidang kunci utama yang disebut sebagai `_ID` yang diharapkan ada oleh beberapa kelas Android, seperti adaptor kursor. Kelas dalam ini tidak diperlukan, tetapi bisa membantu database Anda bekerja baik dengan framework Android.

2. Buat kelas dalam WordList yang mengimplementasikan `BaseColumns`.

```
public static abstract class WordList implements BaseColumns {  
}
```

3. Pindahkan nama `WORD_LIST_TABLE`, serta nama kolom `KEY_ID` dan `KEY_WORD` dari WordListOpenHelper ke dalam kelas WordList di Contract dan buat menjadi publik.
4. Kembali ke WordListOpenHelper dan tunggu agar Android Studio mengimpor konstanta dari Contract atau mengimpornya secara manual, jika Anda belum menyiapkan impor otomatis.

Gunakan **File > Settings > Editor > General > Auto Import** di Windows/Linux atau **Android Studio > Preferences >Editor >General > Auto Import** di Mac untuk mengonfigurasi impor otomatis.)

2.3 Definsikan Konstanta URI

1. Deklarasikan skema URI untuk penyedia konten Anda.

Menggunakan Contract di MinimalistContentProvider sebagai contoh, deklarasikan AUTHORITY, CONTENT_PATH. Tambahkan CONTENT_PATH_URI untuk mengembalikan semua item dan ROW_COUNT_URI yang mengembalikan nomor entri. Di AUTHORITY, gunakan nama aplikasi Anda.

```
public static final int ALL_ITEMS = -2;  
public static final String COUNT = "count";  
  
public static final String AUTHORITY =  
        "com.android.example.wordlistsqllwithcontentprovider.provider";  
  
public static final String CONTENT_PATH = "words";  
  
public static final Uri CONTENT_URI =  
        Uri.parse("content://" + AUTHORITY + "/" + CONTENT_PATH);  
public static final Uri ROW_COUNT_URI =  
        Uri.parse("content://" + AUTHORITY + "/" + CONTENT_PATH + "/" + COUNT);
```

2.4 Deklarasikan tipe MIME

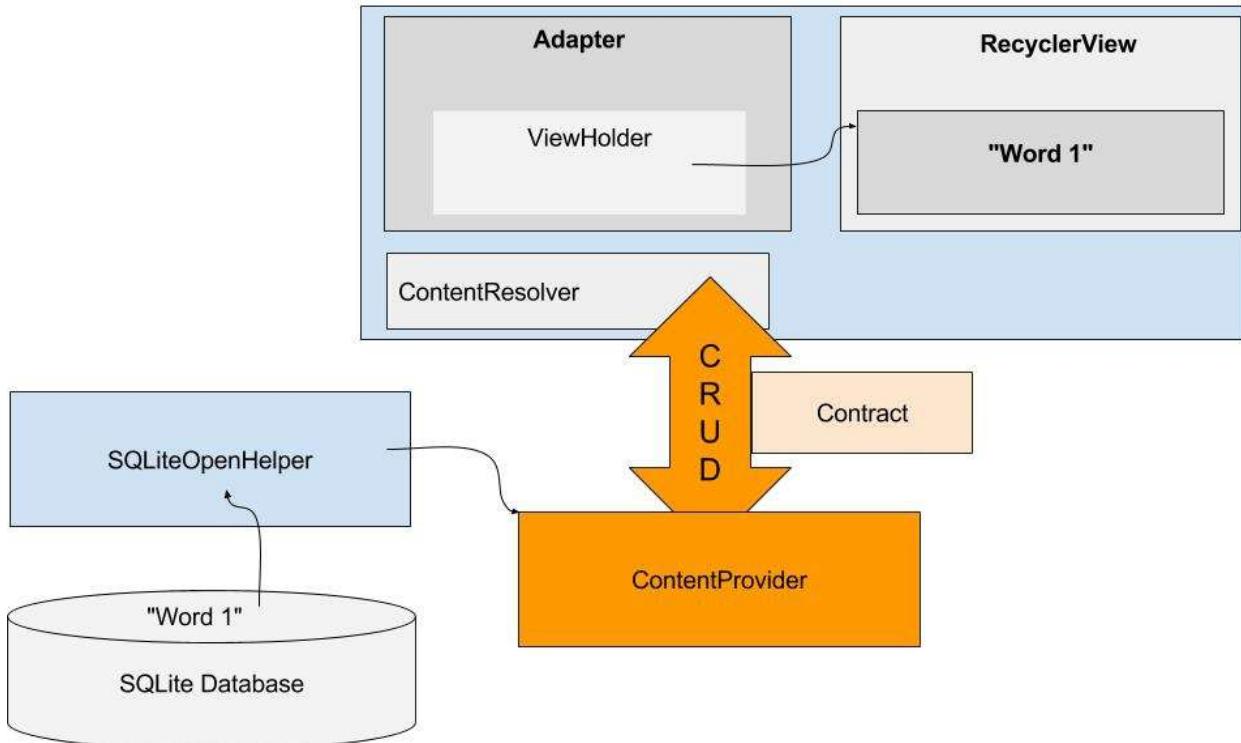
Tipe MIME menjelaskan tipe dan format data. Tipe MIME digunakan untuk memproses data dengan benar. Tipe MIME umum misalnya `text/html` untuk laman web dan `application/json`. Baca selengkapnya tentang [tipe MIME untuk penyedia konten](#) di dokumentasi Android.

1. Deklarasikan tipe MIME untuk satu atau beberapa respons catatan:

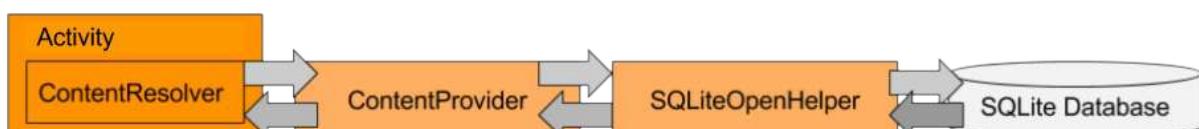
```
static final String SINGLE_RECORD_MIME_TYPE =
    "vnd.android.cursor.item/vnd.com.example.provider.words";
static final String MULTIPLE_RECORDS_MIME_TYPE =
    "vnd.android.cursor.item/vnd.com.example.provider.words";
```

- Jalankan aplikasi Anda. Aplikasi harus berjalan dan tampak dan bertindak persis seperti sebelum Anda mengubahnya.

Tugas 3. Membuat Penyedia Konten



Dalam tugas ini, Anda akan membuat penyedia konten, mengimplementasikan metode kuerinya, dan mengaitkannya dengan WordListAdapter dan WordListOpenHelper. Sebagai ganti melakukan kueri pada WordListOpenHelper, WordListAdapter akan menggunakan resolver konten untuk melakukan kueri pada penyedia konten, yang pada akhirnya akan melakukan kueri pada WordListOpenHelper yang akan melakukan kueri pada database.



3.1 Buat kelas WordListContentProvider

- Buat kelas baru yang memperluas ContentProvider dan beri nama WordListContentProvider.
- Di Android Studio, klik bola lampu merah, pilih "Implement methods", dan klik OK untuk mengimplementasikan semua metode yang dicantumkan.
- Tentukan TAG log.
- Deklarasikan UriMatcher.

Penyedia konten ini menggunakan [UriMatcher](#), sebuah kelas utilitas yang memetakan URI ke angka, sehingga Anda bisa menggunakannya.

```
private static UriMatcher sUriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
```

5. Deklarasikan variabel kelas WordListOpenHelper, mDB.

```
private WordListOpenHelper mDB;
```

6. Deklarasikan kode untuk URI matcher sebagai konstanta.

Ini akan menyatukan kode di satu tempat dan memudahkannya diubah. Gunakan kelipatan sepuluh, sehingga menyisipkan kode tambahan bisa langsung dilakukan.

```
private static final int URI_ALL_ITEMS_CODE = 10;
private static final int URI_ONE_ITEM_CODE = 20;
private static final int URI_COUNT_CODE = 30;
```

7. Ubah metode onCreate() untuk

- menginisialisasi mDB dengan WordListOpenHelper,
- memanggil metode initializeUriMatching() yang akan Anda buat nanti,
- dan mengembalikan true.

```
@Override
public boolean onCreate() {
    mDB = new WordListOpenHelper(getContext());
    initializeUriMatching();
    return true;
}
```

8. Buat metode void pribadi initializeUriMatching().

9. Di initializeUriMatching(), tambahkan URI ke matcher untuk mendapatkan semua item, satu item, dan jumlahnya.

Periksa Contract dan gunakan metode initializeUriMatching() di aplikasi MinimalistContentProver sebagai template.

Solusi:

```
private void initializeUriMatching(){
    sUriMatcher.addURI(Contract.AUTHORITY, Contract.CONTENT_PATH, URI_ALL_ITEMS_CODE);
    sUriMatcher.addURI(Contract.AUTHORITY, Contract.CONTENT_PATH + "/#", URI_ONE_ITEM_CODE);
    sUriMatcher.addURI(Contract.AUTHORITY, Contract.CONTENT_PATH + "/" + Contract.COUNT, URI_COUNT_CODE );
}
```

3.2 Implementasikan WordListContentProvider.query()

Gunakan MiniContentProvider sebagai template untuk mengimplementasikan metode query().

1. Modifikasi WordListContentProvider.query().
2. Gunakan pernyataan Switch untuk kode yang dikembalikan oleh sUriMatcher.
3. Untuk URI_ALL_ITEMS_CODE, URI_ONE_ITEM_CODE, URI_COUNT_CODE, panggil yang sesuai di WordListOpenHelper (mDB).

Perhatikan bagaimana menetapkan hasil dari mDB.query() ke kursor, akan menghasilkan kesalahan, karena WordListOpenHelper.query() mengembalikan WordItem.

Perhatikan bagaimana menetapkan hasil dari mDB.count() ke kursor, akan menghasilkan kesalahan, karena WordListOpenHelper.count() mengembalikan sebuah long.

Anda akan memperbaiki kedua kesalahan ini nanti.

Solusi:

```

@NoArgsConstructor
@Override
public Cursor query(Uri uri, String[] projection, String selection,
    String[] selectionArgs, String sortOrder) {

    Cursor cursor = null;

    switch (suriMatcher.match(uri)) {
        case URI_ALL_ITEMS_CODE:
            cursor = mDB.query(ALL_ITEMS);
            break;

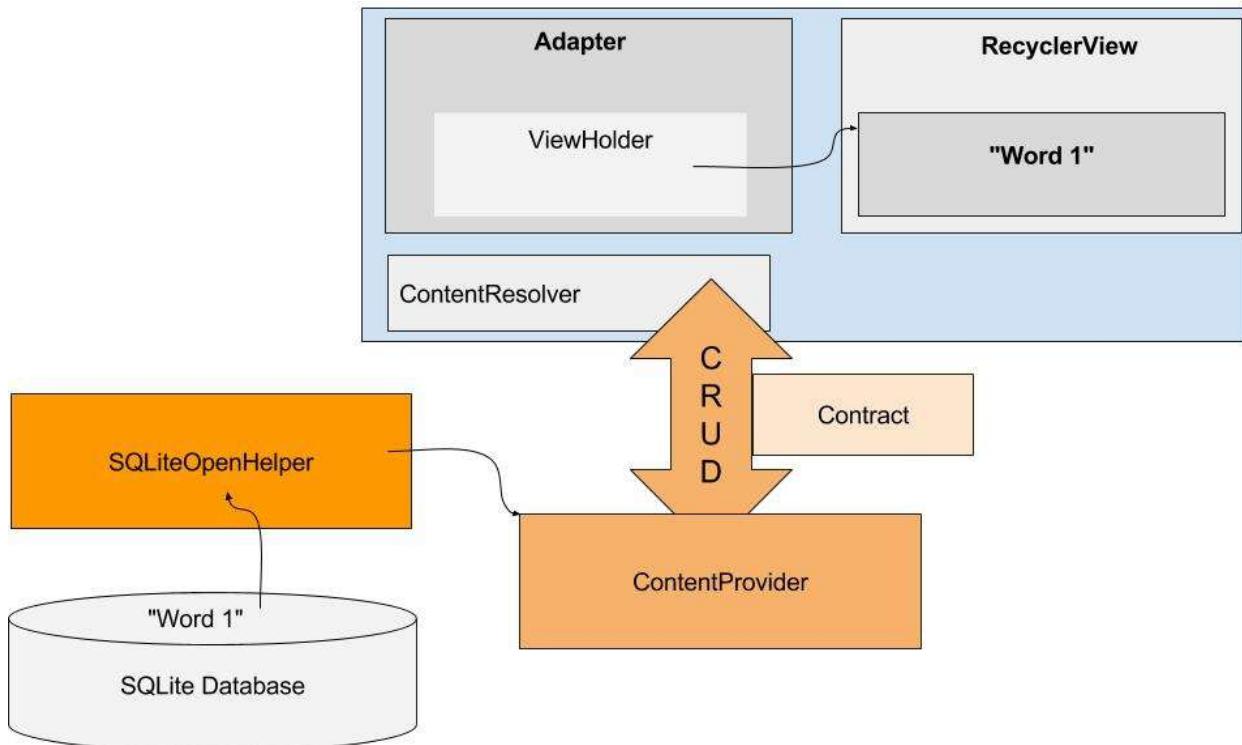
        case URI_ONE_ITEM_CODE:
            cursor = mDB.query(parseInt(uri.getLastPathSegment()));
            break;

        case URI_COUNT_CODE:
            cursor = mDB.count();
            break;

        case UriMatcher.NO_MATCH:
            // You should do some error handling here.
            Log.d(TAG, "NO MATCH FOR THIS URI IN SCHEME: " + uri);
            break;
        default:
            // You should do some error handling here.
            Log.d(TAG, "INVALID URI - URI NOT RECOGNIZED: " + uri);
    }
    return cursor;
}

```

3.3 Perbaiki WordListOpenHelper.query() untuk mengembalikan Cursor dan menangani pengembalian semua item.



Karena penyedia konten bekerja dengan kursor, Anda bisa menyederhanakan metode WordListOpenHelper.query() untuk mengembalikan kursor.

1. Tambahkan kode dengan sebuah kueri untuk mengembalikan semua item dari database untuk menangani kasus `cursor = mDB.query(ALL_ITEMS)` dari pernyataan Switch di atas.

2. Sederhanakan WordListOpenHelper.query() agar mengembalikan kursor.

Ini memperbaiki kesalahan di WordListContentProvider.query().

Akan tetapi, ini merusak WordListOpenHelper.OnBindViewHolder(), yang mengharapkan WordItem dari WordListOpenHelper.

Untuk mengatasi ini, WordListContentProvider.onBindViewHolder() harus menggunakan resolver konten, sebagai ganti memanggil database secara langsung, yang akan Anda lakukan setelah memperbaiki WordListContentProvider.count().

Catatan: Jenis kesalahan dan perbaikan berjenjang seperti ini umum ditemui saat mengerjakan aplikasi dunia nyata. Jika aplikasi yang sedang Anda kerjakan dirancang dengan baik, Anda bisa memperbaiki kesalahan satu demi satu.

Solusi:

```
/**
 * Queries the database for an entry at a given position.
 *
 * @param position The Nth row in the table.
 * @return a WordItem with the requested database entry.
 */
public Cursor query(int position) {
    String query;
    if (position != ALL_ITEMS) {
        position++; // Because database starts counting at 1.
        query = "SELECT " + KEY_ID + "," + KEY_WORD + " FROM "
            + WORD_LIST_TABLE
            +" WHERE " + KEY_ID + "=" + position + ";";
    } else {
        query = "SELECT * FROM " + WORD_LIST_TABLE
            + " ORDER BY " + KEY_WORD + " ASC ";
    }

    Cursor cursor = null;
    try {
        if (mReadableDB == null) {
            mReadableDB = this.getReadableDatabase();
        }
        cursor = mReadableDB.rawQuery(query, null);
    } catch (Exception e) {
        Log.d(TAG, "QUERY EXCEPTION! " + e);
    } finally {
        return cursor;
    }
}
```

3.4 Perbaiki WordListOpenHelper.count() untuk mengembalikan sebuah Cursor

Karena penyedia konten bekerja dengan kursor, Anda juga harus mengubah metode WordListOpenHelper.count() untuk mengembalikan kursor.

Gunakan [MatrixCursor](#) yang merupakan kursor dari baris dan kolom yang bisa diubah.

1. Buat MatrixCursor menggunakan Contract.CONTENT_PATH.
2. Di dalam blok try, dapatkan jumlah dan tambahkan sebagai baris ke kursor.
3. Kembalikan kursor.

Solusi:

```

public Cursor count(){
    MatrixCursor cursor = new MatrixCursor(new String[] {Contract.CONTENT_PATH});
    try {
        if (mReadableDB == null) {
            mReadableDB = getReadableDatabase();
        }
        int count = (int) Databaseutils.queryNumEntries(mReadableDB, WORD_LIST_TABLE);
        cursor.addRow(new Object[]{count});
    } catch (Exception e) {
        Log.d(TAG, "EXCEPTION " + e);
    }
    return cursor;
}

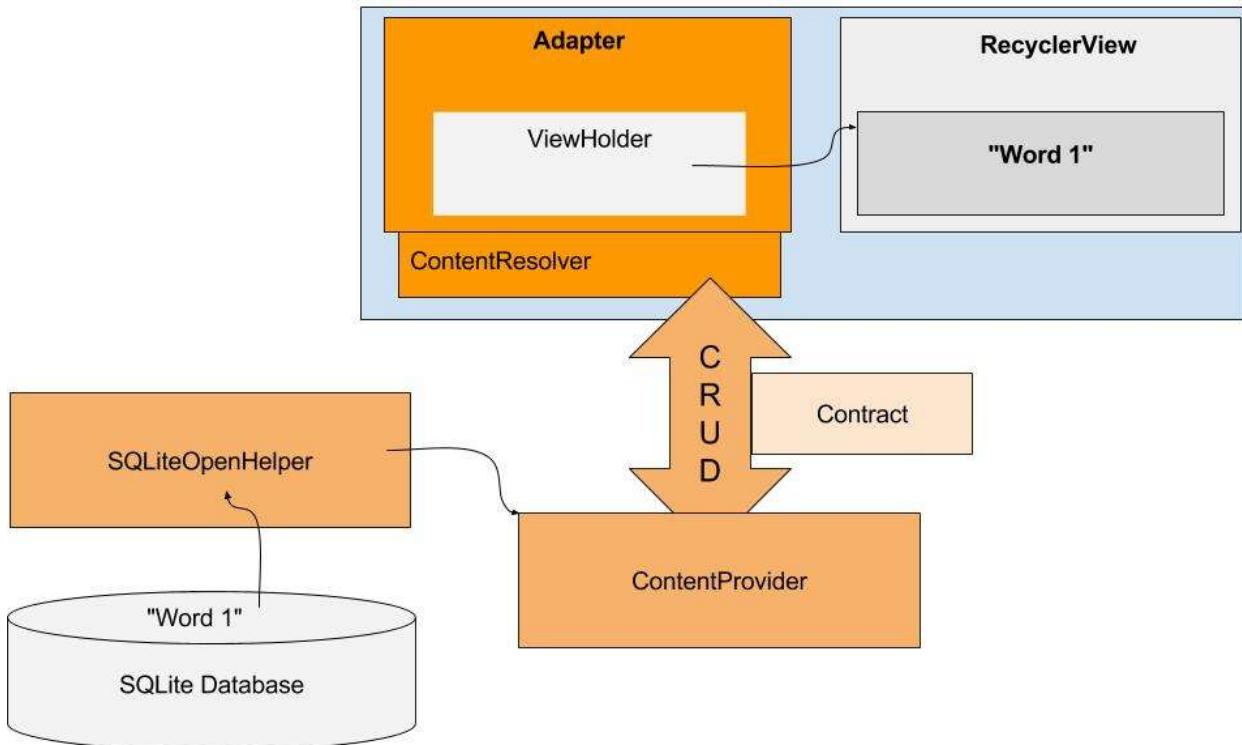
```

Ini akan memperbaiki kesalahan di WordListContentProvider.count(), tetapi merusak WordListAdapter.getItemCount(), yang mengharapkan sebuah long dari WordListOpenHelper.

Di WordListAdapter.onBindViewHolder(), sebagai ganti memanggil database secara langsung, Anda harus menggunakan resolver konten, yang akan Anda lakukan nanti.

3.5 Perbaiki WordListAdapter.onBindViewHolder() untuk menggunakan resolver konten

Selanjutnya, Anda akan memperbaiki WordListAdapter.onBindViewHolder() untuk menggunakan resolver konten, bukan memanggil WordListOpenHelper secara langsung.



1. Di WordListAdapter, hapus variabel mDB, karena Anda tidak lagi secara langsung mengacu ke database. Ini menampilkan kesalahan di Android Studio yang akan memandu perubahan Anda selanjutnya.
2. Di constructor, hapus penetapan ke mDB.
3. Refactor > Change tanda tangan constructor dan hapus parameter db.
4. Tambahkan variabel instance untuk parameter kueri karena akan digunakan lebih dari sekali.

Resolver konten mengambil parameter kueri, yang harus Anda bangun. Pembuatan struktur kueri sama dengan kueri SQL, tetapi sebagai ganti pernyataan Selection, kueri menggunakan URI. Parameter kueri sangat mirip dengan kueri SQL.

```

private String queryUri = Contract.CONTENT_URI.toString(); // base uri
private static final String[] projection = new String[] {Contract.CONTENT_PATH}; //table
private String selectionClause = null;
private String selectionArgs[] = null;
private String sortOrder = "ASC";

```

5. Di onBindViewHolder(), hapus dua baris kode pertama.
 - o WordItem current = mDB.query(position);
 - o holder.wordItemView.setText(current.getWord());
6. Define an empty String variable named word.
7. Definisikan variabel integer yang disebut sebagai id dan setel menjadi -1.
8. Buat resolver konten dengan parameter kueri yang ditetapkan dan simpan hasilnya di sebuah Cursor yang disebut sebagai cursor. (Lihat MainActivity aplikasi MinimalistContentProvider untuk contohnya.)

```

String word = "";
int id = -1;

Cursor cursor = mContext.getContentResolver().query(Uri.parse(
    queryUri), null, null, null, sortOrder);

```

9. Sebagai ganti hanya mengirimkan WordItem, WordListAdapter.onBindViewHolder() harus melakukan pekerjaan ekstra yaitu mengekstrak kata dari kursor yang dikembalikan oleh resolver konten.
 - o Jika kursor yang dikembalikan berisi data, ekstrak kata dan setel teks holder tampilan.
 - o Ekstrak id-nya, karena Anda akan memerlukannya untuk listener klik.
 - o Tutup kursor. Ingatlah bahwa Anda tidak menutup kursor di WordListOpenHelper.query(), karena Anda telah mengembalikannya.
 - o Tangani jika ada kejadian tanpa data di kursor.
 - o Implementasikan sumber daya string apa pun yang direferensikan.

```

if (cursor != null) {
    if (cursor.moveToPosition(position)) {
        int indexWord = cursor.getColumnIndex(Contract.WordList.KEY_WORD);
        word = cursor.getString(indexWord);
        holder.wordItemView.setText(word);
        int indexId = cursor.getColumnIndex(Contract.WordList.KEY_ID);
        id = cursor.getInt(indexId);
    } else {
        holder.wordItemView.setText(R.string.error_no_word);
    }
    cursor.close();
} else {
    Log.e(TAG, "onBindViewHolder: Cursor is null.");
}

```

1. Perbaiki parameter untuk listener klik untuk dua tombol:
 - o current.getId() => id
 - o current.getWord() => word
- Listener klik yang diperbarui untuk tombol DELETE tampak seperti ini:

```

@Override
public void onClick(View v) {
    selectionArgs = new String[]{Integer.toString(id)};
    int deleted = mContext.getContentResolver().delete(
        Contract.CONTENT_URI, Contract.CONTENT_PATH, selectionArgs);
    if (deleted > 0) {
        // Need both calls
        notifyItemRemoved(h.getAdapterPosition());
        notifyItemRangeChanged(
            h.getAdapterPosition(), getItemCount());
    } else {
        Log.d(TAG, mContext.getString(R.string.not_deleted) + deleted);
    }
}

```

2. Ganti panggilan ke mDB.delete(id) di callback tombol DELETE dengan panggilan resolver konten yang akan dihapus.

```

selectionArgs = new String[]{Integer.toString(id)};
int deleted = mContext.getContentResolver().delete(
    Contract.CONTENT_URI, Contract.CONTENT_PATH, selectionArgs);

```

3.6 Ubah WordListAdapter.getCount() untuk menggunakan resolver konten

Sebagai ganti meminta hitungan dari database, getCount() harus tersambung ke resolver konten dan meminta hitungan. Di Contract, Anda mendefinisikan URI untuk mendapatkan hitungan tersebut:

```

public static final String COUNT = "count";
public static final Uri ROW_COUNT_URI =
    Uri.parse("content://" + AUTHORITY + "/" + CONTENT_PATH + "/" + COUNT)

```

Ubah WordListAdapter.getCount() ke:

- Menggunakan kueri resolver konten untuk mendapatkan hitungan item
- Menggunakan ROW_COUNT_URI di kueri Anda
- Hitungan adalah tipe integer dan merupakan elemen pertama Cursor yang dikembalikan
- Mengekstrak count dari kursor dan mengembalikannya
- Mengembalikan -1
- Menutup kursor

Gunakan kode yang baru saja Anda tulis untuk onBindViewHolder sebagai panduan

Solusi:

```

@Override
public int getItemCount() {
    Cursor cursor = mContext.getContentResolver().query(
        Contract.ROW_COUNT_URI, new String[] {"count(*) AS count"},
        selectionClause, selectionArgs, sortOrder);
    try {
        cursor.moveToFirst();
        int count = cursor.getInt(0);
        cursor.close();
        return count;
    } catch (Exception e){
        Log.d(TAG, "EXCEPTION getItemCount: " + e);
        return -1;
    }
}

```

3.7 Tambahkan penyedia konten ke Manifes Android

1. Jalankan aplikasi Anda.

2. Periksa logcat untuk melihat penyebab kesalahan (yang sangat umum).
3. Tambahkan penyedia konten ke Manifes Android di dalam tag `<application>`.

```
<provider
    android:name=".WordListContentProvider"  android:authorities="com.android.example.wordlistsqllwithcontentprovider.provider">
</provider>
```

4. Jalankan aplikasi Anda.

Aplikasi Anda harus berjalan dan berfungsi penuh. Jika tidak, bandingkan kode Anda dengan kode solusi yang diberikan dan gunakan debugger dan log untuk menemukan masalah.

3.8 Apa berikutnya?

- Anda telah mengimplementasikan penyedia konten dan metode `query()`-nya.
- Anda telah mengikuti kesalahan untuk memperbarui metode di kelas `WordListOpenHelper` dan `WordListAdapter` untuk bekerja dengan penyedia konten.
- Saat Anda menjalankan aplikasi, panggilan metode masuk ke penyedia konten.
- Untuk operasi penyisipan, penghapusan, dan pembaruan, aplikasi Anda masih memanggil `WordListOpenHelper`.

Dengan infrastruktur yang telah Anda bangun, Anda tidak perlu lagi berupaya keras mengimplementasikan metode yang tersisa.

Tugas 4. Mengimplementasikan Metode Penyedia Konten

4.1 `getType()`

Metode `getType()` dipanggil oleh aplikasi lain yang ingin menggunakan penyedia konten ini, untuk melihat jenis data seperti apa yang dikembalikan oleh aplikasi Anda.

Gunakan pernyataan `Switch` untuk mengembalikan tipe MIME yang sesuai.

- Tipe MIME dicantumkan dalam `Contract`.
- `SINGLE_RECORD_MIME_TYPE` adalah untuk `URI_ALL_ITEMS_CODE`
- `MULTIPLE_RECORDS_MIME_TYPE` adalah untuk `URI_ONE_ITEM_CODE`

Solusi:

```
@Nullable
@Override
public String getType(Uri uri) {
    switch (sUriMatcher.match(uri)) {
        case URI_ALL_ITEMS_CODE:
            return MULTIPLE_RECORDS_MIME_TYPE;
        case URI_ONE_ITEM_CODE:
            return SINGLE_RECORD_MIME_TYPE;
        default:
            return null;
    }
}
```

Tantangan: Bagaimana Anda bisa menguji metode ini, karena metode ini tidak dipanggil oleh aplikasi Anda. Bisakah Anda menyebutkan tiga cara yang berbeda untuk menguji apakah metode ini bekerja dengan benar?

4.2 Panggil penyedia konten untuk menyisipkan dan memperbarui kata di `MainActivity`.

Untuk memperbaiki operasi penyisipan, `MainActivity().onActivityResult` perlu memanggil penyedia konten, bukan database untuk menyisipkan dan memperbarui kata.

1. Di `MainActivity`, hapus deklarasi `mDB` dan pembuatan instance-nya.

Di `OnActivityResult()`

Menyisipkan:

1. Jika panjang kata tidak nol, buat variabel `ContentValues` bernama "values" dan tambahkan kata yang dimasukkan oleh pengguna ke dalamnya menggunakan string "word" sebagai kunci.
2. Ganti `mDB.insert(word);` dengan permintaan insert ke sebuah resolver konten.

Memperbarui:

1. Ganti `mDB.update(id, word);` dengan permintaan update ke sebuah resolver konten.

Cuplikan solusi:

```
// Update the database
if (word.length() != 0) {
    ContentValues values = new ContentValues();
    values.put(Contract.WordList.KEY_WORD, word);
    int id = data.getIntExtra(WordListAdapter.EXTRA_ID, -99);

    if (id == WORD_ADD) {
        getContentResolver().insert(Contract.CONTENT_URI, values);
    } else if (id >= 0) {
        String[] selectionArgs = {Integer.toString(id)};
        getContentResolver().update(Contract.CONTENT_URI, values, Contract.WordList.KEY_ID, selectionArgs
    );
    }
    // Update the UI
    mAdapter.notifyDataSetChanged();
}
```

4.3 Implementasikan `insert()` di penyedia konten

Metode `insert()` di penyedia konten merupakan pass-through. Jadi, Anda

1. memanggil metode `insert()` `OpenHelper`,
2. mengonversi long id yang dikembalikan ke sebuah URI konten ke item yang disisipkan,
3. dan mengembalikan URI tersebut.

Android Studio melaporkan kesalahan untuk parameter `values`, yang akan Anda perbaiki di langkah selanjutnya.

Solusi:

```
public Uri insert(Uri uri, ContentValues values) {
    long id = mDB.insert(values);
    return Uri.parse(CONTENT_URI + "/" + id);
}
```

4.4 Perbaiki `insert()` di `WordListOpenHelper`

Android Studio melaporkan kesalahan untuk parameter `values`.

1. Buka `WordListOpenHelper`. Metode `insert()` ditulis untuk mengambil parameter `String`.
2. Ubah parameter menjadi tipe `ContentValues`.
3. Hapus deklarasi dan penetapan `values` di isi metode.

4.5 Implementasikan `update()` di penyedia konten

Perbaiki metode `update` dengan cara yang sama seperti yang Anda pakai untuk memperbaiki metode `insert`.

- Di WordListContentProvider, implementasikan update(), yang merupakan satu baris kode yang meneruskan id dan word sebagai argumen.

```
return mDB.update(parseInt(selectionArgs[0]),
    values.getAsString(Contract.WordList.KEY_WORD));
```

- Anda tidak perlu membuat perubahan apa pun untuk memperbarui WordListOpenHelper.

4.6 Implementasikan delete() di penyedia konten

Di WordListContentProvider, implementasikan metode delete() dengan memanggil metode delete() di WordListOpenHelper dengan di kata untuk dihapus.

```
return mDB.delete(parseInt(selectionArgs[0]));
```

4.7. Jalankan aplikasi Anda.

Ya. Demikian saja. Jalankan aplikasi Anda dan pastikan semua berjalan dengan baik.

Dan jika aplikasi Anda masih belum bekerja, Anda harus memperbaiki semua masalah. Anda memerlukan kode yang berfungsi dengan baik di praktik selanjutnya. Di pelajaran tersebut, Anda akan menulis aplikasi yang menggunakan penyedia konten untuk memuat data daftar kata ke dalam antarmuka pengguna.

Kode solusi

Proyek Android Studio: [word_list_sql_with_content_provider](#)

Tantangan penyusunan kode

Catatan: Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk pelajaran berikutnya.

- Daftar kata hanyalah sebuah daftar yang berisi kata-kata tunggal, yang tidak terlalu berguna. Perluas aplikasi untuk menampilkan definisi, serta tautan ke informasi yang berguna, misalnya developer.android.com, stackoverflow, atau wikipedia.
- Tambahkan aktivitas yang memungkinkan pengguna mencari kata.
- Tambahkan pengujian dasar untuk semua fungsi di WordListContentProvider.

Rangkuman

- Dalam produksi, sebagian besar developer aplikasi biasanya akan mengoptimalkan aplikasi untuk mengakomodasi penyedia konten.
- Selama pengoptimalan, developer biasanya akan mengalami perubahan dan kesalahan bertahap.
- Anda perlu memisahkan UI dari database menggunakan penyedia konten dan resolver konten.
- UI harus tidak berubah selama proses pengoptimalan dari database yang disematkan ke sumber data eksternal.
- Kelas Contract mendefinisikan konstanta umum untuk semua komponen di aplikasi yang dioptimalkan.
- Kelas Contract melokalkan semua konstanta umum agar memudahkan pengelolaan.
- Saat mengoptimalkan, memiliki diagram kelas akses database akan berguna bagi Anda.
- Berhati-hatilah saat merancang URI akses data yang diperlukan aplikasi lain untuk mengakses data.
- Semua akses ke database Anda harus diubah untuk mengakses Resolver Konten, bukan mengakses langsung kelas helper (misalnya: WordListOpenHelper)
- Jika data yang mendasari diubah, penting bagi Anda untuk memberi tanda kepada UI untuk menyegarkan menggunakan notifyDataSetChanged().

[Konsep terkait]

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Penyedia Konten](#)

Ketahui selengkapnya

Dokumentasi Developer

- [Uniform Resource Identifier atau URI](#)
- [Tipe MIME](#)
- [MatrixCursor dan Cursor](#)
- [Penyedia Konten](#)

Video:

- [Arsitektur Aplikasi Android](#)
- [Arsitektur Aplikasi Android: Miliaran Pengguna Berikutnya](#)

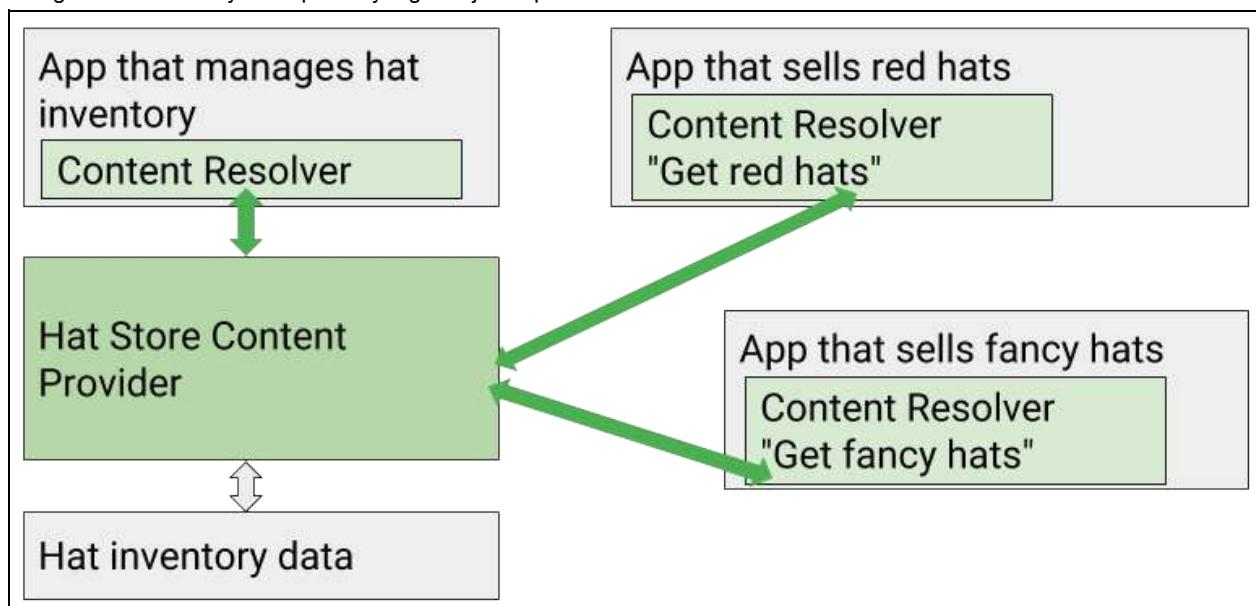
11.1C: Berbagi konten dengan aplikasi lain

Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1. Menjadikan penyedia konten Anda tersedia untuk aplikasi lain
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Untuk melindungi aplikasi dan data pengguna, aplikasi tidak bisa berbagi data dengan aplikasi lain secara langsung. Akan tetapi, aplikasi bisa menyediakan data ke aplikasi lain dengan menggunakan penyedia konten. Aplikasi klien kemudian bisa menggunakan resolver konten untuk mengakses data melalui antarmuka publik penyedia konten.

Diagram berikut menampilkan bagaimana penjual grosir topi bisa menggunakan penyedia konten untuk berbagi informasi mengenai inventarisnya ke aplikasi yang menjual topi.



Dalam praktik ini Anda akan memodifikasi WordListSQLWithContentProvider untuk memungkinkan aplikasi lain mengakses data di penyedia kontennya. Kemudian, Anda akan membuat aplikasi kedua, WordListClient, yang tidak memiliki data sendiri, tetapi mengambil data dari penyedia konten WordListSQLWithContentProvider.

Yang harus sudah Anda KETAHUI

Untuk praktik ini, Anda harus sudah mengenal:

- Penyedia dan resolver konten
- Cara mengganti nama paket
- Aplikasi WordListSQLWithContentProvider dari praktik sebelumnya

Yang akan Anda PELAJARI

Anda akan mempelajari cara:

- Menyetel izin, sehingga aplikasi lain bisa menggunakan penyedia konten aplikasi Anda.
- Membangun aplikasi klien yang mengambil data dari penyedia konten aplikasi Anda.

Yang akan Anda LAKUKAN

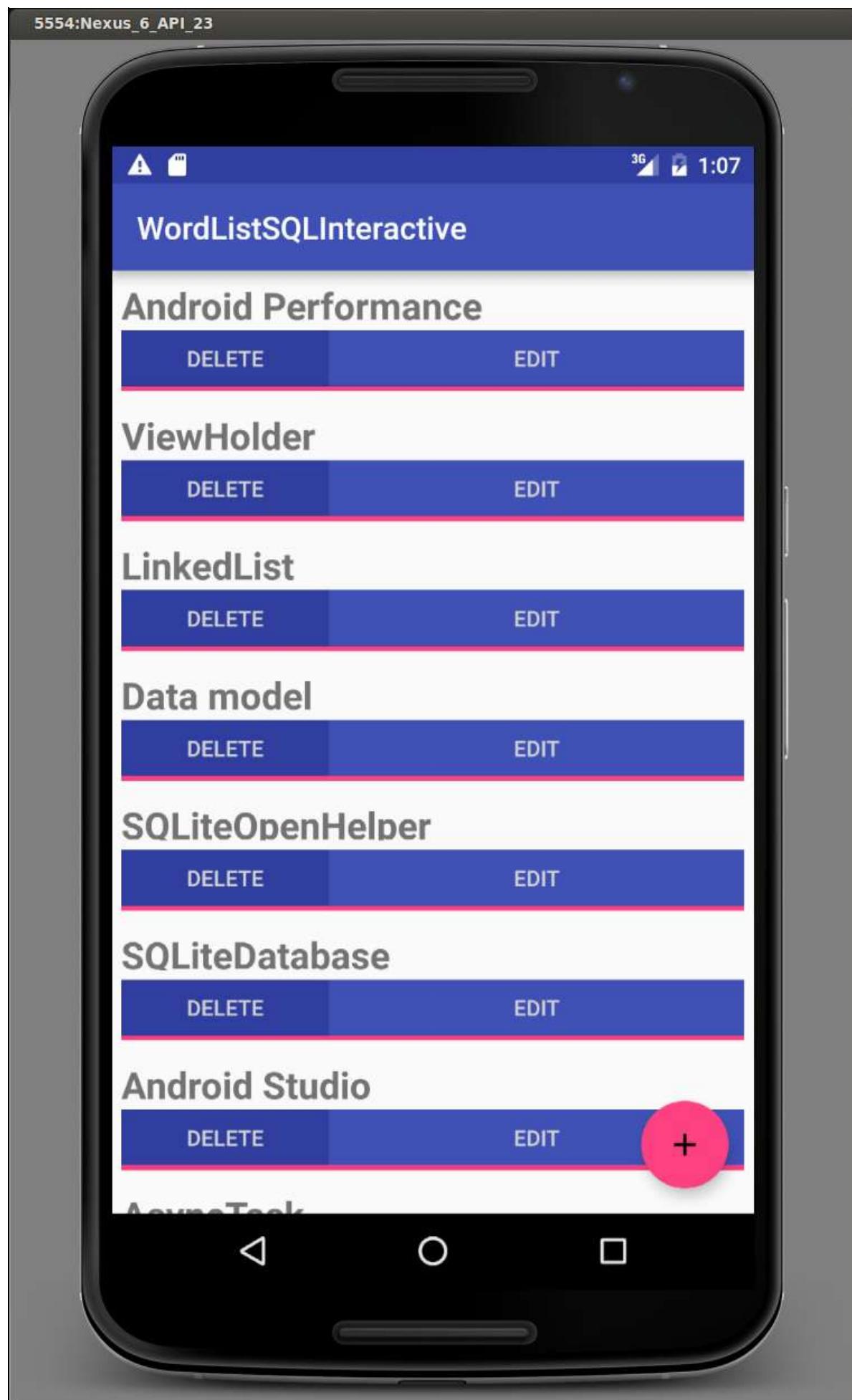
Anda akan:

- Mengaktifkan WordListSQLWithContentProvider untuk berbagi datanya.
- Membuat aplikasi klien yang mengambil data dari penyedia konten WordListSQLWithContentProvider.

Ringkasan Aplikasi

Anda akan menggunakan dua aplikasi di praktik ini.

- Aplikasi WordListSQLWithContentProvider yang ada yang telah Anda buat di praktik sebelumnya.
- Aplikasi WordListClient baru yang akan melakukan kueri pada penyedia konten WordListSQLWithContentProvider. UI untuk aplikasi ini sama dengan WordListInteractive.



Tugas 1. Menjadikan penyedia konten Anda tersedia untuk aplikasi lain

Secara default, aplikasi tidak bisa mengakses data aplikasi lain.

Agar penyedia konten tersedia untuk aplikasi lain, Anda harus menentukan izin di AndroidManifest di aplikasi Anda. Ini berlaku untuk aplikasi apa pun yang memiliki penyedia konten. Setiap penyedia konten memerlukan izin yang ditetapkan di AndroidManifest.

Izin tidak dicakup secara detail dalam praktik ini. Anda bisa mengetahui selengkapnya di [Mengimplementasikan Izin Penyedia Konten](#).

1.1 Modifikasi WordListSQLWithContentProvider untuk mengizinkan akses aplikasi

1. Buka [WordListSQLWithContentProvider](#) di Android Studio.
2. Buka file AndroidManifest.xml.
3. Tambahkan pernyataan eksport di dalam `<provider>` .

```
    android:exported="true"
```

4. Di bagian atas, di dalam tag `<manifest>`, tambahkan izin untuk penyedia konten.

Anda disarankan untuk menggunakan nama paket yang unik agar izin juga tetap unik.

```
<permission  
    android:name="com.android.example.wordlistsqllwithcontentprovider.PERMISSION" />
```

5. Jalankan aplikasi untuk memastikan tidak ada kesalahan dan biarkan terpasang di perangkat.

Agar aplikasi lain bisa mengakses penyedia konten WordListWithContentProvider, aplikasi yang memiliki penyedia konten harus dipasang di perangkat. Aplikasi tidak perlu berjalan.

Anda sekarang memiliki penyedia konten di perangkat Anda yang bisa diakses oleh aplikasi lain. Selanjutnya, Anda akan membangun aplikasi yaitu WordListClient, yang mengambil kata dari penyedia konten dan menampilkannya.

1.2 Buat aplikasi WordListClient

Sebagai ganti membangun aplikasi klien dari awal, Anda akan membuat WordListClient dari salinan [WordListSQLWithContentProvider](#). Anda akan mempertahankan antarmuka pengguna dan adaptor untuk menampilkan data. Anda akan membuang penyedia konten dan database, dan sebagai gantinya, mengambil data dari penyedia konten WordListSQLWithContentProvider.

1. Buat salinan **folder** [WordListSQLWithContentProvider](#) dan beri nama WordListClient.
2. Buka aplikasi yang disalin di folder WordListClient di Android Studio.
3. Ubah nama paket (Refactor > Rename) menjadi wordlistclient.
4. Buka build.gradle(Module:app) dan ubah id aplikasi menjadi wordlistclient.
5. Di strings.xml, ubah nama aplikasi menjadi WordListClient.
6. Di Manifes Android WordListClient, hapus deklarasi `<provider>` sehingga tidak ada lagi penyedia di WordListClient.
7. Di bagian atas, di dalam izin `<manifest>` tag, add a `<uses-permission>` untuk menggunakan penyedia konten WordListSQLWithContentProvider.

```
<uses-permission android:name = "com.android.example.wordlistsqllwithcontentprovider.PERMISSION"/>
```

8. Hapus kelas WordListContentProvider karena aplikasi akan mengakses penyedia konten

- WordListSQLWithContentProvider.
9. Hapus kelas WordListOpenHelper karena aplikasi Anda tidak memerlukan database atau helper yang terbuka sendiri.
 10. Perhatikan MainActivity dan WordListAdapter. Perhatikan bahwa kode untuk menyisipkan, menghapus, dan memperbarui kata tetap tidak berubah, memanggil penyedia konten WordListSQLWithContentProvider menggunakan URI yang ditetapkan di kelas Contract.
 11. Jalankan WordListClient.
 - Meskipun tidak memiliki data sendiri, WordListClient menampilkan data, yang merupakan data yang diambil dari penyedia konten aplikasi WordListSQLWithContentProvider.
 12. Masukkan beberapa kata di WordListClient.
 13. Mulai WordListSQLWithContentProvider.
 - Perhatikan bahwa kata yang Anda masukkan di WordListClient juga ditampilkan di WordListSQLWithContentProvider karena keduanya berbagai sumber data.
 - Hapus kata di WordListSQLWithContentProvider dan perhatikan bahwa kata juga dihapus dari tampilan WordListClient. (Anda mungkin perlu menggulir agar perubahan muncul.)
 14. Saat berinteraksi dengan aplikasi lain, perubahan yang dibuat oleh satu aplikasi direfleksikan di aplikasi lain.

Pada contoh toko topi sebelumnya, pemilik gudang bisa memperbarui inventaris topi dengan topi merah atau cantik yang baru dan aplikasi toko akan langsung bisa menampilkan topi baru ini kepada pelanggannya. Dan jika toko topi merah berhasil menjual semua topi merah cantik, toko topi cantik akan mengetahui bahwa topi merah cantik telah habis terjual.

Kode solusi

Proyek Android Studio: [WordListClient](#)

Rangkuman

- Aplikasi Anda bisa mengizinkan aplikasi lain untuk berinteraksi dengan penyedia kontennya dan mengambil atau mengedit data.
- Anda bisa membangun aplikasi klien yang bukan menyediakan datanya sendiri, tetapi mengambil data dari penyedia konten aplikasi lain.
- Anda bisa memisahkan pengelolaan data dari tampilan data. Yaitu, Anda bisa memiliki satu aplikasi yang mengelola penyedia konten dan beberapa aplikasi klien yang menggunakan data yang disediakan oleh penyedia konten.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Penyedia Konten](#)

Ketahui selengkapnya

Dokumentasi Developer

- [Bekerja dengan Izin Sistem](#)
- [Mengimplementasikan Izin Penyedia Konten](#)

12.1: Memuat dan menampilkan data yang diambil dari penyedia konten

Daftar Isi:

- [Yang harus sudah Anda KETAHUI](#)
- [Yang akan Anda PELAJARI](#)
- [Yang akan Anda LAKUKAN](#)
- [Ringkasan Aplikasi](#)
- [Tugas 1: Membuat aplikasi dasar untuk WordListLoader](#)
- [Tugas 2: MainActivity: Menambahkan LoaderManager dan LoaderCallbacks](#)
- [Tugas 3: WordListAdapter: Implementasikan setData\(\), getItemCount\(\), dan onBindViewHolder\(\)](#)
- [Rangkuman](#)
- [Konsep terkait](#)
- [Ketahui selengkapnya](#)

Dalam praktik ini Anda akan mempelajari cara memuat data yang disediakan oleh penyedia konten aplikasi lain latar belakang dan menampilkannya kepada pengguna, saat sudah siap.

Meminta [ContentProvider](#) untuk menyediakan data yang ingin Anda tampilkan bisa memakan waktu. Jika Anda meminta data dari penyedia konten dari sebuah [Aktivitas](#) (dan menjalankannya di thread UI], aplikasi bisa diblokir cukup lama sehingga penundaan cukup terlihat untuk pengguna dan sistem bahkan bisa mengeluarkan pesan "Application Not Responding". Dengan demikian, Anda harus memuat data di thread yang terpisah, latar belakang, dan menampilkan hasilnya setelah pemuatan selesai.

Untuk menjalankan kueri di thread yang terpisah, Anda harus menggunakan loader yang berjalan secara asinkron latar belakang dan menghubungkan ulang ke [Aktivitas](#) setelah selesai. Khususnya, [CursorLoader](#) menjalankan kueri di latar belakang, dan otomatis menjalankan ulang kueri saat data yang terkait dengan kueri berubah.

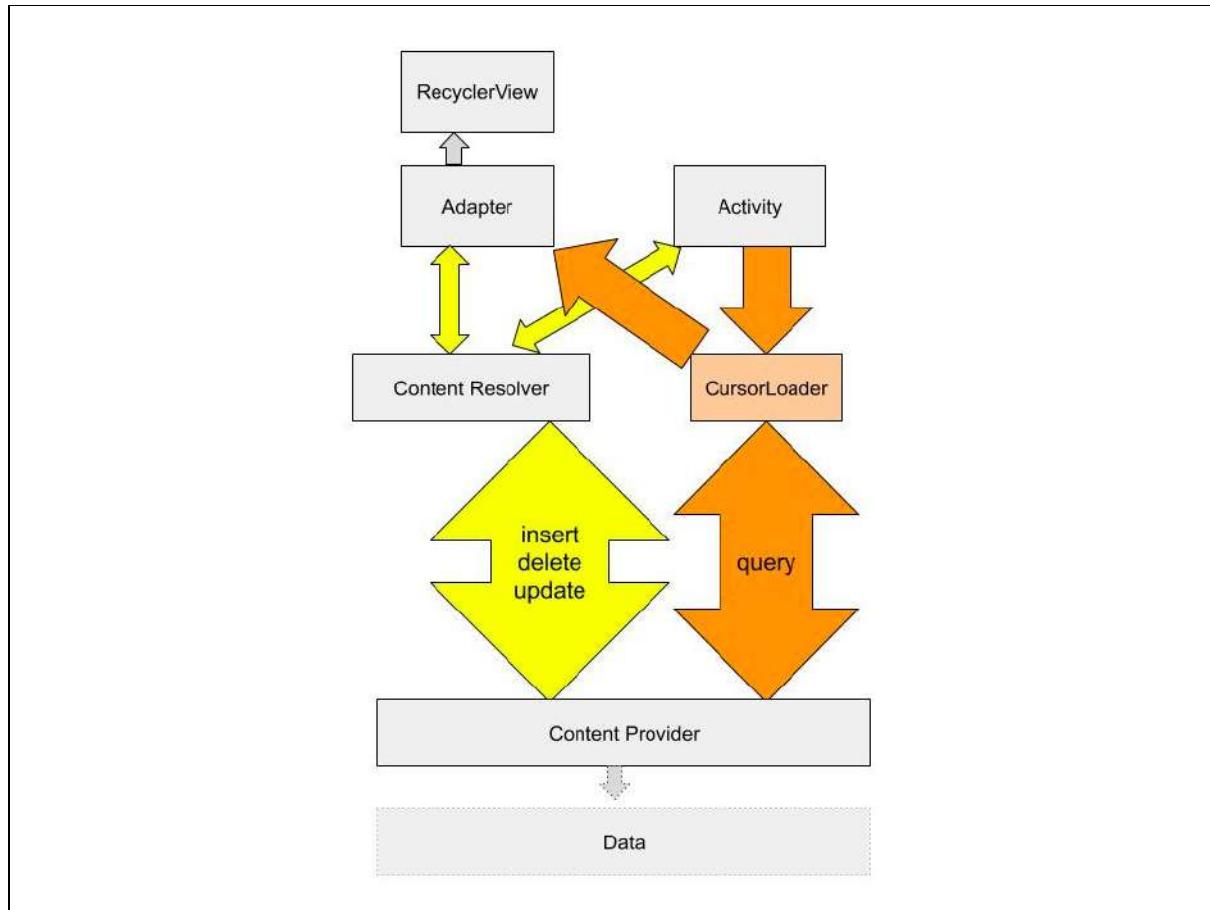
Anda telah menggunakan [AsyncTaskLoader](#) di praktik sebelumnya. CursorLoader memperluas AsyncTaskLoader untuk bekerja dengan penyedia konten.

Di tingkat tinggi, Anda memerlukan bagian ini untuk menggunakan loader untuk menampilkan data dari penyedia konten:

- Sebuah [Aktivitas](#) atau fragmen.
- Sebuah instance [LoaderManager](#) di Aktivitas.
- Sebuah [CursorLoader](#) untuk memuat data yang didukung oleh [ContentProvider](#).
- Sebuah implementasi untuk [LoaderCallbacks.LoaderCallbacks](#), sebuah antarmuka callback bagi klien untuk berinteraksi dengan LoaderManager.
- Cara untuk menampilkan data loader, umumnya menggunakan sebuah adaptor. Misalnya, Anda bisa menampilkan data di RecyclerView.

Diagram berikut menampilkan arsitektur aplikasi lengkap dengan sebuah loader.

- Loader melakukan kueri untuk item di latar belakang. Jika data berubah, loader otomatis mendapatkan serangkaian data baru untuk adaptor.
- Operasi penyisipan, penghapusan, dan pembaruan tidak menggunakan loader. Akan tetapi, setelah data berubah karena operasi penyisipan, penghapusan, atau pembaruan, loader mengambil data yang diperbarui dan memberi tahu adaptor.



Yang harus sudah Anda KETAHUI

Untuk praktik ini Anda harus bisa:

- Menampilkan data di RecyclerView.
- Bekerja dengan Adaptor sederhana.
- Memahami Kursor (lihat praktik dan konsep sebelumnya.)
- Bekerja dengan AsyncTaskLoader.
- Memahami cara bekerja dengan Penyedia Konten.

Yang akan Anda PELAJARI

Anda akan belajar:

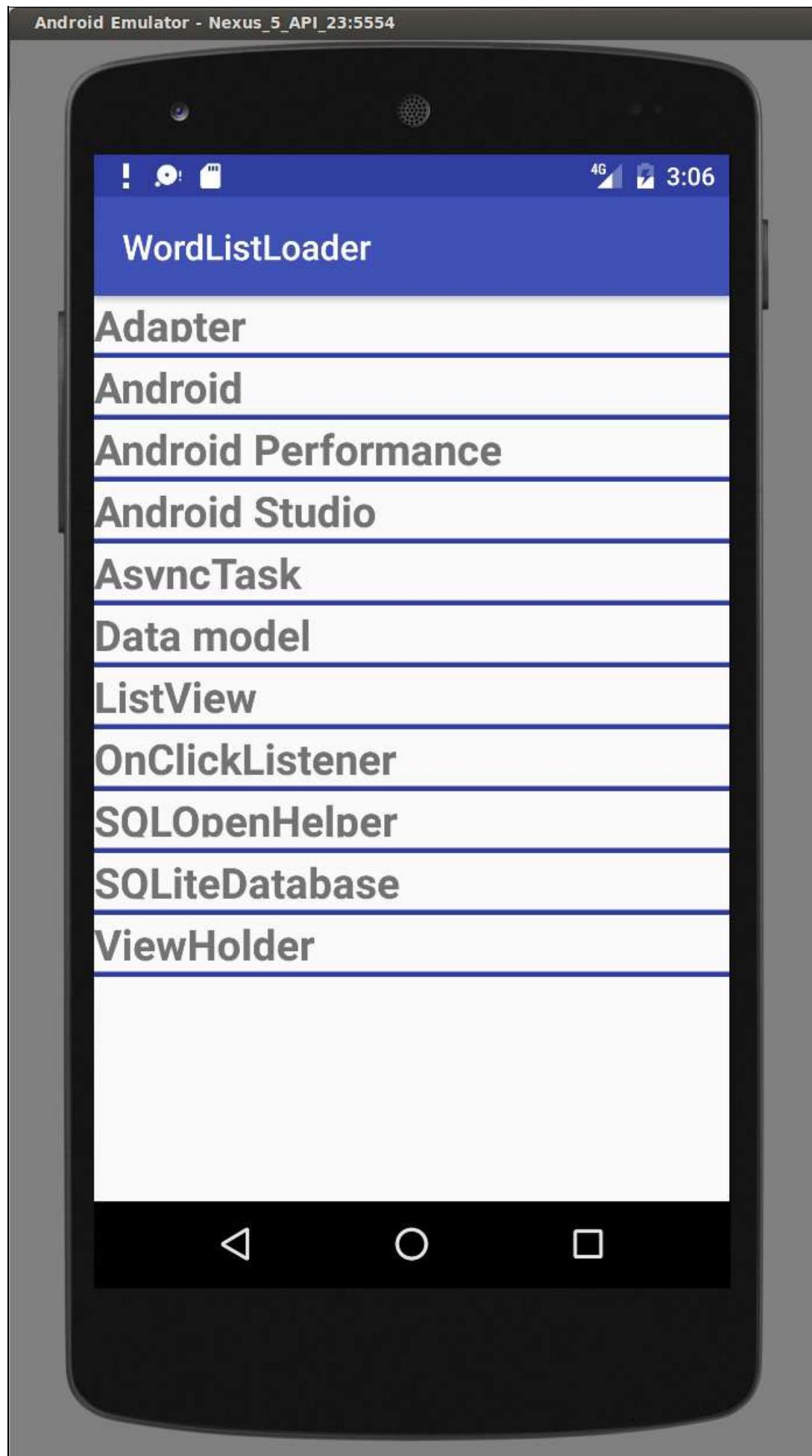
- Memuat data dari penyedia konten menggunakan CursorLoader.
- Menggunakan kode dari aplikasi yang telah selesai untuk membangun aplikasi secara cepat dengan fungsionalitas terkait.

Yang akan Anda LAKUKAN

- Anda akan membuat aplikasi dasar yang menggunakan CursorLoader untuk melakukan kueri pada penyedia konten WordListSQLWithContentProvider dan menampilkan data di RecyclerView.
- Gunakan WordListClient sebagai referensi untuk beberapa kode. Khususnya, Anda bisa menggunakan Kelas Contract dan WordItem, serta bagian dari kelas MainActivity dan WordListAdapter.
- Aplikasi yang akan Anda buat akan memiliki UI yang sangat dasar. Tidak seperti WordListClient, ini tidak akan memiliki fungsionalitas penyisipan, penghapusan, atau pembaruan.

Ringkasan Aplikasi

Menggunakan WordListClient dari praktik sebelumnya sebagai sumber untuk beberapa kode, Anda akan membuat aplikasi baru, WordListLoader yang memuat dan menampilkan data dari penyedia konten untuk WordListSQLWithContentProvider. Tangkapan layar berikut menampilkan bagaimana aplikasi yang telah selesai akan menampilkan kata.



PENTING:

- Anda harus memasang aplikasi [WordListWithContentProvider](#) yang membagikan penyedia kontennya sehingga tersedia penyedia konten untuk WordListLoader.
- Gunakan aplikasi [WordListClient](#) yang telah Anda bangun di praktik sebelumnya sebagai referensi dan untuk menggunakan ulang kode.

Tugas 1. Membuat aplikasi dasar untuk WordListLoader

Dalam tugas ini, Anda akan memuat proyek dan bagian aplikasi yang tidak spesifik untuk loader. Anda harus memuat aplikasi WordListClient di Android Studio, sehingga Anda bisa menyalin kode darinya.

1.1 Buat proyek dengan kelas Contract dan WordListItem dan file layout.

1. Mulailah Android Studio dan muat aplikasi WordListClient yang sudah selesai dari praktik sebelumnya.
2. Buat proyek baru dengan template Empty Activity dan beri nama WordListLoader.
3. Tambahkan izin untuk penyedia konten WordListSQLWithContentProvider ke Manifes Android.

```
<uses-permission android:name =
    "com.android.example.wordlistsqlwithcontentprovider.PERMISSION"/>
```

4. Buat kelas Java baru dan beri nama Contract.
5. Salin kelas Contract dari WordListClient ke kelas Contract baru WordListLoader. Pastikan Anda tidak menyalin nama paket.
6. Di WordListLoader, buat kelas Java baru dan beri nama WordListItem.
7. Salin kelas WordItem dari WordListClient ke kelas WordItem baru WordListLoader.
8. Salin layout tampilan recycler dari activity_main.xml dari WordListClient ke WordListLoader. Buang tombol aksi mengambang.
9. Buat layout baru untuk WordListItem, wordlist_item.xml.
10. Menggunakan wordlist_item.xml dari WordListClient sebagai referensi Anda, buat LinearLayout dengan satu TextView.
 - o Id TextView harus `android:id="@+id/word"`.
 - o Selesaikan string, dimensi, dan gaya dan Anda gunakan kembali. Perhatikan bahwa Anda bisa menyalin/menempel antar proyek. Ganti file XML yang sudah ada di WordListLoader.
 - o Ubah app_name ke WordListLoader di strings.xml.
11. Di tahap ini, Anda seharusnya tidak melihat kesalahan di Android Studio.

1.2 Tambahkan sebuah RecyclerView ke MainActivity

Untuk menampilkan data, tambahkan RecyclerView ke MainActivity. Anda bisa melakukan ini sendiri atau menggunakan kembali kode dari WordListClient.

1. Tambahkan RecyclerView dan Layout Coordinator dari pustaka dukungan ke file build.gradle Anda.

```
compile 'com.android.support:recyclerview-v7:24.1.1'
compile 'com.android.support:design:24.1.1'
```

2. Impor versi pustaka dukungan RecyclerView dan LinearLayoutManager ke MainActivity Anda.

```
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
```

3. Buat TAG untuk MainActivity.
4. Buat variabel pribadi mRecyclerView untuk RecyclerView.
5. Buat variabel pribadi mWordListAdapter untuk adaptor. Ini akan tetap berwarna merah, sampai Anda membuat kelas adaptor.
6. Di onCreate() MainActivity, buat RecyclerView, buat WordListAdapter, setel adaptor di RecyclerView, dan lampirkan

sebuah LinearLayoutManager. Lihat WordListClient untuk kode contoh.

```
// Create recycler view.
mRecyclerView = (RecyclerView) findViewById(R.id.recyclerview);
// Create an adapter and supply the data to be displayed.
mAdapter = new WordListAdapter(this);
// Connect the adapter with the recycler view.
mRecyclerView.setAdapter(mAdapter);
// Give the recycler view a default layout manager.
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
```

7. Jika Anda membangun aplikasi Anda sekarang, hanya WordListAdapter yang berwarna merah. Aplikasi belum berjalan.

1.3 Buat WordListAdapter

Gunakan WordListAdapter dari WordListClient dan cuplikan berikut sebagai referensi untuk membuat adaptori ini. Jika memerlukan informasi pengingat, baca lagi bab RecyclerView di kursus ini.

1. Buat kelas Java baru WordListAdapter yang memperluas RecyclerView.Adapter.

```
public class WordListAdapter
    extends RecyclerView.Adapter<WordListAdapter.WordViewHolder> {}
```

Menggunakan WordListAdapter sebagai referensi, tambahkan yang berikut ini:

2. Tambahkan kelas ViewHolder dalam dengan satu TextView, yang bernama wordItemView dan mekarkan dari tampilan teks dengan id "word".

```
class WordViewHolder extends RecyclerView.ViewHolder {
    public final TextView wordItemView;

    public WordViewHolder(View itemView) {
        super(itemView);
        wordItemView = (TextView) itemView.findViewById(word);
    }
}
```

3. Tambahkan TAG untuk pesan log.

```
private static final String TAG = WordListAdapter.class.getSimpleName();
```

4. Tambahkan variabel anggota untuk LayoutInflater dan konteks.

```
private final LayoutInflater mInflater;
private Context mContext;
```

5. Implementasikan constructor untuk WordListAdapter.

```
public WordListAdapter(Context context) {
    mInflater = LayoutInflater.from(context);
    this.mContext = context;
}
```

6. Implementasikan (atau salin) metode onCreateViewHolder untuk memekarkan tampilan wordlist_item.

```
@Override  
public WordViewHolder onCreateViewHolder(  
    ViewGroup parent, int viewType) {  
    View mItemView = mInflater.inflate(  
        R.layout.wordlist_item, parent, false);  
    return new WordViewHolder(mItemView);  
}
```

7. Tekan Alt-Enter pada header kelas adaptor dan "choose implement methods" untuk membuat stub metode untuk metode `getItemCount()` dan `onBindViewHolder()`.
8. Pada tahap ini, seharusnya sudah tidak ada lagi garis bawah merah atau kata di kode Anda.
9. Jalankan aplikasi Anda yang seharusnya akan menampilkan aktivitas kosong seperti yang ditampilkan di tangkapan layar berikut, karena Anda belum memuat data apa pun. Anda akan menambahkan data di tugas berikutnya.



Tugas 2. MainActivity: Menambahkan LoaderManager dan LoaderCallbacks

Ketika Anda menggunakan sebuah loader untuk memuat data, gunakan pengelola loader untuk menangani detail berjalannya loader.

LoaderManager adalah kelas bantu yang mengelola semua loader Anda. Anda hanya perlu satu pengelola loader per aktivitas. Misalnya, pengelola loader menangani pendaftaran sebuah observer dengan penyedia konten, yang menerima callback ketika data di penyedia konten berubah.

2.1 Tambahkan Pengelola Loader

1. Buka MainActivity.java
2. Perluas tanda tangan kelas untuk mengimplementasikan LoaderManager.LoaderCallbacks. Impor versi pustaka dukungan.

```
public class MainActivity extends AppCompatActivity implements LoaderManager.LoaderCallbacks<Cursor>
```

3. Implementasikan stub metode untuk `onCreateLoader()`, `onLoadFinished()`, dan `onLoaderReset()`.
4. Di `onCreate()`, buat LoaderManager dari pustaka dukungan dan daftarkan sebuah loader dengannya.
 - o Argumen pertama adalah tag numerik; karena Anda hanya memiliki satu loader, nomor yang Anda pilih tidaklah penting.
 - o Anda tidak meneruskan data apa pun, sehingga argumen kedua adalah null.
 - o Dan Anda melekatkan loader ke `MainActivity(this)` saat ini.

```
getSupportLoaderManager().initLoader(0, null, this);
```

2.2. Implementasikan onCreateLoader()

LoaderManager memanggil metode `onCreateLoader()` untuk membuat loader, jika belum ada.

Anda membuat loader dengan memberikannya konteks dan URI asal data dimuat—dalam hal ini, untuk penyedia konten WordListSQLWithContentProvider, URI yang ditetapkan di Contract.

1. Di `onCreateLoader()`, buat sebuah `queryUri` dan `projection`. Gunakan URI yang sama dengan yang digunakan oleh resolver konten untuk melakukan kueri pada penyedia konten. Anda bisa menemukan ini di `Contract` dan ini digunakan di `WordListClient`.
2. Buat dan kembalikan `CursorLoader` baru dari argumen ini. Impor `CursorLoader` dari pustaka dukungan.

```
@Override
public Loader<Cursor> onCreateLoader(int id, Bundle args) {
    String queryUri = Contract.CONTENT_URI.toString();
    String[] projection = new String[] {Contract.CONTENT_PATH};
    return new CursorLoader(this, Uri.parse(queryUri), projection, null, null);
}
```

2.3 Implementasikan onLoadFinished()

Setelah pemuatannya selesai, Anda perlu mengirimkan data ke adaptor.

1. Panggil `setData()` di `onLoadFinished()`. Kode akan berubah menjadi merah, dan Anda akan mengimplementasikannya di tugas berikutnya. Argumen untuk `setData()` adalah kurSOR dengan "data" yang dikembalikan oleh loader setelah selesai dimuat.

```

@Override
public void onLoadFinished(Loader<Cursor> loader, Cursor data) {
    mAdapter.setData(data);
}

```

2.4 Implementasikan onLoaderReset()

Saat menyetel ulang loader, beri tahu adaptor bahwa data sudah tidak tersedia dengan meneruskan null ke setData().

```

@Override
public void onLoaderReset(Loader<Cursor> loader) {
    mAdapter.setData(null);
}

```

Tugas 3: WordListAdapter: Mengimplementasikan setData(), getItemCount(), dan onBindViewHolder()

Sebagai tugas akhir, Anda perlu mengimplementasikan metode setData() yang direferensikan di atas, dan mengimplementasikan onBindViewHolder() untuk bekerja dengan loader untuk menampilkan data. Berikut ini prosesnya:

- Saat loader selesai memuat data baru atau yang diubah latar belakang, metode onLoadFinished() yang Anda implementasikan di MainActivity dieksekusi.
- onLoadFinished() memanggil WordListAdapter.setData(), yang memperbarui variabel mCursor di adaptor dengan data terakhir loader dan memberi tahu adaptor bahwa data telah diubah.
- Adaptor memperbarui UI dengan data baru, dengan melekatkan tampilan ke data di onBindViewHolder().

3.1 Implementasikan setData()

Anda memerlukan cara untuk menyetel dan menyimpan versi data yang terakhir dimuat dengan adaptor. Untuk aplikasi ini, loader mengembalikan data sebagai kursor, sehingga Anda perlu membuat variabel anggota Cursor mCursor yang akan selalu menampung rangkaian data terakhir.

Metode setData() dipanggil oleh loader ketika selesai dimuat atau disetel ulang, dan perlu memperbarui mCursor.

1. Buat variabel anggota pribadi tipe Cursor. Beri nama "mCursor" dan inisialisasi ke null.
2. Implementasikan metode publik setData(). Ini mengambil argumen Cursor dan tidak mengembalikan apa pun.
3. Di isi, setel mCursor ke argumen Cursor yang diteruskan dan panggil notifyDataSetChanged(), sehingga adaptor memperbarui tampilan.

```

public void setData(Cursor cursor) {
    mCursor = cursor;
    notifyDataSetChanged();
}

```

3.2. Implementasikan getItemCount()

Sebagai ganti 0, getItemCount() perlu mengembalikan jumlah item di mCursor. Jika mCursor null, kembalikan -1.

```

@Override
public int getItemCount() {
    if (mCursor != null) {
        return mCursor.getCount();
    } else {
        return -1;
    }
}

```

3.3. Implementasikan onBindViewHolder()

Di WordListClient, metode onBindViewHolder() menggunakan resolver konten untuk mengambil data dari penyedia konten WordListSQLWithContentProvider. Di aplikasi ini, onBindViewHolder() menggunakan data yang disediakan loader dan menyimpannya di mCursor.

Di onBindViewHolder, tangani situasi berikut.

1. Jika mCursor null, jangan lakukan apa pun, kecuali menampilkan pesan log. Di aplikasi nyata, Anda juga perlu memberi tahu pengguna dengan cara yang mudah dipahami.
2. Jika mCursor tidak null tetapi tidak berisi kata, setel teks di TextView ke ERROR: NO WORD. Sekali lagi, di aplikasi nyata, Anda akan menangani ini bergantung pada tipe aplikasi yang Anda miliki.
3. Jika tidak, dapatkan indeks kolom untuk kolom "word" (Anda tidak bisa mengasumsikan kolom berada di lokasi tetap di baris kursor) dan menggunakan indeks, mengambil kata Setel teks di tampilan teks ke kata.

```
@Override
public void onBindViewHolder(WordViewHolder holder, int position) {

    String word = "";

    if (mCursor != null) {
        if (mCursor.moveToPosition(position)) {
            int indexWord = mCursor.getColumnIndex(Contract.WordList.KEY_WORD);
            word = mCursor.getString(indexWord);
            holder.wordItemView.setText(word);
        } else {
            holder.wordItemView.setText(R.string.error_no_word);
        }
    } else {
        Log.e(TAG, "onBindViewHolder: Cursor is null.");
    }
}
```

3.4 Jalankan dan uji aplikasi Anda

Aplikasi WordListLoader Anda harus bekerja sama persis dengan aplikasi WordListClient untuk menampilkan daftar kata. Untuk menguji aplikasi Anda, lakukan yang berikut.

1. Pastikan WordListSQLWithContentProvider dipasang di perangkat, sehingga aplikasi Anda memiliki penyedia konten yang menjadi sumber pemuatan. Jika tidak, aplikasi Anda akan menampilkan tampilan teks kosong.
2. Jalankan WordListLoader. Anda akan melihat daftar kata yang sama seperti di WordListSQLWithContentProvider.

Kode solusi

Proyek Android Studio: [WordListLoader](#)

Rangkuman

- Di bab ini, Anda telah belajar cara menggunakan loader untuk memuat data dari penyedia konten yang bukan bagian dari aplikasi Anda.

Konsep terkait

Dokumentasi konsep terkait ada di [Dasar-Dasar Developer Android: Konsep](#).

- [Loader](#)

Ketahui selengkapnya

- [Loader](#)
- [Menjalankan kueri dengan CursorLoader](#)
- [CursorLoader](#)

Apendiks: Tugas Pekerjaan Rumah

Apendiks ini berisi daftar tugas pekerjaan rumah yang bisa diisi siswa di akhir setiap praktik. Tanggung jawab instruktur adalah:

- menugaskan pekerjaan rumah bila diperlukan
- mengkomunikasikan kepada siswa tentang cara menyerahkan tugas pekerjaan rumah
- memeriksa tugas pekerjaan rumah

Instruktur bisa menggunakan saran ini sesuai kebutuhan, dan bebas menugaskan pekerjaan rumah lain yang dirasa cocok.

Tugas Pekerjaan Rumah: Pelajaran 1

Daftar Isi:

- [1.1: Memasang Android Studio dan Menjalankan Hello World](#)
- [1.2 A, B: Membuat UI Interaktif Pertama Anda / Menggunakan Layout](#)
- [1.3: Tampilan Bergulir dan Teks](#)
- [1.4: Sumber Daya](#)

1.1: Memasang Android Studio dan Menjalankan Hello World

Membangun dan menjalankan aplikasi

1. Buat proyek Android baru dari Template Kosong.
2. Tambahkan laporan pembuatan log untuk berbagai tingkat log di `onCreate()` dalam aktivitas utama.
3. Buat emulator untuk perangkat, targetkan versi Android yang Anda suka, kemudian jalankan aplikasi tersebut.
4. Gunakan pemfilteran di logcat untuk menemukan laporan log Anda dan atur tingkat agar hanya menampilkan laporan pembuatan log kesalahan atau debug.

Jawablah pertanyaan berikut

Pertanyaan 1

Apa nama file layout untuk aktivitas utama?

Pertanyaan 2

Apa nama sumber daya string yang menentukan nama aplikasi?

Pertanyaan 3

Apa alat (bantu) yang Anda gunakan untuk membuat emulator baru?

- Android Device Monitor
- AVD Manager
- SDK Manager
- Theme Editor

Pertanyaan 4

Perangkat apa yang memiliki spesifikasi seperti berikut? Anda bisa melihat spesifikasi perangkat yang berbeda saat membuat emulator perangkat yang baru.

Bagaimana spesifikasi perangkat 1?

- Ukuran = 4,7 inci
- Resolusi = 768 x 1280

Bagaimana spesifikasi perangkat 2?

- Ukuran = 8,86 inci
- Resolusi = 2048 x 1536

Pertanyaan 5

Anggaplah bahwa aplikasi Anda menyertakan laporan pembuatan log ini:

```
Log.i("MainActivity", "MainActivity layout is complete");
```

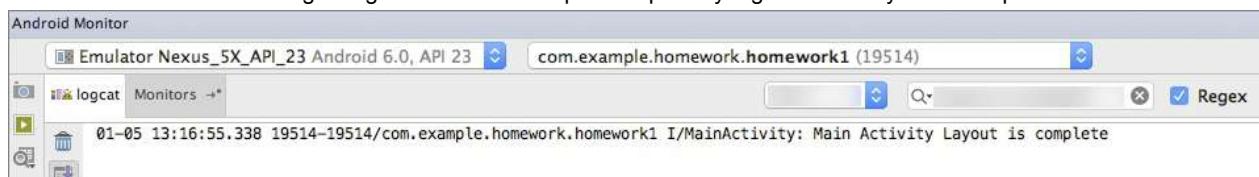
Anda melihat laporan "MainActivity layout is complete" di konsol logcat jika menu tingkat Log disetel ke status? (Petunjuk: boleh memilih beberapa jawaban.)

- Verbose
- Debug
- Info
- Warn
- Error
- Assert



Pertanyaan 6

Jika aplikasi Anda mencatat log pesan "XX Activity layout is complete" setiap kali aktivitas baru ditampilkan, bagaimana Anda bisa meminta konsol logcat agar HANYA menampilkan laporan yang memuat "layout is complete"?



Mengirimkan aplikasi Anda untuk dinilai

Tidak ada aplikasi untuk mengirimkan tugas pekerjaan rumah ini.

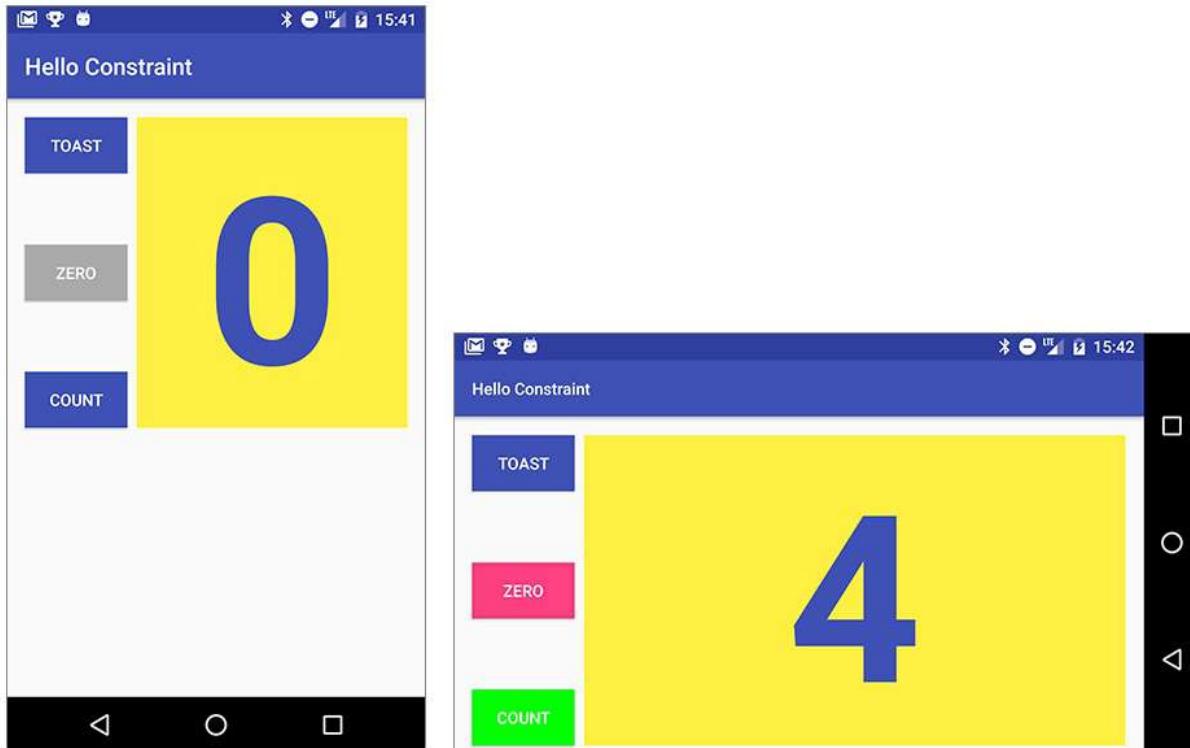
1.2 A, B: Membuat UI Interaktif Pertama Anda / Menggunakan Layout

Membangun dan menjalankan aplikasi

Buka aplikasi [HelloConstraint](#) yang Anda buat di pelajaran [Menggunakan Layout](#).

1. Modifikasi layout activity_main.xml agar menyertakan tombol ketiga yang disebut **Zero** yang muncul di antara tombol "Toast" dan "Count".
2. Pada awalnya tombol **Zero** memiliki latar belakang berwarna abu-abu.
3. Tampilkan seluruh tiga tombol di sebelah kiri dari show_count TextView. Distribusikan tombol secara vertikal di antara bagian atas dan bawah dari show_count TextView.
4. Pastikan Anda menyertakan tombol **Zero** untuk orientasi lanskap di activity_main.xml (land), dan untuk layar berukuran tablet di activity_main (xlarge).
5. Buatlah tombol **Zero** agar mengubah nilai pada show_count TextView ke 0.
6. Perbarui penanganan klik untuk tombol **Count** sehingga ia bisa mengubah *sendiri* warna latar belakangnya, tergantung pada apakah hitungan baru bernilai ganjil atau genap.
 - o Petunjuk: Jangan gunakan `findViewById` untuk menemukan tombol **Count**. Apakah ada hal lain yang bisa Anda gunakan?

- Jangan ragu untuk menggunakan konstanta pada kelas `color` untuk dua warna latar belakang yang berbeda.
7. Perbarui juga penangan klik bagi tombol **Count** untuk menyetel warna latar belakang bagi tombol **Zero** ke warna lain selain abu-abu untuk menunjukkan bahwa itu sekarang aktif. Petunjuk: Anda bisa menggunakan `findViewById` dalam kasus ini.
8. Perbarui penangan klik bagi tombol **Zero** untuk menyetel ulang warna menjadi abu-abu (sehingga berwarna abu-abu ketika hitungannya nol).



Jawablah pertanyaan berikut

Pertanyaan 1

Apa atribut pembatas layout pada tombol **Zero** untuk memosisikannya secara vertikal di tengah-tengah dua tombol lainnya?

Pertanyaan 2

Apa atribut pembatas layout pada tombol **Zero** untuk memosisikannya secara horizontal sejajar dengan dua tombol lainnya?

Pertanyaan 3

Manakah dari operasi berikut yang bisa Anda lakukan untuk memasukkan tombol **Zero** di layout `xlarge` (tablet) dan `land` (lanskap) yang telah dibuat?

- Ulangi prosedur yang digunakan pada layout pertama: Buka layout kedua, klik tab **Design**, seret tombol dari panel **Palette**, setel batasnya dalam panel design, lalu setel ID, lebar, tinggi, warna, dan teks di panel **Properties**.
- Pada layout pertama klik tab **Text**, pilih dan **Salin** kode XML untuk tombol Zero, buka layout kedua, lalu **Tempel** kode XML untuk tombol tersebut.
- Gunakan salah satu cara tersebut.

Pertanyaan 4

Apa tanda tangan yang benar bagi metode yang digunakan sebagai nilai atribut XML `android:onClick` ?

- `public void callMethod()`
- `public void callMethod(View view)`
- `private void callMethod(View view)`
- `public boolean callMethod(View view)`

Pertanyaan 5

Penangan klik untuk tombol **Count** dimulai dengan tanda tangan metode berikut:

```
public void countUp(View view)
```

Manakah dari teknik berikut yang lebih efisien penggunaannya dalam penangan ini untuk mengubah warna latar belakang tombol? Pilih salah satu:

- Gunakan `findViewById` untuk menemukan tampilan tombol **Count**. Tetapkan hasilnya ke variabel `view`, lalu gunakan `setBackgroundColor()`.
- Gunakan parameter `view` yang diteruskan ke penangan klik dengan `setBackgroundColor()`.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Aplikasi menampilkan tombol **Zero**.
- Tombol **Zero** menggunakan pembatas layout untuk memosisikan dirinya di antara tombol **Toast** dan **Count**.
- Aplikasi mencakup implementasi `activity_main.xml`, `activity_main.xml (land)`, dan `activity_main.xml (xlarge)`, termasuk penyesuaian tombol toast di `activity_main.xml (land)`.
- Aplikasi mencakup implementasi metode penangan klik bagi tombol **Zero** untuk menyetel ulang hitungan ke 0. Metode harus menunjukkan hitungan nol pada tampilan `show_count`. Penangan klik juga harus menyetel ulang warna latar belakang tombol **Zero** menjadi abu-abu.
- Metode penangan klik untuk tombol **Count** telah diperbarui sehingga ia bisa mengubah sendiri warna latar belakangnya tergantung pada apakah hitungan baru berupa ganjil atau genap. Metode ini harus menggunakan parameter `view` untuk mengakses tombol. Metode ini juga harus mengubah latar belakang tombol **Zero** menjadi warna selain abu-abu.

1.3: Tampilan Bergulir dan Teks

Membangun dan menjalankan aplikasi

Buka aplikasi `ScrollingText2` yang Anda buat dalam pelajaran [Bekerja dengan Elemen TextView](#).

1. Ubah subjudul sehingga terbungkus dalam kolom di sebelah kiri yang berlebar 100dp, seperti yang ditampilkan di bawah ini.
2. Tempatkan teks artikel di sebelah kanan subjudul seperti yang ditampilkan di bawah ini.



4G



4:01

Scrolling Text

Beatles Anthology Vol. 1

Behind
That
Locked
Door:
Beatles
Rarities!

In a vault deep inside Abbey Road Studios in London – protected by an unmarked, triple-locked, police-alarmed door – are something like 400 hours of unreleased Beatles recordings, starting from June 2, 1962 and ending with the very last tracks recorded for the *Let It Be* album. The best of the best were released by Apple Records in the form of the 3-volume Anthology series. For more information, see the Beatles Time Capsule at www.rockument.com.

This volume starts with the first new Beatle song, “Free as a Bird” (based on a John Lennon demo, found only on the bootleg *The Lost Lennon Tapes Vol. 28*) and covers the very



Jawablah pertanyaan berikut

Pertanyaan 1

Berapa banyak Tampilan yang bisa dimuat sebuah ScrollView? Pilih salah satu:

- Hanya satu Tampilan
- Satu Tampilan atau satu ViewGroup
- Sebanyak yang Anda butuhkan

Pertanyaan 2

Apa atribut XML yang Anda gunakan dalam LinearLayout untuk menampilkan tampilan secara berdampingan? Pilih salah satu:

- android:orientation="horizontal"
- android:orientation="vertical"
- android:layout_width="wrap_content"

Pertanyaan 3

Apa atribut XML yang Anda gunakan untuk menentukan lebar LinearLayout dalam tampilan bergulir? Pilih salah satu:

- android:layout_width="wrap_content"
- android:layout_width="match_parent"
- android:layout_width="200dp"

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Layout memperlihatkan subjudul di kolom kiri dan teks artikel di kolom kanan, seperti yang ditunjukkan pada gambar di atas.
 - ScrollView mencakup sebuah LinearLayout dengan dua TextView.
 - Orientasi LinearLayout disetel ke horizontal.
-

1.4: Sumber Daya

Memuat dan menjalankan aplikasi yang sudah ada, mengeksplorasi sumber daya

1. Muat salah satu aplikasi contoh ke dalam Android Studio.
2. Buka salah satu file aktivitas Java di aplikasi. Carilah kelas, tipe, atau prosedur yang tidak Anda pahami dan carilah di dokumentasi Developer Android.
3. Masuklah ke Stackoverflow dan telusuri pertanyaan serta jawaban untuk topik yang sama.
4. Cari saluran Google Developers di YouTube. Cari daftar putar atau video tentang Android Studio dan tonton video tersebut.

Jawablah pertanyaan berikut

Pertanyaan 1

Pada Android Studio, apa perintah menu untuk membuka daftar aplikasi contoh?

Pertanyaan 2

Apa yang Anda cari, dan apa URL untuk dokumentasi yang Anda temukan?

Pertanyaan 3

Apa 2 perbedaan antara jenis informasi yang Anda temukan dalam dokumentasi Developer Android dan Stackoverflow?
Kapan Anda menggunakan dokumentasi Developer Android? Kapan Anda menggunakan Stackoverflow?

Pertanyaan 4

Apa URL ke Android Studio Playlist atau video yang Anda tonton? Apa yang Anda pelajari?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Tidak ada aplikasi untuk mengirimkan tugas pekerjaan rumah ini.

Tugas Pekerjaan Rumah: Pelajaran 2

Daftar Isi:

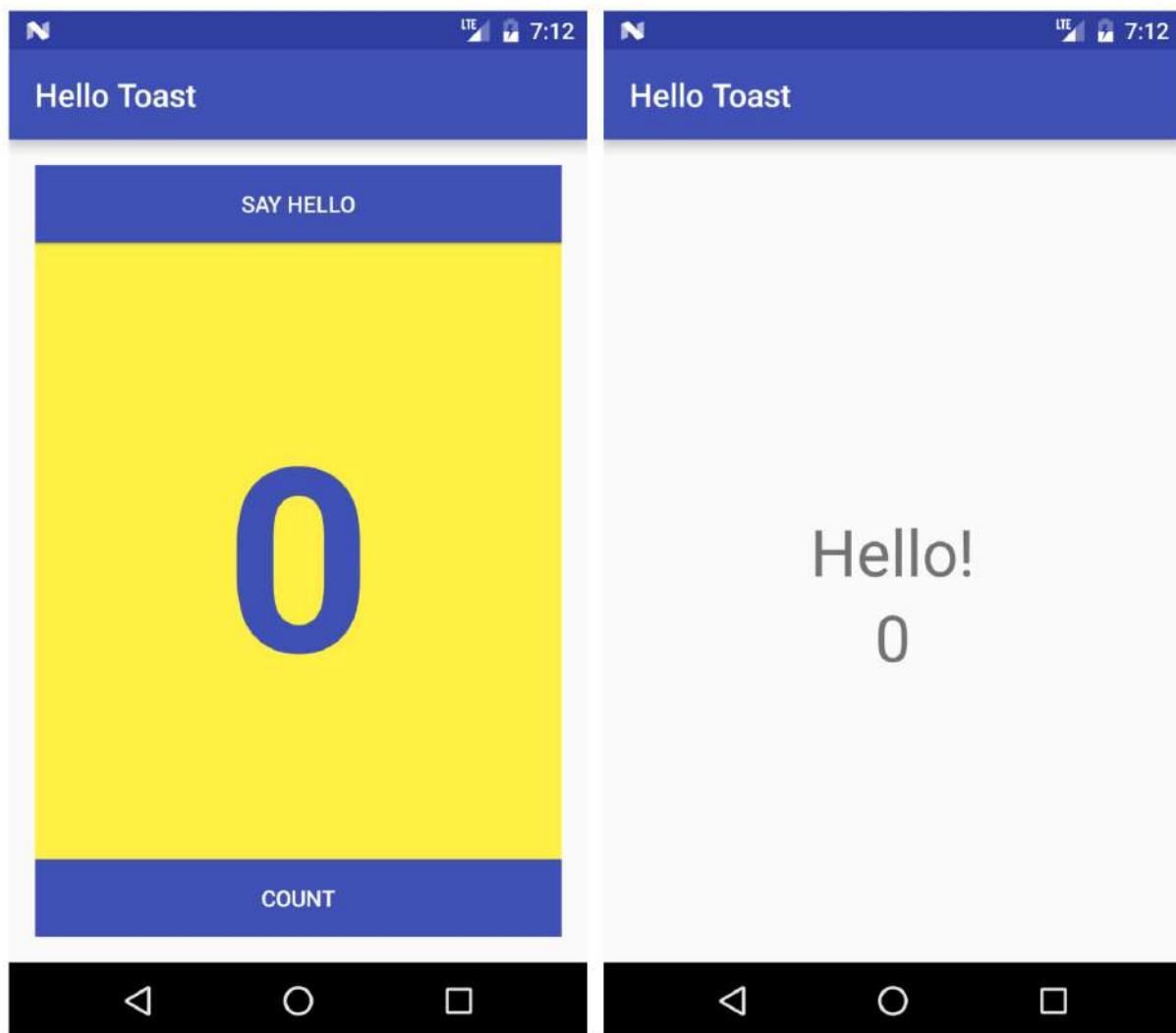
- 2.1: Membuat dan Memulai Aktivitas
- 2.2: Daur Hidup Aktivitas dan Mengelola Status
- 2.3: Memulai Aktivitas dengan Maksud Implisit

2.1: Membuat dan Memulai Aktivitas

Membangun dan menjalankan aplikasi

Buka aplikasi [HelloToast](#) yang Anda buat pada pelajaran [Membuat UI Interaktif Pertama Anda](#).

1. Modifikasi tombol toast sehingga itu meluncurkan aktivitas baru untuk menampilkan kata "Hello!" dan penghitungan saat ini, seperti yang ditampilkan di bawah ini.
2. Ubah teks pada tombol toast.



Jawablah pertanyaan berikut

Pertanyaan 1

Apakah perintah menu yang Anda gunakan untuk menambahkan aktivitas baru ke dalam aplikasi?

Pertanyaan 2

Apa file yang ditambahkan ketika Anda menambahkan aktivitas baru yang disebut HelloActivity ke dalam aplikasi? Apa perubahan yang dibuat ke file yang sudah ada?

Pertanyaan 3

Apa metode konstruktur yang Anda gunakan untuk membuat maksud eksplisit yang baru?

- `new Intent()`
- `new Intent(Context context, Class<?> class)`
- `new Intent(String action, Uri uri)`
- `new Intent(String action)`

Pertanyaan 4

Bagaimana Anda menambahkan nilai hitungan saat ini ke dalam maksud?

- Sebagai data maksud
- Sebagai tindakan maksud
- Sebagai ekstra maksud

Pertanyaan 5

Bagaimana Anda memperbarui hitungan di HelloActivity untuk menampilkan hitungan saat ini?

- Mendapatkan maksud yang digunakan untuk meluncurkan aktivitas.
- Mendapatkan nilai hitungan saat ini dari maksud.
- Memperbarui tampilan teks untuk hitungan.
- Semua hal di atas.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

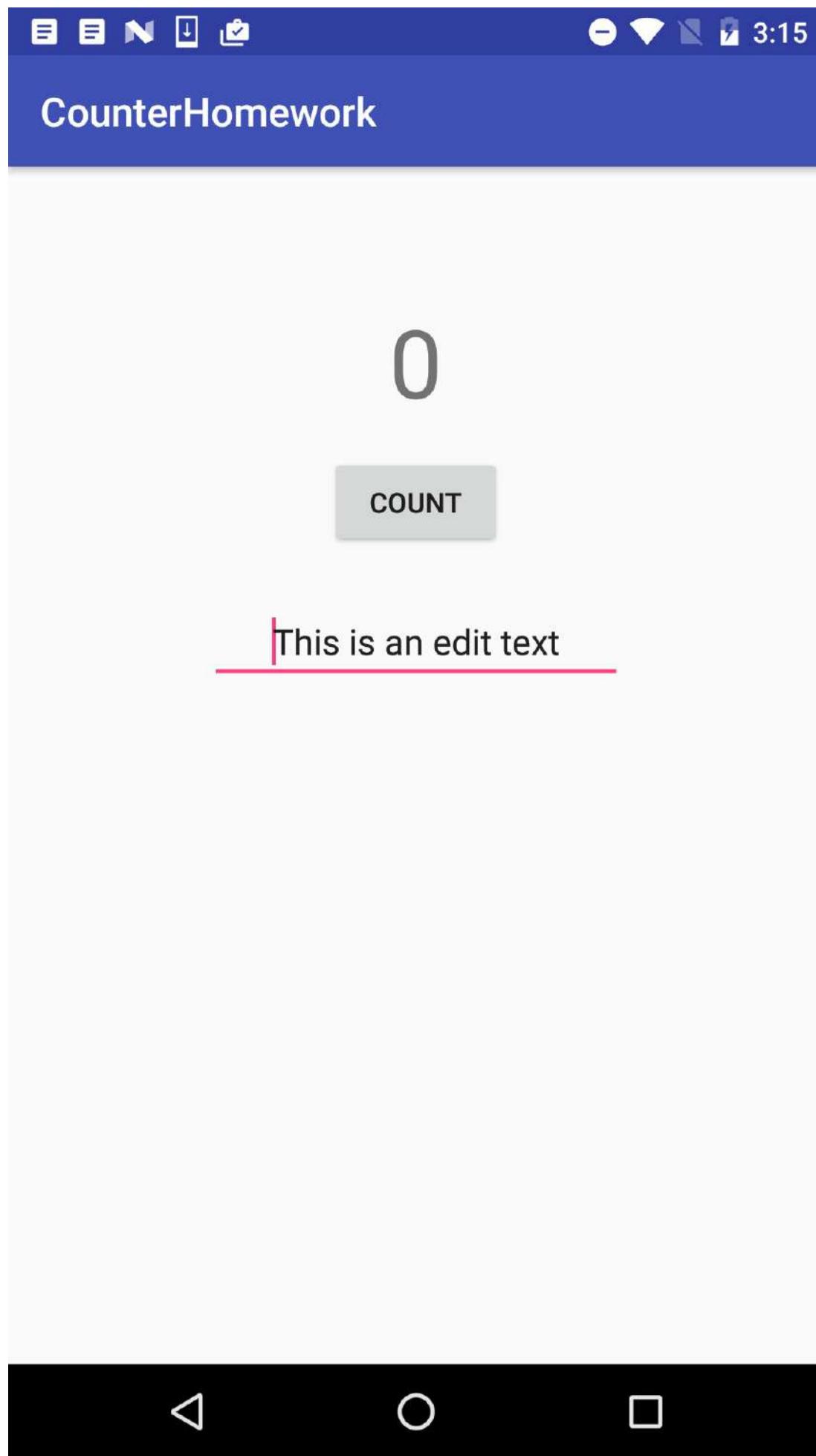
- Aplikasi menampilkan tombol **Hello** sebagai ganti tombol **Hello Toast**.
 - HelloActivity dimulai ketika tombol **Hello** ditekan, dan aktivitas baru menampilkan pesan "Hello!" dan hitungan saat ini dari aktivitas utama.
 - File HelloActivity.java dan activity_hello.xml telah ditambahkan ke proyek.
 - File activity_hello.xml berisi dua objek tampilan teks, satu dengan string Hello! dan yang kedua dengan hitungan.
 - Aplikasi mencakup implementasi metode penanganan klik untuk tombol **Hello** (di MainActivity).
 - Aplikasi mencakup implementasi metode `onCreate()` untuk HelloActivity dan memperbarui TextView hitungan dengan hitungan dari MainActivity.
-

2.2: Daur Hidup Aktivitas dan Mengelola Status

Membangun dan menjalankan aplikasi

1. Membuat aplikasi dengan layout yang berisi penghitung, tombol untuk menaikkan hitungan, dan teks edit. Lihat tangkapan layar di bawah sebagai contoh. Anda tidak harus secara persis membuat duplikat layout.

2. Tambahkan penangan klik untuk tombol yang menaikkan nilai penghitung.
3. Jalankan aplikasi dan naikkan nilai penghitung. Masukkan beberapa teks ke dalam teks edit.
4. Putar perangkat. Perhatikan bahwa penghitung akan disetel ulang, namun materi dari teks edit tidak.
5. Implementasikan `onSaveInstanceState()` untuk menyimpan keadaan aplikasi saat ini.
6. Perbarui `onCreate()` untuk memulihkan keadaan aplikasi.
7. Putar perangkat dan pastikan bahwa keadaan aplikasi tetap dipertahankan.



Jawablah pertanyaan berikut

Pertanyaan 1

Bila Anda memutar perangkat (sebelum Anda mengimplementasikan `onSaveInstanceState()`), penghitung akan disetel ulang ke 0 namun materi dari teks edit tetap dipertahankan. Mengapa?

Pertanyaan 2

Apa metode daur hidup Aktivitas yang dipanggil ketika terjadi perubahan konfigurasi-perangkat (seperti rotasi) terjadi?

Pertanyaan 3

Kapan `onSaveInstanceState()` dipanggil dalam daur hidup aktivitas?

Pertanyaan 4

Manakah yang merupakan tanda tangan metode yang tepat untuk `onSaveInstanceState()` :

- `void onSaveInstanceState(Bundle outState)`
- `void onSaveInstanceState()`
- `void onSaveInstanceState(Bundle outState, PersistableBundle outPersistentState)`

Pertanyaan 5

Apa perbedaan antara memulihkan status aktivitas dalam `onCreate()` dibandingkan dalam `onRestoreInstanceState()` ?

Pertanyaan 6

Jika Anda menghentikan dan memulai ulang aplikasi, apa yang terjadi pada status Aktivitas?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Aplikasi menampilkan penghitung, tombol untuk menaikkan penghitung tersebut, dan sebuah teks edit.
 - Mengeklik tombol akan menaikkan penghitung sebesar 1.
 - Bila perangkat diputar, keadaan penghitung dan teks edit tetap dipertahankan.
 - Implementasi `MainActivity.java` menggunakan metode `onSaveInstanceState()` untuk menyimpan nilai penghitung.
 - Implementasi pengujian `onCreate()` untuk eksistensi bundel `outState`. Jika bundel tersebut ada, nilai penghitung akan dikembalikan dan disimpan ke tampilan teks.
-

2.3: Memulai Aktivitas dengan Maksud Implisit

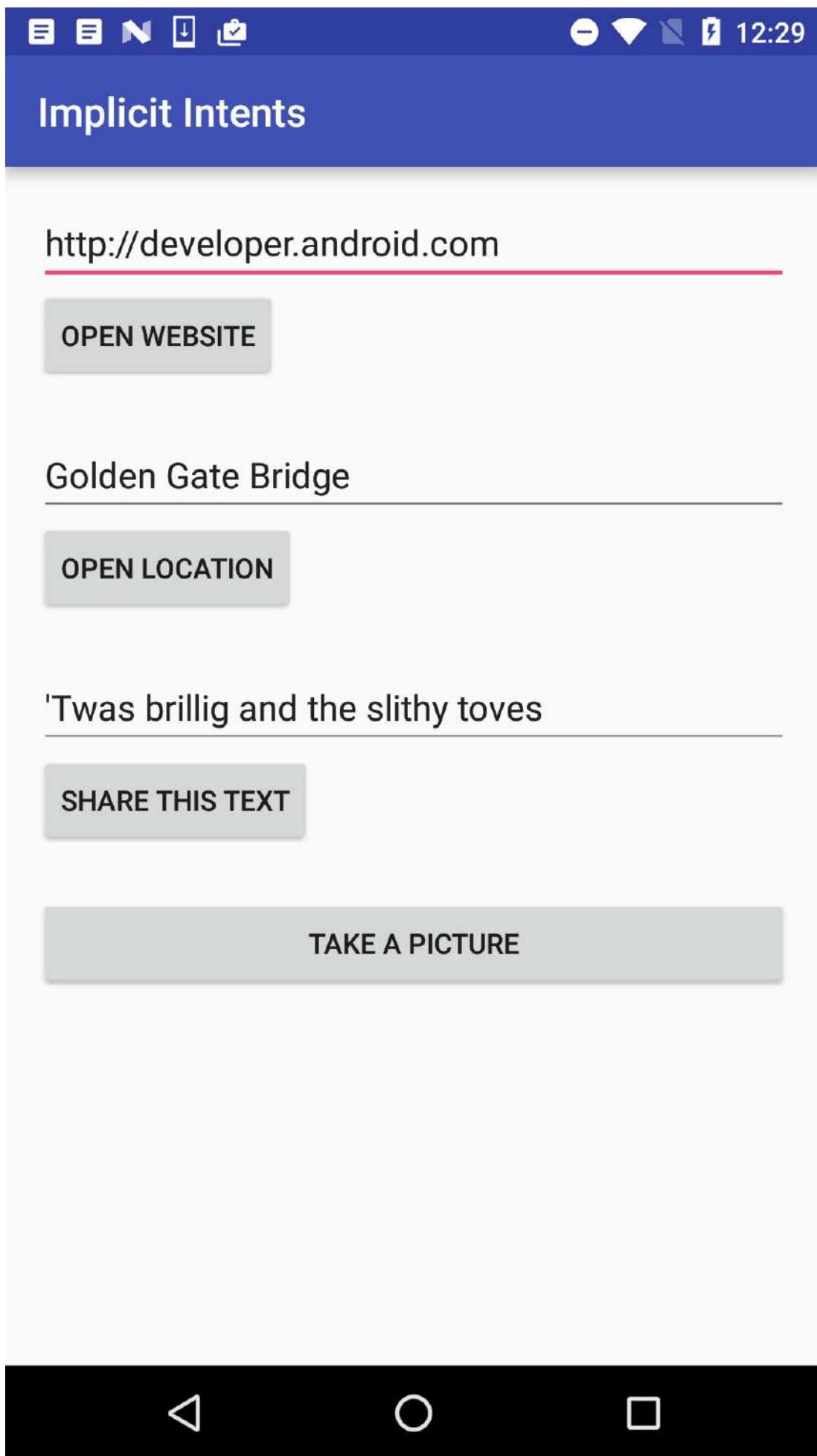
Membangun dan menjalankan aplikasi

Buka [aplikasi ImplicitIntents](#) yang Anda buat dalam pelajaran [Memulai Aktivitas dengan Maksud Implisit](#).

1. Tambahkan tombol lain di bagian bawah layar.
2. Ketika tombol diklik, luncurkan aplikasi kamera untuk mengambil gambar. (Anda tidak perlu mengembalikan gambar

ke aplikasi asli.)

Catatan: Jika Anda menggunakan emulator Android untuk menguji kamera, buka konfigurasi emulator di Android AVD manager, pilih Advanced Settings, kemudian pilih "Emulated" untuk kamera depan dan belakang. Mulai ulang emulator Anda jika dibutuhkan.



Jawablah pertanyaan berikut

Pertanyaan 1

Apa metode konstruktur yang Anda gunakan untuk membuat maksud implisit guna meluncurkan aplikasi kamera?

- new Intent()
- new Intent(Context context, Class<?> class)
- new Intent(String action, Uri uri)
- new Intent(String action)

Pertanyaan 2

Apa tindakan maksud yang Anda gunakan untuk meminta aplikasi kamera?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Aplikasi menampilkan tombol "Take a Picture" di bagian bawah aplikasi.
- Ketika diklik, tombol meluncurkan aplikasi kamera pada perangkat.
- Metode on click untuk tombol Take a Picture memastikan bahwa tersedia aplikasi pada perangkat (dengan metode resolveActivity() dan getPackageManager()) sebelum mengirim maksud.

Tugas Pekerjaan Rumah: Pelajaran 3 & 4

Daftar Isi:

- [3.1: Debugging](#)
- [3.2: Pengujian](#)
- [3.3: Pustaka Dukungan dan Kompatibilitas Mundur](#)
- [4.1: Kontrol Masukan Pengguna](#)
- [4.2: Menu](#)
- [4.3: Navigasi Layar](#)
- [4.4: RecyclerView](#)

3.1: Debugging

Membangun dan menjalankan aplikasi

Buka aplikasi [SimpleCalc](#) dari pelajaran [Menggunakan Debugger](#).

1. Dalam MainActivity, tempatkan breakpoint pada baris pertama dari metode `onAdd()`.
2. Jalankan aplikasi di debugger. Lakukan operasi tambah di aplikasi. Eksekusi berhenti di breakpoint.
3. Gunakan tombol **Step Into** untuk mengikuti langkah demi langkah eksekusi aplikasi. Perhatikan bahwa **Step Into** membuka dan menjalankan file dari kerangka kerja Android, memungkinkan Anda untuk melihat bagaimana Android beroperasi pada kode Anda.
4. Periksa bagaimana jendela debugger berubah saat Anda melangkah melalui kode dalam bingkai tumpukan dan variabel lokal saat ini.
5. Periksa bagaimana kode tersebut beranotasi di jendela editor ketika setiap baris dieksekusi.
6. Gunakan tombol **Step Out** untuk kembali ke aplikasi Anda jika tumpukan eksekusi terlalu dalam untuk dipahami.

Jawablah pertanyaan berikut

Pertanyaan 1

Apa perbedaan antara **Step Over** dan **Step Into**?

Pertanyaan 2

Bagaimana setiap bagian dari tampilan debugger berubah ketika Anda masuk ke metode yang baru?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Tidak ada aplikasi untuk mengirimkan tugas pekerjaan rumah ini.

3.2: Pengujian

Membangun dan menjalankan aplikasi

Buka aplikasi `SimpleCalcTest` yang Anda buat dalam pelajaran [Menguji Aplikasi dengan Pengujian Unit](#). Anda akan menambahkan tombol **POW** ke layout. Tombol ini menghitung operand pertama yang diberi pangkat operand kedua. Misalnya, saat diberikan operand 5 dan 4, aplikasi menghitung 5 pangkat 4, atau 625.

SEBELUM Anda menulis implementasi tombol daya, pertimbangkan jenis pengujian yang ingin Anda lakukan dengan penghitungan ini. Apa nilai-nilai tidak biasa yang bisa muncul dalam penghitungan ini?

1. Perbarui kelas Kalkulator di aplikasi agar menyertakan metode `pow()`. (Petunjuk: Bacalah dokumentasi kelas `java.lang.Math`.)
2. Perbarui kelas MainActivity untuk menghubungkan tombol **POW** ke dalam penghitungan.
3. Tulis masing-masing pengujian ke metode `pow()` Anda. Jalankan suite pengujian setiap kali Anda menulis pengujian, dan memperbaiki penghitungan awal dalam aplikasi bila diperlukan.
 - Pengujian dengan operand integer positif.
 - Pengujian dengan integer negatif sebagai operand pertama.
 - Pengujian dengan integer negatif sebagai operand kedua.
 - Pengujian dengan 0 sebagai operand pertama dan integer positif sebagai operand kedua.
 - Pengujian dengan 0 sebagai operand kedua.
 - Pengujian dengan 0 sebagai operand pertama dan -1 sebagai operand kedua. (Petunjuk: baca dokumentasi untuk `Double.POSITIVE_INFINITY`.)
 - Pengujian dengan -0 sebagai operand pertama dan semua angka negatif sebagai operand kedua.

Jawablah pertanyaan berikut

Tidak ada pertanyaan.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Aplikasi menampilkan tombol **POW** yang menyediakan penghitungan eksponensial ("pangkat").
- Implementasi MainActivity menyertakan metode on click untuk tombol **POW**.
- Implementasi Kalkulator menyertakan metode `pow()` yang melakukan penghitungan.
- Metode `CalculatorTest` menyertakan metode pengujian terpisah untuk metode `pow()` pada kelas Kalkulator yang melakukan pengujian untuk operand negatif dan 0, dan saat 0 dan -1 bertindak sebagai operand.

3.3: Pustaka Dukungan dan Kompatibilitas Mundur

Menjalankan aplikasi

Buka aplikasi `HelloCompat` yang dibuat dalam pelajaran [Menggunakan Pustaka Dukungan Android](#).

1. Setel breakpoint debugger pada baris dalam metode `changeColor()` yang benar-benar mengubah warna:

```
int colorRes = ContextCompat.getColor(this, colorResourceName);
```

2. Jalankan aplikasi dalam mode debug pada perangkat atau emulator yang menjalankan versi API 23 atau yang lebih tinggi. Melangkah masuk ke metode `getColor()`, mengikuti panggilan metode yang masuk lebih dalam ke tumpukan. Periksa bagaimana kelas `ContextCompat` menentukan cara mendapatkan warna dari sumber daya, dan kelas kerangka kerja lain yang digunakannya.

Catatan: Beberapa kelas mungkin menghasilkan peringatan "source code does not match the bytecode." Klik **Step Out** untuk kembali ke file sumber yang dikenal, atau terus mengeklik **Step Into** sampai debugger kembali dengan

- sendirinya.
3. Ulangi langkah sebelumnya untuk perangkat atau emulator yang menjalankan versi API lebih rendah dari 23. Perhatikan jalur berbeda yang digunakan kerangka kerja untuk mendapatkan warna.

Jawablah pertanyaan berikut

Pertanyaan 1

Berdasarkan eksplorasi Anda di debugger, bagaimana platform Android memutuskan implementasi yang akan digunakan dalam kelas Compat?

Pertanyaan 2

Selain perbedaan dalam tanda tangan metode, apa perbedaan di antara implementasi tersebut? Mengapa kelas kompatibilitas diperlukan?

Mengirimkan aplikasi Anda untuk dinilai

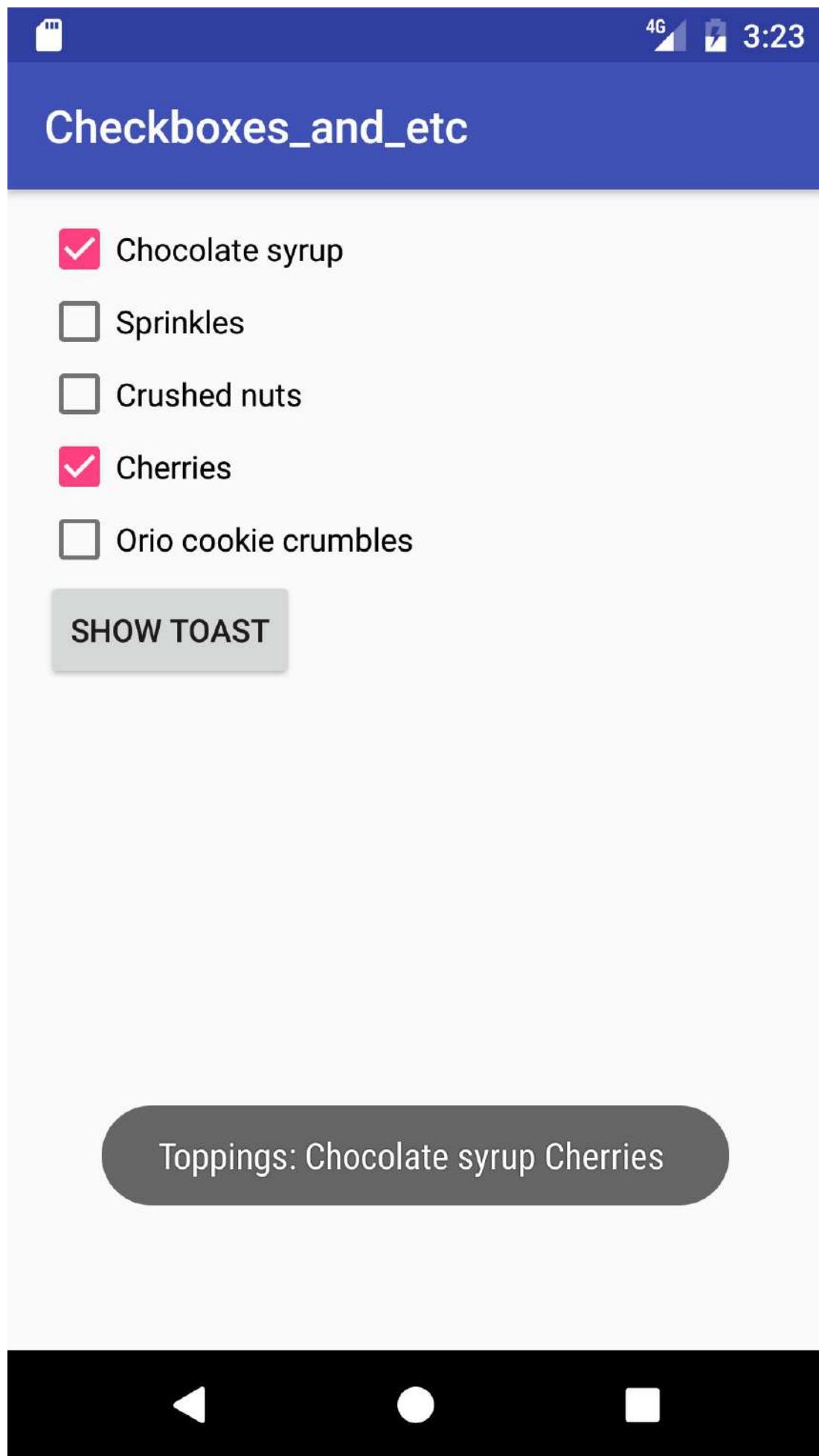
Panduan untuk penilai

Tidak ada aplikasi untuk mengirimkan tugas pekerjaan rumah ini.

4.1: Kontrol Masukan Pengguna

Membangun dan menjalankan aplikasi

1. Buat aplikasi dengan 5 kotak centang dan tombol **Show Toast**, seperti yang ditampilkan di bawah ini.
2. Ketika pengguna mengeklik satu kotak centang kemudian **Show Toast**, tampilkan pesan toast yang menampilkan kotak centang yang dipilih.
3. Bila pengguna memilih lebih dari satu kotak centang kemudian **Show Toast**, tampilkan toast yang memuat pesan untuk semua kotak centang yang dipilih, seperti yang ditunjukkan pada gambar di bawah ini.



Jawablah pertanyaan berikut

Pertanyaan 1

Apa perbedaan paling penting antara kotak centang dan RadioGroup pada tombol radio? Pilih salah satu:

- Satu-satunya perbedaan adalah bagaimana semuanya terlihat: kotak centang menampilkan tanda centang saat dipilih, sementara tombol "radio" bundar akan terisi warna saat dipilih.
- Elemen `CheckBox` dalam layout bisa menggunakan atribut `android:onClick` untuk memanggil penangan saat dipilih.
- Perbedaan utamanya adalah bahwa kotak centang memungkinkan beberapa pilihan, sementara RadioGroup hanya mengizinkan satu pilihan.

Pertanyaan 2

Grup layout manakah yang lebih disukai untuk menyelaraskan kumpulan elemen `CheckBox` secara vertikal? Pilih salah satu:

- `RelativeLayout`
- `LinearLayout`
- `ScrollView`

Pertanyaan 3

Apa metode [antarmuka Checkable](#) yang Anda gunakan untuk memeriksa keadaan kotak centang (apakah dicentang atau tidak)?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Layout memuat lima tampilan CheckBox yang sejajar secara vertikal di layar, dan tombol **Show Toast**.
 - Metode `onSubmit()` menentukan kotak centang yang dicentang dengan menggunakan `findViewById()` bersama `isChecked()`.
 - String yang menjelaskan topping digabungkan menjadi pesan toast.
-

4.2: Menu

Membangun dan menjalankan aplikasi

Buka aplikasi [ScrollingText](#) yang Anda buat dalam pelajaran [Bekerja dengan Elemen TextView](#).

1. Tambahkan menu konteks mengambang untuk menunjukkan tiga pilihan menu: **Edit**, **Share**, dan **Delete**, seperti yang terlihat pada gambar di bawah ini. Menu akan muncul ketika pengguna melakukan klik lama pada TextView.
2. Tambahkan pesan log untuk menampilkan item menu yang diklik.

The image shows a smartphone screen displaying a news article. The top status bar indicates a 4G signal and the time 4:04. The title of the article is "Scrolling Text" in a blue header bar. Below the title, the main content features a large yellow header "Beatles Anthology Vol. 1". The main text discusses unreleased Beatles recordings from June 1962 to the end of 1966, mentioning the "Let It Be" album and the Beatles Time Capsule. A vertical ellipsis on the right side of the text block reveals the continuation of the story. To the right of the text block is a white rectangular menu with three options: "Edit", "Share", and "Delete". At the bottom of the screen is a black navigation bar with standard Android icons for back, home, and recent apps.

Behind That Locked Door: Beatles Rarities!

In a vault deep inside Abbey Road Studios in London – protected by an unmarked, triple-locked, police-alarmed door – are something like 400 hours of unreleased Beatles recordings, starting from June 2, 1962 and ending with the *Let It Be* album for the *Let It Be* album released by Apple Records. This volume Anthology set is the Beatles Time Capsule.

This volume starts with the first new Beatle song, “Free as a Bird” (based on a John Lennon demo, found only on the bootleg *The Lost Lennon Tapes* Vol. 28), and covers the very earliest historical recordings, outtakes from the first albums, and live recordings from early concerts and BBC Radio sessions.

Edit
Share
Delete

Jawablah pertanyaan berikut

Pertanyaan 1

Apa nama dan lokasi file yang menjadi tempat Anda membuat item menu konteks?

Pertanyaan 2

Apa yang terjadi ketika ketuk lama (juga dikenal sebagai klik lama) terjadi? Pilih salah satu:

- Bila tampilan menerima kejadian klik-lama, sistem akan memanggil metode `onCreateContextMenu()`, yang tidak bisa Anda ubah.
- Bila tampilan yang telah didaftarkan menerima kejadian klik-lama, sistem akan memanggil metode `onCreateContextMenu()` yang bisa Anda ganti dalam aktivitas atau fragmen.
- Bila tampilan yang telah didaftarkan menerima kejadian klik-lama, sistem akan memanggil metode `onContextItemSelected()` yang bisa Anda ganti dalam aktivitas atau fragmen.

Pertanyaan 3

Di mana Anda mendaftarkan menu konteks untuk tampilan? Pilih salah satu:

- Menggunakan `registerForContextMenu()` dalam metode `onCreate()`.
- Menggunakan `registerForContextMenu()` dalam metode `onCreateContextMenu()`.
- Menggunakan `getMenuInflater()` dalam metode `onCreateContextMenu()`.

Pertanyaan 4

Di mana Anda mengembangkan menu konteks menggunakan MenuInflater? Pilih salah satu:

- Dalam metode `onCreate()`.
- Dalam metode `onCreateContextMenu()`.
- Dalam metode `onContextItemSelected()`.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

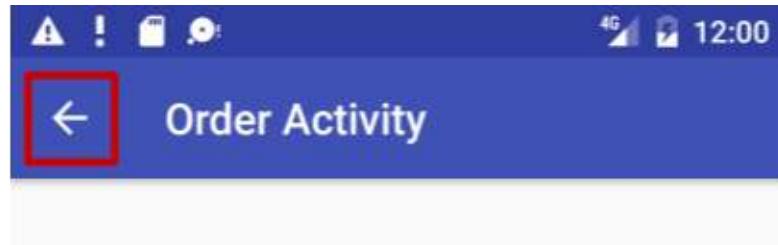
- Metode `onCreateContextMenu()` diimplementasikan dalam kelas MainActivity dan menggunakan MenuInflater untuk mengembangkan menu konteks.
- File menu_context.xml memuat tiga pilihan: **Edit, Share, dan Delete**.
- Metode `onContextItemSelected()` diimplementasikan dan menggunakan `getItemId()` untuk menentukan item menu yang dipilih.

4.3: Navigasi Layar

Membangun dan menjalankan aplikasi

Membuat aplikasi dengan sebuah aktivitas utama dan minimal tiga aktivitas lainnya. Semua aktivitas memiliki menu Opsi dasar dan menggunakan Bilah Alat [pustaka dukungan](#) v7 appcompat sebagai bilah aplikasi, seperti yang ditampilkan di bawah ini.

1. Pada aktivitas utama, bangun layout grid dengan gambar pilihan Anda sendiri (atau menggunakan gambar dalam [4_1_P_starter_images.zip](#)). Ubah ukuran gambar sehingga ketiganya pas secara horizontal pada layar dalam layout grid.
2. Aktifkan setiap gambar agar menyediakan navigasi ke aktivitas lain.
 - Bila pengguna mengetuk gambar, itu akan memulai aktivitas lainnya.
 - Dari aktivitas lain, pengguna bisa mengetuk tombol **Up** dalam bilah aplikasi (disorot dalam gambar di bawah ini)



untuk kembali ke aktivitas utama.

Jawablah pertanyaan berikut

Pertanyaan 1

Template manakah yang menyediakan aktivitas dengan menu opsi, tombol **Up**, dan Bilah Alat [pustaka dukungan v7 appcompat](#) sebagai bilah aplikasi?

Pertanyaan 2

Mengapa Anda menggunakan [GridLayout](#) dibandingkan dengan [LinearLayout](#) atau [RelativeLayout](#) untuk menyediakan navigasi menggunakan gambar?

Pertanyaan 3

Di mana Anda menempatkan [GridLayout](#) gambar? Pilih salah satu:

- Pada `activity_main.xml` untuk `MainActivity`.
- Pada `content_main.xml` untuk `MainActivity`.
- Pada file layout XML "utama" untuk setiap aktivitas lainnya.
- Pada file layout XML "materi" untuk setiap aktivitas lainnya.

Pertanyaan 4

Di mana Anda menetapkan aktivitas aplikasi dan aktivitas induk untuk menyediakan navigasi Up? Pilih salah satu:

- Untuk menyediakan tombol **Up** bagi aktivitas layar anak, deklarasikan induk aktivitas anak dalam file `activity_main.xml`.
- Untuk menyediakan tombol **Up** bagi aktivitas layar anak, deklarasikan induk aktivitas dalam file `AndroidManifest.xml`.
- Untuk menyediakan tombol **Up** bagi aktivitas layar anak, deklarasikan induk aktivitas anak dalam file layout XML "utama" untuk aktivitas layar anak.

Pertanyaan 5

Teknik manakah yang Anda gunakan untuk memulai aktivitas lain dari gambar navigasi? Pilih salah satu:

- Menggunakan atribut `android:onClick` dengan `ImageView` dalam layout XML untuk memanggil metode publik dalam aktivitas yang berhubungan dengan layout.
- Menggunakan kode berikut dalam metode publik (dengan asumsi bahwa aktivitas lain disebut `OtherActivity`):
`Intent intent = new Intent(this, OtherActivity.class); startActivity(intent);`
- Kedua hal di atas.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

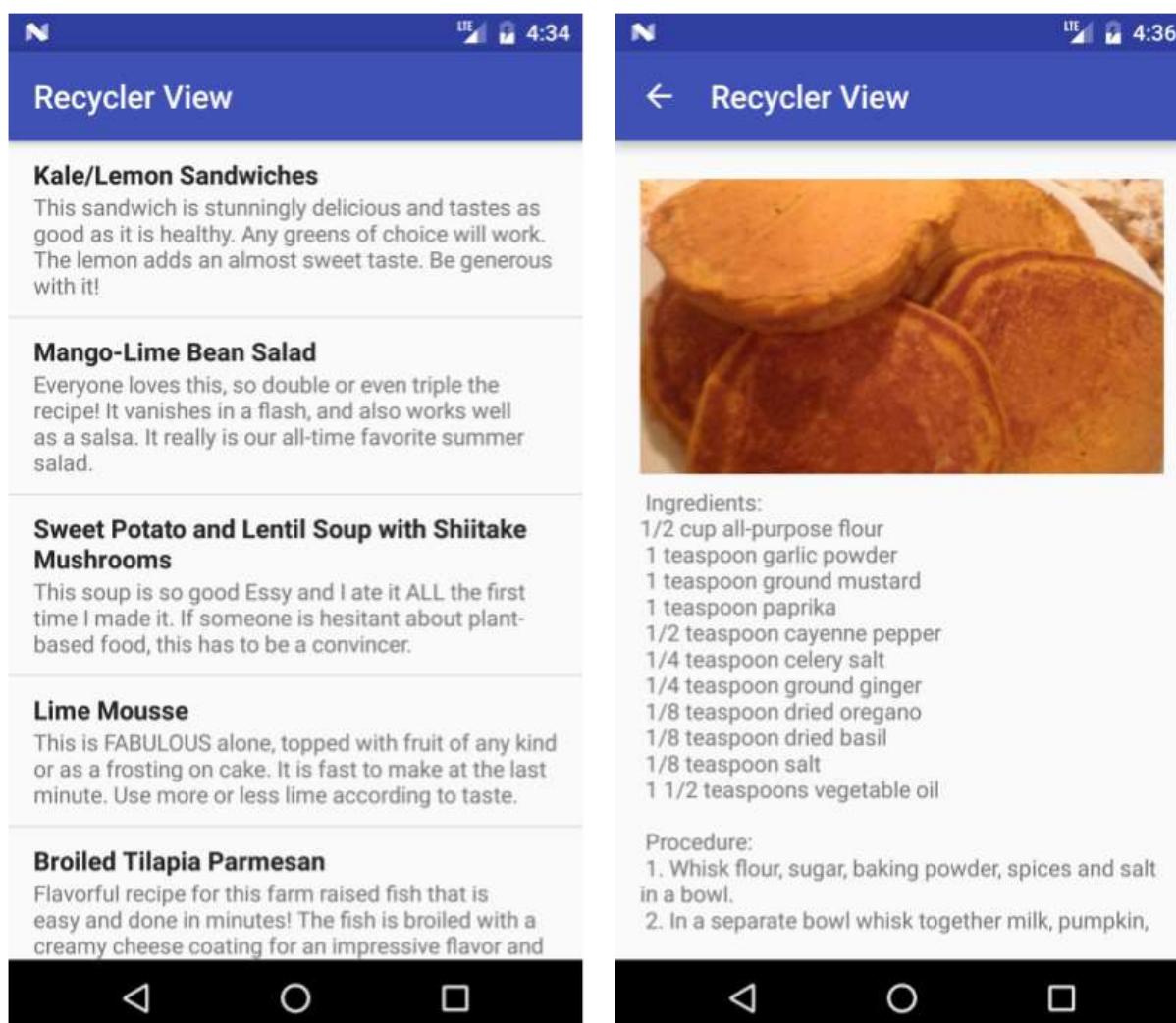
- GridLayout dalam file content_main.xml.
- Sebuah Intent baru dan metode startActivity() untuk setiap elemen navigasi dalam grid.
- Sebuah kelas aktivitas terpisah untuk setiap elemen navigasi dalam grid.

4.4: RecyclerView

Membangun dan menjalankan aplikasi

1. Membuat aplikasi yang menggunakan RecyclerView untuk menampilkan daftar resep.
 - Setiap item daftar menampilkan nama resep dengan deskripsi singkat. Menggunakan tampilan TextView dan penataan gaya terpisah untuk nama dan deskripsi resep.
2. Bila pengguna mengetuk resep (item dalam daftar), jalankan aktivitas yang menunjukkan teks resep penuh.
 - Anda bisa menggunakan teks Placeholder untuk resep penuh.
 - Opsional, tambahkan foto masakan untuk setiap resep yang sudah selesai dimasak.
 - Mengklik tombol up akan membawa pengguna kembali ke daftar resep.

Tangkapan layar di bawah ini menunjukkan contoh implementasi sederhana. Aplikasi Anda bisa terlihat sangat berbeda, asalkan memiliki fungsionalitas yang diperlukan.



Jawablah pertanyaan berikut

Pertanyaan 1

Apa saja komponen utama yang Anda butuhkan untuk menampilkan daftar resep? Cek semua yang berlaku.

- RecyclerView
- RecyclerView.Adapter
- RecyclerView.ViewHolder
- AppCompatActivity

Pertanyaan 2

Apa kelas yang harus Anda implementasikan untuk mendengarkan dan merespons klik pengguna?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Mengimplementasikan RecyclerView yang menampilkan daftar judul resep dapat digulir dan deskripsi singkat.
- Kode yang memperluas atau mengimplementasikan RecyclerView, RecyclerView.Adapter, RecyclerView.ViewHolder, dan View.OnClickListener.
- Mengklik pada item daftar akan memulai aktivitas yang menampilkan resep penuh.
- File manifest mendefinisikan hubungan induk sehingga mengeklik tombol **Up** dalam tampilan resep akan mengembalikan ke daftar resep.
- ViewHolder berisi sebuah layout dengan dua TextViews; misalnya, sebuah LinearLayout dengan dua TextViews.

Tugas Pekerjaan Rumah: Pelajaran 5 & 6

Daftar Isi:

- 5.1: Sumber Daya Dapat Digambar, Tema, Gaya
- 5.2: Desain Material
- 5.3: Menyediakan Sumber Daya untuk Layout Adaptif
- 6.1: Menguji Antarmuka Pengguna

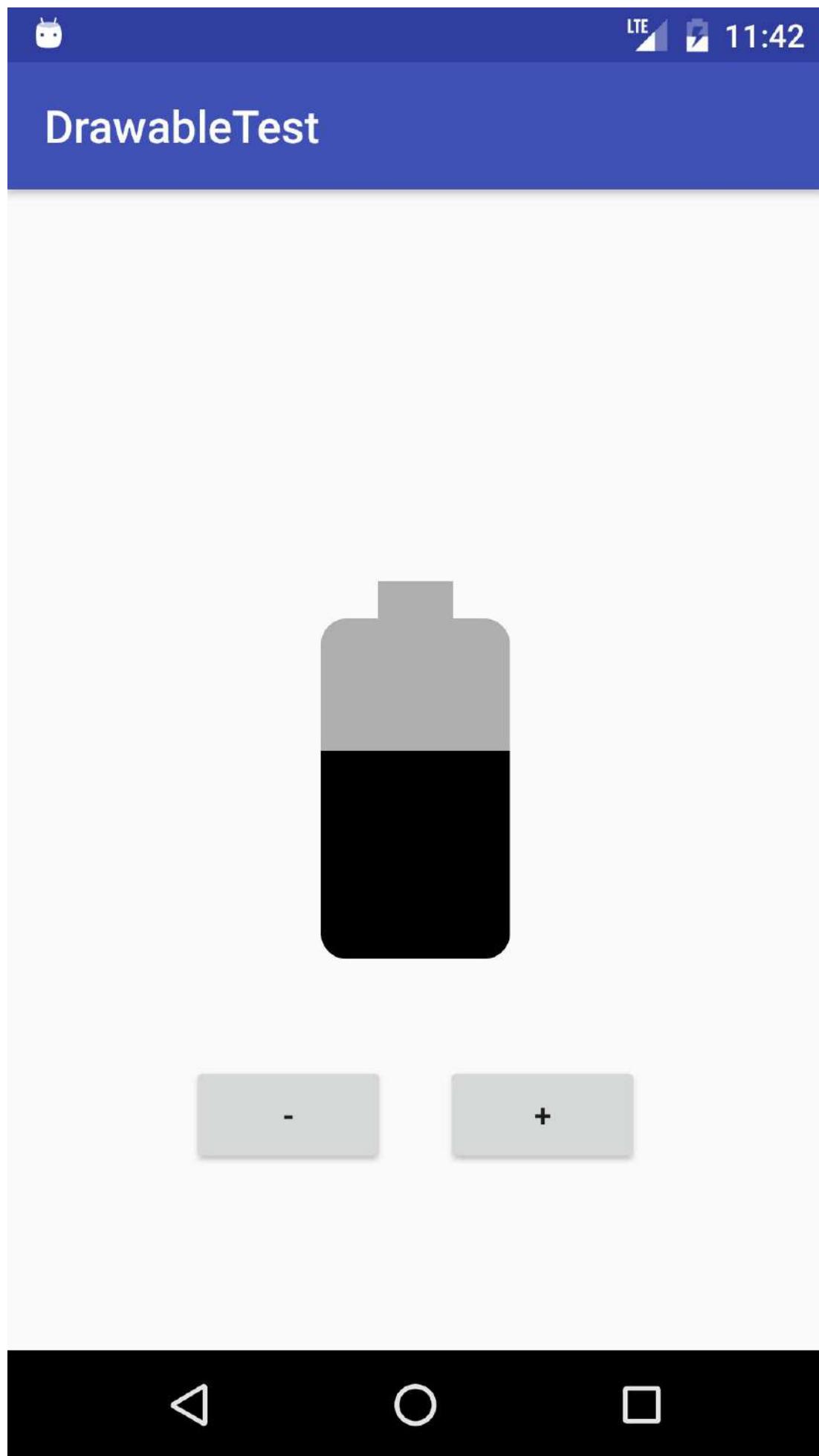
5.1: Sumber Daya Dapat Digambar, Tema, Gaya

Membangun dan menjalankan aplikasi

Buat aplikasi yang menampilkan ImageView serta tombol plus dan minus, seperti yang terlihat di bawah ini. ImageView berisi [daftar tingkat dapat digambar](#) yang merupakan indikator tingkat daya baterai. Menekan tombol plus atau minus akan mengubah tingkat indikator. Gunakan ikon baterai dari Vector Asset Studio untuk merepresentasikan 7 nilai berbeda untuk tingkat daya baterai.

Aplikasi memiliki properti berikut:

- Tombol plus akan menaikkan tingkat, menyebabkan indikator baterai terlihat lebih penuh.
- Tombol minus akan mengurangi tingkat, menyebabkan indikator mengosongkan satu tingkat.



Jawablah pertanyaan berikut

Pertanyaan 1

Apa dua tipe sumber daya dapat digambar yang Anda gunakan untuk membuat tombol yang menampilkan teks, dengan tombol yang memiliki sebuah latar belakang saat aktif dan latar belakang yang berbeda ketika dinonaktifkan, dan kedua latar belakang akan terentang ketika ukuran tombol lebih besar dari teks di dalamnya?

- LevelListDrawable
- TransitionDrawable
- StateListDrawable
- NinePatchDrawable

Pertanyaan 2

Misalkan Anda membuat sebuah aplikasi yang memiliki latar belakang gelap serta teks berwarna terang, dan aplikasi tersebut tidak membutuhkan ActionBar. Apa gaya dasar yang diwarisi oleh gaya aplikasi Anda?

- Theme.AppCompat.Light
- Theme.AppCompat.Dark.NoActionBar
- Theme.AppCompat.NoActionBar
- Theme.NoActionBar

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

- Tombol menaikkan variabel hitungan yang digunakan untuk menyetel tingkat pada ImageView menggunakan metode `setImageLevel()`.
 - Tingkatan dalam sumber daya dapat digambar LevelList berupa 0 sampai 6.
 - Tombol metode `onclick` akan memeriksa apakah variabel hitungan berada dalam rentang daftar tingkat sumber daya dapat digambar (0-6) sebelum menaikkan atau menurunkan tingkat gambar, sehingga Anda tidak bisa menyetel tingkat yang tidak ada.
-

5.2: Desain Material

Membangun dan menjalankan aplikasi

Buka aplikasi [MaterialMe](#) yang Anda buat dalam pelajaran [Lanskap Lesson Supporting](#), Beberapa Ukuran Layar dan Pelokalan.

1. Membuat [transisi elemen bersama](#) antara MainActivity dan DetailActivity, dengan gambar spanduk olahraga sebagai elemen bersama.
2. Mengklik item daftar dalam aplikasi MaterialMe akan memicu transisi. Gambar spanduk dari kartu akan bergerak ke bagian atas layar dalam tampilan Detail.

Jawablah pertanyaan berikut

Pertanyaan 1

Apa atribut warna dalam gaya Anda yang mendefinisikan warna bilah status?

- colorPrimary
- colorPrimaryDark
- colorAccent
- colorAccentDark

Pertanyaan 2

Apa pustaka dukungan yang memiliki Tombol Tindakan Mengambang?

- v4 Support Library
- v7 Support Library
- Design Support Library
- Custom Button Support Library

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

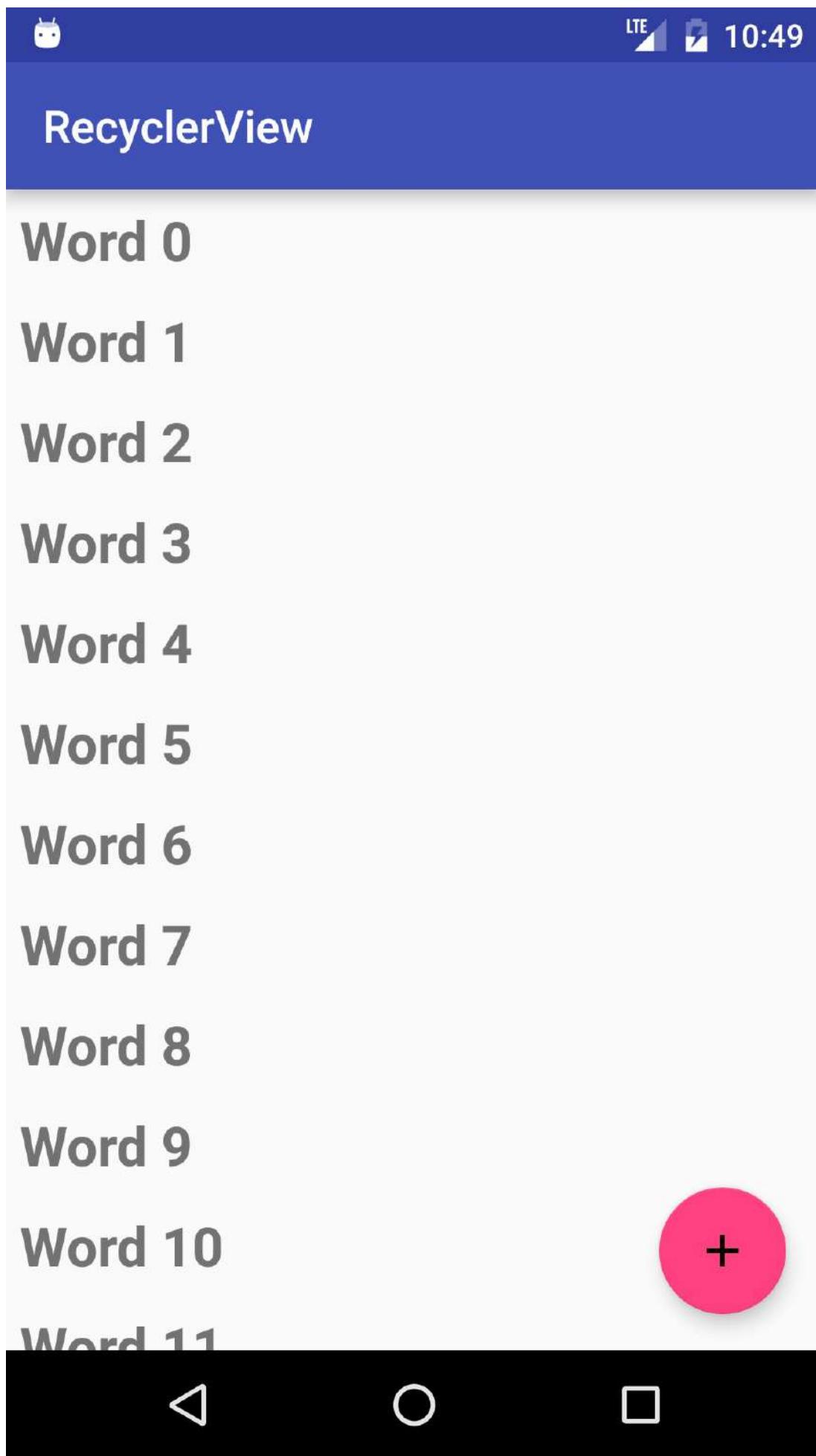
- Transisi materi-jendela yang diaktifkan dalam tema aplikasi.
 - Transisi elemen bersama yang ditetapkan dalam gaya aplikasi.
 - Transisi yang didefinisikan sebagai sumber daya XML.
 - Nama umum yang diberikan ke elemen bersama dalam kedua layout dengan atribut `android:transitionName`.
 - Kode yang menggunakan metode `ActivityOptions.makeSceneTransitionAnimation()`.
-

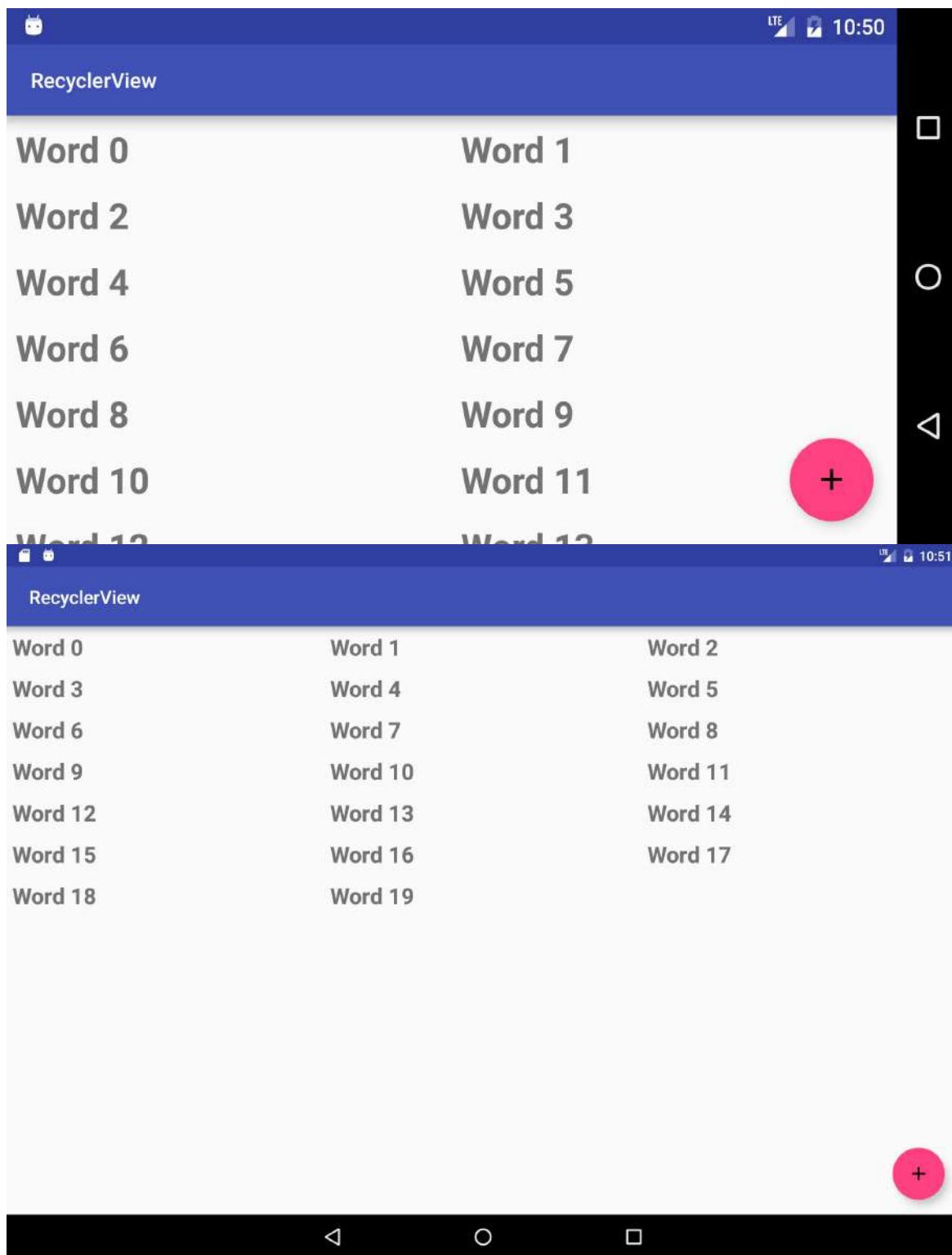
5.3: Menyediakan Sumber Daya untuk Layout Adaptif

Membangun dan menjalankan aplikasi

Buka [aplikasi RecyclerView](#) yang Anda buat dalam pelajaran [Membuat Recycler View](#). Memodifikasi aplikasi agar menggunakan LinearLayoutManager dengan hitungan kolom berikut:

1. Untuk ponsel:
 - i. 1 kolom pada potret
 - ii. 2 kolom pada lanskap
2. Untuk tablet:
 - i. 2 kolom pada potret
 - ii. 3 kolom pada lanskap





Jawablah pertanyaan berikut

Pertanyaan 1

Apa qualifier sumber daya yang dipakai untuk menentukan sumber daya yang akan digunakan ketika aplikasi Anda dalam mode malam?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Untuk ponsel dan tablet pada mode potret dan lanskap, kode memuat file nilai berkualifikasi sumber daya yang berisi integer untuk hitungan kolom.
 - Aplikasi menggunakan `getResources().getInteger()` untuk mengambil nilai dari file sumber daya, kemudian menggunakan nilai tersebut sebagai hitungan kolom untuk layout grid.
-

6.1: Menguji Antarmuka Pengguna

Menulis pengujian Espresso bagi aplikasi DroidCafe (dibuat dalam Bab 4.3P) yang menguji gambar dalam aktivitas utama untuk memastikan bahwa pengguna dibawa ke aktivitas kedua.

Membangun dan menjalankan aplikasi

Buka [aplikasi DroidCafe](#) yang Anda buat dalam pelajaran sebelumnya.

1. Membuat pengujian Espresso sebagai kelas Java dalam folder `com.example.android.droidcafe (androidTest)` (ditunjukkan dalam Project: Tampilan Android di folder `java`).
2. Membuat pengujian untuk setiap gambar dalam MainActivity yang:
 - i. Mengklik gambar.
 - ii. Memeriksa jika muncul Order Activity.

Jawablah pertanyaan berikut

Pertanyaan 1

Langkah-langkah manakah yang Anda lakukan untuk menguji interaksi, dan bagaimana urutannya? Masukkan nomor untuk setiap langkah, dari 1 sampai 3, untuk menentukan urutannya:

- Mencocokan tampilan: Mencari tampilan untuk menjalankan pengujian.
- Menyatakan dan memverifikasi hasil: Memeriksa status tampilan untuk memeriksa apakah statusnya mencerminkan keadaan atau perilaku yang diharapkan yang didefinisikan oleh pernyataan.
- Melakukan tindakan: Melakukan klik atau tindakan lain yang memicu sebuah kejadian dengan tampilan.

Pertanyaan 2

Manakah dari anotasi berikut yang mengaktifkan kelas pengujian JUnit 4 terinstrumentasi? Pilih salah satu:

- `@RunWith`
- `@Rule`
- `@Test`

Pertanyaan 3

Manakah dari anotasi berikut yang membentuk konteks untuk kode pengujian? Pilih salah satu:

- `@RunWith`
- `@Rule`
- `@Test`

Pertanyaan 4

Dalam tugas ini, Anda harus menguji setiap tampilan gambar yang digunakan untuk navigasi layar utama aplikasi DroidCafe dengan mengekliknya. Apakah Anda menggunakan `onView()` untuk mencari tampilan gambar, atau `onData()`, dan mengapa? Pilih salah satu:

- Saya menggunakan `onData()` karena tampilan yang ingin saya cari adalah tampilan gambar.
- Saya menggunakan `onView()` karena tampilan yang ingin saya cari terletak dalam hierarki tampilan saat ini dan ditampilkan pada layar. Metode `onData()` berguna untuk menemukan tampilan anak dalam AdapterView dengan terlebih dahulu memuat adaptor tampilan, kemudian mengaktifkan tampilan anak agar ditampilkan pada layar.
- Saya tidak akan menggunakan keduanya karena tampilan tersebut sudah terdapat dalam hierarki tampilan saat ini.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Menyertakan kelas pengujian di folder **com.example.android.droidcafe (androidTest)** dengan anotasi `@RunWith(AndroidJUnit4.class)`.
- Menyertakan pengujian terpisah (diberi keterangan `@Test`) untuk setiap gambar.
- Menggunakan metode `onView()`, `check()`, dan `perform()`.
- Lulus semua pengujian.

Tugas Pekerjaan Rumah: Pelajaran 7 & 8

Daftar Isi:

- [7.1: Membuat AsyncTask](#)
- [7.2: Terhubung ke Internet](#)
- [7.3: Penerima Siaran](#)
- [8.1: Notifikasi](#)
- [8.2: Alarm Manager](#)
- [8.3: JobScheduler](#)

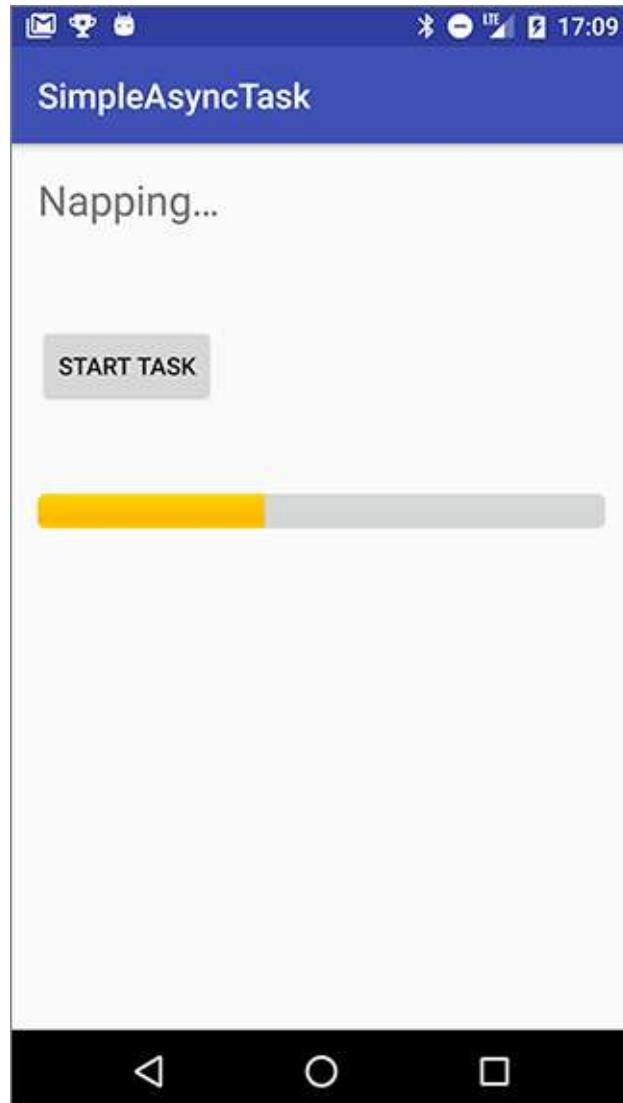
7.1: Membuat AsyncTask

Membangun dan menjalankan aplikasi

Buka aplikasi [SimpleAsyncTask](#) yang Anda buat dalam pelajaran [Membuat AsyncTask](#). Tambahkan [ProgressBar](#) yang menampilkan persentase total waktu tidur. Bilah kemajuan akan terisi saat thread AsyncTask tidur dengan nilai rentang 0 hingga 100 (persen).

Petunjuk: Pecah waktu tidur menjadi bagian yang lebih kecil.

Referensi AsyncTask: developer.android.com/reference/android/os/AsyncTask.html



Jawablah pertanyaan berikut

Pertanyaan 1

Untuk `ProgressBar` :

1. Bagaimana Anda menentukan rentang nilai yang bisa ditunjukkan `ProgressBar` ?
2. Bagaimana Anda mengubah berapa banyak bilah kemajuan yang terisi?

Pertanyaan 2

Jika `AsyncTask` ditetapkan seperti berikut:

```
private class DownloadFilesTask extends AsyncTask<URL, Integer, Long>
```

1. Apa tipe nilai yang diberikan ke `doInBackground()` di `AsyncTask`?
2. Apa tipe nilai yang diberikan ke callback yang melaporkan kemajuan tugas?
3. Apa tipe nilai yang diberikan ke callback yang dijalankan ketika tugas selesai?

Pertanyaan 3

Untuk melaporkan kemajuan pekerjaan yang dieksekusi oleh AsyncTask, apa metode callback yang Anda *implementasikan*, dan metode yang Anda *panggil* di subkelas AsyncTask?

- Mengimplementasikan `publishProgress()`. Memanggil `publishProgress()`.
- Mengimplementasikan `publishProgress()`. Memanggil `onProgressUpdate()`.
- Mengimplementasikan `onProgressUpdate()`. Memanggil `publishProgress()`.
- Mengimplementasikan `onProgressUpdate()`. Memanggil `onProgressUpdate()`.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Layout mencakup ProgressBar yang menyetel atribut tepat untuk menentukan rentang nilai.
 - AsyncTask memecah waktu tidur total menjadi beberapa bagian dan memperbarui bilah kemajuan pada setiap bagian tersebut.
 - AsyncTask memanggil metode tepat dan mengimplementasikan callback tepat untuk memperbarui bilah kemajuan.
 - AsyncTask harus mengetahui tampilan yang perlu diperbarui. Tergantung pada apakah AsyncTask diimplementasikan sebagai kelas inti atau tidak, tampilan bisa diteruskan baik ke konstruktur AsyncTask atau didefinisikan sebagai variabel anggota pada Aktivitas.
-

7.2: Terhubung ke Internet

Membangun dan menjalankan aplikasi

Membuat aplikasi yang mengambil dan menampilkan materi laman web di URL. Aplikasi menampilkan:

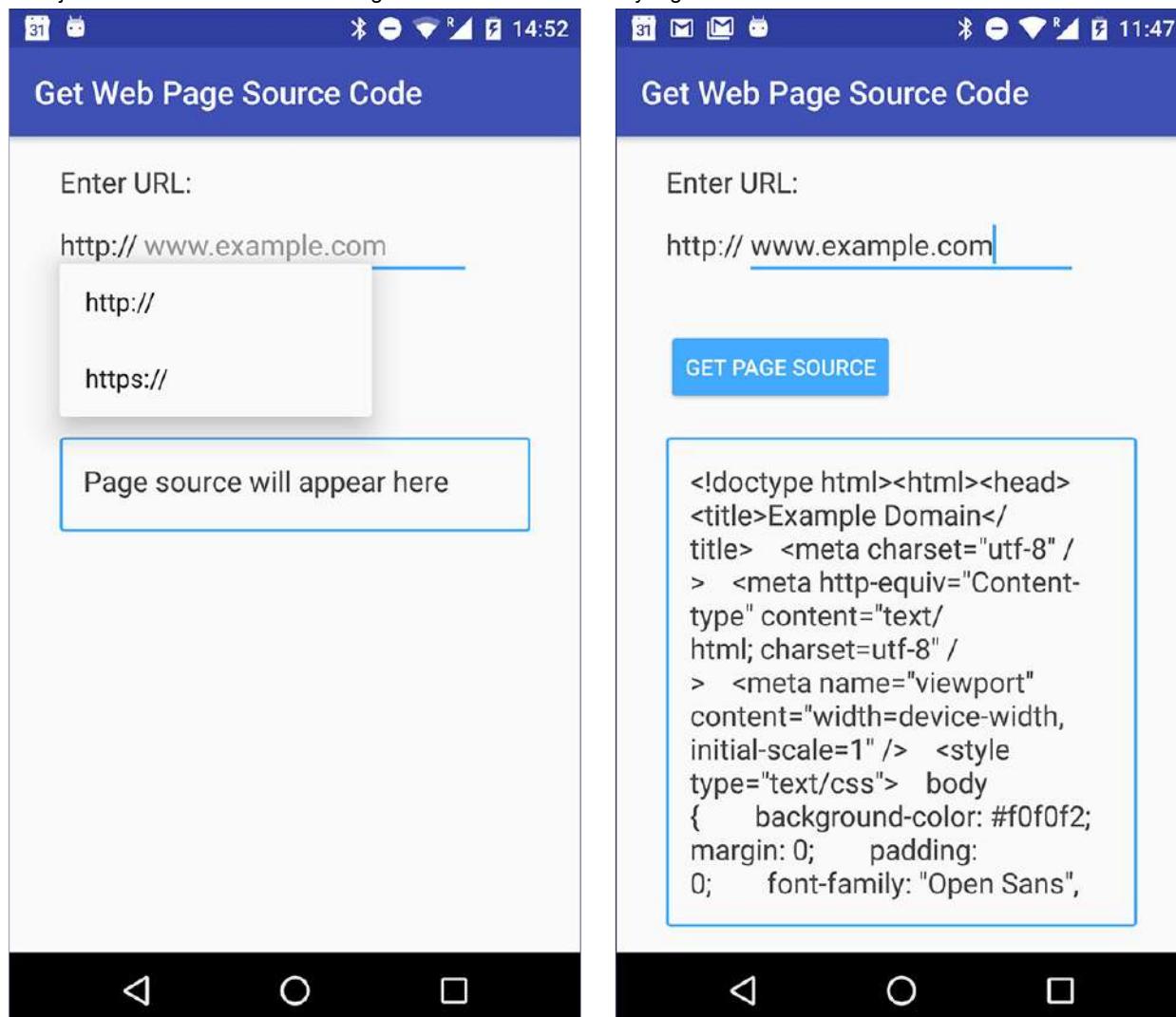
- Bidang tempat pengguna memasukkan URL
- Bidang seperti menu atau spinner yang memungkinkan pengguna memilih protokol (HTTP atau HTTPS)
- Tombol yang mengeksekusi tugas saat diklik
- Tampilan bergulir dari kode sumber laman web di URL

Gunakan AsyncTaskLoader untuk mengambil kode sumber laman web di URL. Anda harus mengimplementasikan subkelas AsyncTaskLoader.

Jika koneksi ke Internet tidak tersedia ketika pengguna mengeklik tombol, aplikasi harus memberikan respons yang tepat kepada pengguna. Misalnya, aplikasi bisa menampilkan pesan seperti "Check your Internet connection and try again."

Tampilan harus berisi TextView dalam ScrollView yang menampilkan kode sumber, namun tampilan sesungguhnya dari antarmuka sepenuhnya terserah Anda. Layar Anda bisa saja terlihat berbeda daripada tangkapan layar di bawah. Anda bisa menggunakan menu munculan, spinner, atau kotak centang untuk mengizinkan pengguna memilih HTTP atau HTTPS.

Gambar di sebelah kiri menunjukkan layar awal, dengan menu munculan untuk protokol. Gambar di sebelah kanan menunjukkan contoh hasil setelah mengambil sumber laman URL yang diberikan.



Jawablah pertanyaan berikut

Pertanyaan 1

Izin apa yang dibutuhkan aplikasi Anda untuk terhubung ke Internet?

- android.permission.CONNECTIVITY
- android.permission.INTERNET
- Aplikasi tidak memerlukan izin khusus; semua aplikasi diizinkan untuk terhubung ke Internet.

Pertanyaan 2

Bagaimana aplikasi Anda memeriksa bahwa tersedia konektivitas Internet?

Dalam manifes:

- permintaan izin ACCESS_NETWORK_STATE
- permintaan izin ALL_NETWORK_STATE
- permintaan izin NETWORK_CONNECT

Dalam kode:

- Membungkus kode untuk terhubung ke Internet dalam blok coba/tangkap, dan menangkap kesalahan NO_NETWORK .

- Menggunakan ConnectivityManager untuk memeriksa jaringan aktif sebelum terhubung ke jaringan.
- Menampilkan dialog ke pengguna untuk mengingatkan mereka serta memastikan bahwa koneksi Internet sudah tersedia sebelum mencoba untuk terhubung ke Internet.

Pertanyaan 3

Di mana Anda mengimplementasikan metode callback loader yang terpilih ketika loader selesai melaksanakan tugasnya?

- Pada subkelas AsyncTaskLoader. AsyncTaskLoader harus mengimplementasikan LoaderManager.LoaderCallbacks.
- Pada Aktivitas yang menampilkan hasil tugas. Aktivitas harus mengimplementasikan LoaderManager.LoaderCallbacks.
- Pada kelas Utilitas yang memperluas Obyek dan mengimplementasikan LoaderManager.LoaderCallbacks.

Pertanyaan 4

Ketika pengguna memutar perangkat, bagaimana AsyncTask dan AsyncTaskLoader berperilaku berbeda apabila mereka sedang dalam proses menjalankan tugas di latar belakang?

- Pilihan 1
 - AsyncTask yang berjalan menjadi terputus dari Aktivitas meskipun terus berjalan.
 - AsyncTaskLoader yang berjalan menjadi terputus dari Aktivitas namun berhenti berjalan, menjaga sumber daya sistem.
- Pilihan 2
 - AsyncTask yang berjalan menjadi terputus dari Aktivitas namun berhenti berjalan, menjaga sumber daya sistem.
 - AsyncTaskLoader yang berjalan memulai ulang secara otomatis eksekusi tugasnya dari awal. Aktivitas menampilkan hasil.
- Pilihan 3
 - AsyncTask yang berjalan menjadi terputus dari Aktivitas meskipun terus berjalan.
 - AsyncTaskLoader yang berjalan terhubung kembali secara otomatis ke Aktivitas setelah rotasi perangkat. Aktivitas menampilkan hasil.

Pertanyaan 5

Bagaimana Anda melakukan inisialisasi AsyncTaskLoader untuk mengerjakan langkah seperti melakukan inisialisasi variabel, yang harus dilakukan sebelum loader mulai menjalankan tugas latar belakang?

- Pada `onCreateLoader()` dalam Aktivitas, membuat instance dari subkelas AsyncTaskLoader. Dalam konstruktor loader melakukan tugas inisialisasi.
- Pada `onCreateLoader()` dalam Aktivitas, membuat sebuah instance dari subkelas AsyncTaskLoader. Dalam metode `init()` loader, melakukan tugas-tugas inisialisasi.
- Pada Aktivitas, mengimplementasikan `initLoader()` untuk melakukan inisialisasi loader.
- Melakukan tugas inisialisasi loader pada awal `loadInBackground()` dalam Loader.

Pertanyaan 6

Apa metode yang harus diimplementasikan AsyncTaskLoader?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Manifes mencakup permintaan untuk izin yang sesuai.
- Menggunakan subkelas AsyncTaskLoader.
- Merespons dengan tepat jika perangkat tidak bisa terhubung ke Internet.

- Menggabungkan protokol dan laman web untuk membuat URL valid yang digunakan aplikasi untuk terhubung ke Internet.
 - Mengimplementasikan metode callback Loader yang dibutuhkan.
 - Menampilkan hasil dari pengambilan sumber laman web dalam TextView di ScrollView. (Tidak masalah melakukannya dalam Aktivitas yang sama, atau dengan memulai Aktivitas baru.)
-

7.3: Penerima Siaran

Membangun dan menjalankan aplikasi

1. Membuat sebuah aplikasi yang disebut BroadcastCounter menggunakan template Empty Activity.
2. Menggunakan BroadcastReceiver untuk menghitung berapa kali siaran `ACTION_POWER_CONNECTED` diterima. Petunjuk: Menetapkan BroadcastReceiver Anda sebagai kelas inti dan mendaftarkannya secara dinamis.
3. Menampilkan hitungan dalam tampilan TextView.



Jawablah pertanyaan berikut

Pertanyaan 1

Apa perbedaan antara mendaftar penerima siaran secara statis atau dinamis?

- Mendaftar penerima siaran secara dinamis menghubungkan operasi ke daur hidup aktivitas Anda.
- Jika mendaftarkan penerima untuk menerima siaran lokal saja, Anda harus mendaftarkannya secara dinamis; pendaftaran statis bukanlah opsi.
- Mendaftarkan penerima siaran secara statis menciptakan sebuah proses baru dalam menjalankan penerima siaran jika tidak ada proses yang berjalan terkait dengan aplikasi Anda.
- Semua hal di atas.

Pertanyaan 2

Benar atau salah? Bila penerima siaran terdaftar secara statis, penerima akan merespons kejadian siaran bahkan jika aplikasi Anda tidak berjalan.

Pertanyaan 3

Kelas apa yang digunakan untuk mengurangi risiko keamanan BroadcastReceivers ketika siaran tidak lintas-aplikasi (yaitu, siaran dikirim dan diterima oleh aplikasi yang sama)?

- SecureBroadcast
- LocalBroadcastManager
- OrderedBroadcast
- SecureBroadcastManager

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

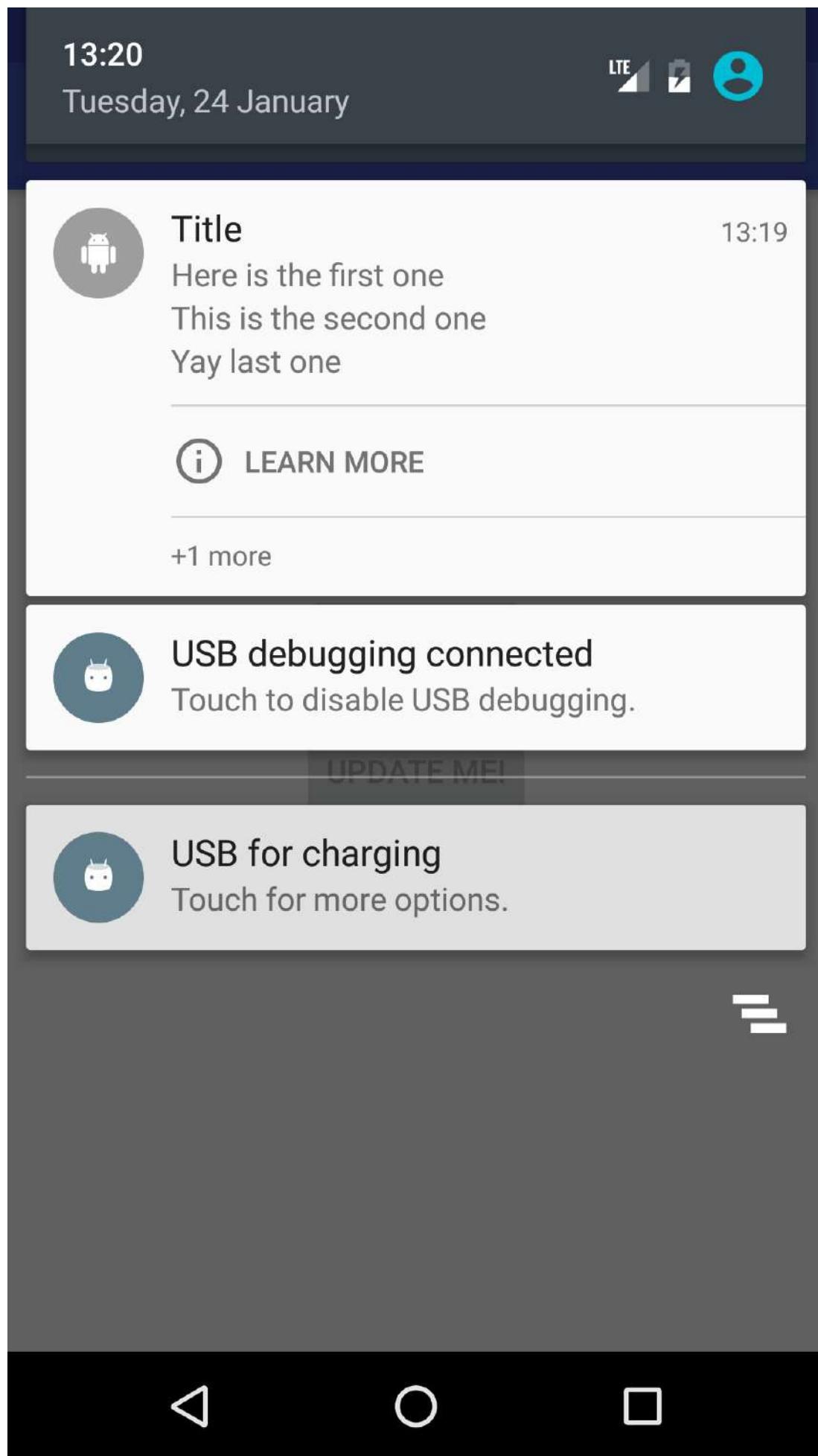
Pastikan aplikasi memiliki beberapa fitur berikut:

- Penerima siaran mendaftar dan membatalkan pendaftaran secara dinamis dalam salah satu pasangan metode daur hidup berikut: OnResume/OnPause, OnCreate/OnDestroy, atau OnStart/OnStop.
 - Penghitung akan ditampilkan dan bertambah saat ponsel dicolokkan.
-

8.1: Notifikasi

Membangun dan menjalankan aplikasi

Buka aplikasi [NotifyMe](#) yang Anda buat dalam pelajaran [Notifikasi](#). Ubah notifikasi yang telah diperbarui dalam aplikasi agar menggunakan layout [InboxStyle](#) yang diperluas sebagai ganti BigPictureStyle. Gunakan data string palsu untuk setiap baris dan teks ringkasan.



Catatan: Notifikasi mungkin terlihat sedikit berbeda, tergantung pada API level perangkat.

Jawablah pertanyaan berikut

Pertanyaan 1

Misalkan Anda membuat sebuah aplikasi yang mengunduh sebuah karya seni setiap harinya. Setelah karya seni tersedia, aplikasi menunjukkan notifikasi kepada pengguna, dan pengguna bisa mengunduh atau melewati karya seni sehari. Apa metode PendingIntent yang Anda gunakan untuk memulai layanan mengunduh gambar?

- `Activity.startService()`
- `PendingIntent.getBroadcast()`
- `PendingIntent.getActivity()`
- `PendingIntent.getService()`

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

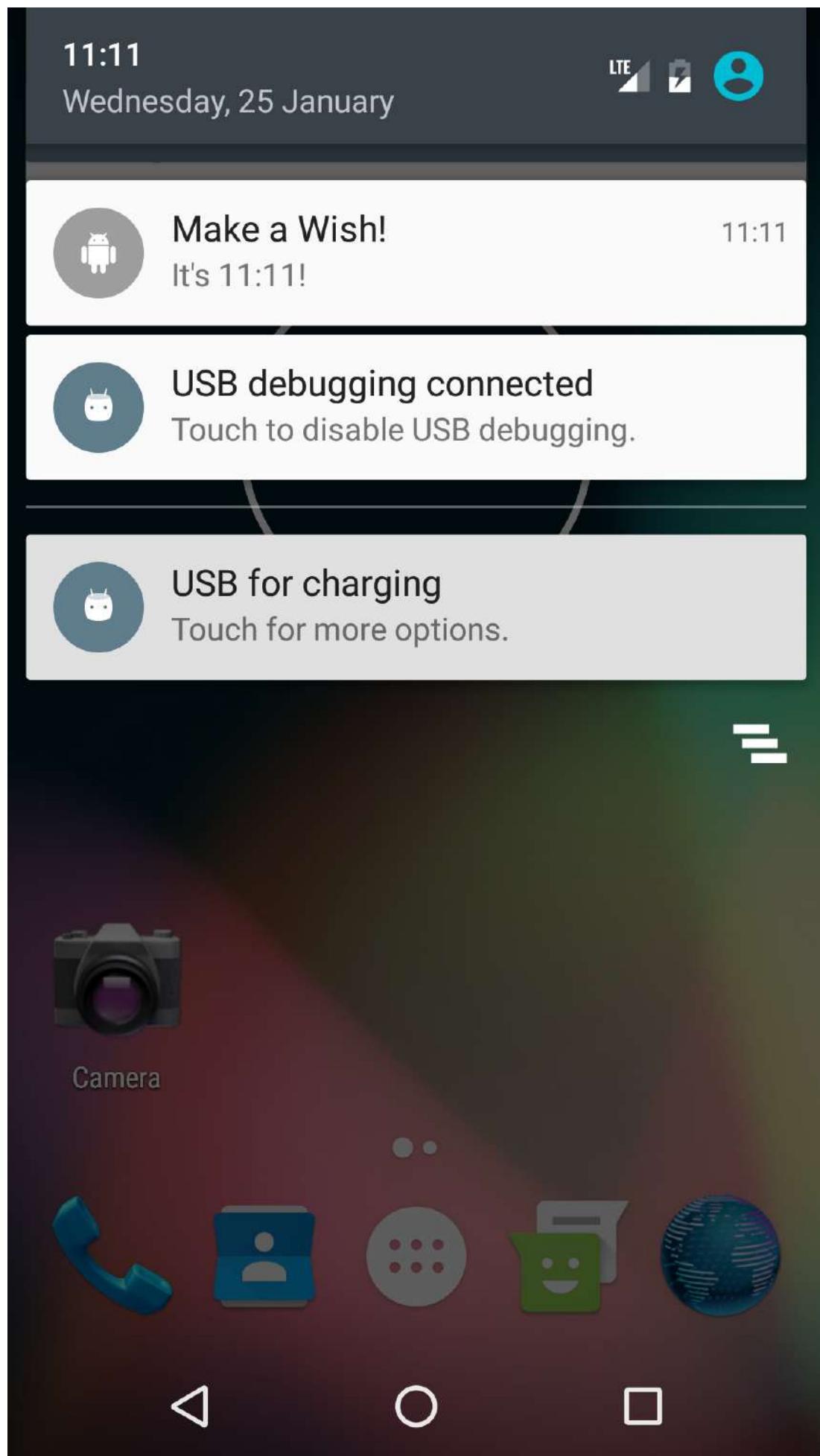
Pastikan aplikasi memiliki beberapa fitur berikut:

- Ketika pengguna mengetuk tombol update, notifikasi akan menjadi notifikasi InboxStyle dengan beberapa baris teks yang merepresentasikan item baris.
- Layar memiliki baris teks judul dan ringkasan, yang berubah posisinya tergantung pada API level. (Lihat [Notifikasi dalam panduan desain material](#).)
- Gunakan kelas `NotificationCompat.InboxStyle` sehingga kompatibel ke belakang.

8.2: Alarm Manager

Membangun dan menjalankan aplikasi

Membuat sebuah aplikasi yang mengirimkan notifikasi saat pukul 11:11 (AM). Layar menampilkan tombol toggle untuk menghidupkan dan mematikan alarm.



Catatan: Notifikasi mungkin terlihat sedikit berbeda, tergantung pada API level perangkat.

Jawablah pertanyaan berikut

Pertanyaan 1

Pengaturan waktu tidak pasti menjadi setelan default AlarmManager pada API level apa? (Semua metode `set()` menggunakan pengaturan waktu tidak pasti, kecuali secara eksplisit dinyatakan lain.)

- API level 16
- API level 18
- API level 19
- API level 17

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

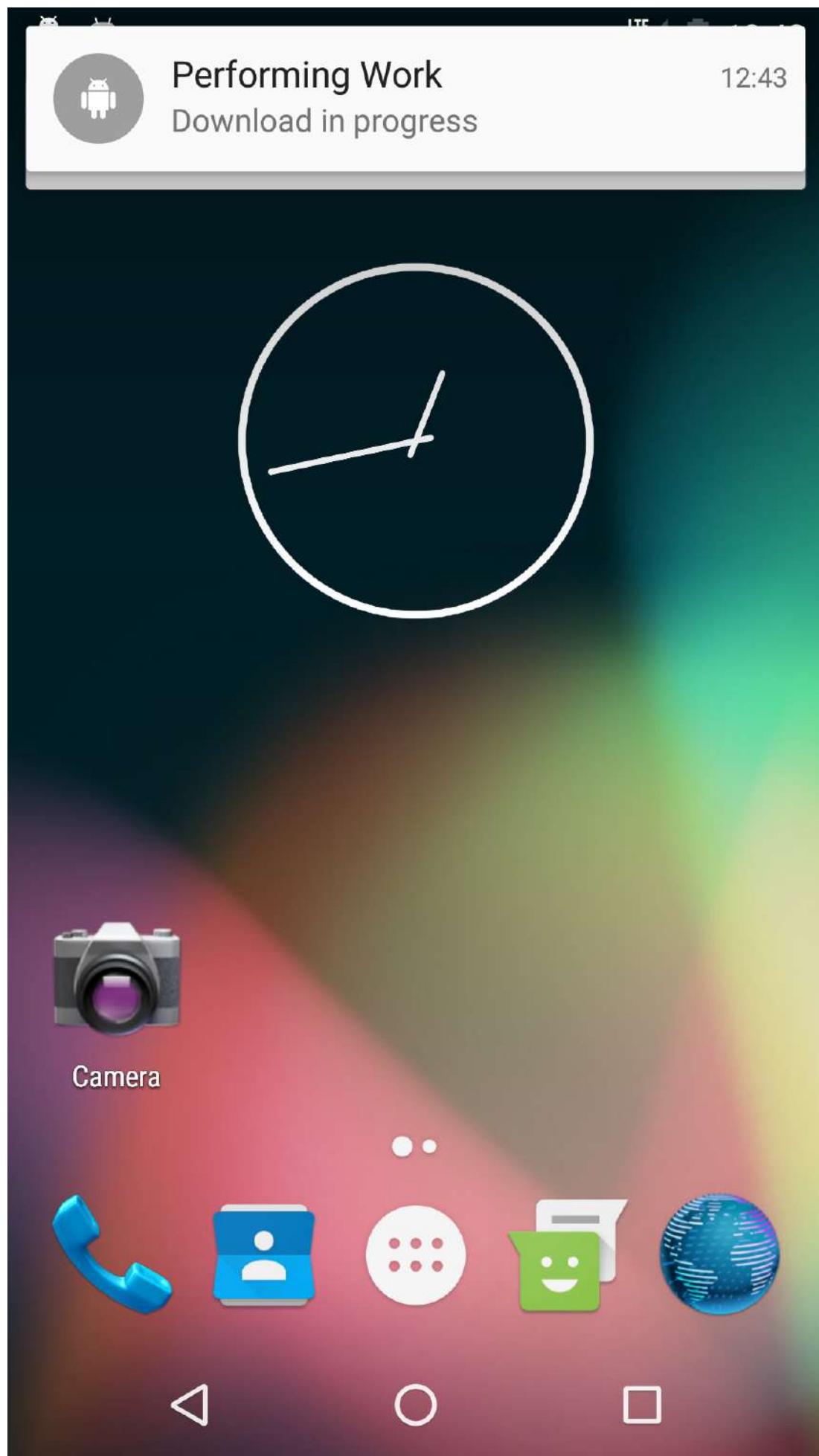
- Alarm menggunakan waktu pasti. Ini berarti bahwa kode memuat pernyataan yang memeriksa bahwa API level perangkat > 19, dan menggunakan metode `setExact()` jika memang begitu.
 - Aplikasi menampilkan notifikasi saat pukul 11:11 AM.
-

8.3: JobScheduler

Membangun dan menjalankan aplikasi

Membuat aplikasi yang menyimulasikan unduhan berukuran besar secara terjadwal dengan baterai dan pemakaian data sebagai pertimbangan. Aplikasi berisi tombol bertuliskan "Download Now" dan memiliki beberapa fitur berikut:

- Aplikasi memberikan notifikasi sebagai pengganti dari melakukan unduhan yang sebenarnya.
- "Unduhan" dilakukan sekali sehari, saat ponsel dalam keadaan diam namun terhubung ke listrik dan WiFi, atau ketika tombol ditekan.
- Ketika pengguna mengetuk tombol **Download Now**, notifikasi "pengunduhan" akan terpicu.
Petunjuk :Menetapkan kelas JobService sebagai kelas inti. Dengan demikian, tombol **Download Now** dan JobService bisa memanggil metode yang sama untuk memberikan notifikasi.



Catatan: Notifikasi mungkin terlihat sedikit berbeda, tergantung pada API level perangkat.

Jawablah pertanyaan berikut

Pertanyaan 1

Apa kelas yang Anda gunakan jika menginginkan fitur seperti yang disediakan oleh JobScheduler, namun menginginkan fitur bekerja untuk perangkat yang menjalankan API level 20 dan di bawahnya?

- JobSchedulerCompat
- FirebaseJobDispatcher
- AlarmManager

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

- Objek JobInfo memiliki 4 set kriteria: `setRequiresCharging()` , `setPeriodic()` , `setRequiresDeviceIdle()` ,
`setRequiredNetworkType()`
- Aplikasi akan mogok bila kelas JobService tidak memiliki konstruktur kosong.

Tugas Pekerjaan Rumah: Pelajaran 9 & 10 & 11

Daftar Isi:

- 9.1: Preferensi Bersama
- 9.2: Setelan Aplikasi
- 10.1: Database SQLite
- 11.1: Penyedia Materi
- 11.2: Loader

9.1: Preferensi Bersama

Membangun dan menjalankan aplikasi

Buka aplikasi ScoreKeeper yang Anda buat dalam pelajaran Sumber Daya Dapat Digambar, Gaya, dan Tema.

1. Ganti status instance yang tersimpan dengan preferensi bersama untuk setiap nilainya.
2. Uji aplikasi:
 - Putar perangkat untuk memastikan bahwa perubahan konfigurasi membaca preferensi yang disimpan dan memperbarui antarmuka pengguna.
 - Hentikan aplikasi dan mulai ulang untuk memastikan bahwa preferensi sudah tersimpan.
3. Tambahkan tombol **Reset** yang menyetel ulang nilai skor ke 0 dan mengosongkan preferensi bersama.

Jawablah pertanyaan berikut

Pertanyaan 1

Anda menyimpan status aplikasi ke preferensi bersama dalam metode daur hidup apa?

Pertanyaan 2

Anda memulihkan status aplikasi dalam metode daur hidup apa?

Pertanyaan 3

Bisakah Anda memikirkan sebuah kejadian saat memerlukan preferensi bersama dan status instance secara bersamaan?

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

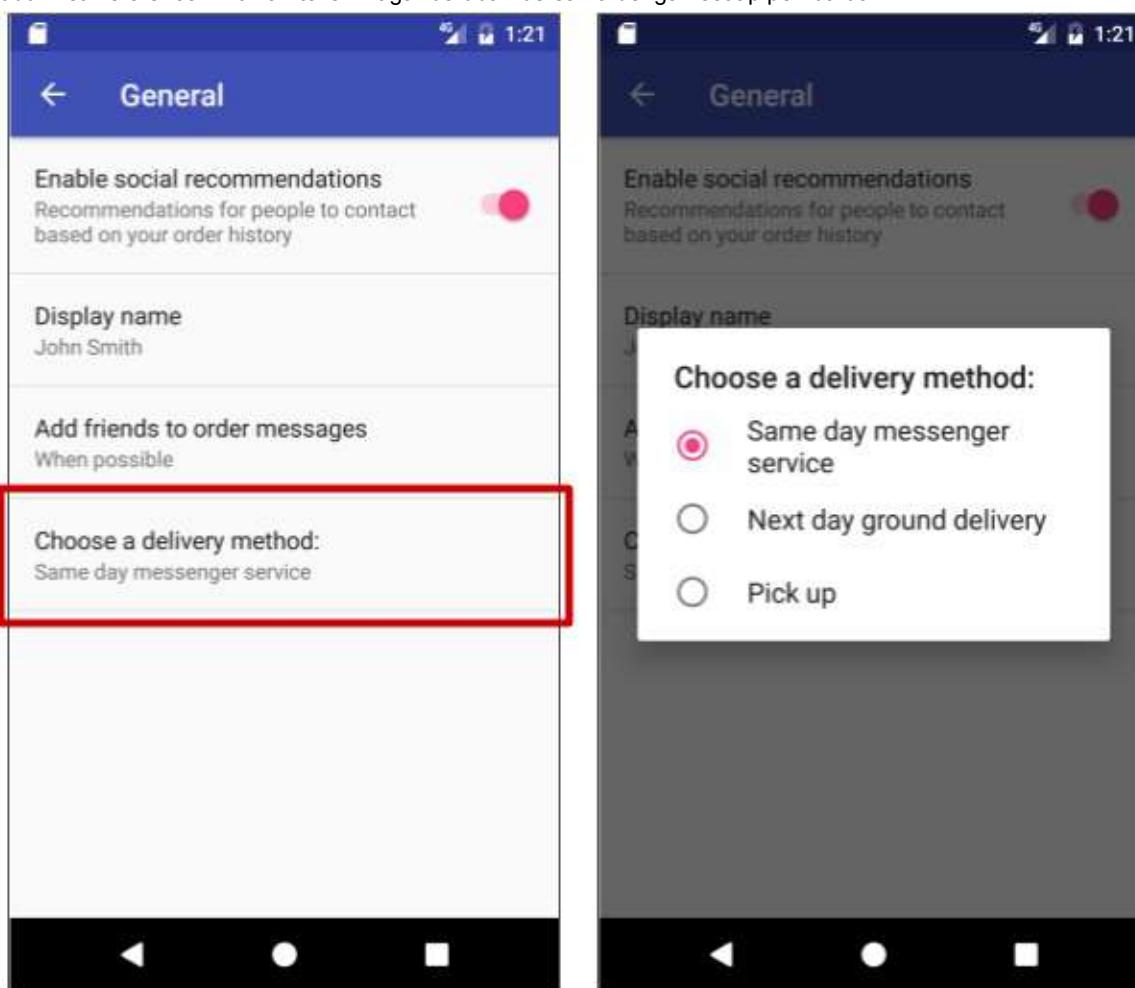
- Aplikasi mempertahankan skor saat rotasi.
- Aplikasi mempertahankan skor setelah dihentikan dan dimulai ulang.
- Aplikasi menyimpan nilai saat ini ke preferensi bersama dalam metode `onPause()`.
- Aplikasi mengembalikan preferensi bersama dalam metode `onCreate()`.
- Aplikasi menampilkan tombol **Reset** yang menyetel ulang skor ke 0.
- Implementasi metode penanganan click untuk tombol reset:
 - Menyetel ulang kedua variabel skor ke 0.
 - Memperbarui kedua tampilan teks
 - Mengosongkan preferensi bersama.

9.2: Setelan Aplikasi

Membangun dan menjalankan aplikasi

Buka aplikasi DroidCafeWithSettings yang Anda buat dalam pelajaran [Menambahkan Setelan ke Aplikasi](#).

1. Tambahkan ListPreference (dialog dengan tombol radio) ke grup setelan "General". Masukkan ke dalam layout layar "General settings", di bawah ListPreference "Add friends to order messages".
2. Edit larik string yang digunakan untuk ListPreference agar memasukkan judul ListPreference "Choose a delivery method." Gunakan pilihan pengiriman yang sama seperti yang digunakan dalam tombol radio di OrderActivity.
3. Pastikan setelan Pengiriman yang dipilih pengguna ditampilkan dalam pesan toast yang sama dengan setelan Pasar dan Rekomendasi yang dipilih.
4. Kredit tambahan: Menampilkan metode pengiriman yang dipilih sebagai setelan teks ringkasan yang muncul di bawah judul ListPreference. Aktifkan teks ini agar berubah bersama dengan setiap pembaruan.



Jawablah pertanyaan berikut

Pertanyaan 1

Anda menetapkan larik masukan dan larik nilai untuk ListPreference pada file apa? Pilih salah satu:

- pref_general.xml
- strings.xml
- menu_main.xml
- content_main.xml

Pertanyaan 2

Anda menggunakan larak masukan dan larak nilai dalam menyiapkan ListPreference, dan menyetel kunci ListPreference serta nilai default pada file apa? Pilih salah satu:

- pref_general.xml
- strings.xml
- menu_main.xml
- SettingsActivity.java

Pertanyaan 3

Bagaimana Anda menyetel nilai default setelan saat aktivitas berjalan untuk pertama kalinya?

- Menetapkan nilai default menggunakan atribut `android:defaultValue` untuk setiap preferensi setelan dalam file XML preferensi.
- Menyetel nilai default dalam metode `onCreate()` untuk aktivitas menggunakan `PreferenceManager.setDefaultValues()`.
- Kedua hal di atas.

Pertanyaan 4

Untuk aplikasi yang mendukung Android 3.0 dan versi yang lebih baru, praktik terbaik setelan adalah menggunakan Aktivitas Setelan yang memperluas Aktivitas, dan fragmen untuk setiap file XML preferensi yang memperluas PreferenceFragment. Namun bagaimana cara agar aplikasi tetap kompatibel dengan [pustaka dukungan v7 appcompat](#) ketika memperluas sebuah Aktivitas dengan [AppCompatActivity](#)?

Pertanyaan 5

Ketika menggunakan antarmuka [SharedPreferences](#) untuk mengakses dan memodifikasi data preferensi seperti setelan, pernyataan berikut membaca preferensi setelan yang ditetapkan oleh kunci `delivery`:

```
String deliveryPref = sharedPref.getString("delivery", "1");
```

- Benar atau salah? Argumen string "1" adalah nilai yang dikembalikan jika tidak ada preferensi setelan. Ini biasanya adalah string untuk nilai default setelan, yang dalam contoh ini bernilai "1".

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Pastikan aplikasi memiliki beberapa fitur berikut:

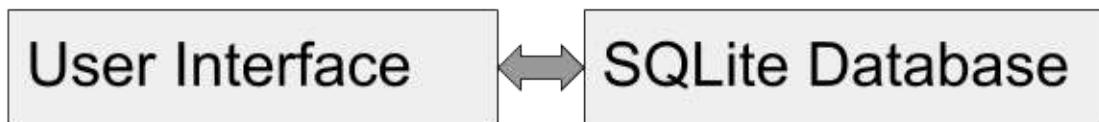
- Metode `onCreate()` membaca setelan `deliveryPref` menggunakan `sharedPref.getString()`.
- File `pref_general.xml` menyertakan ListPreference yang menggunakan entri larak dari pilihan pengiriman.
- Kredit tambahan: Pernyataan `bindPreferenceSummaryToValue(findPreference("delivery"))` telah ditambahkan ke metode `onCreate()` dari kelas GeneralPreferenceFragment di `SettingsActivity` untuk menampilkan pilihan pengiriman dalam ringkasan preferensi.

10.1: Database SQLite

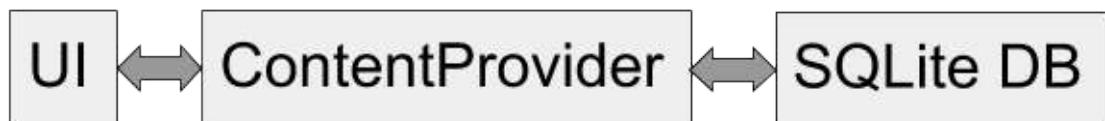
README: Pada grup tugas pekerjaan rumah berikutnya (10.1, 11.1, 11.2), Anda membuat 2 aplikasi. Mereka saling berhubungan satu sama lain dengan cara yang sama seperti aplikasi yang dibuat dalam praktik yang sama, seperti berikut:

1. Aplikasi pertama, pada 10.1, adalah daftar TODO yang menggunakan database SQLite untuk menyimpan item.

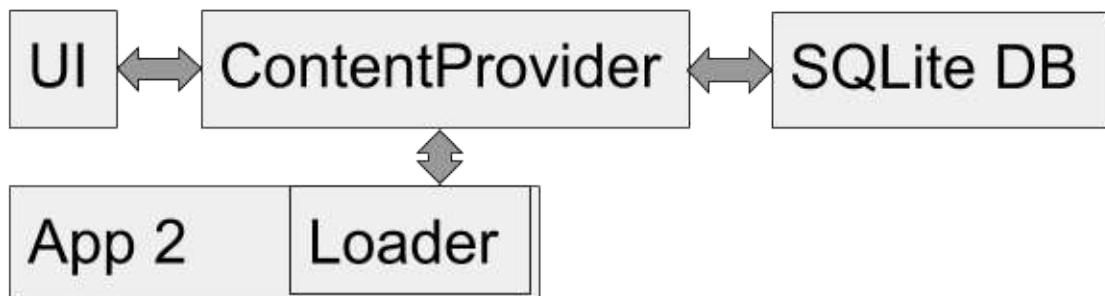
Aplikasi ini juga menyertakan cara untuk menambahkan, menampilkan, dan mengedit item.



2. Pada 11.1, Anda memperluas aplikasi daftar TODO agar menggunakan penyedia materi untuk menyajikan data dari database SQLite ke antarmuka pengguna.



3. Pada 11.2, Anda membuat aplikasi bernama ShowToDoItems yang mengakses daftar TODO penyedia materi dan memuat item to-do menggunakan loader.



Membangun dan menjalankan aplikasi yang menggunakan database SQLite

Membuat aplikasi bernama TODO dengan database SQLite, tempat pengguna bisa membuat dan mengedit item daftar to-do yang disimpan dalam database.

1. Memperluas kelas SQLiteOpenHelper dengan mengimplementasikan metode `query()`, `insert()`, dan `update()`.
2. Mencakup fitur aplikasi yang dijelaskan dibawah ini.

Fitur:

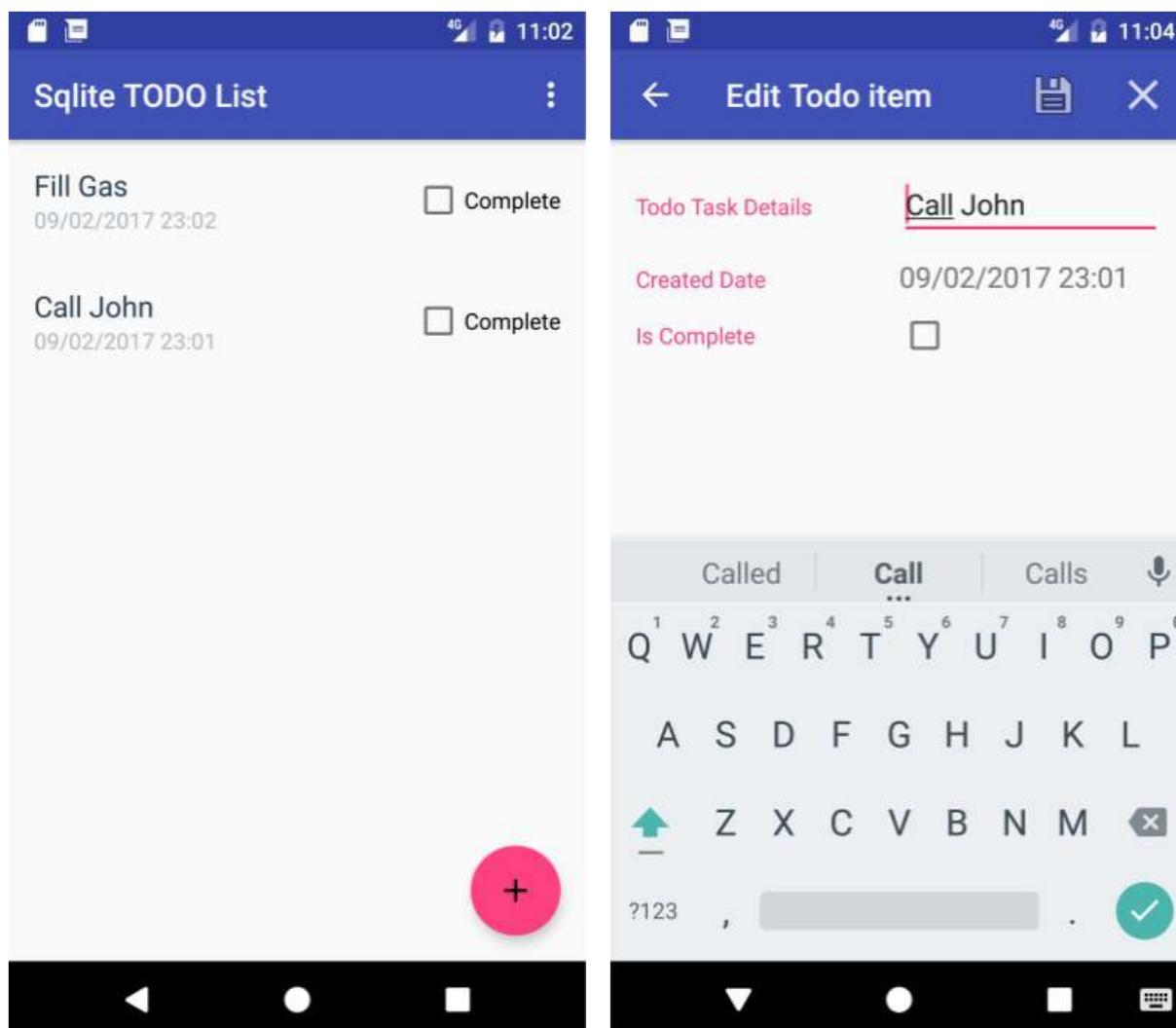
- Pengguna bisa menambahkan item baru ke dalam daftar.
- Setiap item dalam database berisi tugas yang harus dilakukan, tanggal pembuatan dan tanggal selesai, dan apakah tugas telah selesai atau belum.
- Ketika aplikasi dimulai, layar menampilkan daftar item to-do belum selesai yang diurutkan berdasarkan tanggal pembuatan.
- UI berisi item menu Options untuk memulai aktivitas yang menunjukkan tugas yang selesai.
- Pengguna bisa mengubah status selesai sebuah item. Ketika pengguna menandai tugas telah selesai, tugas diberi tanda selesai dan tanggal pembuatan diganti dengan tanggal selesai.
- Ketika pengguna mengetuk item belum selesai, mode edit akan terpicu, dan pengguna bisa mengedit item tersebut.
- Ketika aplikasi dimulai ulang, status terbaru item akan terlihat, yang menunjukkan bahwa data tersebut disimpan dan dimuat ulang.

Tip:

- Salah satu cara untuk mengimplementasikan UI adalah dengan menggunakan RecyclerView yang memulai aktivitas edit ketika item diklik.
- Anda tidak akan dinilai tentang cara Anda mengimplementasikan UI, atau bagaimana UI terlihat, selama UI memperlihatkan fungsionalitas tersebut.
- Aplikasi yang menggunakan RecyclerView dan database SQLite mengikuti sebuah pola. Periksa dan gunakan kembali

kode yang Anda tulis untuk pelajaran [Database SQLite](#) dan pelajaran [Menelusuri Database SQLite](#).

- Pastikan Anda dengan rapi memisahkan data dari antarmuka pengguna. Anda akan memperluas aplikasi to-do ini untuk menyelesaikan tugas pekerjaan rumah penyedia materi dan loader di masa mendatang.



Jawablah pertanyaan berikut

Pertanyaan 1

Berapa banyak kode yang bisa Anda gunakan kembali dari aplikasi lain? Berapa banyak waktu yang bisa Anda hemat? Seberapa besar bantuan yang Anda dapat dengan menggunakan aplikasi lain sebagai contoh dalam menyusun aplikasi? Tidak ada jawaban benar atau salah.

Pertanyaan 2

Apa beberapa keuntungan menggunakan kelas SQLiteOpenHelper? Cek semua yang berlaku.

- Menyediakan utilitas untuk menyederhanakan tugas-tugas pembuatan dan inisialisasi database.
- Menyediakan metode `onUpgrade()`. Yang paling penting, jika peningkatan versi gagal dilakukan, metode ini melakukan rollback bagi Anda.
- Menggunakan pola yang direkomendasikan akan mempermudah dalam memahami, memelihara, dan memperluas aplikasi.

Pertanyaan 3

Manakah dari pernyataan berikut yang menunjukkan keuntungan menggunakan database SQLite untuk menyimpan data Anda? Cek semua yang berlaku.

- Menggunakan kueri SQL untuk mengambil data, memungkinkan Anda untuk mencocokkan pembatas dan ketentuan yang diberikan.
- Data disimpan dengan persisten dan aman, dan bisa diambil dengan efisien.
- Aplikasi lain bisa menggunakan data Anda.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

Tidak ada jawaban benar atau salah untuk Pertanyaan 1. Sangatlah penting agar siswa memikirkan dan menyadari manfaat dari membangun karya mereka sendiri dan karya orang lain.

UI yang dipilih siswa bukanlah faktor untuk penilaian, selama aplikasi menunjukkan fungsionalitas database. Siswa memiliki banyak kebebasan dalam bagaimana mengimplementasikan fungsionalitas ini.

Pastikan aplikasi memiliki beberapa fitur berikut:

- Menyertakan kelas SQLiteOpenHelper dengan mengimplementasikan metode `query()`, `insert()`, dan `update()` untuk mendukung fungsionalitas yang diperlukan.
- Ketika aplikasi dimulai, layar menampilkan daftar item to-do yang diurutkan berdasarkan tanggal pembuatan.
- Ketika pengguna berinteraksi dengan item, mode edit akan terpilih, dan pengguna bisa mengedit item tersebut.
- Pengguna bisa mengubah status selesai sebuah item.
- Menyertakan item menu Opsi yang memungkinkan pengguna untuk melihat item yang sudah selesai.
- Ketika item berubah, UI menunjukkan perubahan.
- Ketika aplikasi dimulai ulang, status terbaru item akan terlihat, yang menunjukkan bahwa data tersebut disimpan dan dimuat ulang.

11.1: Penyedia Materi

Memperluas aplikasi daftar TODO dari pekerjaan rumah 10.1 agar menggunakan penyedia materi.

1. Menambahkan kelas Contract untuk konstanta biasa dan publik, URI, dan skema database.
2. Menambahkan kelas ContentProvider yang menangani URI serta mengimplementasikan metode `query()`, `insert()`, dan `update()`.
3. Menambahkan penyedia materi ke file AndroidManifest.xml.

Fitur:

- Dari perspektif pengguna, aplikasi harus memiliki fungsionalitas yang sama persis dengan aplikasi TODO yang dibuat untuk 10.1 tugas pekerjaan rumah.
- Semua kueri masuk melalui penyedia materi, kemudian ke database SQLite.

Tip:

- Jika Anda membutuhkan bantuan, lihat [aplikasi word_list_sql_with_content_provider](#) dari pelajaran [Berbagi Materi dengan Aplikasi Lain](#).

Jawablah pertanyaan berikut

Pertanyaan 1

Apa tujuan utama penyedia materi?

- Memisahkan data dari antarmuka pengguna.
- Menyediakan data untuk aplikasi lain.
- Memisahkan back end dari antarmuka pengguna.

Pertanyaan 2

Apa beberapa keuntungan menggunakan kelas Contract?

- Contract bisa bersifat publik sehingga aplikasi lain dapat mengetahui cara mengakses penyedia materi.
- Anda hanya perlu mendefinisikan konstanta umum sekali.
- Kelas ini mendefinisikan bagian dari aplikasi yang tidak bisa diubah.
- Untuk aplikasi yang lebih besar dan kompleks, kumpulkan konstanta ke dalam sebuah tempat untuk mempermudah pemeliharaan.

Pertanyaan 3

Mengapa penyedia materi perlu dideklarasikan dalam Manifes Android?

- Untuk memberi tahu Android Framework ID unik dari penyedia materi.
- Untuk memastikan bahwa hanya ada satu penyedia materi untuk setiap aplikasi.
- Untuk memberi tahu Kerangka Kerja Android propertinya, seperti izin bagi penyedia materi ini.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

UI yang dipilih siswa bukanlah faktor untuk penilaian, selama itu menunjukkan fungsionalitas aplikasi.

Pastikan bahwa aplikasi mengimplementasikan hal berikut:

- Menggunakan kelas Contract untuk URI dan konstanta lainnya.
- Menyertakan kelas ContentProvider yang menangani URI dan mengimplementasikan metode `query()`, `insert()`, dan `update()` yang berinteraksi dengan database.
- Aplikasi telah didesain sedemikian rupa sehingga aktivitas yang menggunakan metode `query()`, `insert()`, dan `update()` penyedia materi akan berinteraksi dengan database.
- Ketika aplikasi dimulai ulang, status terbaru item akan terlihat, yang menunjukkan bahwa data tersebut disimpan dan dimuat ulang.

11.2: Loader

Membangun dan menjalankan aplikasi bernama ShowToDoItems yang menggunakan loader untuk mengambil dan menampilkan data dari penyedia materi yang Anda gunakan dalam aplikasi TODO dari pekerjaan rumah 11.1.

1. Buat UI dasar untuk menampilkan item daftar TODO yang belum selesai.
2. Implementasikan kelas Contract bagi konstanta untuk penyedia materi aplikasi TODO.
3. Implementasikan `LoaderManager.LoaderCallbacks<>` untuk memuat data dari penyedia materi.
4. Tambahkan pengelola loader untuk mengelola loader Anda.

Fitur:

- Ketika aplikasi dimulai, layar menampilkan daftar item to-do yang diambil dari penyedia materi aplikasi TODO yang sedang berjalan.
- Ketika data dalam database aplikasi TODO berubah, pembaruan juga dilakukan di aplikasi ShowToDoItems.

Tip:

- Gunakan kembali elemen UI dari aplikasi sebelumnya.
- Jangan lupa untuk menambahkan izin ke file manifest.
- Jika Anda membutuhkan bantuan, lihat aplikasi [WordListClient](#) dan [WordListLoader](#) pada GitHub.

Jawablah pertanyaan berikut

Pertanyaan 1

Manakah dari pernyataan berikut yang menunjukkan keuntungan menggunakan loader?

- Loader cepat.
- Loader berjalan di thread terpisah untuk mencegah UI tidak responsif atau tersendat.
- Loader menyederhanakan pengelolaan thread dengan menyediakan metode callback bila terjadi peristiwa.
- Loader mempertahankan dan meng-cache hasil ketika konfigurasi berubah untuk mencegah duplikasi pertanyaan.
- Loader bisa mengimplementasikan pengamat untuk memantau perubahan dalam sumber data dasarnya.

Pertanyaan 2

Pada saat membuat aplikasi ShowToDoItems, mengapa Anda tidak harus mengimplementasikan pengamat materi?

- Pengelola loader menangani pengamatan data bagi Anda.
- CursorLoader secara otomatis mendaftar ContentObserver untuk memicu muat ulang ketika data berubah.
- Aplikasi daftar TODO mengirimkan data baru ke aplikasi ShowToDoItems bila item dalam database berubah.

Mengirimkan aplikasi Anda untuk dinilai

Panduan untuk penilai

UI yang dipilih siswa bukanlah faktor untuk penilaian, selama itu menunjukkan fungsionalitas aplikasi.

Pastikan aplikasi memiliki beberapa fitur berikut:

- Menggunakan kelas Contract.
- Mengimplementasikan `LoaderManager.LoaderCallbacks<>` dan menggunakan pengelola loader.
- Ketika aplikasi dimulai, layar menampilkan daftar item to-do yang diambil dari penyedia materi aplikasi TODO yang sedang berjalan.
- Ketika data dalam database aplikasi TODO berubah, pembaruan juga dilakukan di aplikasi ShowToDoItems.

Apendiks: Utilitas

Daftar Isi:

- [Menyalin dan mengubah nama proyek](#)
- [Menghapus proyek](#)
- [Mengekstrak sumber daya](#)
- [Menambahkan pustaka dukungan](#)
- [Membuat gambar di Asset Studio](#)
- [Membandingkan objek khusus](#)
- [Menyimpan keadaan objek khusus](#)

Apendiks ini adalah kumpulan tugas yang mungkin perlu Anda lakukan selama proses development aplikasi di praktik ini. Tugas-tugas ini tidak khusus untuk satu praktik.

Menyalin dan mengubah nama proyek

Untuk beberapa pelajaran, Anda perlu membuat salinan proyek sebelum membuat perubahan baru. Anda juga disarankan menyalin proyek untuk menggunakan sebagian kodennya di proyek baru. Apa pun itu, Anda bisa menyalin proyek yang sudah ada (**ExistingProject**), lalu mengubah name dan mengoptimalkan komponen proyek baru (**NewProject**) untuk menggunakan nama proyek yang baru. (Di petunjuk di bawah ini, ganti nama proyek aktual Anda untuk **ExistingProject** dan **NewProject**.

1. Salin proyek

1. Di sistem file komputer Anda (bukan di Android Studio), buat salinan direktori **ExistingProject**.
2. Ubah nama direktori yang disalin menjadi **NewProject**.

2. Ubah nama dan optimalkan komponen proyek

Nama proyek yang lama, **ExistingProject**, masih ada di paket dan file di salinan proyek baru Anda. Ubah referensi file dan paket di aplikasi Anda ke nama baru, sebagai berikut:

1. Mulai Android Studio
2. Klik **Open an existing Android Studio project**.
3. Masuk ke direktori **NewProject**, pilih, dan klik **OK**.
4. Pilih **Build > Clean Project** untuk membuang file yang otomatis dibuat.
5. Klik tab samping **1:Project** dan pilih **Android** dari menu tarik-turun untuk melihat file Anda di tampilan Project.
6. Perluas **app > java**.
7. Klik kanan **com.example.android.existingproject** dan pilih **Refactor > Rename**. Ini akan membuka dialog Rename.
8. Ubah **existingproject** menjadi **newproject**.
9. Centang **Search in comments and strings** dan **Search for text occurrences** dan klik **Refactor**.
10. Panel **Find Refactoring Preview** akan muncul yang menampilkan kode yang akan dioptimalkan.
11. Klik **Do Refactor**.
12. Perluas **res > values** dan klik dua kali file **strings.xml**.
13. Ubah string **name="app_name"** menjadi **New Project**.

3. Perbarui file build.gradle dan AndroidManifest.xml

Setiap aplikasi yang Anda buat harus memiliki ID aplikasi yang unik, seperti yang didefinisikan di file build.gradle aplikasi. Meskipun langkah-langkah di atas harus diubah di file build.gradle, Anda harus memeriksanya untuk memastikan dan juga menyinkronkan proyek dengan file gradle:

1. Perluas **Gradle Scripts** dan klik dua kali **build.gradle (Module: app)**.
2. Di defaultConfig, periksa untuk memastikan nilai kunci applicationID telah diubah menjadi **"com.example.android.newproject"**. Jika belum berubah, ubah secara manual sekarang.
3. Klik **Sync Now** di sudut kanan atas jendela Android Studio.

Tip: Anda juga memilih **Tools > Android > Sync Project** dengan File Gradle untuk menyinkronkan file gradle Anda.

Sebagai tambahan, beberapa aplikasi menyertakan nama aplikasi dalam format yang bisa dibaca (misalnya, "New Project" bukan newproject), sebagai label di file AndroidManifest.xml.

1. Perluas **app > manifests** and double-click **AndroidManifest.xml**.
2. Temukan pernyataan di bawah ini, dan jika perlu, ubah label if menjadi sumber daya string untuk nama aplikasi baru:

```
android:label="@string/app_name"
```

Menghapus proyek

Semua file untuk proyek Android berada dalam folder proyek di sistem file komputer. Untuk menghapus proyek, hapus foldernya.

Android Studio juga menyimpan daftar proyek terbaru yang telah Anda buka. Anda bisa menghapus proyek dari daftar proyek terbaru di Android Studio. (Menghapus proyek dari daftar proyek terbaru tidak memengaruhi file proyek sebenarnya.)

Untuk membuang proyek dari daftar proyek terbaru, lakukan salah satu dari yang berikut:

- Di layar mulai Android Studio, klik nama proyek dan tekan tombol Delete.
- Pilih File > Open Recent > Manage Projects, klik nama proyek dan tekan tombol Delete.

Mengekstrak String dan Dimensi

Mengekstrak String

Agar aplikasi Anda bisa diterjemahkan ke berbagai bahasa, Anda harus mempertahankan semua sumber daya string Anda di file res/values/strings.xml.

Membuat sumber daya string

Ada beberapa cara untuk membuat sumber daya string:

- Tambahkan secara manual di file strings.xml menggunakan sintaks berikut:

```
<string name="string_name">String Value</string>
```

- Kapan pun string akan digunakan, seperti atribut teks sebuah TextView:

1. Ketikkan nama yang diinginkan untuk sumber daya string dalam format berikut: `@string/string_name`. Ini akan disorot dengan warna merah karena sumber daya belum ada.
2. Pastikan kursor Anda berada di teks yang disorot.
3. Tekan **Alt + Enter** dan pilih **Create string value resource**.
4. Masukkan string yang Anda inginkan dan tekan **OK**, dan string akan ditambahkan ke file strings.xml Anda.

- Anda bisa memilih string yang di-harcode yang sudah ada dalam XML atau Java, tekan **Alt + Enter** dan pilih **Extract string resource**.

Mengakses sumber daya string:

- Di **XML**, buat referensi sumber daya string menggunakan sintaks berikut: `@string/string_name`
- Di **Java**, buat referensi sumber daya string menggunakan sintaks berikut: `getString(R.string.string_name)`

2. Ekstrak Dimensi

Dimensi secara umum harus tidak di-hardcode tetapi disimpan di file dimens.xml. Ini memungkinkan Anda menetapkan dimensi yang berbeda menggunakan [qualifier sumber daya](#)

Ekstrak dimensi dengan cara yang sama sebagai string (Alt-Enter) dan dimensi akan disimpan di dimens.xml.

3. Ekstrak Gaya

Jika Anda memiliki beberapa elemen yang beratribut sama, Anda bisa membuat gaya di file style.xml. Untuk mengetahui selengkapnya tentang gaya, lihat pelajaran [Gaya dan Tema](#).

Untuk mengekstrak atribut yang sudah ke dalam gaya, lakukan yang berikut:

1. Letakkan kursor Anda di tampilan yang atributnya ingin Anda ubah menjadi gaya.
2. Klik kanan dan pilih **Refactor > Extract > Style**.
3. Beri nama gaya dan pilih atribut. Jika **Launch 'Use Style Where Possible' refactoring after the style is extracted** dicentang, Android Studio akan menelusuri semua file untuk atribut yang dipilih dan menerapkan gaya ke tampilan yang cocok dengan atribut.
4. Klik **OK**.

Tambahkan pustaka dukungan Android ke file build.

[Pustaka Dukungan Android](#) menyediakan versi Android Framework API yang kompatibel dengan versi lama, komponen UI tambahan, dan serangkaian utilitas yang berguna.

Misalnya, untuk menggunakan kelas RecyclerView, yang berlokasi di paket Dukungan Android, Anda harus menyertakan dua dependensi di file build.gradle proyek Anda. Prosesnya sama dengan komponen pustaka dukungan lainnya.

Ikuti langkah-langkah berikut dan lihat tangkapan layar di bawah ini:

1. Di Android Studio, di proyek Anda, pastikan Anda berada dalam panel **Project** (1) dan di tampilan **Android** (2).
2. Di hierarki file, temukan folder **Gradle Scripts** (3).
3. Perluas **Gradle Scripts**, jika perlu, dan buka file **build.gradle (Module: app)** (4). 
4. Di akhir file **build.gradle (Module: app)**, temukan bagian dependensi.
5. Tambahkan kedua dependensi pustaka ini sebagai dua baris (di dalam kurung kurawal):

```
compile 'com.android.support:recyclerview-v7:23.1.1'
compile 'com.android.support:design:23.1.1'
```

- Mungkin terdapat baris yang sudah ada yang sama dengan ini:

```
compile 'com.android.support:appcompat-v7:23.1.1'
```

- Tambahkan baris Anda bawah baris tersebut.

- **Pastikan nomor versi baris Anda sesuai dengan nomor versi baris yang sudah ada tersebut.**

- Pastikan nomor versi semua pustaka sama dan cocok dengan `compiledSdkVersion` di bagian atas file. (Jika tidak cocok, Anda akan mendapatkan kesalahan build time.)

6. Jika diminta, sinkronkan aplikasi Anda sekarang.
7. Bangun dan jalankan aplikasi Anda.

Berikut ini adalah contoh bagian dependensi file build.gradle dengan pustaka dukungan yang ditambahkan.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:recyclerview-v7:23.1.1'
    compile 'com.android.support:design:23.1.1'
}
```

Membuat gambar di Asset Studio

Gunakan [Image Asset Studio](#) untuk membuat dan menambahkan ikon peluncur.

1. Buka aplikasi Anda di Android Studio.
2. Klik kanan folder **res** di proyek Anda dan pilih **New > Image Asset** dari menu.

Ini akan membuka jendela Image Asset Studio, tempat Anda bisa membuat ikon teks, memilih dari clipart yang tersedia, atau menambahkan ikon khusus Anda sendiri.

Perhatikan bahwa panel di kiri atas bisa digulir; gulir untuk melihat penyesuaian tambahan.

Untuk menambahkan ikon teks khusus:

1. Ubah **Nama** ikon menjadi `ic_launcher_text`, jika Anda tidak ingin mengganti ikon `ic_launcher` Android default yang ada di proyek Anda.
2. Di baris **Asset Type**, pilih Text.
3. Ketikkan "Hello World!" di kotak teks.
4. Coba bereksperimen dengan menyesuaikan font.
5. Gulir ke bawah dan ubah warna font dan latar belakang.
6. Klik **Next**.
7. Jendela **Confirm Icon Path** menampilkan bagaimana sebuah ikon dengan teks yang Anda tetapkan akan dibuat untuk setiap resolusi, serta lokasi storage default dan path di aplikasi Anda.
8. Klik **Finish**.
9. Masuk ke folder **res/mipmap**. Jika sekarang terdapat ikon baru, dengan versi default di tingkat atas, dan versi yang ukurannya disesuaikan untuk berbagai resolusi.
10. Untuk menggunakan ikon baru, buka manifes Android. Ubah baris ikon Android dari `ic_launcher` ke `ic_launcher_text`.

```
    android:icon="@mipmap/ic_launcher_text"
```

11. Jalankan aplikasi Anda.
12. Setelah aplikasi diluncurkan, masuk ke layar beranda, dan buka daftar aplikasi.
13. Gulir dan Anda akan melihat ikon Anda dicantumkan bersama dengan aplikasi lain yang dipasang.

Untuk menambahkan ikon clipart:

Ikuti langkah-langkah sebelumnya kecuali:

1. Ubah **Nama** menjadi `ic_launcher_clipart`.
2. Pilih **Clip Art** sebagai **Asset Type**.
3. Di baris Clip Art, klik tombol yang menampilkan ikon saat ini, Android default.
4. Pilih ikon dari aplikasi dari jendela munculan clip art.

Untuk menambahkan ikon khusus:

Ikuti langkah-langkah sebelumnya kecuali:

1. Ubah **Nama** menjadi `ic_launcher_image`.
2. Pilih **Image** sebagai **Asset Type**.
3. Di baris **Path**, pilih gambar. Ini bisa berupa gambar yang telah Anda tambahkan ke proyek Anda atau gambar di komputer Anda.

Membandingkan objek khusus

Kapan pun model data Anda memanggil objek untuk diurutkan, Anda perlu mendefinisikan bagaimana objek ini bisa dibandingkan satu sama lain.

Antarmuka [Comparable](#) memungkinkan Anda untuk menetapkan cara membandingkan dua objek dan menentukan apakah salah satunya lebih besar, lebih kecil, atau sama dengan yang lain.

Antarmuka Comparable mengharuskan Anda mengimplementasikan satu metode: `compareTo(<T> another)` adalah tipe yang berparameter yang Anda implementasikan bisa dibandingkan dan tipe objek yang Anda bandingkan (misalnya, Anda ingin membandingkan instance Foobar Anda dengan instance Foobar lainnya, Ada akan mengimplementasikan `Comparable<Foobar>` dan metode `compareTo` Anda akan mengambil Foobar sebagai parameter).

Metode Compare akan melakukan yang berikut:

- Mengembalikan integer negatif jika objek kurang dari parameter.
- Mengembalikan integer positif jika objek lebih besar dari parameter.
- Mengembalikan nol jika objek sama.

Misalnya, untuk membandingkan daftar buku menurut tanggal publikasi:

```
@Override
public int compareTo(Book book) {
    if (this.publication == book.publication) { return 0; }
    else { return this.publication > book.publication ? 1 : -1; }
}
```

Menyimpan keadaan objek khusus

Di Android, Anda akan sering membuat objek khusus untuk merepresentasikan model data khusus Anda. Untuk mempertahankan keadaan objek ini, Anda harus bisa meneruskannya ke dalam bundel `savedInstanceState`. Untuk melakukannya, kelas khusus Anda harus mengimplementasikan antarmuka [Parcelable](#). Ini memungkinkan tipe primitif (int, string, byte, dsb.) untuk disimpan di callback `savedInstanceState`.

Lakukan yang berikut:

1. Setelah menyiapkan data di kelas khusus (hanya tipe data primitif akan disimpan), tambahkan implementasi `Parcelable` ke deklarasi kelas Anda.
2. Deklarasi akan digarisbawahi dengan warna merah, karena Anda harus mengimplementasikan metode antarmuka. Arahkan cursor ke teks yang digarisbawahi, tekan **Alt + Enter** dan pilih **Implement methods**.
3. Pilih `describeContents()` dan `writeToParcel(Parcel dest, int flags)`. Klik **OK**.
4. Nama kelas akan tetap digarisbawahi, yang menandakan bahwa antarmuka belum sepenuhnya diimplementasikan. Pilih nama kelas dan tekan lagi **Alt + Enter** dan pilih **Add Parcelable implementation**. Android Studio akan otomatis menambahkan kode yang diperlukan. Perhatikan variabel yang keadaannya (tipe primitif) yang ingin Anda pertahankan sudah dituliskan ke `Parcel` di metode `writeToParcel`.
5. Anda sekarang bisa menyimpan keadaan objek ini menggunakan metode bundle `savedInstanceState`: `putParcelable`, `putParcelableArray`, dan `putParcelableArrayList` dan getter yang terkait.

