

MODUL

MOBILE PROGRAMMING



PERTEMUAN 1

Dasar Pengembangan Aplikasi Mobile/Bergerak

Tujuan

- Mahasiswa mengetahui dasar dari pengembangan aplikasi *mobile*.
- Mahasiswa memahami berbagai platform dari sistem operasi *mobile*.
- Mahasiswa memahami Platform Android.
- Mahasiswa memahami Arsitektur dari Sistem operasi Android.
- Mahasiswa memahami Building Block dari aplikasi Android.

Perkembangan Mobile

Perkembangan teknologi yang pesat membuat masyarakat haus akan informasi secara cepat, karena itulah perangkat mobile/bergerak mengalami perkembangan yang sangat pesat tidak hanya dari segi perangkat keras, tetapi juga dari segi perangkat lunak. Perangkat mobile sudah menjadi bagian dari gaya hidup masyarakat saat ini. Terlebih lagi banyak aplikasi yang dapat membantu dalam kehidupan sehari-hari.

Aplikasi mobile merupakan sebuah perangkat lunak yang berjalan pada perangkat mobile. Aplikasi mobile dapat berupa hiburan, pembelajaran, maupun membantu pengguna dalam kehidupan sehari-hari. Aplikasi mobile menyediakan berbagai layanan seperti komunikasi dan pesan, audio/video, permainan, manajemen kontak dan koneksi.

Aplikasi mobile dapat dibagi menjadi beberapa kategori tergantung dari penggunaannya. Beberapa pengkategorian dari aplikasi mobile antara lain.

- Aplikasi Multimedia, seperti pemutar video, pemutar musik, penampil gambar.
- Aplikasi utilitas, seperti pengelola berkas, pengelola kontak dan pengelola task.
- Aplikasi komunikasi, seperti aplikasi chat, aplikasi e-mail dan aplikasi video/voice call.
- Aplikasi media sosial, seperti aplikasi facebook, twitter dan instagram.
- Aplikasi Produktifitas, seperti aplikasi untuk mengelola tugas, mengelola keuangan dan kalkulator.
- Aplikasi Enterprise, aplikasi yang digunakan untuk bisnis.
- Aplikasi berbasis lokasi, aplikasi yang memanfaatkan aplikasi peta seperti google map dan waze.
- Aplikasi perjalanan, aplikasi yang digunakan dalam perjalanan seperti aplikasi penerjemah dan aplikasi pemesanan angkutan.
- Aplikasi Permainan/games.

Platform dari Sistem Operasi Mobile

Terdapat bermacam-macam jenis dari platform sistem operasi mobile. Sistem operasi mobile yang populer antara lain.

- Android (open source, Apache)

Sistem operasi yang berbasis linux dan banyak digunakan oleh para pembuat/pabrikan telepon genggam sekarang ini (Samsung, HTC, Motorola, Sony Ericsson, LG, Huawei, dll). Android OS adalah sebuah sistem operasi mobile yang diperuntukan bagi smartphone dan komputer tablet. Android OS yang berbasis linux kernel ini dikembangkan oleh Open Handset Alliance dibawah bendera Google. Pengembangan aplikasi-aplikasi Android dan menguploadnya di Android Market, dengan tujuan aplikasi-aplikasi tersebut dapat didownload oleh para pengguna smartphone android. Dalam empat tahun terakhir (rilis tahun 2008) Android berhasil merilis versi terbaru mulai dari : Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb , Ice Cream Sandwich, jelly bean, Kitkat, Lolipop, Marshmallow, Nougat, dan oreo.

- BlackBerry OS (closed source, proprietary)

BlackBerry menggunakan BlackBerry OS sebagai sistem operasinya. BlackBerry OS, dikembangkan oleh Research in Motion (RIM), untuk BlackBerry, perusahaan telekomunikasi asal Kanada ini. Diawali kemunculannya BlackBerry booming dengan layanan push-email dan sebagai smartphone yang sukses dipasaran. Versi terakhir dari BlackBerry OS adalah BlackBerry OS 10.

- iOS (closed source, proprietary)

iOS adalah sistem operasi mobile milik Apple yang dikembangkan dan hanya diaplikasikan untuk perangkat Apple Inc seperti iPhone, iPod Touch, iPad dan Apple TV. iOS memiliki lebih dari 500.000 aplikasi di App Store dengan angka penjualan perangkat smartphone terbaik di dunia pada tahun 2011 disusul Google Android dan Nokia Symbian. iOS menjadi “leading” bagi sistem operasi mobile yang lain dengan mengusung interface multitouch yang bisa digeser (slider), ditekan (switch), dan tombol. Perangkat Apple masih menjadi nomor satu dalam hal teknologi dan angka penjualan di dunia. Produk smartphone Apple yang terbaru adalah iPhone x dan iOS11 adalah versi terbaru.

- Windows Phone (closed source, proprietary)

Windows Mobile merupakan sistem operasi mobile yang dikembangkan oleh Microsoft, produk Mobile Windows seperti Sagem myS-7, O2 Xphone dll. Sistem operasi

yang berhasil dikembangkan dan diaplikasikan mulai dari Pocket PC 2000 hingga Windows Mobile versi 6.5.5. Meskipun ada beberapa kelebihan seperti fitur GPS, dapat menonton televisi, mobile blog, mobile database namun kelemahan dari Windows Mobile adalah minim developer independen. Karena pihak Microsoft yang menerapkan peraturan bahwa Windows Mobile bersifat tertutup. Hal ini tidak menguntungkan bagi pasar Windows Mobile karena produk mereka yang minim developer berakibat pada minimnya aplikasi-aplikasi yang dapat dikembangkan. Sehingga Windows Mobile kalah bersaing dengan smartphone lain, terutama bagi vendor yang menerapkan sistem terbuka bagi developer manapun yang bersedia mengembangkan sebuah sistem operasi, seperti Andorid.

Menjelajah Platform Sistem Operasi Android

Android merupakan platform mobile yang paling populer di dunia. Android menjadi sistem operasi dari jutaan perangkat mobile di lebih dari 190 negara. Perkembangan Android dimulai dengan berdirinya Android, Inc. Pada Oktober 2003 dengan tujuan membuat perangkat mobile untuk menyaingi sistem operasi mobile yang populer saat itu yaitu symbian dan windows mobile. Tahun 2005 Google mengakusisi dari Android inc. Versi beta Android diluncurkan pada tanggal 5 Nopember 2007 dan pada tanggal 12 nopolember 2007 Android Software Development Kit (SDK) diluncurkan sehingga membuat pengguna dapat mengembangkan aplikasi berbasis Android mereka sendiri.

Kelebihan dari sistem operasi Android adalah banyaknya dukungan dari penyedia perangkat mobile. Android banyak digunakan sebagai sistem operasi untuk berbagai perangkat mobile. Kemudahan dalam distribusi melalui google playstore menjadi keunggulan lain dari Android. Kekurangan dari Android ada pada Google tidak menyaring aplikasi yang masuk pada google playstore sehingga masih bisa menemukan aplikasi yang kualitasnya dibawah standar.

Pada perkembangannya, Android telah merilis bermacam versi dengan penambahan fitur disetiap rilisnya. Berikut adalah versi Android yang telah dirilis.

Android 1.1

Pertama kali platform Android diluncurkan pada tahun 2009, setahun sebelumnya Google telah merilis versi beta yang diperkenalkan kepada khalayak ramai. Android versi pertama dikenal dengan Android 1.1.



Android 1.1

Berikut adalah fitur-fitur android 1.1,Mampu menyimpan attachment dalam pesan dan Waktu default screen in-call lebih lama bila menggunakan speakerphone plus kemampuan untuk menampilkan atau menyembunyikan dialpad.

Android1.5 (Cupkace)

Pada tanggal 27 April 2009 Google juga merilis Android versi Cupcake. Cupcake adalah versi Android yang memulai tradisi penamaan kue untuk rilis Android. Ini dapat menambahkan beberapa fitur dan peningkatan baru dibanding versi sebelumnya.

**Android 1.5 Cupcake**

Berikut adalah fitur yang dimilikinya: dukungan untuk tampilan aplikasi widget-miniatur yang bisa disematkan di aplikasi lain (seperti home screen) dan menerima update berkala dan rekaman video ditambahkan ke kamera bersamaan dengan kemampuan untuk langsung mengupload video ke YouTube.

Android 1.6 (Donut)

Google merilis Android 1.6 Donut pada bulan September tahun 2009. Penambahan fitur terbesar adalah dukungan untuk CDMA yang memperluas pasar pengguna baru ke Android. CDMA adalah teknologi yang digunakan oleh jaringan mobile Amerika pada saat itu.

**Android 1.6 Donut**

Berikut adalah fitur yang dimilikinya:

- Navigasi Google Maps ditambahkan bersamaan dengan dukungan navigasi satelit.
- Donut menyertakan fitur gallery untuk memperlancara penangkapan media
- Pengenalan Search Box
- Toggling cepat antara kamera

Android 2.0-2.1 (Eclair)

Pada bulan Oktober 2009, sekitar setahun setelah peluncuran Android 1.0, Google merilis versi 2.0 dari OS, dengan nama Android Eclair. Versi ini adalah pertama yang menambahkan dukungan Text-to-Speech.



Android 2.0 Eclair

Berikut adalah fitur yang menyertainya.

- Dukungan multi-touch ditambahkan ke Android.
- Dukungan untuk mencari di dalam pesan teks.

Eclair membawa kotak masuk terpadu ke Android. Dukungan untuk beberapa akun Google telah ditambahkan.

Android 2.2.3 (Froyo)

Android versi terbaru kembali dirilis pada 20 Mei 2010. Google menamainya dengan Froyo. Nama Froyo ini diambil dari singkatan frozen yogurt. Smartphone pertama yang membawa merek Google Nexus, Nexus One, dirilis dengan Android 2.1 dari kotak pada awal tahun 2010, namun dengan cepat menerima update over-the-air ke Froyo akhir tahun . Pada

android versi ini mulai dilengkapi dengan fitur friendly user seperti opsi untuk mematikan akses data pada jaringan seluler.



Android 2.2 Froyo

Berikut adalah fitur yang menyertainya:

- Hotspot WiFi portabel untuk berbagi koneksi 3G perangkat dengan gadget lainnya.
- Setelan bergabung dengan kontak dan email untuk memback up ke server Google yang memungkinkan pemnbaca mengembalikan segalanya secara otomatis ke perangkat baru.
- flash telepon juga bisa digunakan dalam video.

Android 2.3 - 2.3.7 (Gingerbread)

Gingerbread dirilis pada tahun 2010 . Pada 13 September 2017, Google menunjukkan bahwa hanya 0,6 persen dari semua perangkat Android yang saat ini menjalankan beberapa versi Gingerbread. Fitur utama termasuk dukungan NFC, SIP untuk panggilan Internet.



Android 2.3 Gingerbread

Berikut adalah fitur yang menyertainya.

- UI overhaul untuk menghindari screen burn-in dan meningkatkan daya tahan baterai.
- Dukungan kamera menghadap depan untuk panggilan video.
- Download manager untuk mengawasi download Pembaca.
- Peningkatan pada keyboard layar dengan cara pintas dan kursor untuk membantu copy paste.

Android 3.0 - 3.2.6 (Honeycomb)

Honeycomb diluncurkan pada bulan Mei 2011 pada dasarnya untuk memperluas Android untuk mendukung layar tablet.



Android 3.0 Honeycomb

Versi Android ini paling diabaikan dari semua. Karena dirilis khusus untuk tablet dan tidak pernah sampai ke ponsel. Berikut adalah fitur yang menyertainya:

- Beberapa perbaikan UI memanfaatkan layar besar.
- Tombol perangkat keras dijatuhkan untuk mendukung tombol di layar.
- Browser web mengenalkan tabbed browsing.
- Widget yang lebih besar.
- Aplikasi seperti Gmail dan YouTube dirancang ulang untuk menggunakan layar besar.

Android 4.0 - 4.0.4 (Ice Cream Sandwich)

Android Ice Cream Sandwich dirilis pada bulan Oktober 2011, versi Android Ice Cream Sandwich menghadirkan sejumlah fitur baru bagi pengguna. Ini menggabungkan banyak fitur versi Honeycomb tablet saja dengan smartphone yang berorientasi pada Gingerbread.



Android 4.0 Ice Cream Sandwich

Berikut adalah fitur yang menyertainya.

- Desain ulang terbesar ke Android dengan tema Holo.
- Pengalaman browsing lebih cepat.
- Ruang penyimpanan multi untuk aplikasi.
- Pengenalan wajah untuk membuka kunci telepon

Android 4.1 - 4.3.1 (Jelly Bean)

Jellybean dimulai pada bulan Juni 2012 dengan merilis Android 4.1. Google dengan cepat merilis versi 4.2 dan 4.3, keduanya berada di bawah label Jelly Bean, masing-masing pada bulan Oktober 2012 dan Juli 2013.



Android 4.1 Jelly Bean

Berikut adalah fitur yang menyertainya:

- Google Now, tool bantu yang menampilkan informasi yang relevan berdasarkan riwayat pencarian.
- Project Butter untuk mendukung frame rate yang lebih tinggi saat menggesek menu dan layar rumah.
- Mampu melihat foto dengan menggesek dari kamera untuk menuju ke filmstrip.
- Widget menyetel ulang diri mereka untuk menambahkan yang baru.
- Pemberitahuan fitur yang lebih banyak
- Fitur gerak dan aksesibilitas baru.

Android 4.4 (KitKat)

Android KitKat dirilis pada November 2013 ,Android 4.4 adalah satu-satunya versi OS yang benar-benar menggunakan nama sepotong permen.dan menjadi salah satu versi Android yang paling disukai oleh pengguna Smartphone di dunia.



Android 4.4 KitKat

KitKat memiliki fitur yang istimewa dari OS Android sebelumnya. Berikut adalah fitur yang menyertainya:

- Immersive mode untuk konsumsi konten yang lebih baik.
- Bar navigasi yang lebih baik untuk masuk dan keluar dari mode Immersive.
- Dukungan widget layar kunci.
- Dialer baru dengan fitur Caller ID.
- Wallpaper layar penuh.
- Emoji keyboard untuk emoticon.
- Aplikasi Hangouts dan perpesanan terpadu
- Dukungan cloud print yang lebih baik.
- Integrasi Google Now yang lebih cerdas dan handsfree.

Android 5.0 (Lollipop)

Android 5.0 Lollipop pertama kali diperkenalkan pada Mei 2014.,Android lollipop merupakan perancangan ulang terbesar untuk Android. Smartphone Google Nexus 6, bersama dengan tablet Nexus 9-nya, merupakan perangkat pertama yang memiliki Lollipop yang telah terpasang sebelumnya.



Android Lollipop 5.0

Peningkatan terbesar yang dilakukan oleh Lollipop adalah pengenalan Material Design yang dengan cepat menjadi bahasa desain terpadu yang diterapkan di seluruh produk Google.

- Berikut adalah fitur yang dimilikinya:
- Dukungan pengaturan cepat yang lebih baik.
- Masa pakai baterai yang disempurnakan dengan mode Battery Saver yang baru.
- Layar kunci baru
- Fitur Smart Lock melalui Layanan Google Play.
- Mode tamu untuk berbagi perangkat.
- Pemasangan tombol.

Android 6.0 (Marshmallow)

Android 6.0 (Marshmallow) Di rilis pada tahun 2015. Ini Perangkat pertama yang dikirim bersama Marshmallow yang telah terpasang sebelumnya adalah smartphone Google Nexus 6P dan Nexus 5X, dengan tablet Pixel C-nya. Tujuan marshmallow memoles sudut kasar dan membuat versi Lollipop lebih baik lagi.



Android 6.2 Marshmallow

Berikut adalah fitur yang dimilikinya:

- Dukungan sidik jari resmi untuk perangkat.
- Dukungan untuk pembayaran seluler melalui Android Pay.
- Model perizinan yang lebih baik untuk aplikasi.
- Google Now di Tap.
- Deep menghubungkan Apps.

Android 7.0 (Nougat)

Android 7.0 (Nougat) Dirilis pada Tahun ,2016. Sebelum Nougat terungkap "Android N" dirujuk secara internal oleh Google sebagai "New York Cheesecake".



Android 7.0 Nougat

Berikut adalah fitur yang dimilikinya:

- Doze on the Go untuk waktu siaga yang lebih baik lagi.
- Multi Window untuk penggunaan dua aplikasi secara bersamaan.
- Aplikasi Setelan yang Lebih Baik
- Hapus semua di layar aplikasi baru-baru ini.
- Balas langsung ke pemberitahuan
- Notifikasi dibundel
- Pengaturan Cepat akan mengubah kustomisasi.

Android 8.0 (Oreo)

Pada bulan Maret 2017, Google Rilis Android 8.0 Oreo,bulan Agustus, Google mengkonfirmasi Oreo akan menjadi nama publik untuk Android 8.0. Sperti yang kita ketahui Ini adalah kedua kalinya Google memilih nama merek dagan untuk Android (Oreo dimiliki oleh Nabisco).



Android 8.0 Oreo

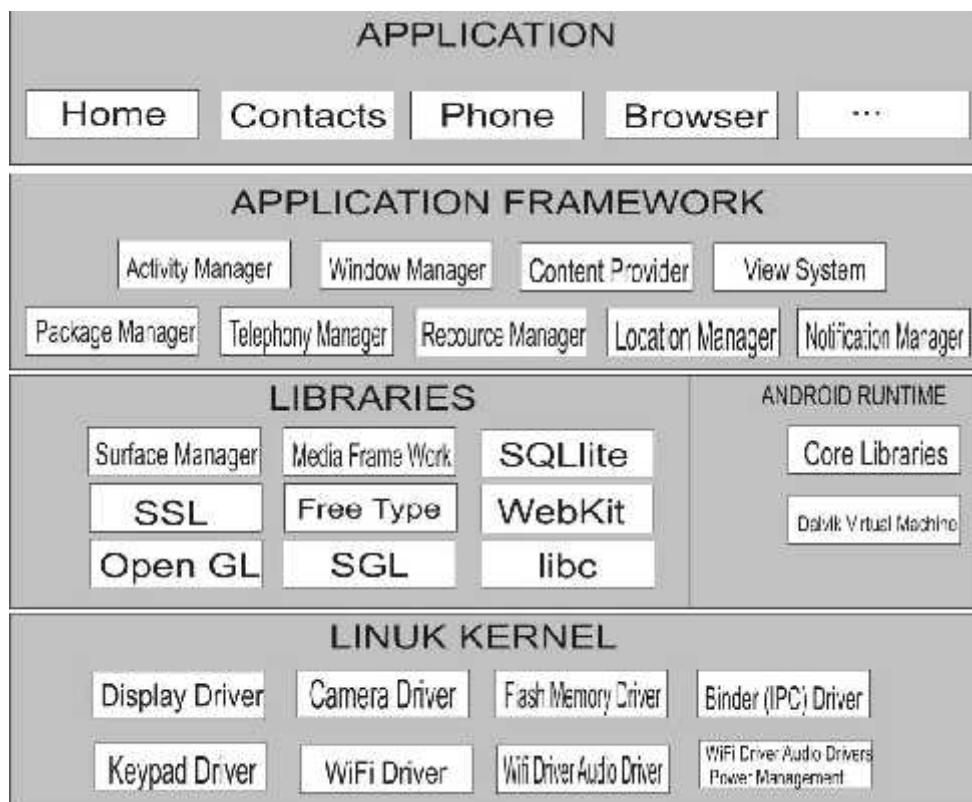
Adapaun sekarang versi ini adalah semua yang baru yang dimilikinya,diantaranya adalah sebagai berikut:

- Pemberitahuan untuk prioritas dan kategorisasi yang lebih baik.
- Pengelolaan warna lebih baik.

- Android O memiliki koleksi emoji baru yang telah didesain ulang.
 - Waktu boot lebih cepat: Pada perangkat Pixel, sekarang bisa mengalami waktu boot dua kali lebih cepat dibandingkan dengan Nougat.
 - Mengisi otomatis dan mengingat kata sandi dalam aplikasi.

Arsitektur Sistem Operasi Android

Android adalah tumpukan perangkat lunak berbasis Linux dengan sumber terbuka yang dibuat untuk berbagai perangkat. Diagram berikut menunjukkan arsitektur dari sistem operasi Android.



Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux

menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (Interprocess Communication) untuk mengatur aplikasi dan lubang keamanan.

Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar Berkeley Software Distribution (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

- Media Library untuk memutar dan merekam berbagai macam format audio dan video.
- Surface Manager untuk mengatur hak akses layer dari berbagai aplikasi.
- Graphic Library termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
- SQLite untuk mengatur relasi database yang digunakan pada aplikasi.
- SSL dan WebKit untuk browser dan keamanan internet.

Android Runtime

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. Android Runtime merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada.

Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan user interface dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut

- Activity Manager, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan "BackStack" untuk navigasi penggunaan.
- Content Providers, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.

- Resuource Manager, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan file layout.
- Location Manager, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
- Notification Manager, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.

Application Layer

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan widget. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android runtime dengan menggunakan kelas dan service yang tersedia pada framework aplikasi. Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti (native) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (Application Programming Interface) yang sama.

Building Block Dari Aplikasi Android

Aplikasi android dibangun menggunakan building block yaitu

- Activity

Activity mewakili satu layar dengan satu antarmuka pengguna. Misalnya aplikasi email mungkin memiliki satu aktivitas yang menampilkan daftar email baru, activity lain untuk menulis email dan activity lainnya untuk membaca email. Walaupun semua activity bekerja sama untuk membentuk pengalaman pengguna yang kohesif dalam aplikasi email, tetapi masing-masing tidak tergantung. Kerena, aplikasi berbeda bisa memulai salah satu activity (jika aplikasi email mengizinkannya). Misalnya, aplikasi kamera bisa memulai Activity dalam aplikasi meila yang membuat email baru agar pengguna bisa berbagai gambar.

- Services

Services adalah komponen yang berjalan di latar belakang untuk melakukan operasi yang berjalan lama atau untuk melakukan pekerjaan bagi proses jauh. Serivce tidak menyediakan antarmuka untuk pengguna.

- Content Provider

Aplikasi Android dapat menyimpan data pada file system, database SQLite, pada web atau media penyimpanan lainnya. Aplikasi dapat mengakses lokasi penyimpanan dengan bantuan content provider. Sebuah content provider dapat menyimpan ataupun membaca data dan membuatnya dapat diakses oleh semua aplikasi.

- Broadcast Receiver

Broadcast Receiver adalah komponen yang merespons pengumuman siaran dalam lingkup sistem. Banyak siaran yang berasal dari sistem, misalnya siaranya yang mengumumkan bahwa layar telah dinonaktifkan, baterai lemah, atau gambar telah direkam. Aplikasi juga bisa memulai siaran, misalnya untuk menginformasikan ke aplikasi lain bahwa sebagian data telah diunduh ke perangkat dan bisa digunakan aplikasi lain tersebut.

PERTEMUAN 2

Menjelajahi Lingkungan Pengembangan Android

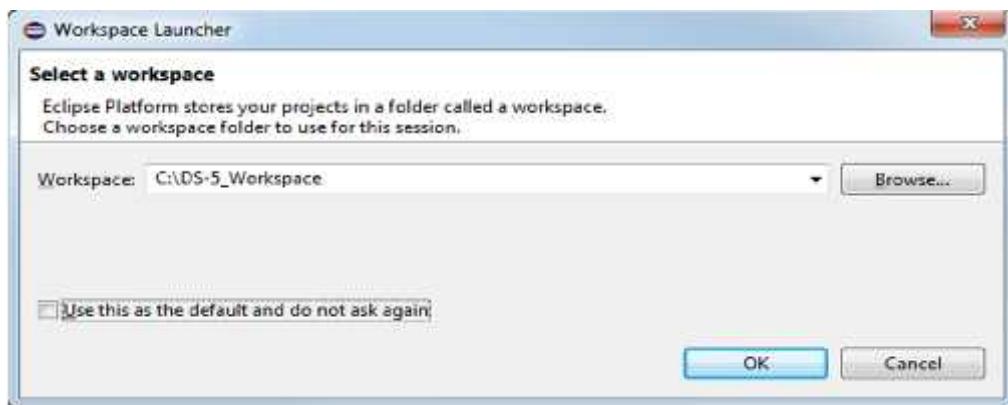
Tujuan

- Mahasiswa mampu membangun aplikasi pertama dengan menggunakan Android Development Tools.
- Mahasiswa memahami struktur dari Proyek Android.

Mempelajari Eclipse IDE

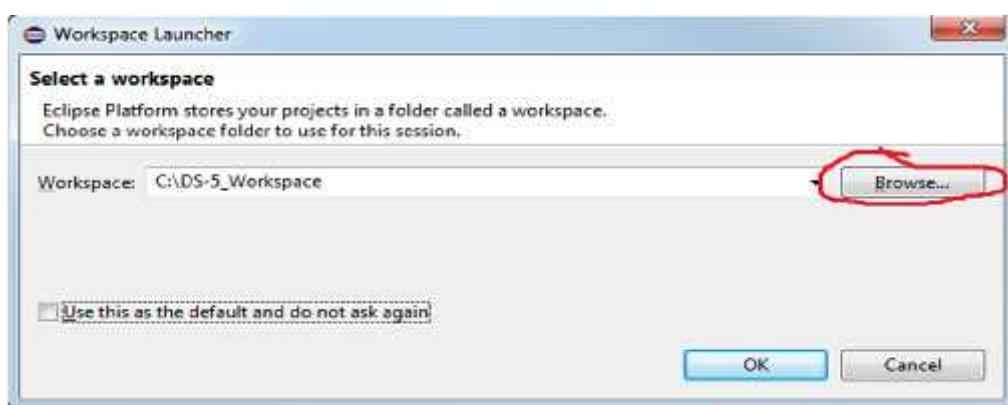
Untuk Mempelajari Eclipse IDE, Setidaknya perlu melakukan langkah-langkah sebagai berikut :

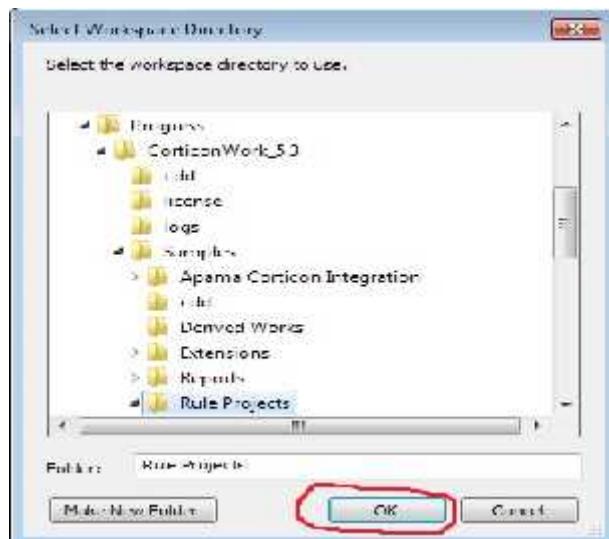
1. Click 2x pada *eclipse* di menu icon desktop. Maka akan muncul kotak dialog *Workspace Launcher*, seperti yang ditunjukkan pada gambar berikut.



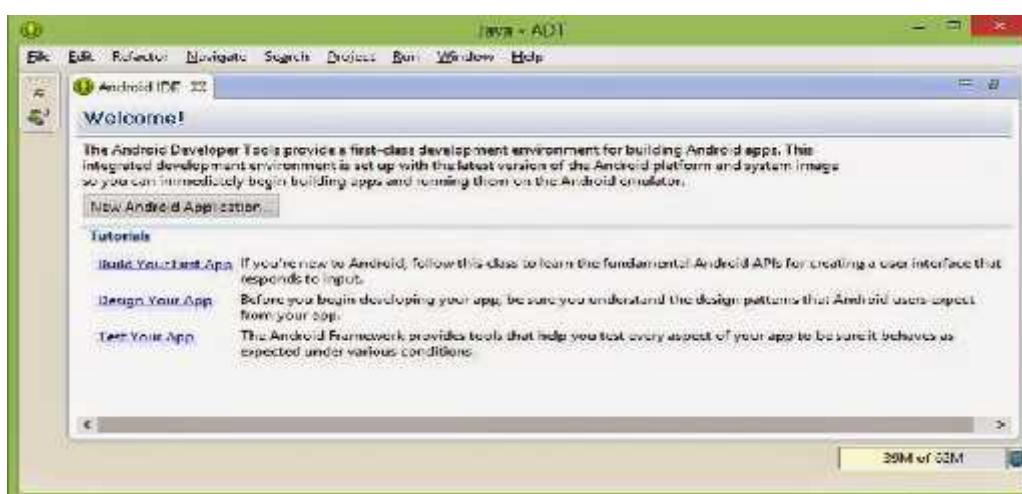
Workspace Launcher

2. Kemudian click *Browse* yang ada pada *Workspace Launcher*. Setelah itu pilih folder yang akan di gunakan terus click Ok, seperti yang ditunjukkan pada gambar berikut.

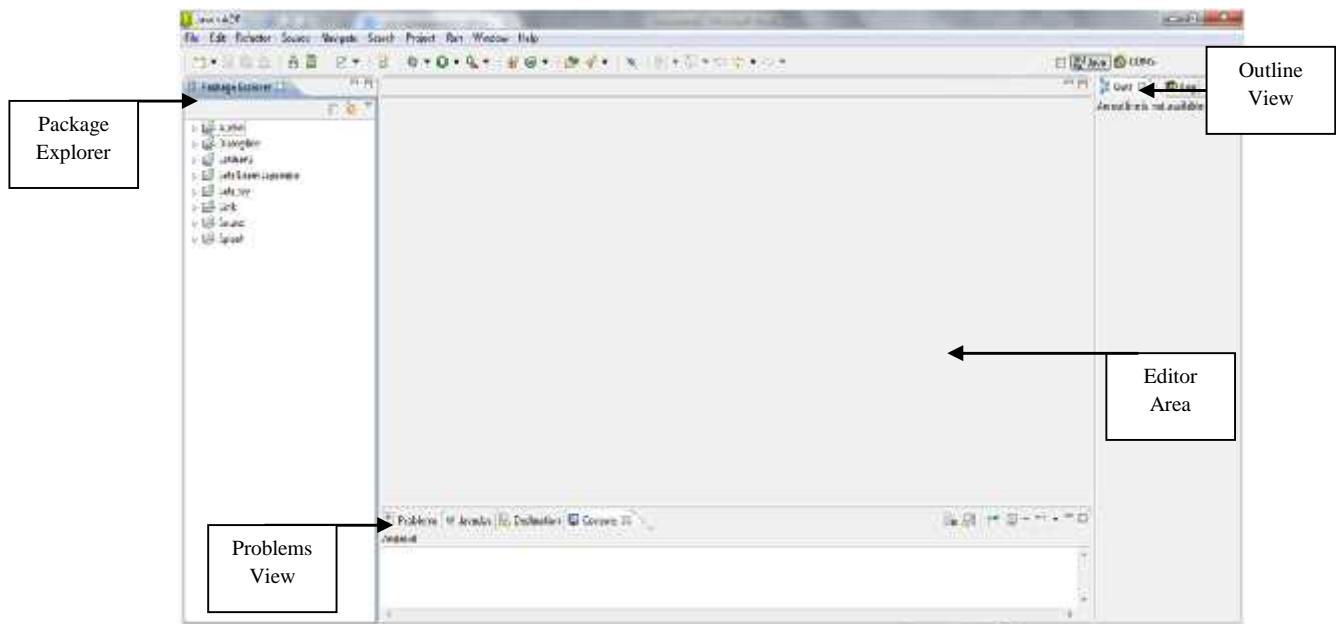


*Select Workspace Directory*

3. Setelah *click ok* maka akan muncul tampilan *Java-ADT* pada *eclipse*, seperti yang ditunjukkan pada gambar berikut.

*Java-ADT*

4. Kemudian close *Android IDE* maka akan muncul tampilan *ADT*, seperti yang ditunjukkan pada gambar berikut.



ADT

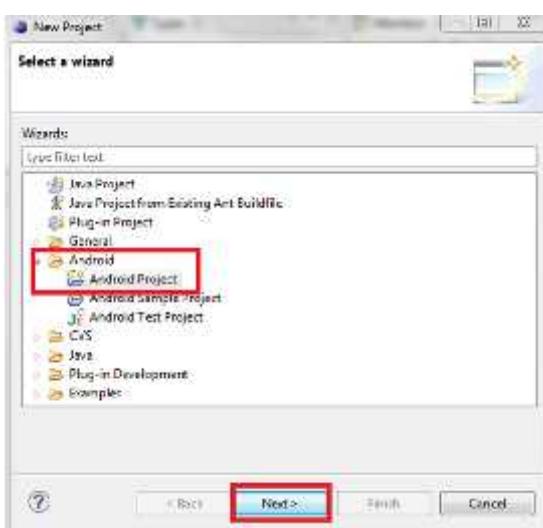
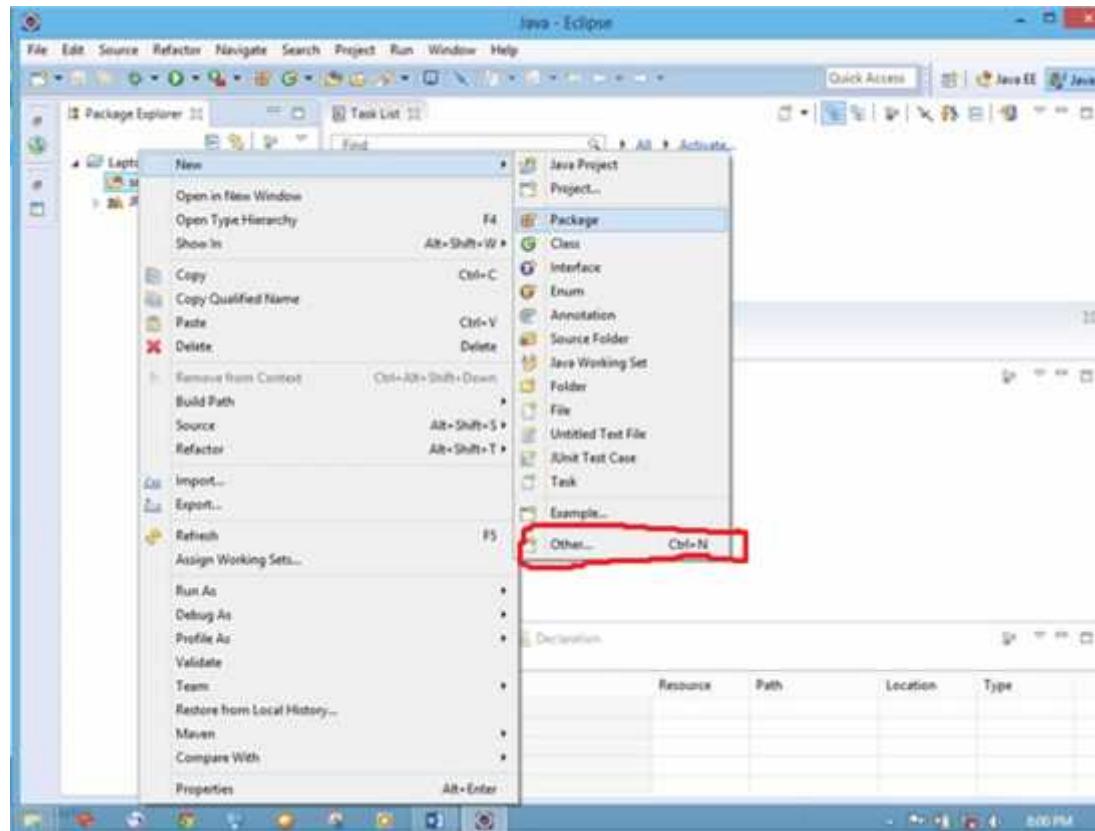
Ada beberapa unsur yang berbeda pada gambar di atas :

- Editor Area: Adalah tampilan isi file yang digunakan untuk memungkinkan Anda mengeditnya.
 - Outline View: Adalah tampilan elemen file yang terstruktur dan ada pada saat Editor Area.
 - Package Explorer: Adalah merupakan jendela yang menampilkan project-project yang sudah di buat dalam sebuah workspace.
 - Problems View: Adalah saat dimana terdapat berbagai macam masalah, kesalahan, atau peringatan yang ada di problems view.
5. Jika sudah selesai menggunakan *eclipse* bisa langsung close program pada tanda **X** di bagian pojok kanan atas dan kemudian tekan *ok*.

Mempelajari Struktur Android

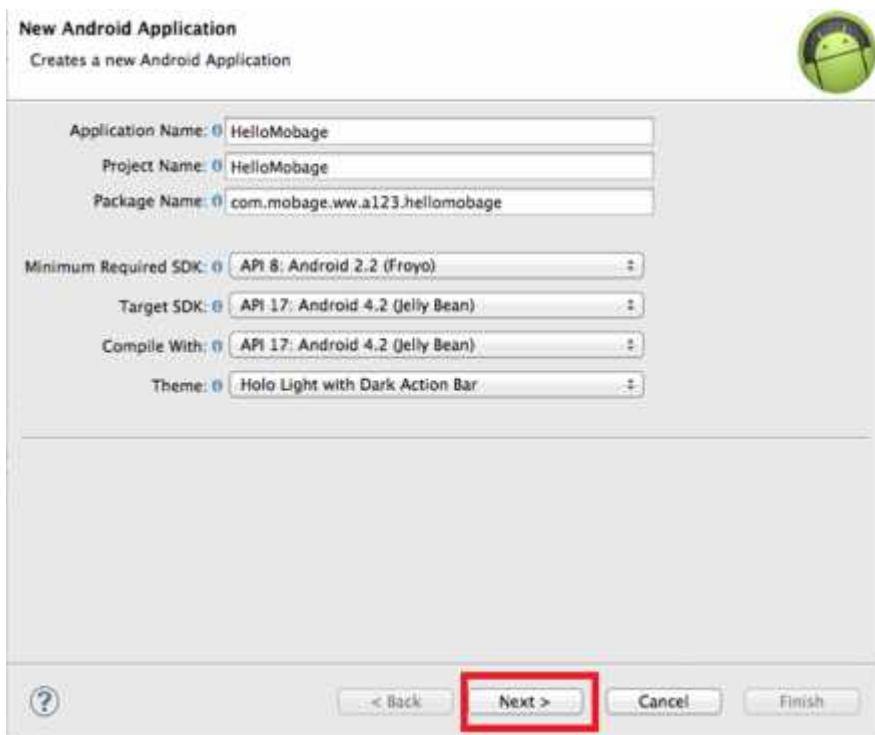
Dalam mempelajari stuktur android, Setidaknya perlu melakukan langkah-langkah sebagai berikut :

1. Click menu *File* pada *ADT* kemudian pilih *New* dan *Other* terus pilih *Android* cari *Android Project*. Maka akan muncul tampilan *new* seperti yang ditunjukkan pada gambar berikut.



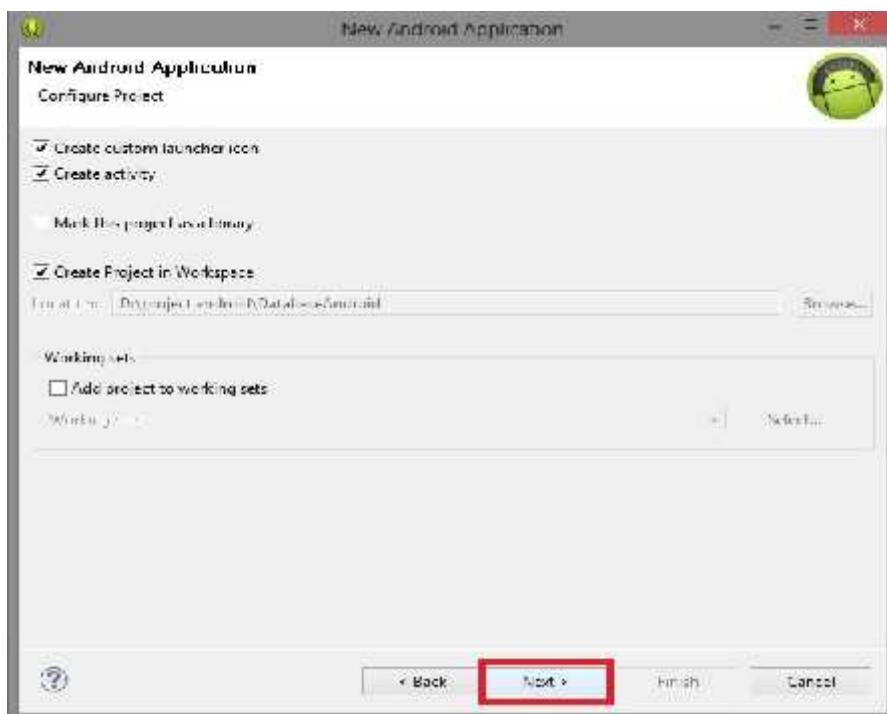
Tampilan New

2. Setelah click *next* pada tampilan *new*. Maka akan muncul tampilan *New Android Application* setelah itu isi bagian *application name*, *project name*, *package name*, seperti yang ditunjukkan pada gambar berikut.



New Android Application

3. Setelah itu akan muncul *Android Application Configure Project* langsung click next aja seperti yang ditunjukkan pada gambar berikut.



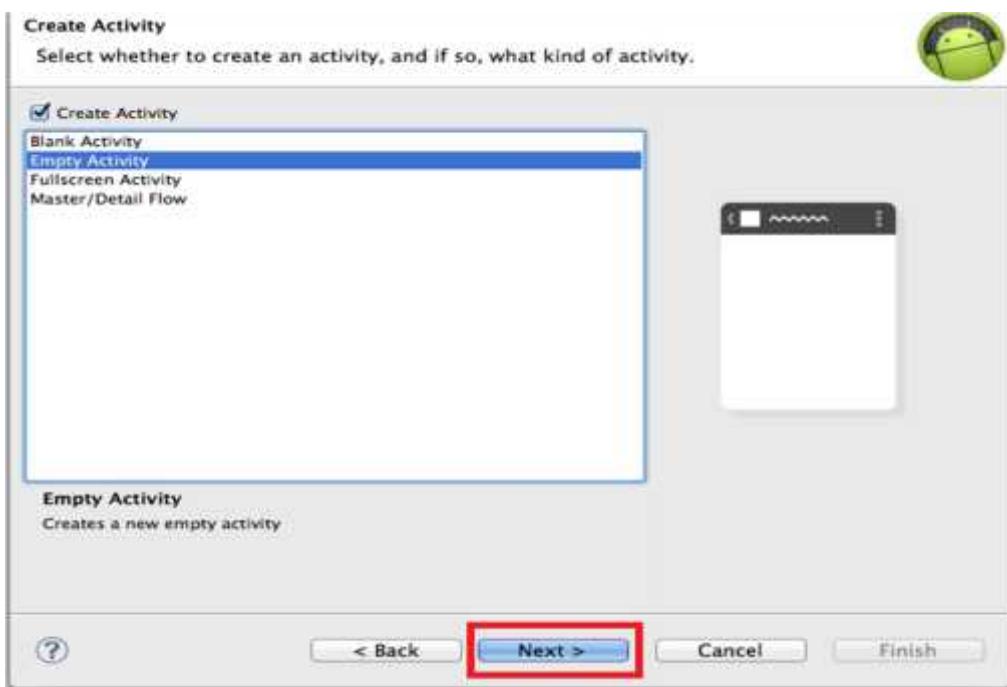
Configure Project

4. Setelah itu muncul tampilan seperti yang ditunjukkan pada gambar click next aja.



Configure the attributes of the icon set

5. Sehabis Next maka akan muncul tampilan dari *Create Activity* untuk memilih activity tampilan dari android, pilih *Blank Activity* / *Empty Activity* kemudian click next seperti yang ditunjukkan pada gambar berikut.



Create Activity

6. Setelah next tadi maka selesai sudah untuk pembuatan *New Android Application* selamat bekerja dan berkreativitas.

Membuat Baru Android Virtual Device (AVD)

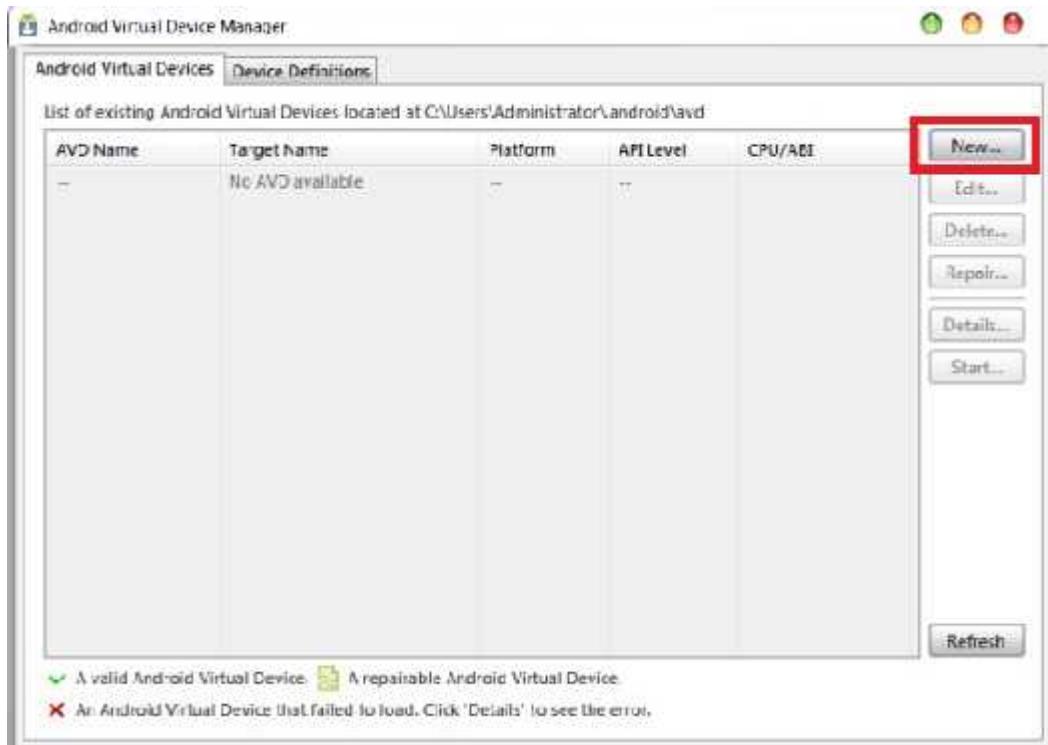
Untuk Membuat Android Virtual Device (AVD), Setidaknya perlu melakukan langkah-langkah sebagai berikut :

1. Click *Android Virtual Devices Manager* pada Toolbar seperti yang ditunjukkan pada gambar berikut.



Android Virtual Devices Manager

2. Berikutnya akan muncul tampilan dimana click *New* pada pojok kanan atas seperti yang ditunjukkan pada gambar berikut.

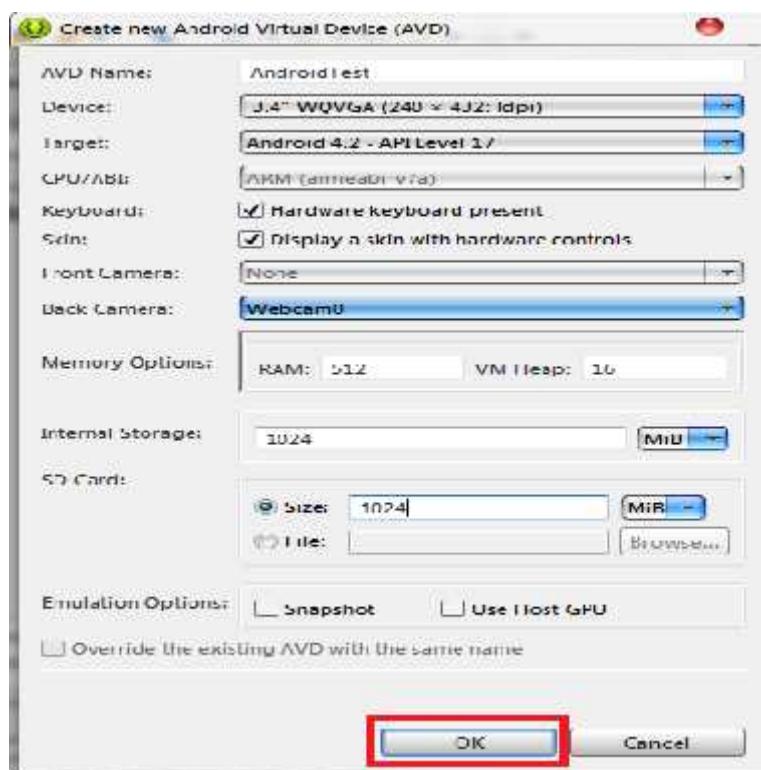


AVD

3. Setelah itu maka akan muncul tampilan *Create new AVD* dimana akan mengisi *AVD Name*, *Device*, *Target*, dan *CPU/ABI* seperti yang ditunjukkan pada gambar berikut.

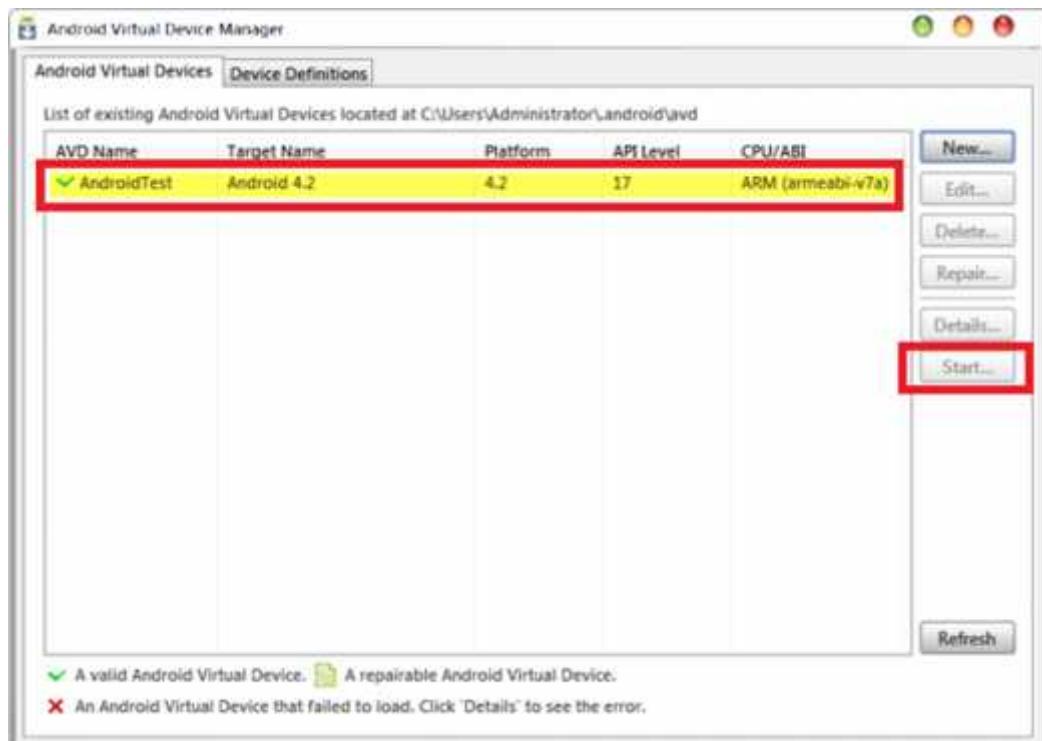


Create new Android Virtual Device



Create new AVD

4. Setelah Click Ok maka akan muncul AVD yang sudah di buat terus click *Start* seperti yang ditunjukkan pada gambar berikut.



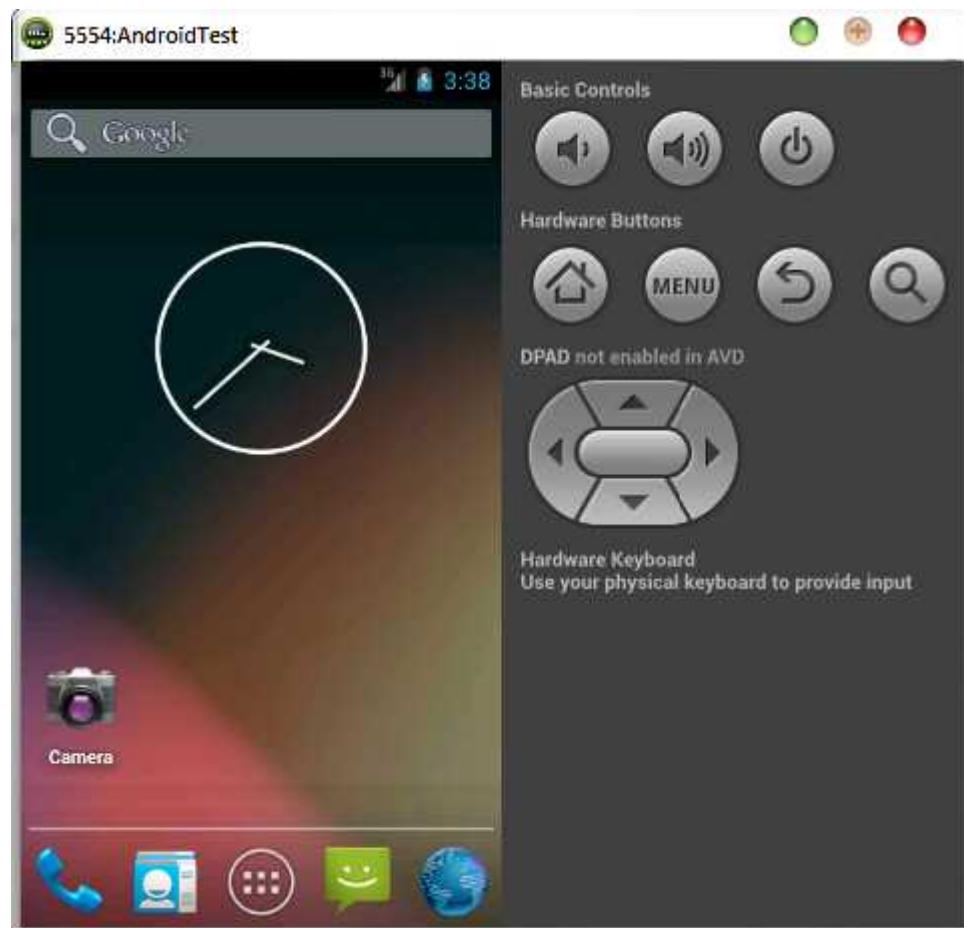
AVD New

5. Setelah itu muncul tampilan *Launch Options* dan click *launch* seperti yang ditunjukkan pada gambar berikut.



Launch Options

6. Setelah di *launch* maka AVD siap untuk di gunakan seperti yang ditunjukkan pada gambar berikut.



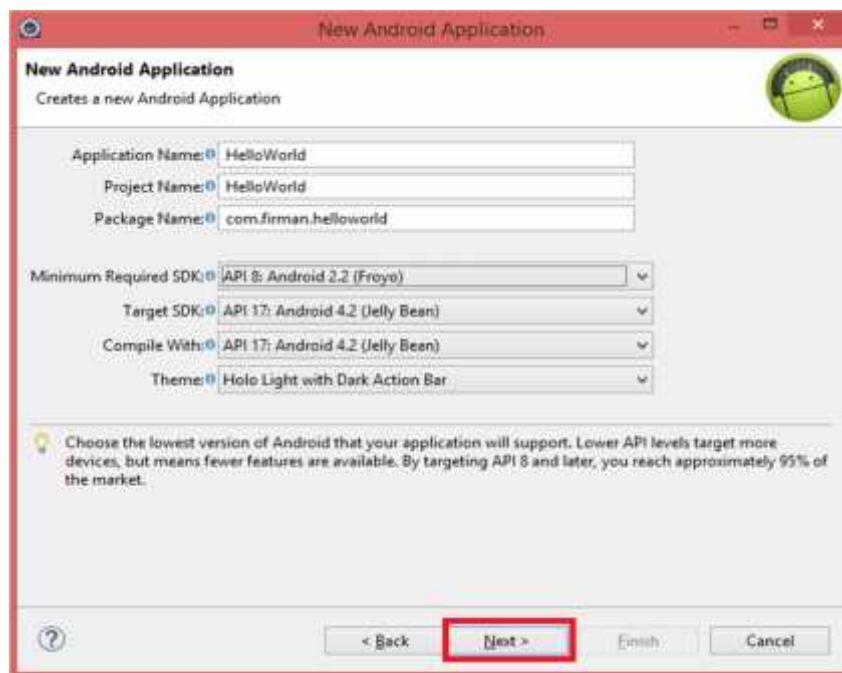
Tampilan AVD

Membuat Hello World

Untuk membuat hallo world, Setidaknya perlu melakukan langkah-langkah sebagai berikut :

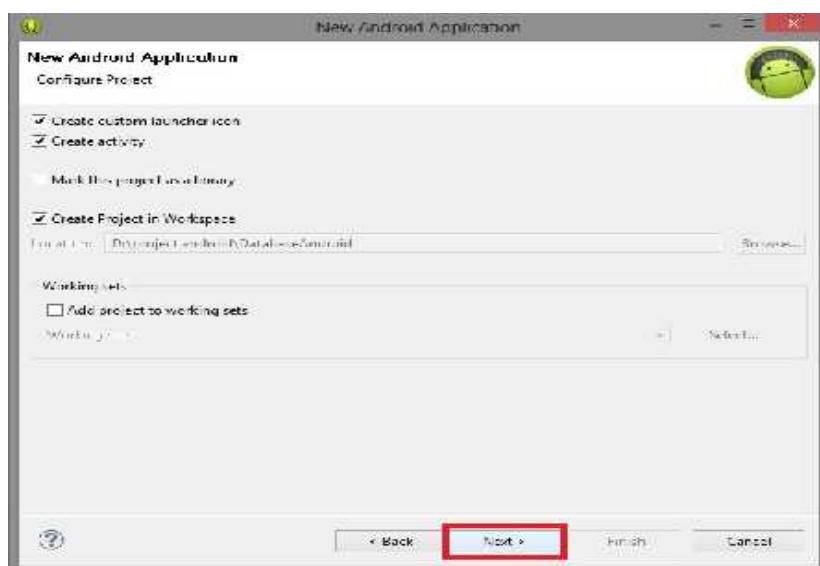
Tahap 1. Membuat Baru Android Project

1. Pilih *File > New > Android Application Project*. Maka akan muncul tampilan *New Android Application* dan isi *Application Name* dengan *Hello World* Kemudian click *Next* seperti yang ditunjukkan pada gambar berikut.



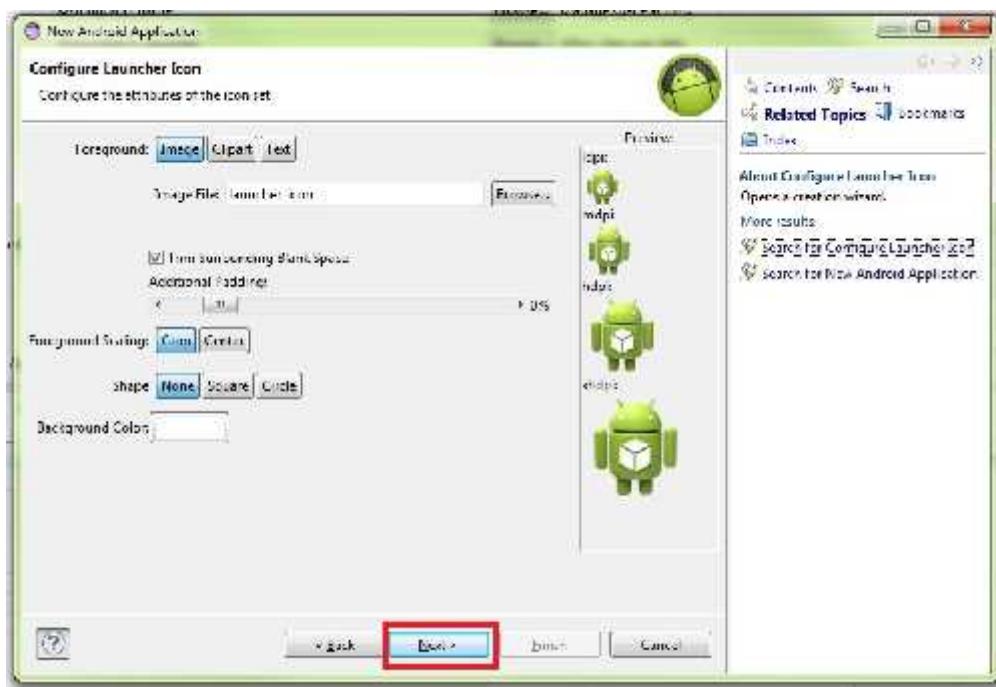
File New Hello World

2. Setelah itu akan muncul *Android Application Configure Project* langsung click next aja seperti yang ditunjukkan pada gambar berikut.



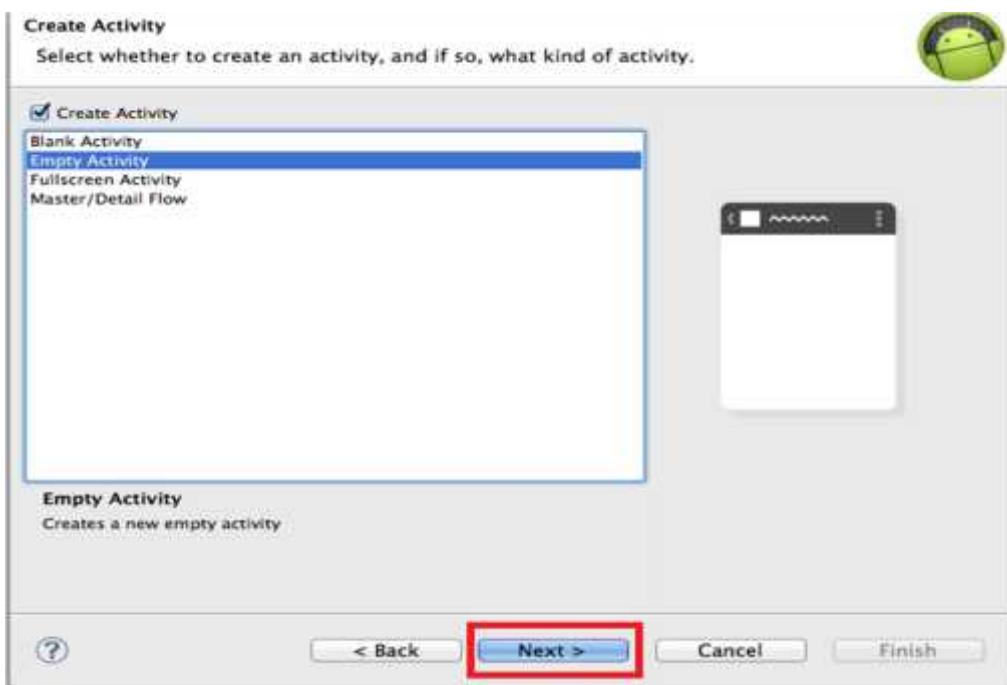
Configure Project

3. Setelah itu muncul tampilan seperti yang ditunjukkan pada gambar click next aja.



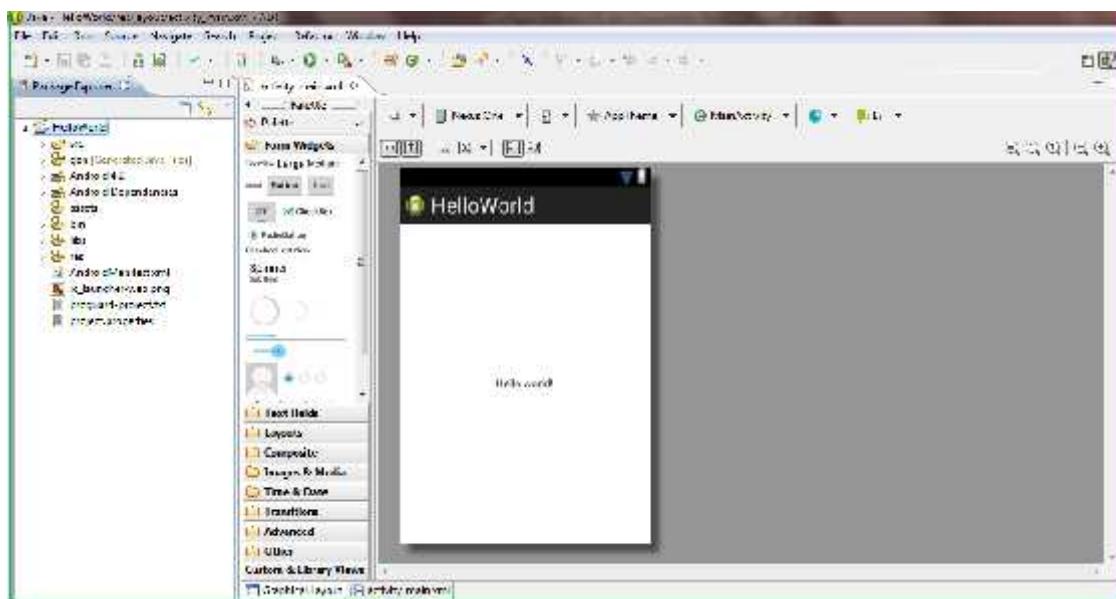
Configure the attributes of the icon set

4. Sehabis Next maka akan muncul tampilan dari *Create Activity* untuk memilih activity tampilan dari android, pilih *Blank Activity* / *Empty Activity* kemudian *click next / Finish* seperti yang ditunjukkan pada gambar berikut.



Create Activity

5. Setelah itu sebuah aplikasi baru dengan nama *hello world* dibuat dalam *package explorer* seperti yang ditunjukkan pada gambar berikut.



Tampilan Hallo World

Tahap 2. Memodifikasi Aplikasi

1. Pilih *res* cari *values* yang ada di dalam *Package Explorer*.
2. Click 2x pada file *string.xml* seperti yang ditunjukkan pada gambar berikut.



string.xml

3. Setelah itu hapus bagian kode yang disorot berikut ini :

```
<string name="hello_world">Hello world!</string>
```

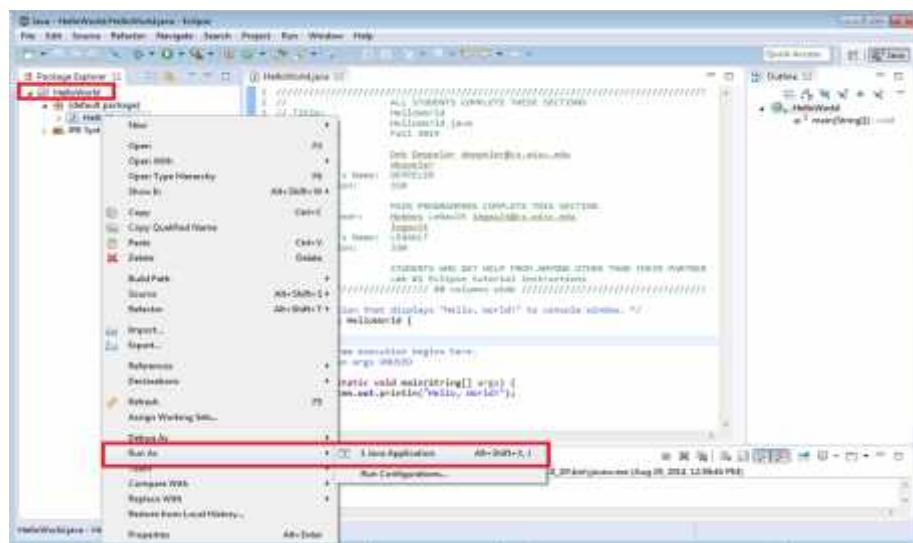
4. Kemudian ketik kode yang disorot berikut ini :

```
<string name="hello_world">Hello! Android</string>
```

5. Setelah selesai pilih *File > Save All.*

Tahap 3. Menjalankan Aplikasi

1. Click kanan pada *package Explorer* kemudian pilih *Run As > Java Application* seperti yang ditunjukkan pada gambar berikut.



Run app

2. Setelah di *Run* maka akan tampil pada *AVD* seperti yang ditunjukkan pada gambar berikut.



Run AVD Hello World!

Latihan

Membuat sebuah project pertama di Android yang menampilkan nama, nim dan jurusan mahasiswa.

PERTEMUAN 3

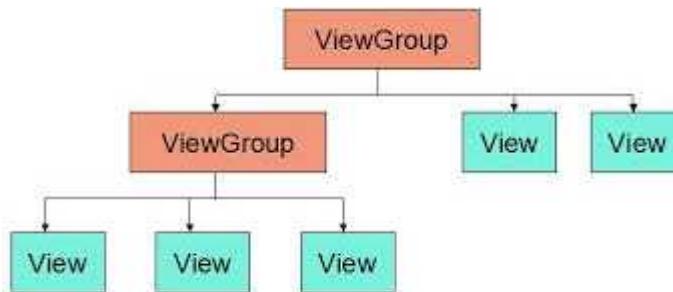
Membangun Tampilan Antar Muka

Tujuan

- Mahasiswa memahami view dan viewgroup pada tampilan antarmuka android.
- Mahasiswa mampu mengaplikasikan dan mengkreasikan view dan viewgroup untuk tampilan antarmuka aplikasi android.
- Mahasiswa mampu menampilkan tampilan antarmuka pada perangkat android.

Tampilan antar muka android terdiri dari dua tipe komponen

- View
- ViewGroup



Gambar diatas menunjukkan hubungan antara view dan viewgroup. ViewGroup berlaku sebagai wadah/kontainer untuk view ataupun ViewGroup lainnya. View merupakan bagian terkecil dan tidak dapat menampung objek lainnya.

Menggunakan Views

View merupakan komponen visual dari android, dimana view sama dengan kontrol pada lingkungan pemrograman desktop. Dengan kata lain, view adalah komponen antar muka dari aplikasi android.

Class view yang merupakan bagian dari paket android.view merupakan dasar dari building block tampilan antar muka aplikasi android. Class view merupakan induk dari semua widget seperti button, label, dan lain sebagainya. Android widget memungkinkan pengguna untuk berinteraksi dengan aplikasi android. Pada android class widget terdapat pada paket android.widget.

Beberapa widget dasar android adalah: TextView, Button, Image, EditText, CheckBox, RadioButton, ListView. Setiap widget memiliki property atau atribut yang mengatur

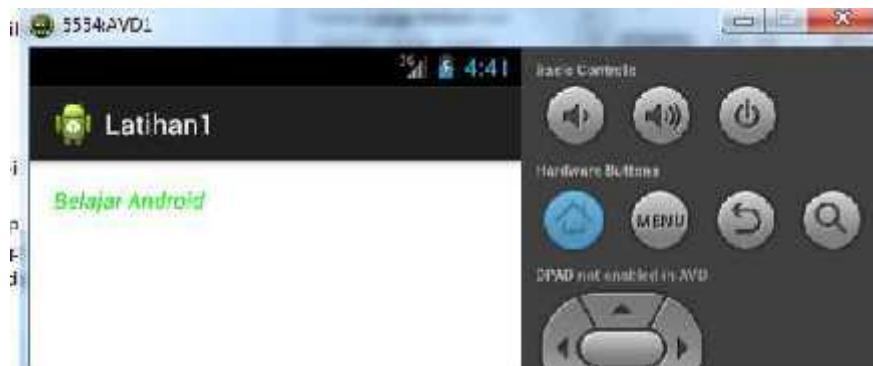
bagaimana widget itu ditampilkan, seperti tinggi dan lebar widget. Property ini dapat diset melalui xml layout, property editor atau melalui program. Beberapa widget memiliki beberapa event yang ter-trigger berdasarkan aksi dari pengguna, misalnya event click pada button. Berikut adalah contoh pengaplikasian widget. Untuk menampilkan cukup drag icon ke layar kerja.

TextView digunakan untuk menampilkan label teks. View ini sudah kita gunakan. Beberapa contoh property dari TextView adalah android:textSize, android:textStyle, android:textColor.

Contoh:

```
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentTop="true"  
    android:text="@string/belajar"  
    android:textColor="#00ff00"  
    android:textStyle="italic"/>
```

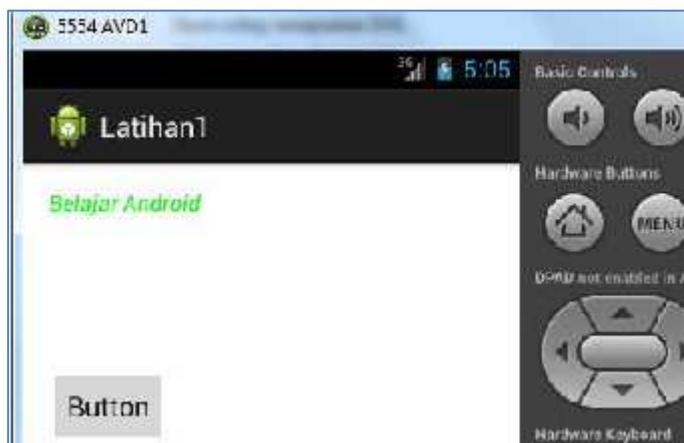
Bila dijalankan maka hasilnya



Button merupakan turunan dari **TextView** sehingga yang berlaku di **textView** juga berlaku di **button**. Contoh :

```
<Button  
    android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignLeft="@+id/textView1"  
    android:layout_below="@+id/textView1"  
    android:layout_marginTop="98dp"  
    android:text="@string/button" />
```

Maka Hasilnya



ImageView adalah widget yang digunakan untuk menampilkan image. Image yang akan ditampilkan dalam button diletakkan di direktori /res sesuai dengan resolusinya. Jadi untuk hasil yang maksimal, untuk setiap image yang akan anda gunakan harus disediakan 4 image untuk setiap resolusi (mdpi sampai dengan xxhdpi). Hal ini untuk mencegah gambar yang digunakan terlihat pecah atau terlalu kecil pada device dengan resolusi berbeda. Pada

folder res sudah terdapat image android. Sebagai contoh kita tampilkan gambar tersebut. Drag image view kelayar kerja.



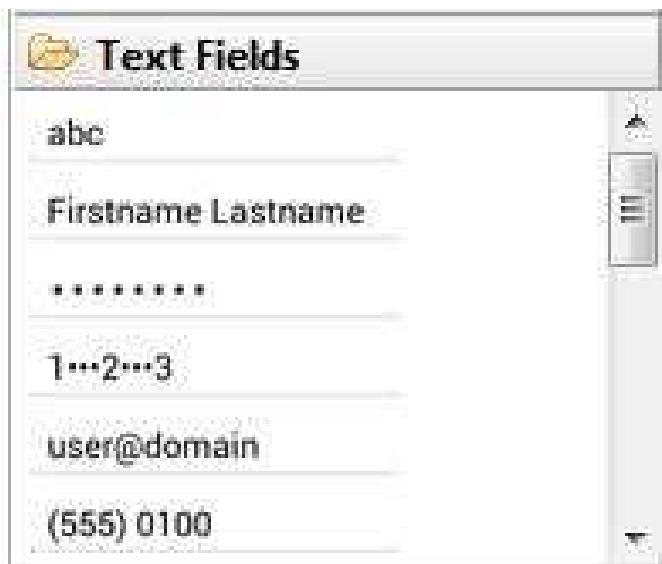
Contoh penerapan pada main.xml

```
<ImageView  
    android:id="@+id/imageView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentRight="true"  
    android:layout_alignTop="@+id/textView1"  
    android:layout_marginRight="14dp"  
    android:src="@drawable/ic_launcher" />
```

setelah dijalankan



EditText digunakan untuk menerima input dari pengguna. Pada palette telah disediakan berbagai jenis EditText, silahkan dicoba satu persatu dan perhatikan XML yang dihasilkan. sebagai contoh adlah sebagai berikut



```
<EditText
```

```
    android:id="@+id/editText1"  
  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
  
    android:layout_alignLeft="@+id/textView1"  
  
    android:layout_below="@+id/imageView1"  
  
    android:ems="10" >
```

setelah dijalankan anda dapat menginput texs di dalamnya



RadioButton hanya satu pilihan yang boleh aktif (mutual exclusive) di dalam satu group yang disebut radioGroup. Sedangkan User dapat memilih lebih dari satu pilihan dengan checkbox. Pada palette, Checkbox ada di bagian FormWidgets.

Contoh script

```
<CheckBox  
    android:id="@+id/checkBox1"  
  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
  
    android:layout_alignBaseline="@+id radioButton1"  
    android:layout_alignBottom="@+id radioButton1"  
  
    android:layout_alignRight="@+id/imageView1"  
  
    android:layout_marginRight="33dp"  
    android:text="@string/cek1" />
```

```
<CheckBox  
    android:id="@+id/checkBox2"  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
    android:layout_alignBaseline="@+id radioButton2"  
  
    android:layout_alignBottom="@+id radioButton2"  
    android:layout_alignLeft="@+id/checkBox1"  
  
    android:text="@string/cek2" />
```

```
<RadioButton
```

```
    android:id="@+id/radioButton1"
```

```
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
  
    android:layout_alignLeft="@+id/editText1"  
  
    android:layout_below="@+id/editText1"  
  
    android:layout_marginTop="22dp"  
  
    android:text="@string/android" />
```

<RadioButton

```
        android:id="@+id/radioButton2"  
  
        android:layout_width="wrap_content"  
  
        android:layout_height="wrap_content"  
  
        android:layout_alignLeft="@+id/editText1"  
  
        android:layout_below="@+id/radioButton1"  
  
        android:text="@string/IoS" />
```

Dari keseluruhan hasilnya adalah



Menggunakan ViewGroup

Ketika membuat tampilan antar muka, kamu memerlukan kontainer untuk menampung dari view. Kontainer/wadah ini disebut dengan ViewGroup. Pada android terdapat layout yang digunakan untuk menyangga widget dengan berbagai kebutuhan.

Linear Layout digunakan untuk mengatur widget dalam baris tunggal. Baris bisa berupa horizontal maupun vertikal. Tinggal mengubah atribut orientasi dari layout. Untuk mengubah layout menjadi liner dapat dilakukan dengan mengubah kode xml.

```
<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="horizontal"
    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=".MainActivity" >

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 2" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 3" />

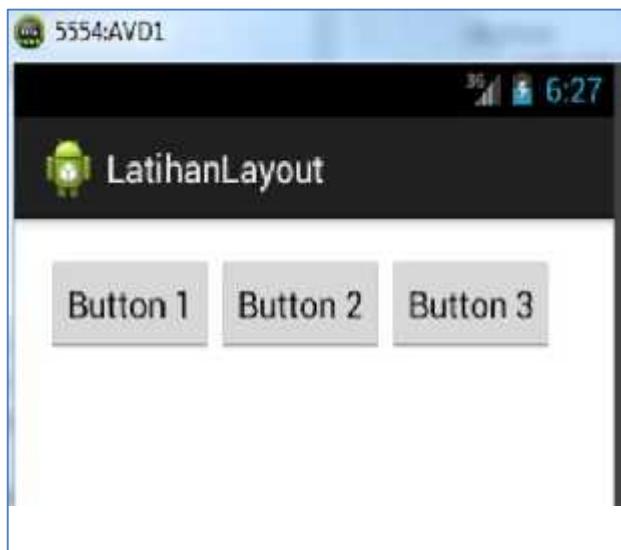
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:text="Button 3" />

</LinearLayout>
```

Tampilan Linear Layout dengan orientasi horizontal



Dengan mengubah orientasi menjadi vertikal `android:orientation="vertical"` Maka tampilan akan menjadi sebagai berikut



Relative layout memungkinkan anda menentukan letak komponen secara relatif dengan komponen lainnya. Layout ini merupakan paling flexible yang memungkinkan meletakan komponen sesuai keinginan. Di relative layout biasa menggunakan, above, below, left, right untuk mengatur isi komponen. Sebagai contoh adalah memasukan 2 button.

Beberapa atribut yang mengatur posisi widget di relative layout adalah:

- Android:layout_above
- Android:layout_below
- Android:layout_toLeftOf
- Android:layout_toRightOf.

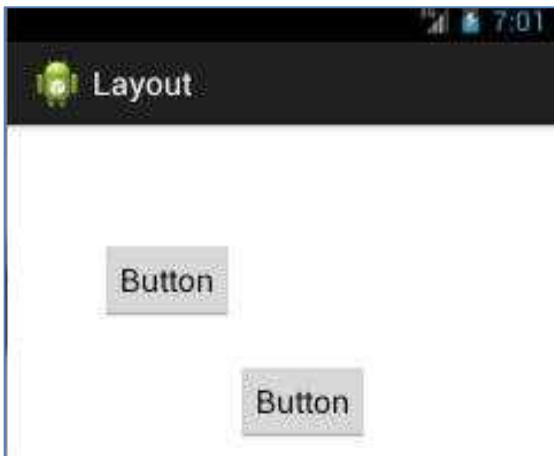
Seperti contoh berikut :

```
<Button  
    android:id="@+id/button1"  
  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentTop="true"  
  
    android:layout_marginLeft="37dp"  
    android:layout_marginTop="50dp"  
    android:text="Button" />
```

```
<Button  
    android:id="@+id/button2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
  
    android:layout_below="@+id/button1"  
    android:layout_marginTop="22dp"  
    android:layout_toRightOf="@+id/button1"
```

```
        android:text="Button" />
```

Tampilan dari Relative Layout



Untuk layout_width dan layout_height ada dua pilihan: match_parent (view berusaha sebesar parent) dan wrap_content (view hanya berusaha sebesar content yang berada di dalamnya). Sedangkan padding menyatakan jarak antara pinggir layar dengan komponen. Perhatikan penggunaan atribut layout_toRightOf menyatakan bahwa button2 berada disebelah kanan button1.

Untuk mengatur posisi ada tambahan lima atribut: android:layout_alignTop, android:layout_alignBottom, android:layout_alignLeft, android:layout_alignBaseline.

Latihan

Membuat program aplikasi android dengan menerapkan salah satu atau lebih layout yang ada dan menggunakan minimal 3 buah widget

PERTEMUAN 4

MENAMBAHKAN FUNSIONALITAS PADA ANTAR MUKA

Tujuan

- Mahasiswa memahami daur hidup dan metode dari activity.
- Mahasiswa mampu mengaplikasikan UI Event Handler pada Activity
- Mahasiswa mampu mengaplikasikan Intent pada Activity

Daur Hidup Activity

Sebuah *activity* menyediakan antar muka untuk satu layar di sebuah aplikasi. Pengguna dapat berinteraksi dengan *activity* untuk melakukan tugas yang diinginkan. *Activity* menyediakan antar muka yang dirancang berbeda untuk tujuan yang terdefinisi baik, seperti *activity* untuk mengambil gambar, mengirim pesan, ataupun melakukan panggilan telepon.

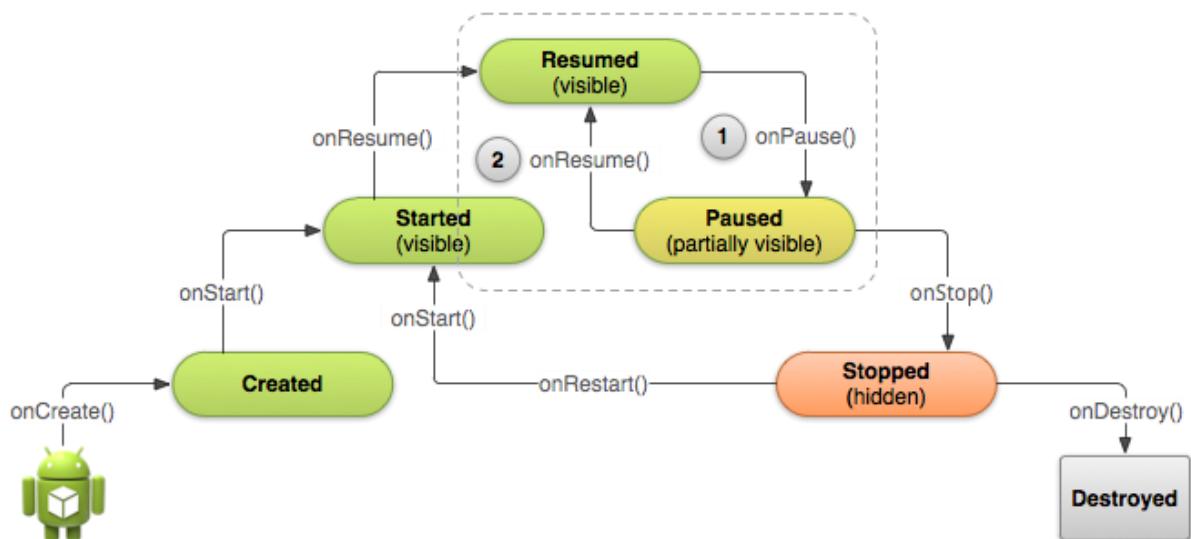
Sebuah aplikasi dapat memiliki banyak *activity*. *Activity* yang ditampilkan pada pengguna saat aplikasi dijalankan disebut dengan *main activity*. Sebuah *activity* memiliki daur hidup yang beragam. Class *activity* memiliki *lifecycle callback methods* untuk setiap *states*/keadaan.

Dalam daur hidupnya, *activity* bisa eksist pada tiga keadaan:

- **Running**, pada keadaan ini, *activity* terlihat oleh pengguna pada layar dan pengguna dapat berinteraksi dengan *activity*.
- **Paused**, pada keadaan ini, *activity* lain sedang ada di *foreground* dan pengguna berfokus pada *activity* tersebut. Tetapi *activity* masih hidup karena sebagian masih terlihat walaupun tidak dalam fokus. Sebagai contoh ketika semi-transparent *activity* terbuka (seperti dialog). Ketika *activity* masih sebagian terlihat tetapi bukan menjadi fokus, *activity* tersebut masuk kedalam paused. Namun, *Activity* yang dipaused bisa di “kill” oleh sistem dalam situasi *low-memory* yang ekstreme.
- **Stopped**, pada keadaan ini, *activity* ditutup keseluruhan oleh *activity* yang lain dan *activity* sekarang ada di *background*. *Activity* pada keadaan stopped tidak terlihat lagi oleh pengguna dan dapat di “kill” oleh sistem jika memory diperlukan untuk aplikasi lain. Sebagai contoh, jika sistem atau pengguna membuka aplikasi baru yang memerlukan memory, semua *activity* dengan keadaan stopped dapat di “kill” secara otomatis oleh sistem.

Metode Daur Hidup Activity

Ketika sebuah activity pada transisi masuk ataupun keluar dari berbagai keadaan, hal ini diberitahukan kedalam berbagai metode callback dari class Activity. Metode Callback ini bisa di overridden untuk menambahkan kode yang diperlukan dieksekusi saat transisi dari setiap keadaan/state. Berikut adalah metode daur hidup activity.



Metode lifecycle callback yang terdapat pada class activity adalah

- **`onCreate()`** : metode ini dipanggil ketika activity dimulai. Setelah metode ini dieksekusi, selanjutnya metode `onStart()` dieksekusi.
- **`onStart()`** : metode ini dipanggil sebelum activity ditampilkan kepada pengguna. Diikuti oleh metode `onResume()` jika activity menuju *foreground* atau metode `onStop()` jika activity menjadi tersembunyi. Pada metode ini, anda dapat memulai proses seperti mengecek koneksi ataupun mulai animasi.
- **`onResume()`** : metode ini dipanggil ketika *activity* mulai berinteraksi dengan pengguna. Selalu diikuti oleh metode `onPause()`.
- **`onPause()`** : metode ini dipanggil ketika sistem akan melanjutkan *activity* sebelumnya. Diikuti oleh `onResume()` jika *activity* ingin dikembalikan ke *foreground*, atau `onStop()` jika *activity* menjadi tidak terlihat oleh pengguna.
- **`onStop()`** : metode ini dipanggil ketika *activity* tidak muncul lagi kepada pengguna, karena *activity* lain sudah dilanjutkan dan menutup *activity* bersangkutan. Diikuti oleh `onRestart()` jika *activity* kembali berinteraksi dengan pengguna atau `onDestroy()` jika *activity* akan di *destroy*.

- **onDestroy()** : method terakhir yang dipanggil ketika activity akan di destroy. Hal ini bisa terjadi karena *activity* di *finish*(method *finish()* dipanggil atau karena sistem mendestroy dari memory untuk menghemat tempat).

Menangani Event Antar muka pada class Activity

Pada android, setiap widget mempunyai bermacam events Antar Muka. Beberapa contoh event tersebut seperti mengklik tombol button ataupun memilih check box. Sebuah aplikasi memerlukan sebuah action untuk menangani kejadian dari event tertentu. Anda dapat mengintersep dan menangani event pada aplikasi android dengan menggunakan dua metode:

- Menggunakan Event handler

Jika anda membangun komponen kustom dari view, maka anda dapat menentukan beberapa metode callback yang digunakan sebagai penangan event default. Contoh dari event handler

- `onKeyDown(int, KeyEvent)` – dipanggil ketika terjadi event key down.
- `onKeyUp(int, KeyEvent)` - dipanggil ketika terjadi event key up..
- `onTrackballEvent(MotionEvent)` – dipanggil ketika terjadi event trackball motion.
- `onTouchEvent(MotionEvent)` – dipanggil ketika terjadi event touch screen motion.
- `onFocusChanged(boolean, int, Rect)` – dipanggil saat melihat perubahan fokus.

- Menggunakan event listener

Event handler memerlukan anda untuk mengextend setiap object view untuk menghandle event. Namun hal ini tidak praktis digunakan. Karena itu class view menyediakan bermacam interface yang mengandung metode single callback. Interface ini dikenal dengan event listener. Beberapa contoh dari event listener dan method callback bersangkutan

Interface	Method Name	Deskripsi
<code>View.OnClickListener.</code>	<code>onClick()</code>	Dipanggil ketika pengguna mengklik atau melakukan tap pada view
<code>View.OnLongClickListener.</code>	<code>onLongClick()</code>	Dipanggil ketika pengguna melakukan menekan lama pada view
<code>View.OnFocusChangeListener.</code>	<code>onFocusChange()</code>	Dipanggil ketika pengguna

		menavigasi menuju ataupun pergi dari sebuah item, menggunakan tombol navigasi ataupun trackball
View.OnKeyListener	onKey ()	Dipanggil ketika pengguna fokus pada item dan menekan atau melepas tombol perangkat keras pada perangkat
View.OnTouchListener	onTouch ()	Dipanggil ketika pengguna melakukan sebuah action yang digolongkan touch event, yang mengandung menekan, melepas, atau pergerakan gesture pada layar.
View.OnCreateContextMenuListener	onCreateContextMenu ()	Dipanggil ketika context menu dibuat sebagai hasil dari penekanan lama pada view

Contoh implementasi Event Listener

Buatlah project baru dengan nama latihan4.

Application Name:

Project Name:

Package Name:

Setelah selesai membuat project baru selanjutnya pada xml file layout tambahkan kode berikut. Pada layout tambahkan widget button dengan id: button1 dan text: "klik".

```

<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.contohonclicklistener.MainActivity" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="54dp"
        android:text="klik" />

</RelativeLayout>
  
```

Pada bagian java tambahkan kode berikut, dimana mengimplementasi interface View.OnClickListener.

```

package com.contohonclicklistener;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

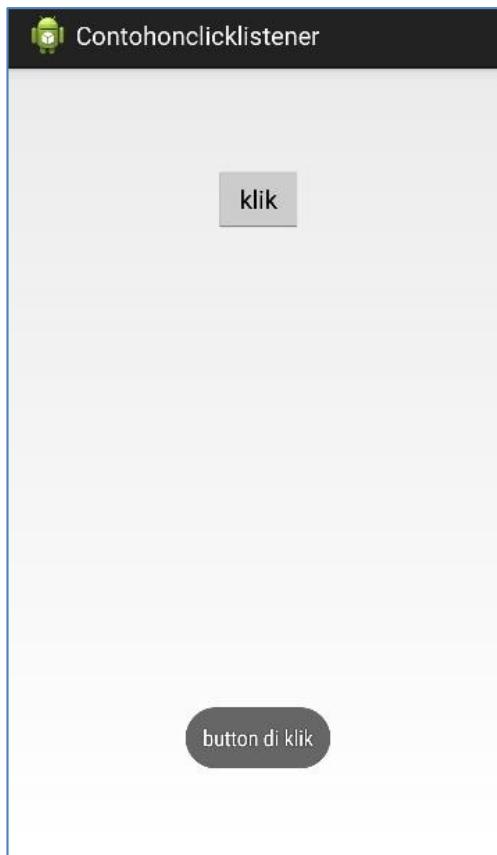
public class MainActivity extends Activity implements View.OnClickListener {

    Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button=(Button)findViewById(R.id.button1);
        button.setOnClickListener(this);
    }

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        Toast.makeText(getApplicationContext(), "button di klik",
        Toast.LENGTH_SHORT).show();
    }
}
  
```

Saat di running, akan tampil gambar berikut. Ketika tombol klik di tekan maka akan muncul pesan toast “button di klik”.



Mengaktifkan komponen aplikasi

Aplikasi android biasanya terdiri dari tidak hanya satu activity tapi terdiri dari banyak activity. Suatu activity dipanggil dari activity lainnya. Untuk dapat mewujudkan hal ini dipergunakanlah intent. Fungsi dari intent adalah untuk mengaktifkan komponen aplikasi seperti activity, service dan broadcast receiver.

Terdapat dua buah jenis intent yaitu

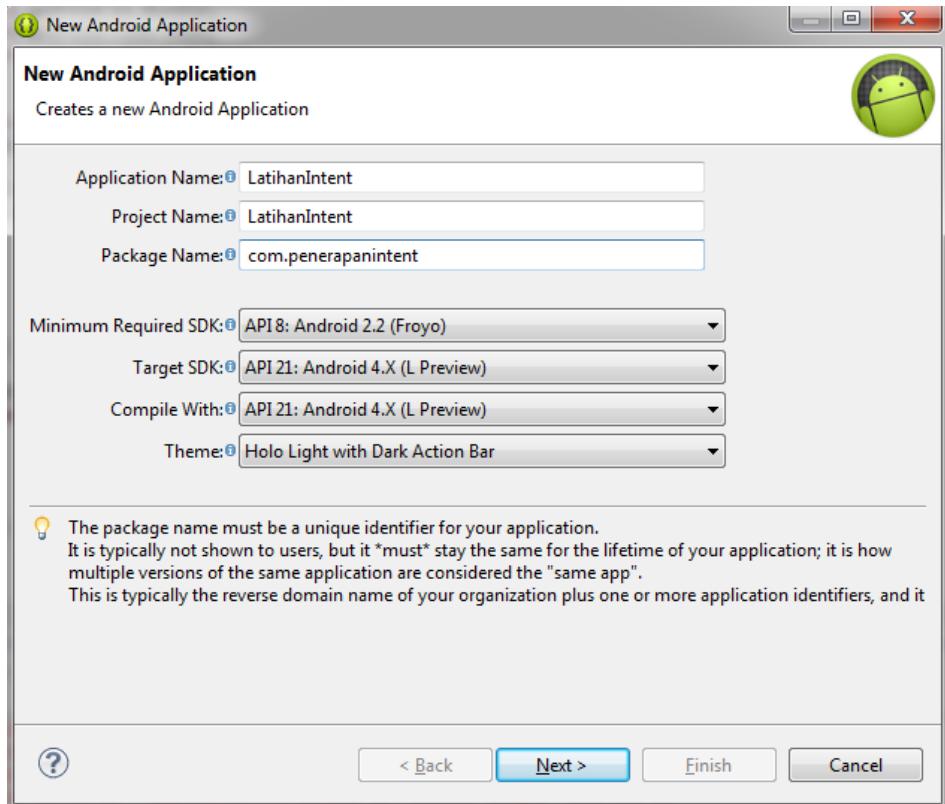
- Explicit Intent
- Implicit Intent

Explicit intent dipergunakan untuk memulai activity atau service yang terdapat pada aplikasi yang sama. Contohnya memanggil Activity2 dari MainActivity

```
Intent ExplicitIntent=new Intent(MainActivity.this, Activity2.class);
startActivity(ExplicitIntent);
```

Contoh Implementasi

Buatlah project dengan nama “LatihanIntent”.



Berikan nama activity “MainActivity” dan layout “activity_main”.

Pada layout tambahkan Button untuk berpindah ke activity selanjutnya. Id Button : “button1”. Berikut kode layout.

```
<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.penerapanintent.MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="39dp" />
```

```

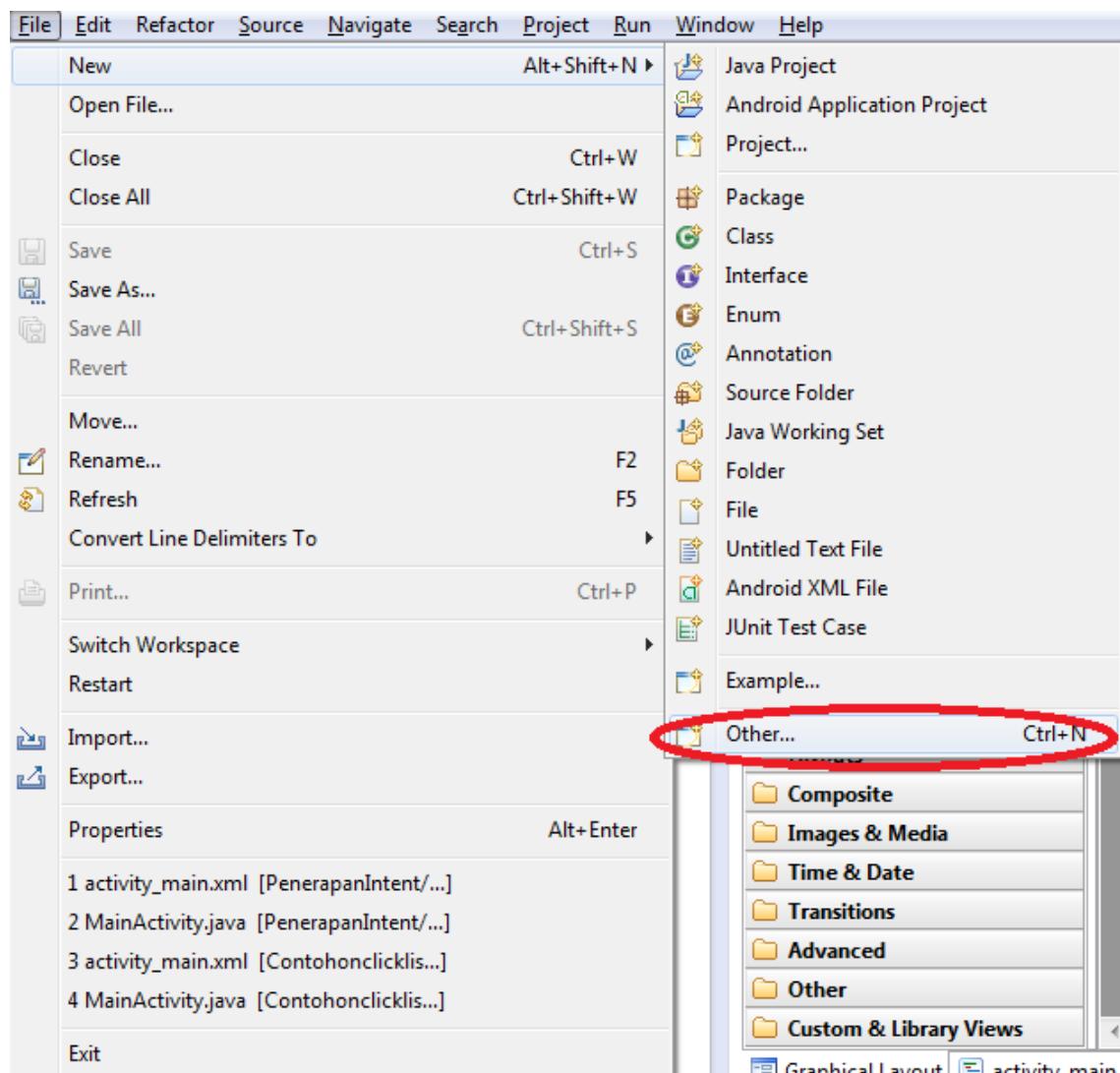
    android:text="Activity 1" />

<Button
    android:id="@+id/button1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="65dp"
    android:text="pindah ke 2" />

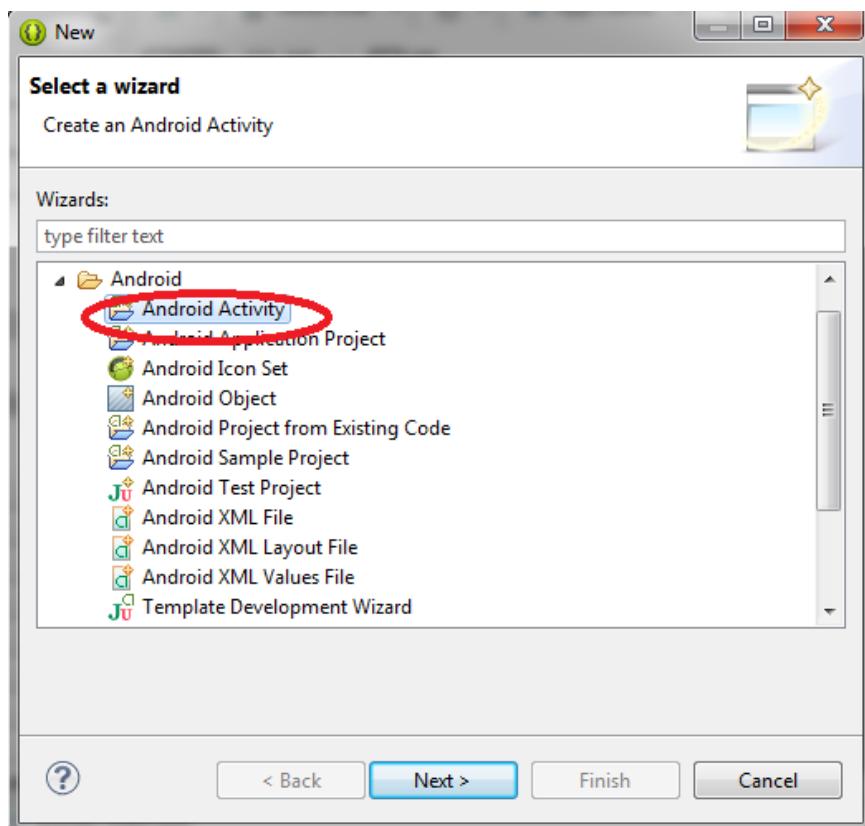
</RelativeLayout>
  
```

Selanjutnya buat activity kedua dengan nama layout “Activity2” dan nama activity “activity_activity2”. Cara membuat activity baru sebagai berikut.

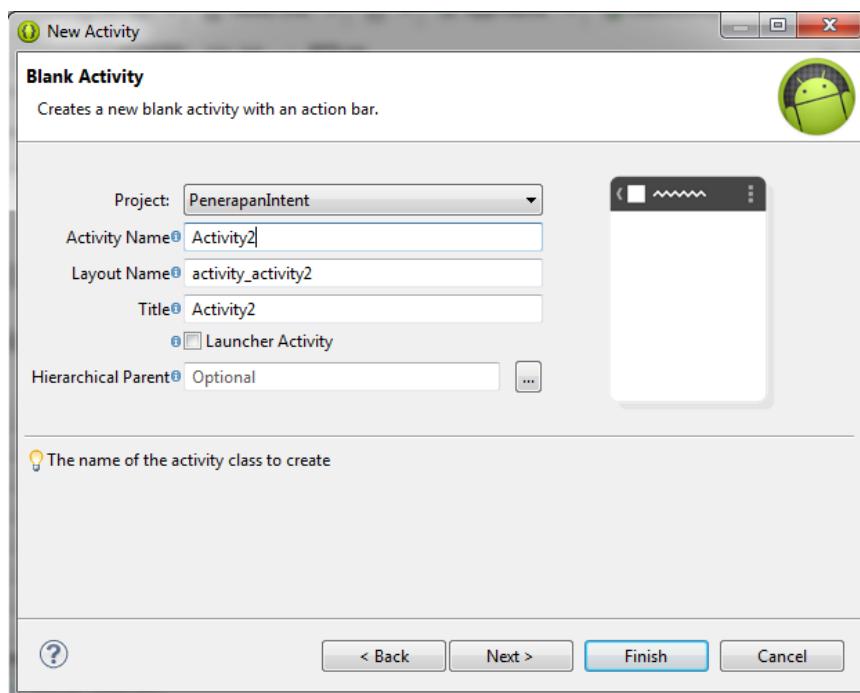
Klik File → New → Other



Pilih menu Android kemudian pilih Android Activity.



Pilih Blank Activity dan berikan nama untuk Activity kedua sebagai berikut. Setelah selai klik finish



Pada “MainActivity” tambahkan kode sebagai berikut:

```
package com.penerapanintent;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity implements View.OnClickListener {
    Button button;

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

        button=(Button)findViewById(R.id.button1);
        button.setOnClickListener(this);
    }

    @Override
    public void onClick(View arg0) {
        //kode pindah activity
        Intent ExplicitIntent=new Intent(MainActivity.this,
Activity2.class);
        startActivity(ExplicitIntent);
    }
}
```

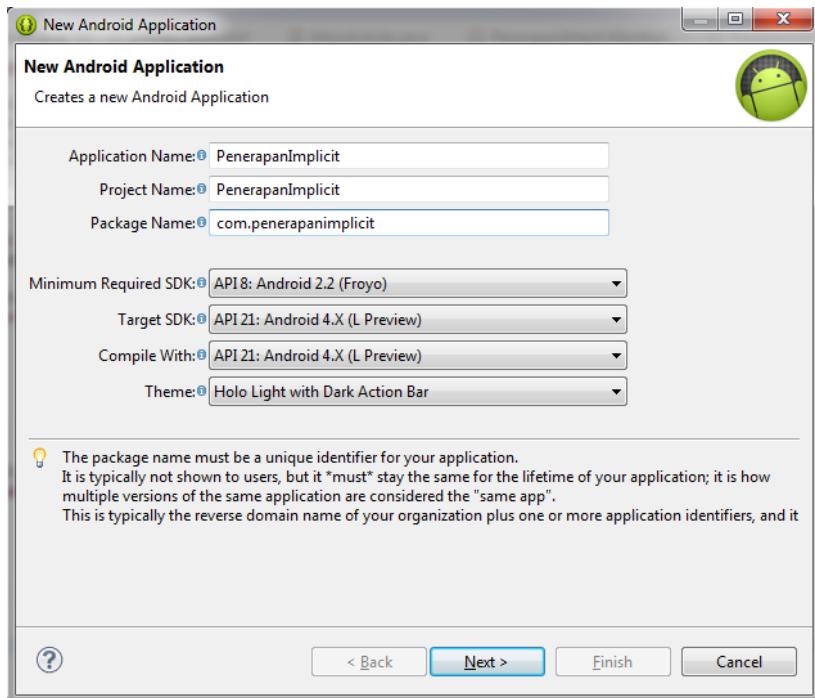
Setelah itu running aplikasi. Jika tombol pada MainActivity di klik maka akan pindah ke Activity2.



Implicit intent digunakan untuk mengaktifkan komponen dari aplikasi lain. Implicit intent tidak menspesifikasi komponen berdasarkan nama. Android sistem harus mencari komponen yang dapat menangani Implicit intent.

Contoh Implementasi

Buatlah project dengan nama “PenerapanImplicit”. Dengan nama Activity “MainActivity” dan Layout “activity_main”.



Pada Layout tambahkan tombol dengan id : button1 yang akan dipergunakan untuk memanggil implicit intent. Dengan kode sebagai berikut.

```
<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.penerapanimplicit.MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="28dp"
        android:text="Implicit Intent" />

    <Button
        android:id="@+id/button1"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="72dp"
    android:text="Telepon" />

</RelativeLayout>
```

Pada MainActivity tambahkan kode berikut

```
package com.penerapanImplicit;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity implements View.OnClickListener {

    Button button;

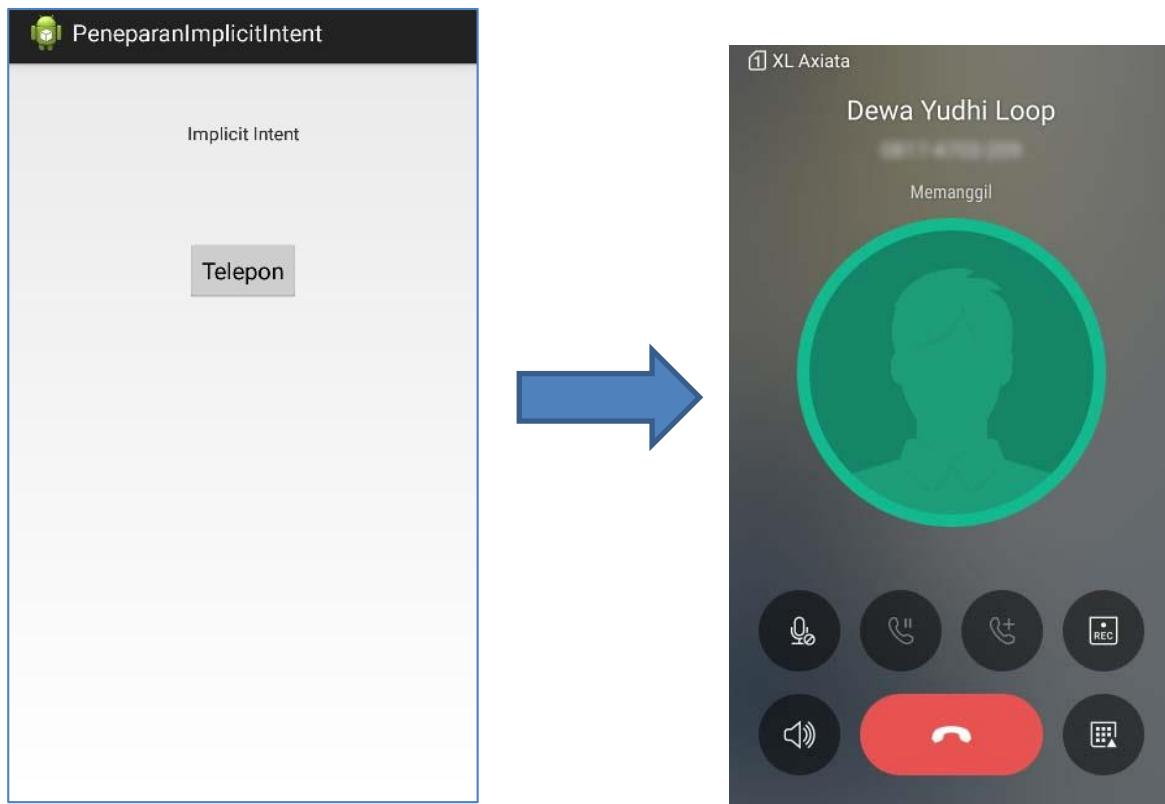
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button=(Button)findViewById(R.id.button1);
        button.setOnClickListener(this);

    }

    @Override
    public void onClick(View arg0) {
        Intent ImplicitIntent=new Intent(Intent.ACTION_CALL,
Uri.parse("tel:0817470320"));
        startActivity(ImplicitIntent);

    }
}
```

Tambahkan `<uses-permission android:name="android.permission.CALL_PHONE" />` pada AndroidManifest untuk perijinan mengakses telepon. Setelah dijalankan akan muncul tampilan sebagai berikut. Jika tombol telepon ditekan maka akan melakukan panggilan.



Latihan

Membuat 3 buah activity dan berpindah dari satu activity dengan activity yang lain.

PERTEMUAN 5

Menambahkan Fungsionalitas Pada Antar Muka

Tujuan

- Mahasiswa memahami peralatan untuk Debugging Aplikasi Android.
- Mahasiswa mampu menggunakan ADT Plug-in untuk mendebug aplikasi.
- Mahasiswa mampu menggunakan Android Debug Bridge

MEMPERKENALKAN LINGKUNGAN DEBUGGING ANDROID

Pengembangan aplikasi mobile memiliki banyak kesamaan dengan pemrograman untuk desktop dan Web. Pengembang kadang-kadang, menganggap bahwa logika pemograman/ teknik yang paling sesuai untuk lingkungan desktop dan web umumnya bekerja di lingkungan mobile juga. Penting bagi pengembang aplikasi seluler untuk memprogram kode dalam pendekatan modular dan berhati-hati terhadap sumber daya yang terlalu over-pereganggan, seperti memori dan penyimpanan yang terus-menerus. Pengembangan aplikasi seluler dan platform pengembangan tradisional menyimpang di bidang pengujian aplikasi dan debugging yang sungguh-sungguh.

Debugging adalah proses mendeteksi kesalahan atau bug pada sebuah aplikasi. Untuk membantu anda membuat aplikasi bebas kesalahan, lingkungan pengembangan android memberi anda berbagai alat debugging yang mudah digunakan. Aspek penting yang berkaitan dengan debugging adalah logging. Terkadang, pengembang perlu membuat dan memeriksa log untuk mengidentifikasi bug di aplikasi.

Debugging Aplikasi Android adalah kegiatan yang menarik dan melibatkan beberapa alat perintah garis/ GUI yang tersedia melalui android SDK. Pertimbangkan contoh dimana pengembang perlu membuat aplikasi pembaca PDF. Saat menjalankan aplikasi, pengembang menemukan kesalahan yang menyatakan bahwa aplikasi tersebut tidak menguraikan teks dalam format yang benar. Dalam aplikasi yang begitu besar, akan sulit untuk mencari setiap baris kode untuk mendeteksi kesalahannya. Dalam skenario seperti itu, alat debugging android beserta log dapat digunakan untuk mendeteksi kode yang keliru di aplikasi dengan cepat dan murah.

Mengidentifikasi alat debugging Android

Untuk melakukan debugging pada aplikasi android, dapat menggunakan JDWP-compilant debugger. Debugger ini sudah ada dalam IDE Eclipse. Berikut beberapa komponen utama dari debugging Android:

- Andrid Debug Bridge (ADB)
ADB adalah alat command-line yang dapat berinteraksi dengan emulator atau perangkat Android yang terhubung.
- Dalvik Debug Monitor Server (DDMS)
DDMS adalah GUI dari ADB yang dapat berinteraksi dengan emulator atau perangkat Android yang terhubung
- Device/AVD
AVD adalah alat debugging dan testing Aplikasi android dengan men-simulasikan bagaimana aplikasi Android bekerja dengan pengaturan hardware yang berbeda
- JDWP Debugger
JDWP adalah protokol yang didukung oleh DVM, yang memberikan hak akses untuk melampirkan (Java Virtual Machine).
- Hierarchy Viewer
Hierarchy Viewer digunakan untuk men-debug tampilan UI dari aplikasi Android
- Traceview
Traceview digunakan untuk men-debug performa dari aplikasi Android

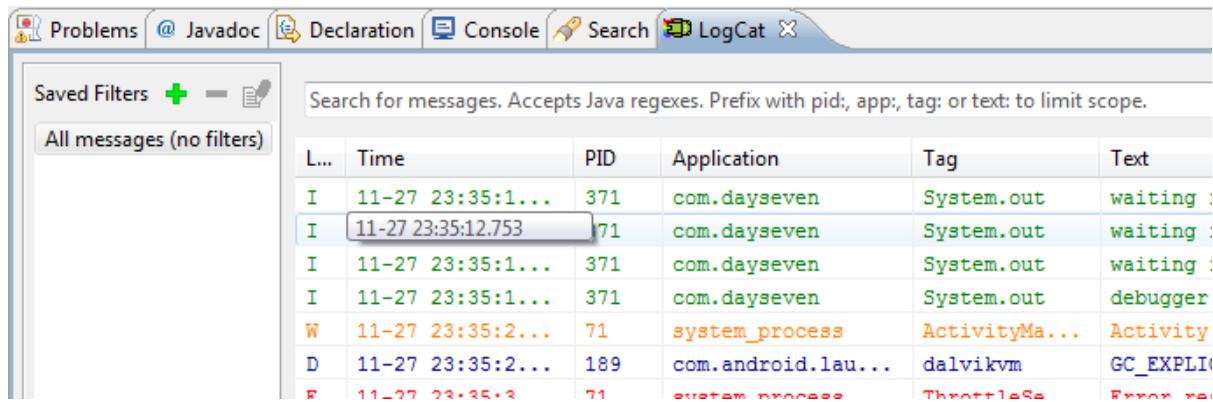
Membangkitkan Dan Memeriksa Pesan Log

Logging adalah teknik debugging dan diagnostik yang berguna yang digunakan oleh pengembang pada platform pengembangan aplikasi seluler. Jika pengembangan aplikasi Android, Anda dapat menggunakan kelas Log yang disediakan sebagai bagian dari SDK Android untuk mencatat informasi penting tentang aplikasi Anda ke utilitas LogCat. Utilitas LogCat memungkinkan Anda memantau keluaran log aplikasi Anda baik pada emulator atau perangkat yang dilampirkan ke Eclipse melalui debugger. Seorang pengembang dapat memeriksa log ini untuk melakukan debug aplikasi.

Anda dapat menjalankan LogCat melalui ADB atau dari DDMS, yang memungkinkan Anda membaca pesan secara real time. Utilitas LogCat mengumpulkan log dari sistem dan berbagai aplikasi. Log ini meliputi:

- Jejak-jejak saat emulator melempar kesalahan.
- Pesan yang Anda cetak dari aplikasi Anda dengan menggunakan metode kelas Log android.util packages.
- Pesan yang dihasilkan oleh sistem Android, seperti pesan yang dihasilkan oleh Pengelola Aktivitas yang mencatat detail tentang aktivitas yang sedang berjalan.

Gambar berikut ini menunjukkan jendela LogCat dengan bermacam-macam pesan log saat menjalankan sebuah aplikasi.



The screenshot shows the Android Studio LogCat tab. The interface includes tabs for Problems, Javadoc, Declaration, Console, Search, and LogCat. The LogCat tab displays a table of log entries with columns: L..., Time, PID, Application, Tag, and Text. The table contains several entries, with the second entry highlighted in blue. The application 'com.dayseven' is listed multiple times under different tags like System.out, debugger, and ActivityManager.

L...	Time	PID	Application	Tag	Text
I	11-27 23:35:1...	371	com.dayseven	System.out	waiting :
I	11-27 23:35:12.753	371	com.dayseven	System.out	waiting :
I	11-27 23:35:1...	371	com.dayseven	System.out	waiting :
I	11-27 23:35:1...	371	com.dayseven	System.out	debugger
W	11-27 23:35:2...	71	system_process	ActivityMa...	Activity
D	11-27 23:35:2...	189	com.android.lau...	dalvikvm	GC_EXPLI
F	11-27 23:35:2...	71	system_process	ThrottleRa...	Error: no

Pesan logging dikategorikan berdasarkan tingkat keparahan (dan verbositas) Dengan kesalahan yang paling parah diikuti oleh peringatan, pesan informasi, pesan debug, dan pesan verbose yang paling parah. Setiap jenis pesan memiliki metode tersendiri, Tingkat keparahan yang bervariasi dalam urutan keparahan yang menurun adalah:

- Kesalahan
- Peringatan
- Pesan informasi
- Pesan debug
- Pesan verbose

Kelas Log menyediakan berbagai metode untuk mencetak pesan atau tingkat keparahan masing-masing. Ini mencakup:

- Log.e (String tag, String msg): Metode ini digunakan untuk mencatat error. Tempat terbaik untuk menempatkan metode Log.e () adalah blok try-catch untuk menampilkan rincian pengecualian, dilemparkan oleh sebuah aplikasi.
- Log.w (): Metode ini digunakan untuk mencatat peringatan. Peringatan, Apakah situasi di mana operasi tertentu dilakukan oleh aplikasi gagal namun kegagalannya tidak penting untuk aplikasi.
- Log.i (): Metode ini digunakan untuk mencatat pesan informasi, seperti informasi yang berkaitan dengan keadaan suatu kegiatan atau layanan.
- Log.d (): Metode ini digunakan untuk log metode debug messages yang harus digunakan pada tahap pengembangan saja.
- Log.v (): Metode ini digunakan untuk mencatat pesan verbose. Informasi tambahan lainnya dicatat melalui metode ini.

pesan verbose tidak boleh dikompilasi ke dalam aplikasi kecuali selama pengembangan. Debug log diajukan tapi dilucuti saat runtime Kesalahan, peringatan, dan log info dapat menjadi bagian dari aplikasi siap produksi.

Selain keparahan, setiap pesan log memiliki tag yang terkait dengannya. tag ini mengacu pada string pendek yang menandakan komponen sistem dari mana pesan itu berasal. Dengan kata lain, dapat mengidentifikasi kelas atau aktivitas atau sebagian kode tempat panggilan log terjadi.

pertimbangkan cuplikan kode berikut dalam aktivitas bernama MyActivity:

```
Log.i ("MyActivity", "x =" + x);  
// dimana x adalah variabel yang nilainya ingin anda monitor  
// pesan log saat aplikasi berjalan
```

dalam cuplikan kode sebelumnya, parameter pertama untuk Log.i () method adalah tag dan parameter kedua adalah pesan yang akan ditulis ke log.

DEBUGGING APLIKASI DARI ECLIPSE DENGAN ADT PLUG-IN

Sementara mengembangkan aplikasi di Eclipse dengan ADT Plug-in, anda dapat menggunakan Java debugger built-in dan alat SDK's DDMS untuk mendebug aplikasi mobile anda. Untuk mengakses debugger dan DDMS, Eclipse menampilkan fitur debugger dan DDMS sebagai perspektif. Perspektif ini dapat disesuaikan dengan tampilan Eclipse, yang menampilkan tab, jendela dan alat yang dipilih untuk memfasilitasi pengembangan aplikasi Android.

Dalam persepektif debug, anda mengikuti pernyataan aplikasi menurut pernyataan untuk memperbaiki kesalahan semantik atau sintaksis. Di sisi lain dengan DDMS, anda dapat mengikuti keseluruhan perilaku aplikasi, memori yang dikonsumsi oleh aplikasi, file yang dibuka, dan seterusnya.

Persepektif Debug

Perspektif debug memberi Anda kemampuan debugging yang serupa dengan lingkungan pengembangan lainnya termasuk kemampuan untuk masuk ke atau melewati pemanggilan metode dan mengintip ke dalam variabel untuk memeriksa nilainya. Ini juga memungkinkan Anda mengatur breakpoint saat kode debugging.

Perspektif Debug memungkinkan Anda men-debug aplikasi yang sedang berjalan dengan menjedanya dan kemudian mengamati kode dan mengakses variabel. Anda dapat

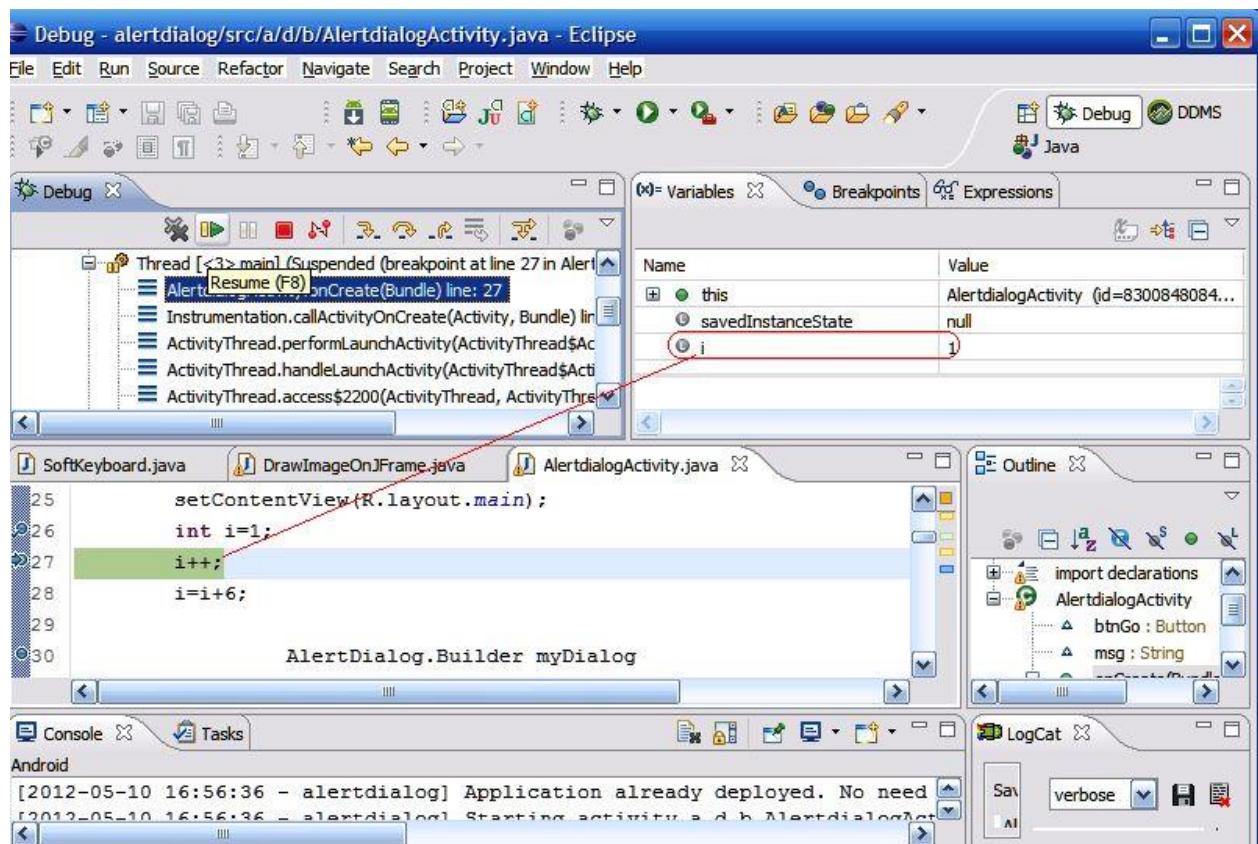
menjeda thread aktif dan mengakses debugger untuk mengatasi kesalahan. Setelah memperbaiki semua kesalahan, Anda dapat melanjutkan aplikasi yang ditangguhkan.

Perspektif debug mencakup tab berikut:

debug: tab ini menampilkan seluruh aplikasi android yang sebelumnya di-debugged di sepanjang aplikasi, yang saat ini sedang debugged. Selain itu, ini menampilkan untaian yang saat ini berjalan di aplikasi.

- Variabel: Saat breakpoint ditetapkan, tab ini menampilkan nilai variabel selama perpanjangan kode.
- Breakpoints: Tab ini menampilkan daftar breakpoint yang ada di kode aplikasi Anda.
- Logcat: Tab ini memungkinkan Anda melihat pesan log sistem secara real time.

Gambar berikut menggambarkan berbagai tab di debug persepektif

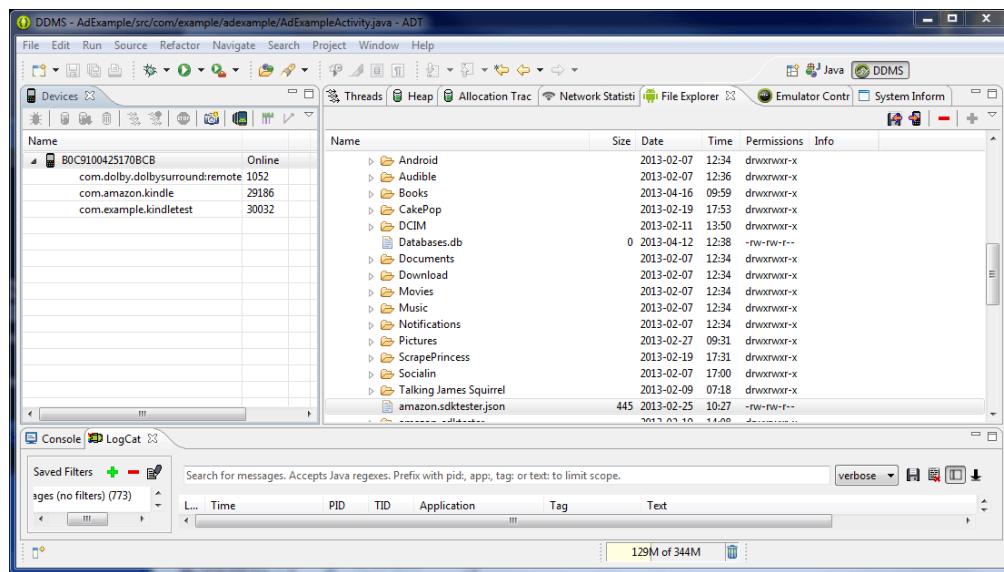


Perspektif DDMS

Perspektif DDMS menyediakan alat untuk mengendalikan emulator / perangkat. Ini juga menyediakan alat untuk memilih proses individual untuk debugging. Berbeda dengan perspektif Debug, ini memungkinkan Anda memberikan nilai untuk pengaturan lokasi, mensimulasikan telepon dan pesan, menjelajahi emulator atau sistem file perangkat, dan banyak lagi. Perspektif DDMS mencakup tab berikut:

- Perangkat: Tab ini menampilkan daftar perangkat dan AVD yang terhubung ke ADB. Ini juga menunjukkan semua proses yang berjalan pada masing-masing perangkat.
- Kontrol Emulator: Tab ini memungkinkan Anda untuk menjalankan fungsi perangkat seperti untuk mensimulasikan pesan dan telepon.
- LogCat: Tab ini memungkinkan Anda melihat pesan log sistem secara real time. Untuk setiap entri dalam tab ini, informasi berikut akan ditampilkan:
 - Tag terkait dengan pesan
 - Waktu ketika pesan log dibuat
 - Tingkat keparahan log
 - ID dari proses yang menghasilkan pesan log
 - Pesan itu sendiri
- Threads: Tab ini menampilkan thread yang sedang berjalan dalam VM. Ini menampilkan untaian utama dan benang anak dari sebuah aplikasi.
- Heap: Tab ini menunjukkan penggunaan tumpukan untuk VM. Objek yang dibuat saat runtime disimpan dalam memori tumpukan dan bisa jadi sampah dikumpulkan jika tidak digunakan oleh aplikasi selama status aplikasi berjalan. Anda bisa klik tombol Cause GC untuk meminta pengumpul sampah.
- Alokasi Tracker: Tab ini membantu Anda bekerja dengan daftar objek yang dialokasikan dari aplikasi yang Anda jalankan dalam sebuah tabel. Dengan memilih pada sebuah garis Anda dapat melihat jejak stack yang mengarah ke alokasi dalam tabel kedua. Dengan informasi ini, sebagai pengembang, Anda akan tahu jenis objek yang dialokasikan, identifikasi benang, kelas, file, dan posisi atau baris dalam kode.
- File Explorer: Tab ini memungkinkan Anda menjelajahi sistem file perangkat. Ini memungkinkan Anda mentransfer file ke dan dari perangkat.

Gambar berikut menggambarkan berbagai tab dalam perspektif DDMS.



Menggunakan Android Debug Bridge

ADB bertindak sebagai media komunikasi antara mesin pengembangan dan perangkat android/ emulator. Ini memungkinkan anda mengelola aplikasi Android (.apk packages), mengumpulkan log melalui LogCat, membunuh proses, mentransfer file ke dan dari perangkat.

Plug-in ADT untuk Eclipse mengoptimalkan dan menyederhankan interaksi dengan ADB. Dengan kata lain, ini membantu anda dengan mudah melakukan berbagai tugas yang anda lakukan melalui ADB, seperti menginstal dan memperbarui aplikasi, memelihara file log, dan mendorong dan menarik file dari IDE itu sendiri.

ADB adalah program client-server yang terdiri dari 3 komponen utama, sebagai berikut :

- Klien : Ia menjalankan mesin pengembangan dan juga dapat dipanggil dari antarmuka command-line sistem oprasi dengan mengeluarkan perintah ADB. Plug-in ADT dan DDMS juga membuat klien ADB.
- Daemon : Dijalankan sebagai proses latar belakang pada perangkat/ emulator.
- Server : Dijalankan sebagai proses latar belakang pada mesin pengembangan. Ini mengelola komunikasi antara klien dan daemon ADB.

Saat klien ADB dimulai, pertama kali memeriksa apakah proses ADB server sedang berjalan. Jika proses server tidak berjalan, sudah dimulai. Saat dimulai, server berikan dengan port TCP lokal, 5037 dan mulai mendengarkan perintah dari klien ADB.

Server kemudian terhubung dengan semua perangkat/ emulator yang berjalan. Emulator dan perangkat menggunakan port diantara 5555 sampai 5585. Masing-masing contoh perangkat/ emulator menggunakan sepasang port berurutan, port bernomor genap untuk koneksi konsol dan port bernomor ganjil dari koneksi ADB. Oleh karena itu untuk menghubungkan ke perangkat/ emulator, server ADB akan memindai port bernomor ganjil diantara 5555 sampai 5585. Begitu server terhubung ke contoh perangkat/ emulator, anda dapat menggunakan perintah ADB untuk mengendalikan kejadian tersebut.

Perintah ADB

Alat ADB memungkinkan pengguna berinteraksi dengan emulator Android langsung dari baris perintah. Ketika sebuah perintah dikeluarkan, klien ADB dipanggil oleh program ini. Sistem berikut menunjukkan penggunaan perintah adb:

```
Adb [-d|-e|-s <serial number>] <command>
```

Dalam sintaks sebelumnya, opsi -d mengarahkan perintah ke emulator yang sedang berjalan. Jika lebih dari satu emulator sedang berjalan, opsi ini mengembalikan sebuah kesalahan. The -a <serial number> option mengarahkan perintah ke perangkat USB atau emulator dengan nomor serial

Juga penting untuk mengetahui kapan contoh emulator / perangkat terhubung ke server sebelum menjalankan perintah ADB. Dengan demikian, untuk menghasilkan daftar emulator / perangkat terpasang, perintah perangkat adb digunakan. Perintah ini mencetak informasi status berikut untuk setiap contoh:

- Nomor seri: ini adalah string yang membantu mengidentifikasi secara unik instance emulator / perangkat dengan nomor port konsolnya.
- State: mengacu pada keadaan instance emulator saat ini. Beberapa negara yang mungkin adalah:
 - offline: Contoh emulator tidak terhubung ke server adb atau tidak merespons.
 - Perangkat: Contoh emulator terhubung ke server adb. Namun, negara ini tidak menginformasikan bahwa perangkat tersebut telah menyelesaikan proses booting awal karena adb terhubung dengan emulator / perangkat saat sistem android berada dalam status boot.

Tips Debugging

Saat melakukan debugging. Anda dapat menggunakan beberapa kiat berikut untuk mengetahui masalah umum di dalam aplikasi Anda

- **Membuang jejak stack:** Anda bisa mendapatkan stack dump dari emulator. Untuk ini, Anda perlu login dengan shell ADB, gunakan perintah ps untuk menemukan proses yang dibutuhkan, kemudian gunakan perintah kill 3, Jejak stack akan muncul di file log, yang menyimpan data, seperti variabel lokal dan fungsi panggilan. Ini juga menunjukkan jumlah memori stack yang digunakan oleh proses tertentu. yang membantu dalam mengoptimalkan aplikasi kami.
- **Menampilkan informasi yang berguna pada layar emulator:** Anda dapat memperoleh informasi yang berguna, seperti penggunaan CPU dengan mengaktifkan fitur pengaturan pengembang. Tum fitur ini on dan off sesuai kebutuhan informasi anda.
- **Dapatkan informasi status aplikasi dan sistem dari emulator:** Anda dapat memperoleh informasi status aplikasi dan sistem dari emulator dengan menggunakan perintah shell ADB.
- **Dapatkan informasi konektivitas nirkabel:** Anda dapat memperoleh informasi tentang konektivitas nirkabel dengan menggunakan **DDMS**. Anda bahkan dapat mencatat pemanggilan metode dan data penelusuran lainnya dalam suatu aktivitas dengan memanggil metode balap awal.
- **Tangkap screenshot:** Anda dapat menggunakan DDM untuk menangkap tangkapan layar dari emulator. Untuk ini, Anda perlu mengklik tombol Screen Capture pada tab Devices pada perspektif DDMs.
- **Pengumpulan sampah:** Debugger dan pengumpul sampah terintegrasi secara longgar. VM memastikan bahwa objek apapun tidak dapat dikumpulkan sampai debugger diputuskan. Hal ini dapat mengakibatkan penumpukan objek selama periode waktu sementara debugger terhubung. Misalnya, jika menemukan benang berjalan, objek Thread yang terkait bukan sampah yang dikumpulkan bahkan setelah benang berakhir.

MODUL 6

MENANGANI DATA PADA APLIKASI ANDROID BAGIAN I

Tujuan

- Mahasiswa mampu mengidentifikasi mekanisme dari Data Storage.
- Mahasiswa mampu menggunakan SQLite sebagai Data Storage.
- Mahasiswa mampu menggunakan Content Provider untuk Data Access.

Mekanisme Data Storage

Aplikasi Android memerlukan untuk menyimpan data untuk berbagai keperluan, seperti menyimpan skor, menyimpan informasi, menyimpan kontak dan berbagai keperluan lainnya. Platform Android menyediakan berbagai cara untuk menyimpan data. Antara lain:

- **SQLite**

Digunakan ketika aplikasi memerlukan untuk menyimpan data dengan ukuran besar. SQLite adalah mesin basis data yang tertanam pada android. SQLite bersifat kode terbuka (open source). Tidak seperti kebanyakan basis data SQL, SQLite membaca dan menulis langsung ke media penyimpanan dan tidak memerlukan proses server yang berbeda.

- **Penyimpanan Internal (Internal Storage)**

Digunakan untuk menyimpan data yang sangat kecil dan tidak memerlukan perubahan yang terlalu sering. Platform Android mengijinkan untuk membuat dan membaca berkas dari penyimpanan internal. Biasanya, berkas yang disimpan pada penyimpanan internal bersifat private untuk suatu aplikasi dan tidak bisa diakses oleh aplikasi lainnya.

- **Penyimpanan Eksternal (External Storage)**

Android menyediakan kemudahan dalam menyimpan data pada media eksternal karena keterbatasan ukuran dari penyimpanan internal.

- **Shared Preferences**

Berguna untuk menyimpan secara cepat nilai default, variabel, kondisi dari Antar muka yang berlangsung dan preferensi pengguna. Shared Preferences adalah mekanisme ringan untuk menyimpan pasangan key-value dari tipe data primitif seperti integer, boolean, string, float dan long.

- **Koneksi Jaringan**

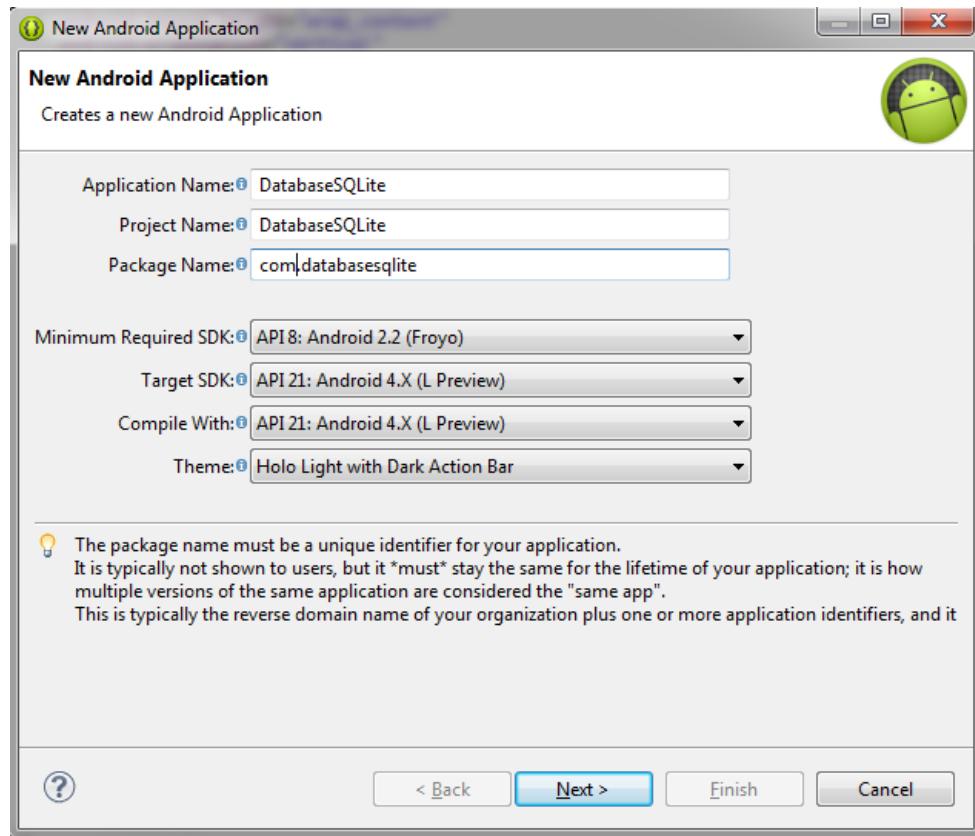
Android mengijinkan aplikasi untuk mengakses yang tersimpan di luar aplikasi seperti pada jaringan lokal ataupun internet. Komponen yang terpisah dari aplikasi yang berada di jaringan menangai permintaan aplikasi untuk melakukan tugas yang kecil.

menggunakan SQLite sebagai Data Storage

Pada platform Android, SQLite membantu untuk membangun basis data yang independen untuk membantuk aplikasi dalam menyimpan ataupun mengelola data. Database SQLite tersimpan di folder **data/data/<nama package>/database** di perangkat Android. Secara default, basis data bersifat private dan hanya bisa diakses oleh aplikasi yang bersangkutan.

Contoh Implementasi

Buatlah project bernama “Database SQLite” dengan nama Activity: “MainActivity” dan layout: “activity_main”



Pada layout *activity_main* tambahkan kode berikut:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    tools:context=".MainActivity" >

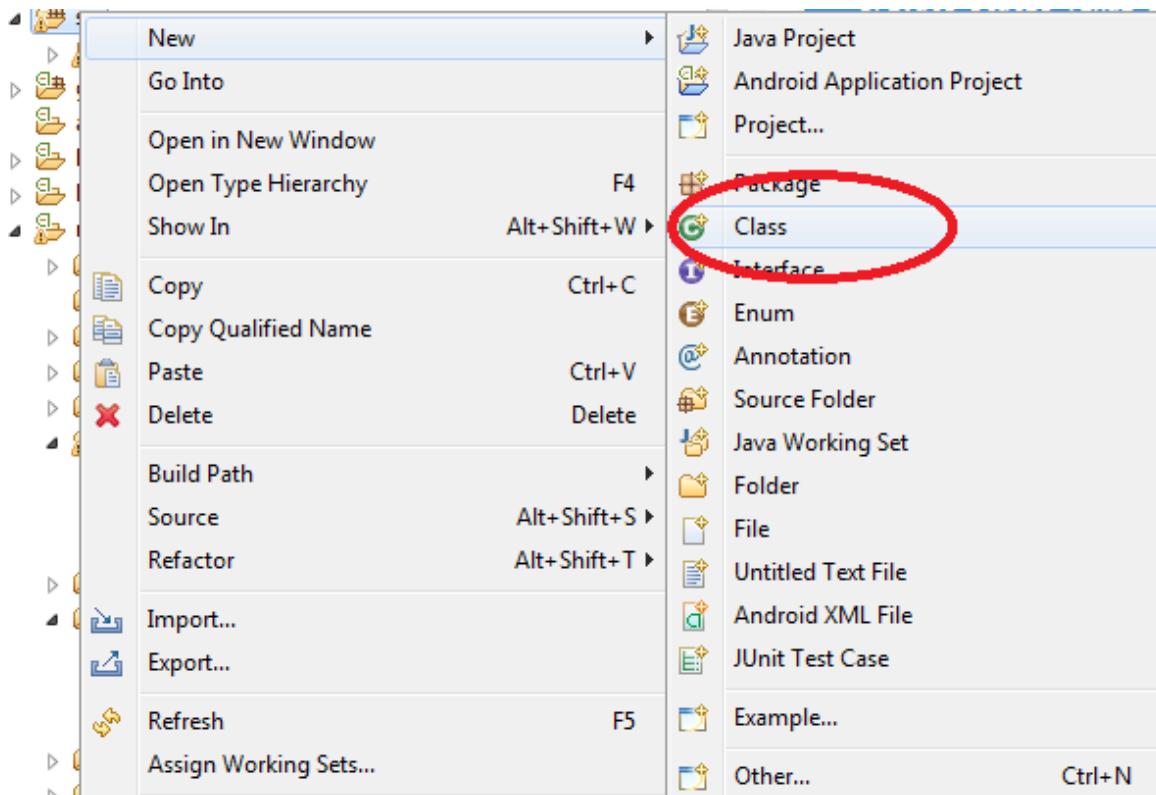
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Daftar Buku"
        />

    <ListView
        android:id="@+id/ListView01"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </ListView>

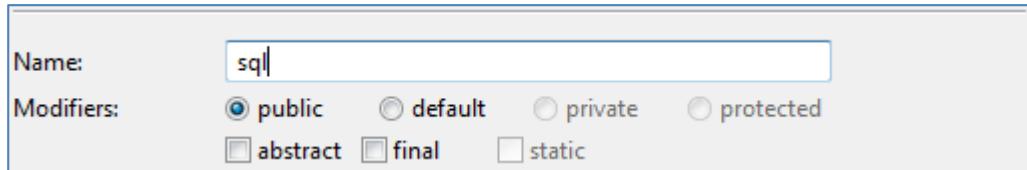
</LinearLayout>
```

Terdapat dua buah widget TextView untuk menampilkan tulisan Daftar Buku dan sebuah ListView yang akan digunakan untuk menampilkan data.

Selanjutnya buat sebuah class dengan nama “sql” yang berisi kode untuk membuat database dan membuat tabel dengan cara mengklik kanan folder src pada package explorer kemudian memilih New → Class.



Beri nama Class tersebut sql.



Berikut kode pada class sql.

```
package com.databasesqlite;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class sql extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "db.db";
    private static final int DATABASE_VERSION = 1;

    public sql(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        // TODO Auto-generated constructor stub
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String sql = "create table buku( id integer primary key
autoincrement, judul_buku text null, " + " nama_pengarang text null, tahun_terbit
text null, penerbit text null);";
        Log.d("Data", "onCreate: " + sql);
        db.execSQL(sql);
        sql = "INSERT INTO buku (id, judul_buku, nama_pengarang,
tahun_terbit, penerbit) VALUES (1, 'Penerapan Sistem', 'Jogiyanto', '2010',
'Andi');";
        db.execSQL(sql);
        sql = "INSERT INTO buku (id, judul_buku, nama_pengarang,
tahun_terbit, penerbit) VALUES (2, 'Pengantar TI', 'Sutarman', '2012', 'Bumi
Aksara');";
        db.execSQL(sql);
        // TODO Auto-generated method stub
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // TODO Auto-generated method stub
    }
}
```

Selanjutnya membuat Activity untuk menambah data pada database SQLite. Buat Activity baru dengan nama Activity “Add” dan nama layout “activity_add”.

Kode pada layout “activity_add”

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:context=".Add" >

    <TextView
        android:text="Judul_Buku"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/edAddJudulBuku"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </EditText>

    <TextView
        android:text="Nama_Pengarang"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/edAddNamaPengarang"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </EditText>

    <TextView
        android:text="Tahun_Terbit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/edAddTahunTerbit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </EditText>

    <TextView
        android:text="Penerbit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/edAddPenerbit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </EditText>
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    
```

```
</EditText>

<Button
    android:text="Simpan"
    android:id="@+id/btnAddSimpan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

```

```
</Button>

</LinearLayout>
```

Kode pada Activity Add

```
package com.databasesqlite;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.app.Activity;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

public class Add extends Activity {
    protected Cursor cursor;
    sql dbHelper;
    Button btnAddSimpan;
    EditText edAddJudulBuku;
    EditText edAddNamaPengarang;
    EditText edAddTahunTerbit;
    EditText edAddPenerbit;

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

        dbHelper = new sql(this);
        edAddJudulBuku = (EditText) findViewById(R.id.edAddJudulBuku);
        edAddNamaPengarang = (EditText) findViewById(R.id.edAddNamaPengarang);
        edAddTahunTerbit = (EditText) findViewById(R.id.edAddTahunTerbit);
        edAddPenerbit = (EditText) findViewById(R.id.edAddPenerbit);

        btnAddSimpan = (Button) findViewById(R.id.btnAddSimpan);
        btnAddSimpan.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick(View arg0) {
```

```
// TODO Auto-generated method stub
SQLiteDatabase db = dbHelper.getWritableDatabase();

db.execSQL("insert into buku(judul_buku, nama_pengarang, tahun_terbit,
penerbit) values('" + edAddJudulBuku.getText().toString() + "','" +
edAddNamaPengarang.getText().toString() + "','" +
edAddTahunTerbit.getText().toString() + "','" + edAddPenerbit.getText().toString()
+ "')");
Toast.makeText(getApplicationContext(), "Berhasil",
Toast.LENGTH_LONG).show();
MainActivity.ma.RefreshList();
finish();
}
});
}
```

Selanjutnya membuat Activity untuk mengubah data. Buatlah Activity dengan nama “Edit” dengan layout “activity_edit”.

Kode untuk layout “activity_edit”

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"

    tools:context=".Edit" >

    <TextView
        android:text="Judul_Buku"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/edEditJudulBuku"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </EditText>

    <TextView
        android:text="Nama_Pengarang"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/edEditNamaPengarang"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </EditText>

    <TextView>
```

```

    android:text="Tahun_Terbit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
  
```

```

<EditText android:text=""
          android:id="@+id/edEditTahunTerbit"
          android:layout_width="match_parent"
          android:layout_height="wrap_content">

```

```

<TextView
    android:text="Penerbit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

```

```

<EditText android:text=""
          android:id="@+id/edEditPenerbit"
          android:layout_width="match_parent"
          android:layout_height="wrap_content">

```

```

<Button
    android:text="Simpan"
    android:id="@+id/btnEditSimpan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

```

```

</LinearLayout>

```

Kode untuk Activity Edit

```

package com.databasesqlite;

import android.app.Activity;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Edit extends Activity {
    protected Cursor cursor;
    sql dbHelper;
    Button btnEditSimpan;
    EditText edEditJudulBuku;
    EditText edEditNamaPengarang;
    EditText edEditTahunTerbit;
    EditText edEditPenerbit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_edit);
dbHelper = new sql(this);
edEditJudulBuku = (EditText) findViewById(R.id.edEditJudulBuku);
edEditNamaPengarang = (EditText)
findViewById(R.id.edEditNamaPengarang);
edEditTahunTerbit = (EditText) findViewById(R.id.edEditTahunTerbit);
edEditPenerbit = (EditText) findViewById(R.id.edEditPenerbit);

SQLiteDatabase db = dbHelper.getReadableDatabase();

cursor = db.rawQuery("SELECT * FROM buku WHERE judul_buku = '" +
getIntent().getStringExtra("judul_buku") + "'",null);

cursor.moveToFirst();
if (cursor.getCount()>0)
{
cursor.moveToPosition(0);
edEditJudulBuku.setText(cursor.getString(1).toString());
edEditNamaPengarang.setText(cursor.getString(2).toString());
edEditTahunTerbit.setText(cursor.getString(3).toString());
edEditPenerbit.setText(cursor.getString(4).toString());
}

btnEditSimpan = (Button) findViewById(R.id.btnEditSimpan);

btnEditSimpan.setOnClickListener(new Button.OnClickListener(){
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
SQLiteDatabase db = dbHelper.getWritableDatabase();

db.execSQL("update buku SET judul_buku='"+ +
edEditJudulBuku.getText().toString()+"', " +"nama_pengarang='"+ +
edEditNamaPengarang.getText().toString() +'", " +"tahun_terbit='"+ +
edEditTahunTerbit.getText().toString() +'", " +"penerbit='"+ +
edEditPenerbit.getText().toString() + "' WHERE " + " judul_buku = '" +
getIntent().getStringExtra("judul_buku") + "'");

Toast.makeText(getApplicationContext(), "Berhasil",
Toast.LENGTH_LONG).show();

MainActivity.ma.RefreshList();
finish();
}
});
}
```

Setelah Activity baru dibuat, terakhir tambahkan koding untuk Activity MainActivity sebagai berikut:

```
package com.databasesqlite;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.AdapterView.OnItemClickListener;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {
    String[] daftar;
    ListView ListView01;
    Menu menu;
    protected Cursor cursor;
    sql dbHelper;
    public static MainActivity ma;

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

        ma = this;
        dbHelper = new sql(this);
        RefreshList();
    }

    public void RefreshList() {
        SQLiteDatabase db = dbHelper.getReadableDatabase();
        cursor = db.rawQuery("SELECT * FROM buku",null);
        daftar = new String[cursor.getCount()];
        cursor.moveToFirst();
        for (int cc=0; cc < cursor.getCount(); cc++)
        {
            cursor.moveToPosition(cc);
            daftar[cc] = cursor.getString(1).toString();
        }

        ListView01 = (ListView)findViewById(R.id.ListView01);
        ListView01.setAdapter(new ArrayAdapter<Object>(this,
        android.R.layout.simple_list_item_1, daftar));
        ListView01.setSelected(true);
        ListView01.setOnItemClickListener(new OnItemClickListener() {
```

```
@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
long arg3) {
final String selection = daftar[arg2];
final CharSequence[] dialogitem = {"Edit", "Delete"};

AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this);
builder.setTitle("Pilih ?");
builder.setItems(dialogitem, new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int item) {
switch(item){
case 0 :

Intent i = new Intent(getApplicationContext(), Edit.class);
i.putExtra("judul_buku", selection);
startActivity(i);
break;
case 1 :
SQLiteDatabase db = dbHelper.getWritableDatabase();
db.execSQL("delete from buku where judul_buku =
""+selection+"");
RefreshList();
break;
}
}
});
builder.create().show();
}});

((ArrayAdapter)ListView01.getAdapter()).notifyDataSetInvalidated();

}

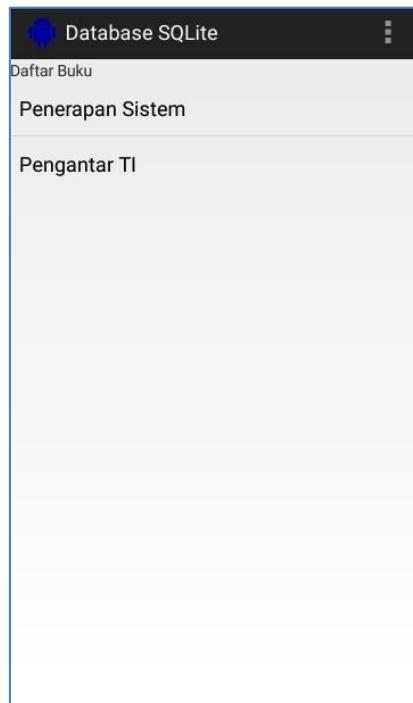
@Override
public boolean onCreateOptionsMenu(Menu menu) {
this.menu = menu;

menu.add(0, 1, 0, "Tambah").setIcon(android.R.drawable.btn_plus);
menu.add(0, 2, 0,
"Refresh").setIcon(android.R.drawable.ic_menu_rotate);
menu.add(0, 3, 0,
"Exit").setIcon(android.R.drawable.ic_menu_close_clear_cancel);
return true;
}

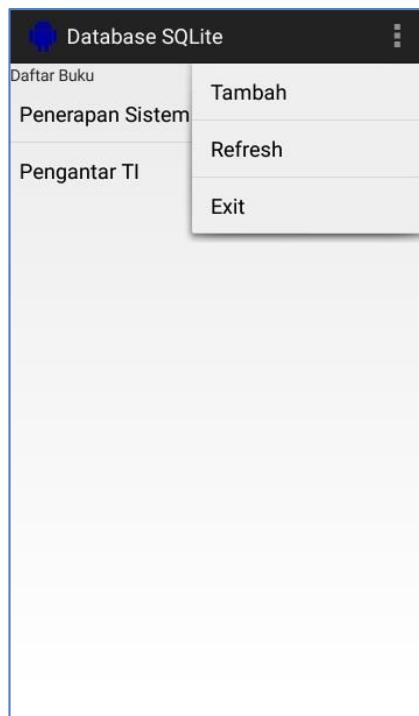
@Override
public boolean onOptionsItemSelected(MenuItem item) {
switch (item.getItemId()) {
case 1:
Intent i = new Intent(MainActivity.this, Add.class);
startActivity(i);
return true;
case 2:
RefreshList();
return true;
case 3:
finish();
return true;
}
}
```

```
    return false;  
}  
}
```

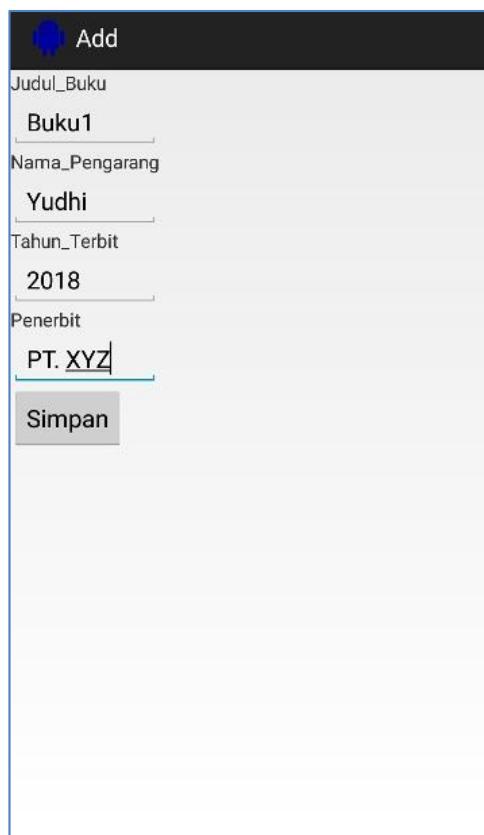
Ketika dirunning akan tampil sebagai berikut.



Tombol menu di pojok kanan atas digunakan untuk menampilkan menu tambah.



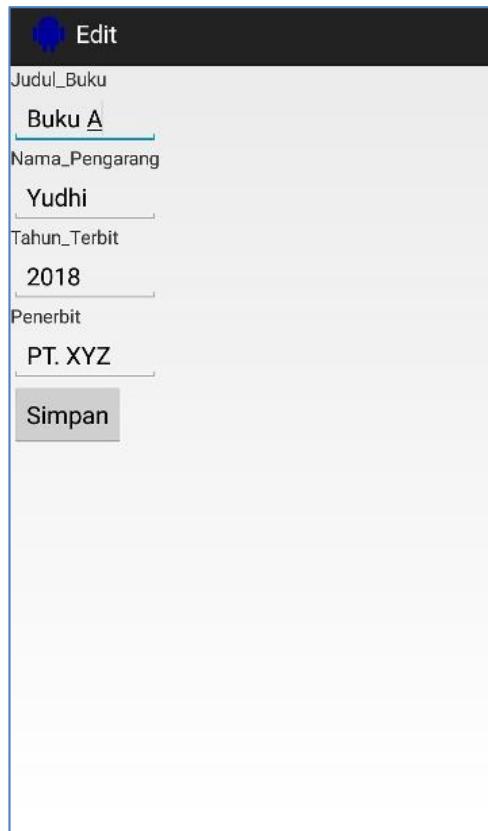
Tampilan Activity Add



Data yang sudah ditambah akan muncul pada halaman utama. Jika diklik akan muncul menu edit dan delete. Jika delete dipilih maka data akan dihapus dari basis data.



Ketika pilihan edit dipilih maka akan tampil Activity Edit



The screenshot shows a mobile application interface titled "Edit". It contains four input fields: "Judul_Buku" with the value "Buku A", "Nama_Pengarang" with the value "Yudhi", "Tahun_Terbit" with the value "2018", and "Penerbit" with the value "PT. XYZ". Below these fields is a grey button labeled "Simpan".

Menggunakan Content Provider untuk Data Access.

Pada aplikasi android kebanyakan data bersifat private dan diperuntukkan hanya untuk aplikasi bersangkutan. Hal ini berarti aplikasi tidak bisa melakukan permintaan data kepada aplikasi lainnya. Untuk bisa melakukan permintaan ataupun manipulasi terhadap data pada aplikasi lain digunakanlah konsep content provider.

Platform Android mendukung dua tipe dari content provider yang dapat digunakan dalam aplikasi.

- Custom Content Provider. Ini dibuat oleh developer untuk keperluan tertentu dari aplikasi.
- Native Content provider. Menyediakan akses pada basis data yang ada pada Android seperti kontak, media player, ataupun basis data native lainnya. Untuk dapat mengaksesnya memerlukan permission pada aplikasi.

Untuk melakukan permintaan pada content provider memerlukan sebuah string dalam bentuk URI. Berikut adalah format URI untuk melakukan permintaan pada content provider.

```
<standard_prefix>://<authority>/<data_path>/<id>
```

Berikut penjelasan bagian-bagian dari URI di atas tersebut:

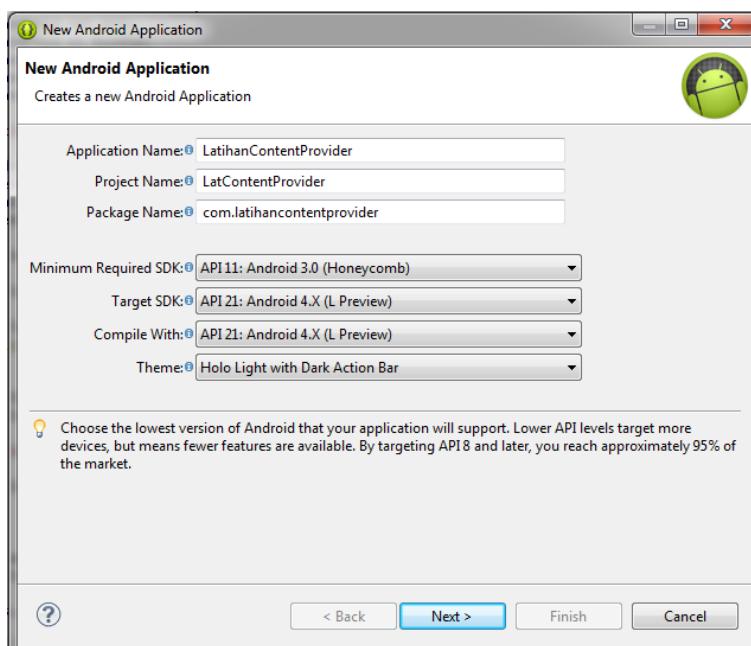
- **standard prefix** untuk 'content provider' selalu *content://*.
- **authority** menunjukkan nama dari 'content provider'. Contohnya contacts untuk 'content provider' Contacts yang sudah built-in. Untuk 'content provider' dari third-party, bisa merupakan nama yang valid dan lengkap, misalnya *com.example.provider*.
- **data path** menunjukkan semacam data yang diminta. Contohnya, bila kita mendapatkan semua daftar contact dari 'content provider' Contact, maka data path tersebut adalah people dan URI nya akan seperti ini: *content://contacts/people*.
- **id** menentukan data/baris tertentu yang diminta. Contohnya, bila kita mencari contact nomor 2 di 'content provider' Contacts, maka URI nya akan menjadi seperti ini: *content://contacts/people/2*

Beberapa contoh string untuk query:

- ***content://media/internal/images*** - mengembalikan/menghasilkan semua images internal di perangkat.
- ***content://media/external/images*** - mengembalikan/menghasilkan semua images di external storage (SD Card) di perangkat.
- ***content://call_log/calls*** - mengembalikan/menghasilkan semua panggilan yang ada di Call Log.
- ***content://browser/bookmarks*** - mengembalikan/menghasilkan semua bookmarks yang ada di browser.

Contoh Implementasi

Menggunakan native content provider untuk mengakses kontak di perangkat telefon. Buatlah sebuah project baru bernama "LatihanContentProvider" dengan nama Activity "MainActivity" dan layout "activity_main".



Pada layout “activity_main” tambahkan kode berikut.

```
<?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:id="@+id/activity_main"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    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:orientation="vertical"
    tools:context="com.example.provider.MainActivity">

    <ListView
        android:id="@+id/android:list"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:stackFromBottom="false"
        android:transcriptMode="normal" />

    <TextView
        android:id="@+id/contactID"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content" />

    <TextView
        android:id="@+id/contactName"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

Pada Activity MainActivity tambahkan kode sebagai berikut.

```
package com.latihancontenprovider;

import android.app.ListActivity;
import android.content.CursorLoader;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.widget.CursorAdapter;
import android.widget.SimpleCursorAdapter;

public class MainActivity extends ListActivity {

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

```
Uri allContacts = Uri.parse("content://contacts/people");

Cursor c;

CursorLoader cursorLoader = new
CursorLoader(this,allContacts,null,null,null,null);
c = cursorLoader.loadInBackground();

String[] columns = new String[] {
    ContactsContract.Contacts.DISPLAY_NAME,
    ContactsContract.Contacts._ID};

int[] views = new int[] {R.id.contactName, R.id.contactID};

SimpleCursorAdapter adapter;

adapter = new SimpleCursorAdapter(this, R.layout.activity_main, c,
columns, views,
        CursorAdapter.FLAG_REGISTER_CONTENT_OBSERVER);

this.setListAdapter(adapter);
}
}
```

Karena Native Content Provider jangan lupa menambahkan permission untuk membaca kontak pada AndroidManifest.

```
<uses-permission android:name="android.permission.READ_CONTACTS" />
```

Ketika di running akan muncul tampilan sebagai berikut.



Latihan

Menerapkan SQLite untuk menyimpan dan membaca data pada aplikasi Android.

MODUL 7

MENANGANI DATA PADA APLIKASI ANDROID BAGIAN II

Tujuan

- Mahasiswa mampu menggunakan Internal Data Storage.
- Mahasiswa mampu menggunakan External Data Storage.
- Mahasiswa mampu menggunakan Shared Preferences.

Menggunakan Internal Data Storage

Pada Platform Android, data dapat disimpan langsung pada media penyimpanan internal perangkat. Secara default, berkas disimpan pada media penyimpanan internal bersifat private yang hanya dapat diakses oleh aplikasi bersangkutan, aplikasi lain tidak dapat mengaksesnya. Ketika aplikasi dihapus oleh pengguna maka data aplikasi yang tersimpan pada media penyimpanan internal akan otomatis terhapus. Berkas disimpan pada folder `/data/data/<nama package>/files`. Berikut adalah kode perintah untuk membuat file pada media penyimpanan internal.

```

String setData = "Halo";
//Digunakan untuk membuat dan menulis File/Data pada Penyimpanan
FileOutputStream fileOutputStream;
try {
    //Membuat Berkas Baru dengan mode Private
    fileOutputStream = openFileOutput("DataSaya.txt", Context.MODE_PRIVATE);

    //Menulis Data Baru dan Mengkonversinya kedalam bentuk byte
    fileOutputStream.write(setData.getBytes());

    //Menutup FileOutputStream
    fileOutputStream.close();
    Toast.makeText(getApplicationContext(), "Data Disimpan di Internal",
    Toast.LENGTH_SHORT).show();
} catch (Exception ex){
    ex.printStackTrace();
}
  
```

Menggunakan External Data Storage

Platform Android menyediakan pilihan kepada pengguna untuk menyimpan datanya pada media penyimpanan luar/External Data Storage. Hal itu dikarenakan data yang akan disimpan mempunyai ukuran yang besar dan kapasitas dari media penyimpanan internal yang terbatas. Umumnya yang digunakan sebagai media penyimpanan eksternal pada Android adalah SD card. Berikut adalah kode perintah untuk menulis data pada media penyimpanan eksternal.

```
//Digunakan Untuk Mengecek Apakah Tersebut Penyimpanan External atau Tidak
String state = Environment.getExternalStorageState();

if (Environment.MEDIA_MOUNTED.equals(state)){
//Digunakan Untuk Mengecek Lokasi Penyimpanan Direktori di Memory External
File dirExternal = Environment.getExternalStorageDirectory();

//Membuat Direktori Baru dengan Nama "ContohDirektori"
File createDir = new File(dirExternal.getAbsolutePath()+"/ContohDirektori");

//Jika Direktori "ContohDirektori" Tidak Ada, Maka akan Membuatnya
if (!createDir.exists()){
    createDir.mkdir();
    //Membuat File Baru didalam Direktori "ContohDirektori"
    File file = new File(createDir, "ContohFile.txt");

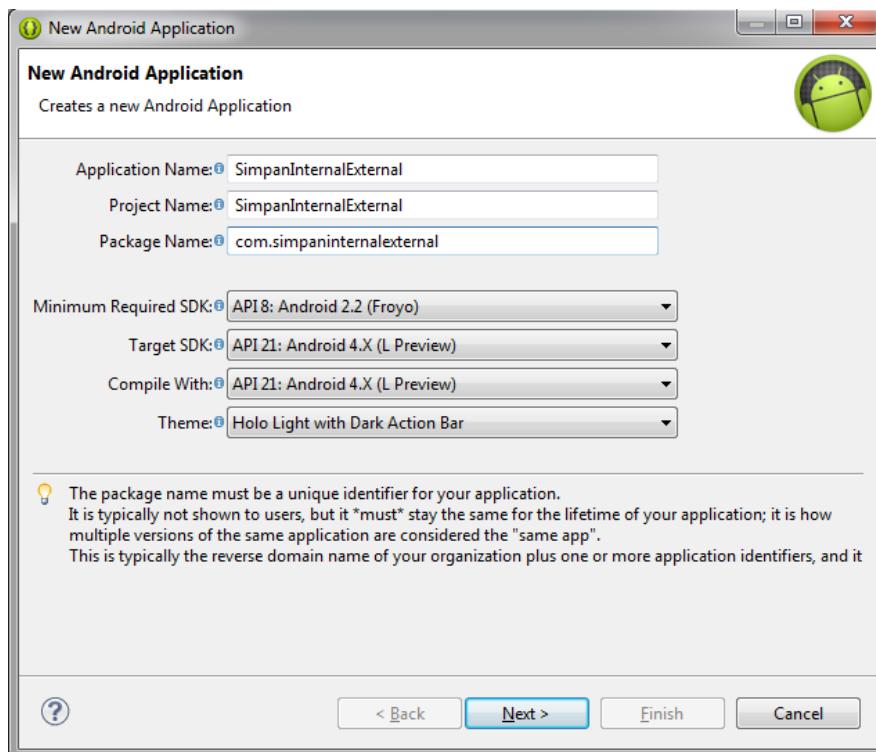
    String setData = InputData.getText().toString();
    //Digunakan untuk membuat dan menulis File/Data di Internal
    FileOutputStream fileOutputStream;
    try {
        //Membuat Berkas Baru dengan mode Private
        fileOutputStream = new FileOutputStream(file);

        fileOutputStream.write(setData.getBytes());
        fileOutputStream.close();

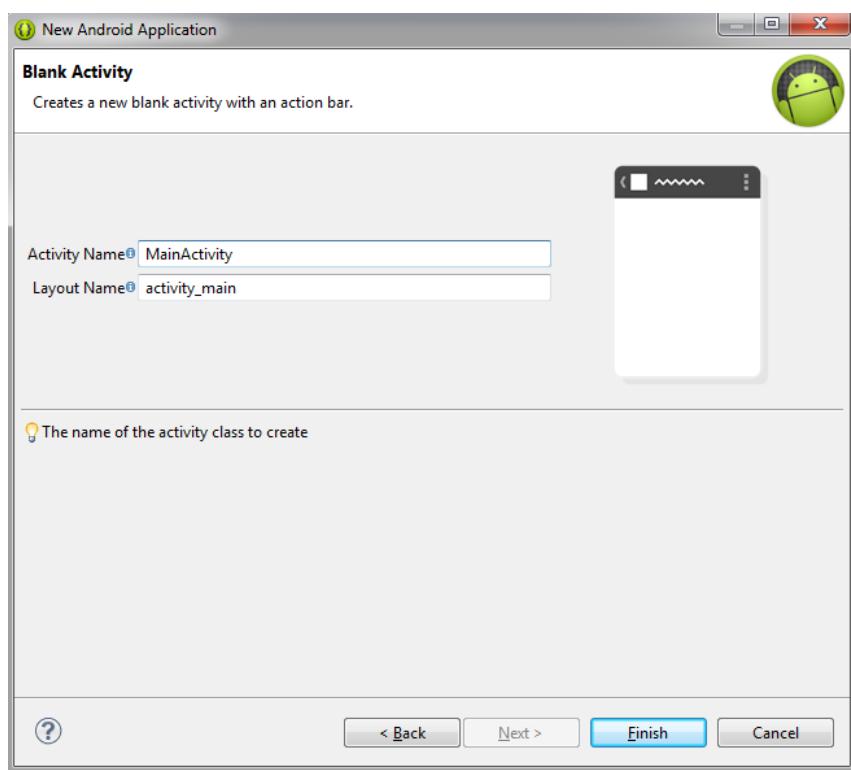
        Toast.makeText(getApplicationContext(), "Data Disimpan di
External", Toast.LENGTH_SHORT).show();
    }catch (IOException ex){
        ex.printStackTrace();
    }
}
}else {
    Toast.makeText(getApplicationContext(), "Penyimpanan External
Tidak Tersedia", Toast.LENGTH_SHORT).show();
}
```

Contoh Implementasi

Buat project baru bernama “SimpanInternalExternal”.



Untuk penamaan Activity dan layout seperti berikut.



Kode untuk Layout

```
<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"
    android:orientation="vertical"
    tools:context="com.simpaninternalexternal.MainActivity" >

    <EditText
        android:id="@+id/input_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Masukan Data"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/save_internal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan ke Internal" />

    <Button
        android:id="@+id/save_external"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan ke External" />

</LinearLayout>
```

Kode untuk Activity

```
package com.simpaninternalexternal;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {
    private Button Internal, External;
```

```

private EditText InputData;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    InputData = (EditText)findViewById(R.id.input_data);
    Internal = (Button)findViewById(R.id.save_internal);
    External = (Button)findViewById(R.id.save_external);

    //menyimpan ke internal
    Internal.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String setData = InputData.getText().toString();
            //Digunakan untuk membuat dan menulis File/Data pada Penyimpanan
            FileOutputStream fileOutputStream;
            try {
                //Membuat Berkas Baru dengan mode Private
                fileOutputStream = openFileOutput("DataSaya",
                    Context.MODE_PRIVATE);

                //Menulis Data Baru dan Mengkonversinya kedalam bentuk byte
                fileOutputStream.write(setData.getBytes());

                //Menutup FileOutputStream
                fileOutputStream.close();

                Toast.makeText(getApplicationContext(), "Data Disimpan di
Internal", Toast.LENGTH_SHORT).show();
            }catch (Exception ex){
                ex.printStackTrace();
            }
        }
    });

    //menyimpan ke eksternal
    External.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            //Digunakan Untuk Mengecek Apakah Tersebut Penyimpanan External
            atau Tidak
            String state = Environment.getExternalStorageState();
            if (Environment.MEDIA_MOUNTED.equals(state)){
                //Digunakan Untuk Mengecek Lokasi Penyimpanan Direktori di
                Memory External
                File dirExternal = Environment.getExternalStorageDirectory();

                //Membuat Direktori Baru dengan Nama "ContohDirektori"
                File createDir = new
File(dirExternal.getAbsolutePath()+" /ContohDirektori");

                //Jika Direktori "ContohDirektori" Tidak Ada, Maka akan
                Membuatnya
                if (!createDir.exists()){
                    createDir.mkdir();
                    //Membuat File Baru didalam Direktori "ContohDirektori"
                    File file = new File(createDir, "ContohFile.txt");
                }
            }
        }
    });
}

```

```
String setData = InputData.getText().toString();
//Digunakan untuk membuat dan menulis File/Data di
Internal
FileOutputStream fileOutputStream;
try {
    //Membuat Berkas Baru dengan mode Private
    fileOutputStream = new FileOutputStream(file);

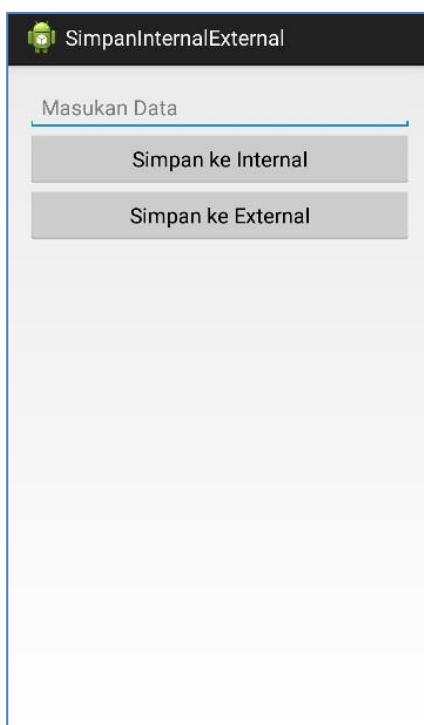
    fileOutputStream.write(setData.getBytes());

    fileOutputStream.close();

    Toast.makeText(getApplicationContext(), "Data Disimpan
di External", Toast.LENGTH_SHORT).show();
} catch (IOException ex){
    ex.printStackTrace();
}
}
}else {
    Toast.makeText(getApplicationContext(), "Penyimpanan External
Tidak Tersedia", Toast.LENGTH_SHORT).show();
}
});
}

}
```

Ketika aplikasi di running

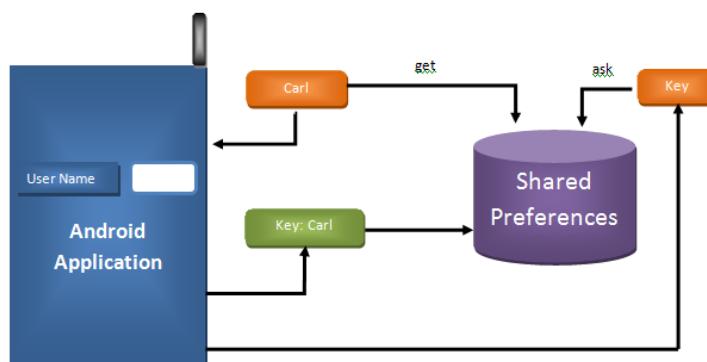


Data yang dibuat dan disimpan pada internal dapat dilihat menggunakan DDMS Perspective dan data yang dibuat dan disimpan pada eksternal/SD card dapat dilihat menggunakan aplikasi file manager. Seperti gambar dibawah.



Menggunakan Shared Preferences.

Shared Preferences merupakan media penyimpanan data primitif pada file internal didalam aplikasi dengan bentuk key-value. Cara kerja shared preferences digambarkan dalam gambar berikut.



Shared preferences biasanya digunakan untuk menyimpan data dalam ukuran kecil seperti user token, setting aplikasi, theme, dan sebagainya. Shared preferences disimpan

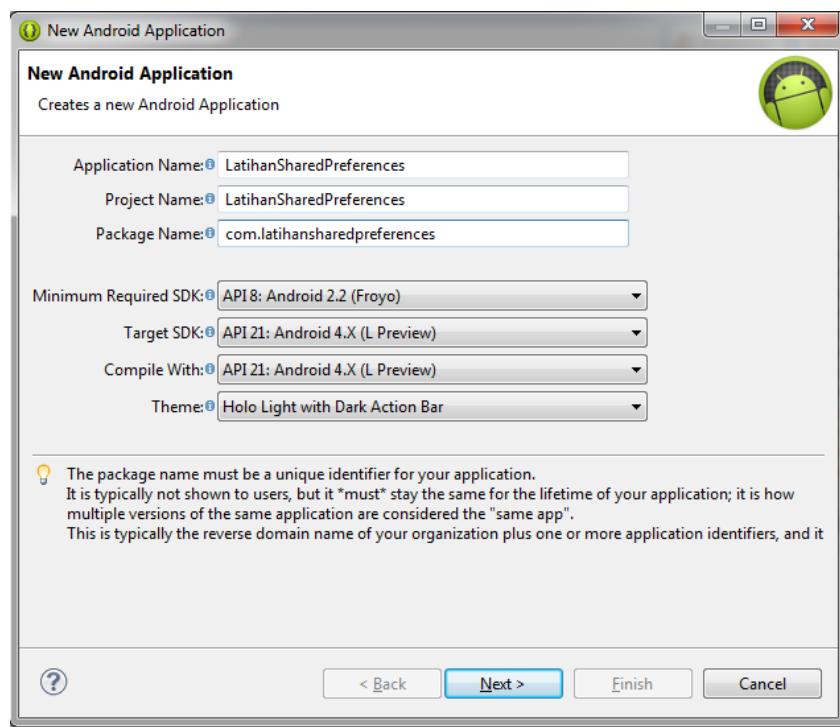
dalam bentuk file XML dan dapat ditemukan pada `data/data/<nama package>/shared_prefs/<nama pacakge>.preferences.xml`.

Untuk mengimplementasikan shared preferences, dapat dengan membuat file shared preference baru atau menggunakan file yang sudah ada dengan dua metode berikut:

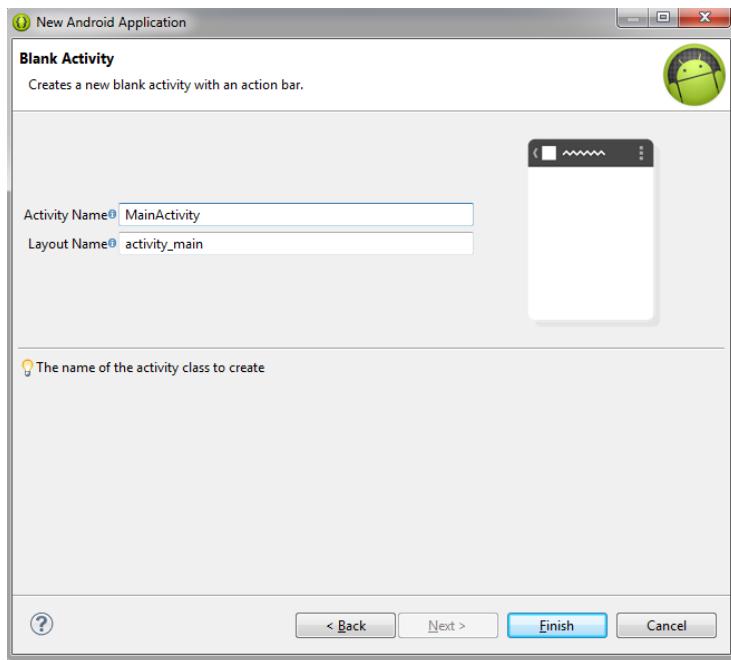
- `getSharedPreferences()`, metode ini digunakan jika ingin membuat file shared preference yang baru dengan parameter yang spesifik dan juga nama filenya spesifik dengunakan Context untuk mengakses file pada aplikasi.
- `getPreferences()`, metode ini digunakan jika ingin menggunakan satu file shared preference pada activity dan tidak perlu menuliskan nama file spesifiknya.

Contoh Implementasi

Buatlah Project dengan nama “LatihanSharedPreferences”



Untuk nama Activity dan layout sebagai berikut.



Kode untuk Layout

```
<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.latihansharedpreferences.MainActivity" >

    <EditText
        android:id="@+id/input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Masukan Data"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan" />

    <TextView
        android:id="@+id/output"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:gravity="center"
        android:text="Output Data :"
        android:textStyle="bold" />
```

```
</LinearLayout>
```

Kode untuk Activity

```
package com.latihansharedpreferences;

import android.app.Activity;
import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {

    //Deklarasi Variable Pendukung
    private TextView Hasil;
    private EditText Masukan;
    private Button Eksekusi;

    //Deklarasi dan Inisialisasi SharedPreferences
    private SharedPreferences preferences;

    //Digunakan Untuk Konfigurasi SharedPreferences
    private SharedPreferences.Editor editor;

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

        Masukan = (EditText) findViewById(R.id.input);
        Hasil =(TextView) findViewById(R.id.output);
        Eksekusi = (Button) findViewById(R.id.save);

        //Membuat File Baru Beserta Modifiernya
        preferences = getSharedPreferences("Belajar_SharedPreferences",
Context.MODE_PRIVATE);

        Eksekusi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getData();
                Toast.makeText(getApplicationContext(), "Data Tersimpan",
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```
private void getData(){
    //Mendapatkan Input dari user
    String getKonten = Masukan.getText().toString();

    //Digunakan Untuk Pengaturan Konfigurasi SharedPreferences
    editor = preferences.edit();

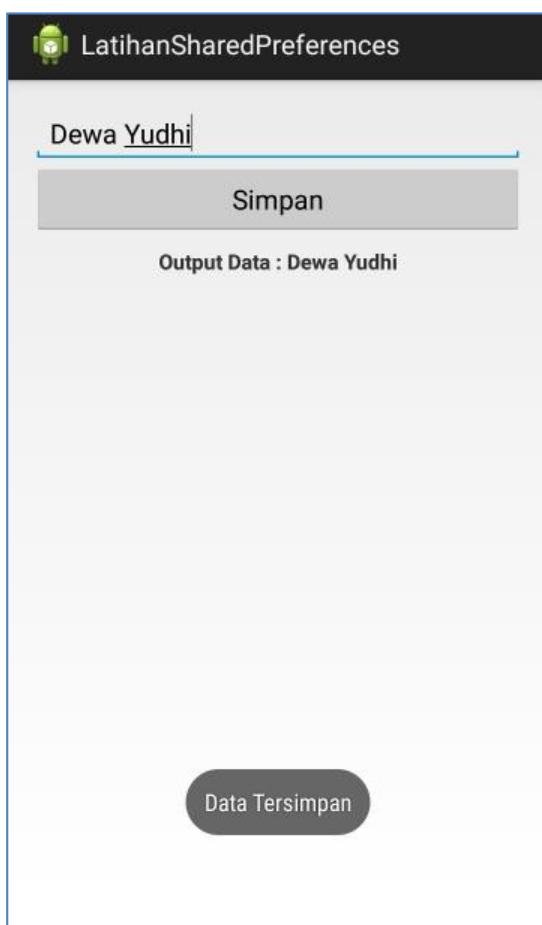
    //Memasukan Data Pada Editor SharedPreferences dengan key (data_saya)
    editor.putString("data_saya", getKonten);

    //Menjalankan Operasi
    editor.apply();

    //Menampilkan Output
    Hasil.setText("Output Data : "+preferences.getString("data_saya", null));
}

}
```

Ketika program dijalankan



Latihan

Menyimpan data menggunakan Shared Preferences.

MODUL 8

BEKERJA DENGAN SERVICES DAN BROADCAST RECEIVER

Tujuan

- Mahasiswa memahami services.
- Mahasiswa mampu menggunakan Services.
- Mahasiswa memahami Broadcast Receivers.
- Mahasiswa mampu menggunakan Broadcast Receivers.

Services

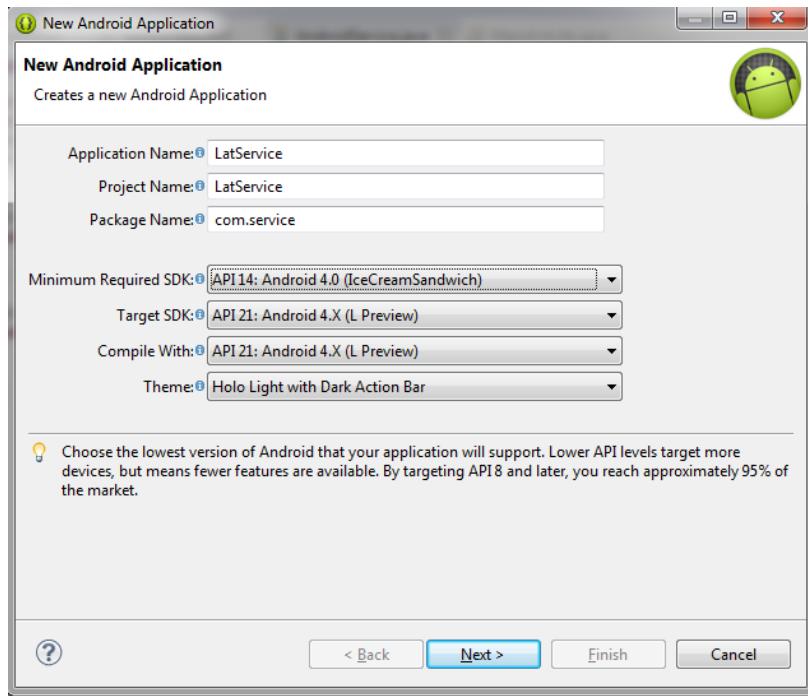
Service merupakan suatu fungsi pada platform Android yang digunakan untuk menjalankan suatu perintah/proses di *background*. *Service* serupa dengan *Asyntask*, perbedaanya adalah *service* digunakan untuk proses yang lebih lama dibandingkan *asyntask*. Perbedaan yang lain *asyntask* berjalan saat *activity* yang menjalankannya masih hidup, sedangkan service masih dapat berjalan meski *activity* yang menjalankannya sudah mati.

Service tidak memiliki *interface* dan menjalankan aplikasi berdasarkan prioritas dari aplikasi itu sendiri. Untuk menggunakan *service*, diperlukan membuat atau menggunakan subclass dari *service*.

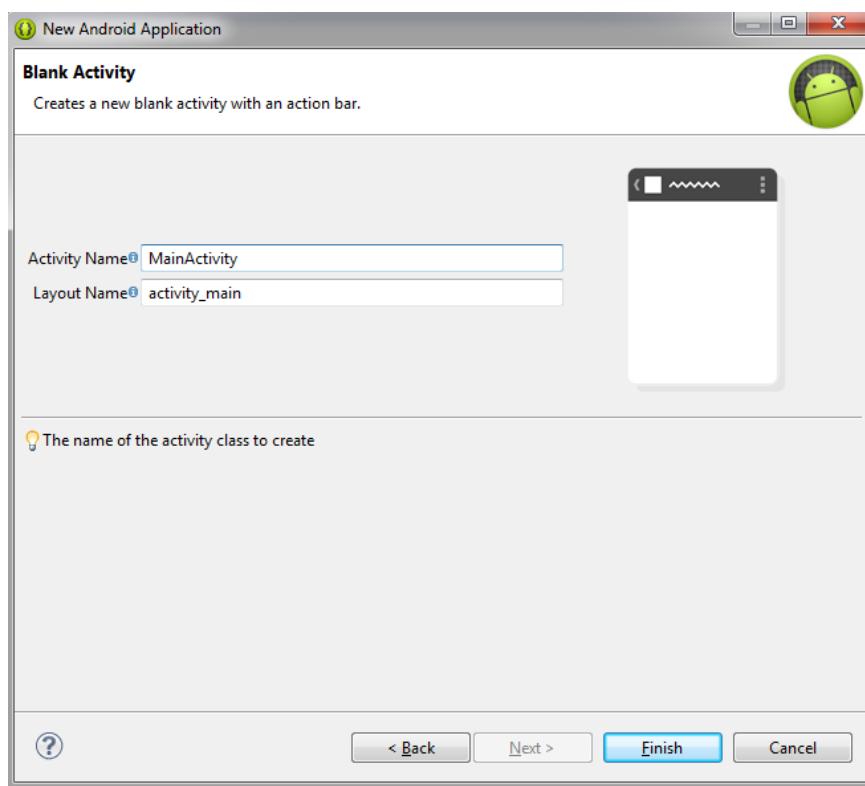
- onStartCommand
- onCreate
- onBind
- onDestroy

Contoh implementasi

Buat sebuah project dengan nama “LatService”.



Berikut nama Activity dan Layout



Buat sebuah class baru untuk service dengan nama “AndroidService” dengan kode sebagai berikut.

```

package com.service;

import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.widget.Toast;

public class AndroidService extends Service {

    private boolean isRunning = false;

    @Override
    public void onCreate() {
        Toast.makeText(this, "Service telah dibuat", Toast.LENGTH_LONG).show();

        isRunning = true;
    }

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        Toast.makeText(this, "Service Started", Toast.LENGTH_LONG).show();
        return START_STICKY;
    }

    @Override
    public void onDestroy() {

        isRunning = false;
        Toast.makeText(this, "Service di Destroy", Toast.LENGTH_LONG).show();
    }

}

```

Kode untuk Layout

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:orientation="horizontal"
    android:padding="10sp">

    <Button
        android:id="@+id/btnStart"
        android:layout_width="wrap_content"

```

```

    android:layout_height="wrap_content"
    android:text="Start Service" />

<Button
    android:id="@+id	btnStop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Stop Service" />
</LinearLayout>

```

Kode untuk MainActivity

```

package com.service;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btnStart = (Button) findViewById(R.id.btnStart);
        Button btnStop = (Button) findViewById(R.id.btnStop);

        btnStart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startService(new Intent(MainActivity.this, AndroidService.class));
            }
        });
        btnStop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                stopService(new Intent(MainActivity.this, AndroidService.class));
            }
        });
    }
}

```

Service yang sudah dibuat harus diregisterkan pada AndroidManifest

```
<service android:name=".AndroidService" />
```

Tempatkan didalam tag <application> </application>

Kode AndroidManifest setelah meregisterkan service

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.service"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

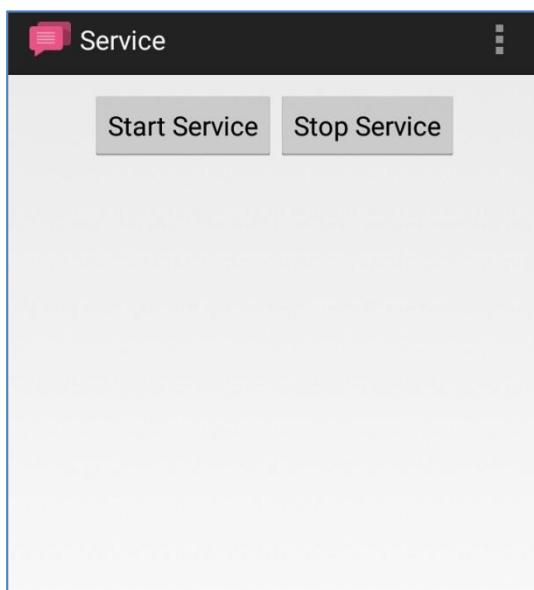
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.service.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

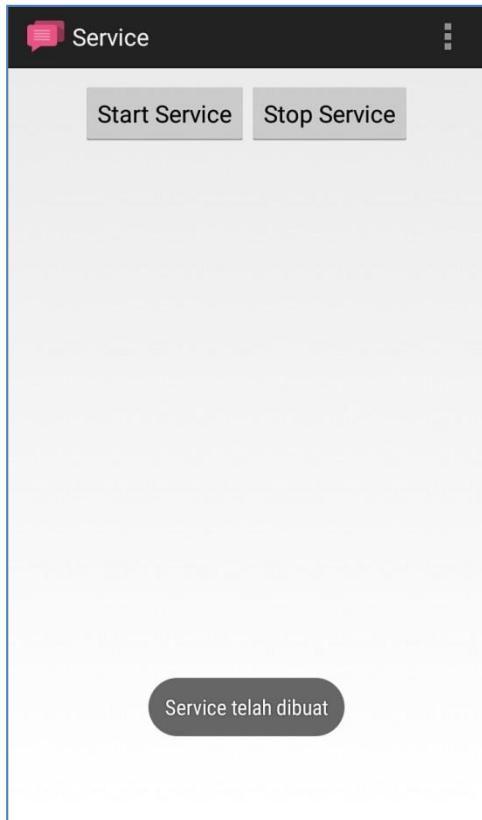
        <service android:name=".AndroidService" />
    </application>

</manifest>
```

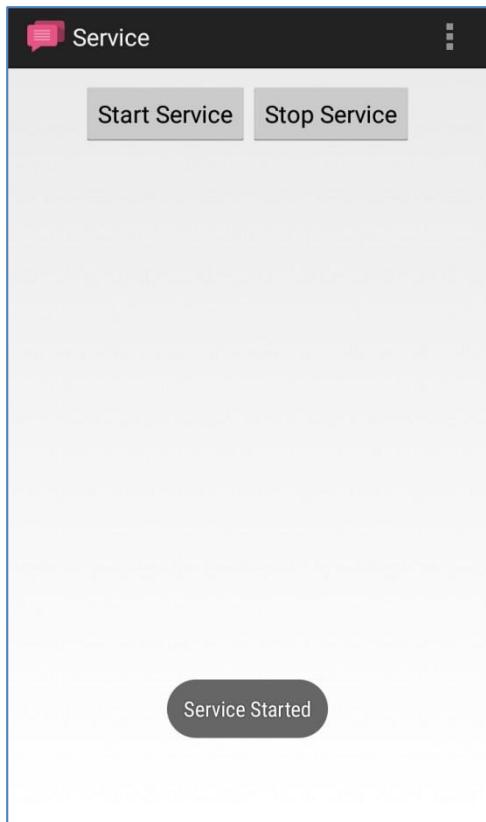
Ketika program dijalankan



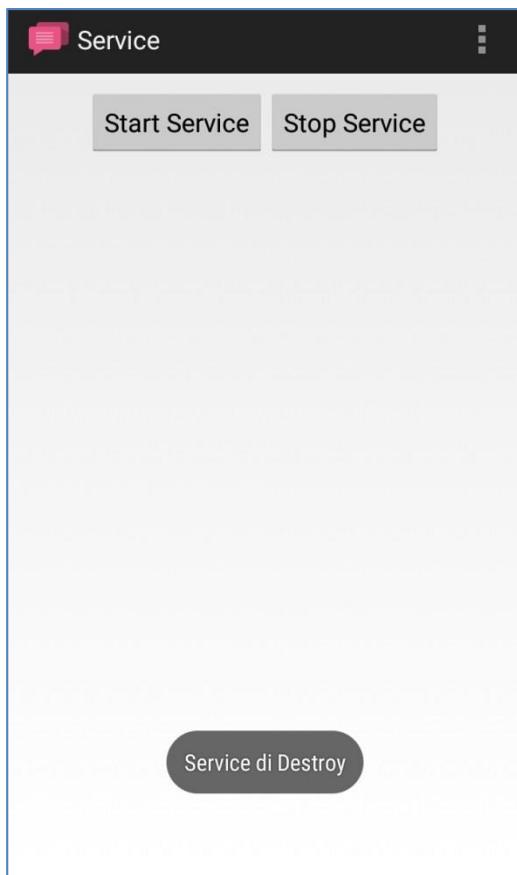
Ketika tombol Start Service ditekan. Service dibuat



Selanjutnya service dimulai setelah dibuat



Ketika service di stop



Broadcast Receiver

Broadcast Receiver adalah sebuah komponen Android yang berfungsi untuk melakukan broadcast pesan dari aplikasi lain atau dari sistem itu sendiri. Pesan di broadcast disebut event atau intent. Aplikasi melakukan broadcast untuk memberi tahu aplikasi lainnya bahwa beberapa data telah di unduh ke perangkat yang tersedia, lalu penerima broadcast akan menerima pesan untuk melakukan transfer informasi.

Untuk membuat broadcast receiver, definisikan subclass dari class `BroadcastReceiver` dan implementasikan method `onReceive()`. Selanjutnya registerkan broadcast receiver, dapat melalui cara dinamis pada java, atau secara statik pada `AndroidManifest`.

Contoh penggunaan broadcast receiver, pada contoh ini class `AlarmReceiver` merupakan subclass/turunan dari `BroadcastReceiver` yang menampilkan pesan `Toast` jika broadcast intent yang datang mempunyai action `ACTION_SHOW_TOAST`.

```
private class AlarmReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(ACTION_SHOW_TOAST)) {
            CharSequence text = "Broadcast Received!";
            int duration = Toast.LENGTH_SHORT;

            Toast toast = Toast.makeText(context, text, duration);
            toast.show();
        }
    }
}
```

Untuk meregisterkan broadcast receiver terdapat dua buah cara yaitu secara statik pada manifest atau secara dinamis pada activity.

Static Registration

Untuk meregister broadcast receiver secara statik, tambahkan `<receiver>` pada `AndroidManifest.xml` pada aplikasi. Didalam elemen `<receiver>` gunakan path dari `BroadcastReceiver` subclas sebagai attribute `android:name`. Untuk mencegah aplikasi lain untuk mengirim broadcast kepada receiver yang dibuat, bisa menset attribute `android:exported` menjadi false, tapi hal ini bersifat opsional. Untuk secara spesifik tipe intents yang sedang di listen oleh komponen dapat menggunakan element `<intent-filter>`.

Contoh penerapan disini nama `subclass` dari `BroadcastReceiver` adalah `AlarmReceiver`. Receiver tidak di eksport, yang berarti tidak ada aplikasi lain yang mengirim broadcast pada aplikasi ini. `Receiver` mengandung intent-filter yang mengecek apakah terdapat intents yang mengandung action dengan nama `ACTION_SHOW_TOAST`.

```
<receiver
  android:name="com.example.myproject.AlarmReceiver"
  android:exported="false">
  <intent-filter>
    <action android:name="com.example.myproject.intent.action.ACTION_SHOW_TOAST"/>
  </intent-filter>
</receiver>
```

Contoh intent-filter dari receiver yang me listent perangkat untuk menyelesaikan booting.

```
<intent-filter>
  <action android:name="android.intent.action.BOOT_COMPLETED"/>
</intent-filter>
```

Dynamic Registration

Regisratsi dari broadcast receiver juga dapat secara dinamis menggunakan `registerReceiver()` yang terikat dengan daur hidup dari activity. Contoh

```
IntentFilter intentFilter = new IntentFilter();
intentFilter.addAction(ACTION_SHOW_TOAST);

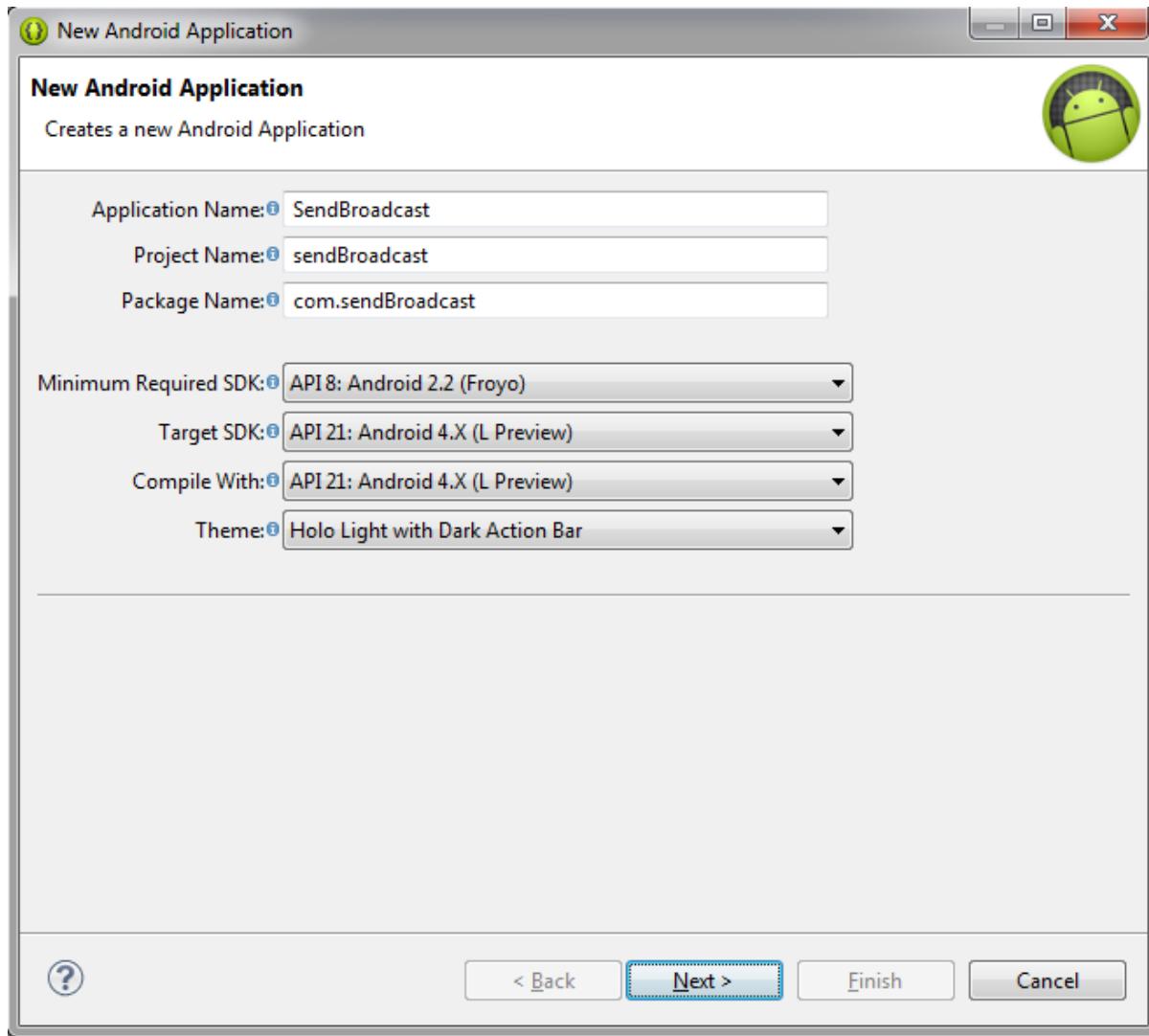
mReceiver = new AlarmReceiver();
registerReceiver(mReceiver, intentFilter);
```

registrasi secara dinamis digunakan hanya untuk local broadcast. Static registration bukanlah opsi. Untuk mengunregister menggunakan `unregisterReceiver`. Sebagai contoh:

```
unregisterReceiver(mReceiver);
```

Contoh Penerapan Broadcast Receiver

Pada contoh kali ini akan dibuat custom broadcast Receiver. Buatlah sebuah project baru bernama “SendBroadcast” seperti gambar berikut.



Untuk penamaan Activity adalah MainActivity dan Layout adalah “activity_main”

Pada Layout, terdapat sebuah tombol yang fungsinya jika di klik akan mengirimkan pesan broadcast. Pada widget button tambahkan atribut `onClick` yang bertugas untuk memanggil method “`kirimBroadcast`” pada `MainActivity`. Berikut kode untuk tampilan layout.

```
<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.sendBroadcast.MainActivity" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="106dp"
        android:onClick="kirimBroadcast"
        android:text="Kirim Broadcast" />

</RelativeLayout>
```

Selanjutnya pada `MainActivity` tambahkan method `kirimBroadcast` yang bertugas mengirimkan broadcast.

```
package com.sendBroadcast;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends Activity {

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

    //method on click untuk button yang berfungsi mengirimkan broadcast
    public void kirimBroadcast(View view){
        Intent i=new Intent();
        i.setAction("com.latihanbroadcast.CUSTOM_INTENT");
        sendBroadcast(i);
    }
}
```

Untuk menerima broadcast yang dikirimkan. Selanjutnya membuat sebuah class baru (bukan activity) yang dipergunakan untuk menerima pesan tersebut. Beri nama class tersebut dengan nama “MyReceiver”. Class MyReceiver yang merupakan subclass dari Class BroadcastReceiver sehingga harus mengimplementasikan method `onReceive()`. Berikut adalah kode dari MyReceiver.

```
package com.sendBroadcast;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class MyReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        Toast.makeText(context, "Broadcast diterima",
        Toast.LENGTH_LONG).show();
    }
}
```

Tidak lupa untuk meregisterkan dari broadcast receiver yang dibuat pada AndroidManifest. Berikut kode untuk meregister broadcast receiver secara statis yang akan ditempatkan diantara tag `<application></application>`.

```
<receiver android:name="MyReceiver" >
    <intent-filter>
        <action android:name="com.Latihanbroadcast.CUSTOM_INTENT" ></action>
    </intent-filter>
</receiver>
```

Berikut AndroidManifest setelah diregisterkan broadcast receiver

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sendBroadcast"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
```

```
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

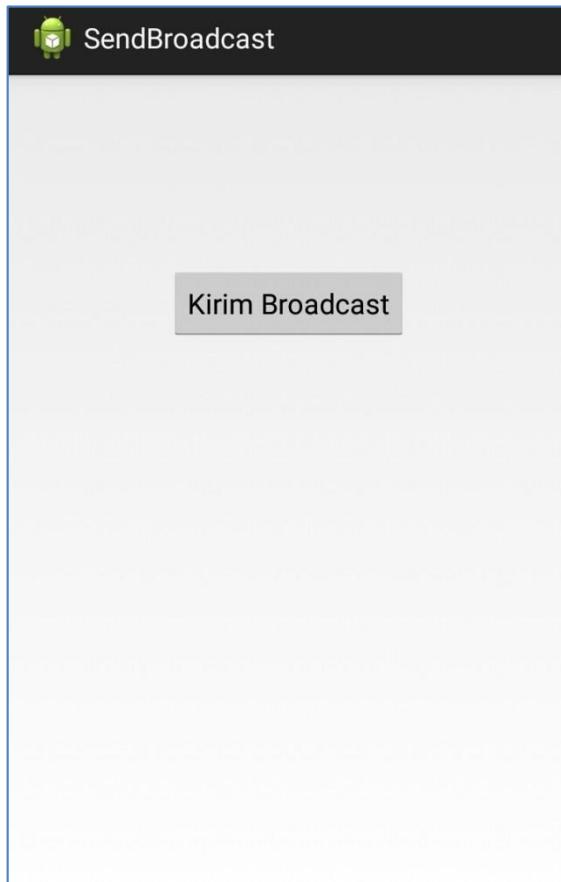
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<receiver android:name="MyReceiver" >
    <intent-filter>
        <action android:name="com.Latihanbroadcast.CUSTOM_INTENT" ></action>
    </intent-filter>
</receiver>

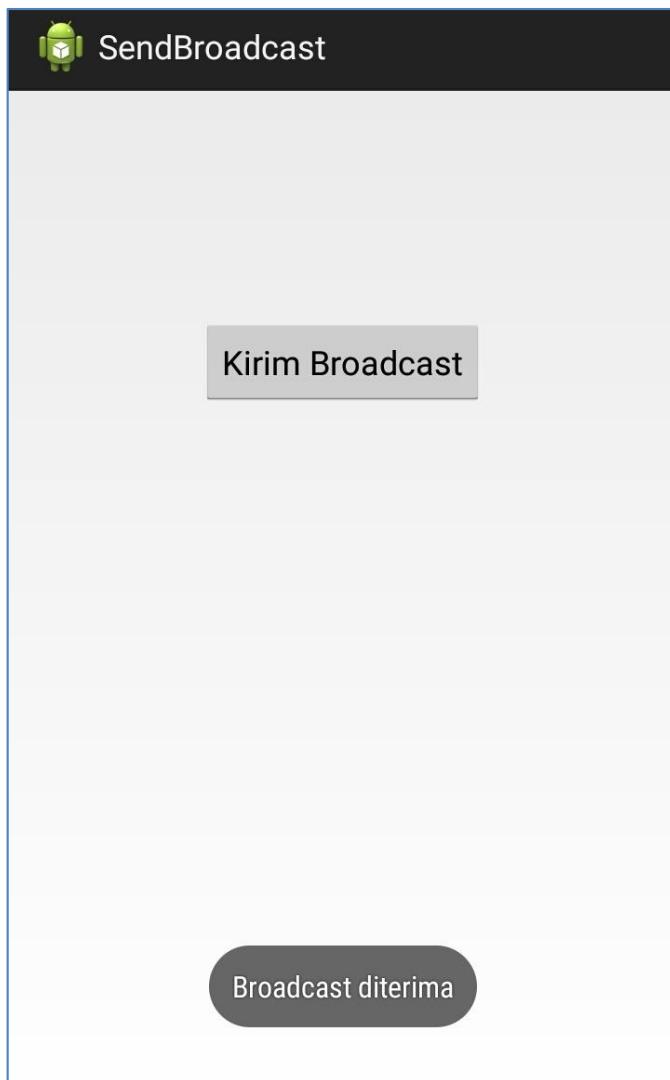
</application>

</manifest>
```

Setelah program di running akan muncul tampilan sebagai berikut



Saat tombol diklik maka broadcast akan dikirimkan dan diterima oleh MyReceiver dan menampilkan pesan Toast seperti gambar berikut.



Latihan

Membuat dan menggunakan services dan Broadcast Receivers

MODUL 9

MENGKREASIKAN TAMPILAN ANTAR MUKA

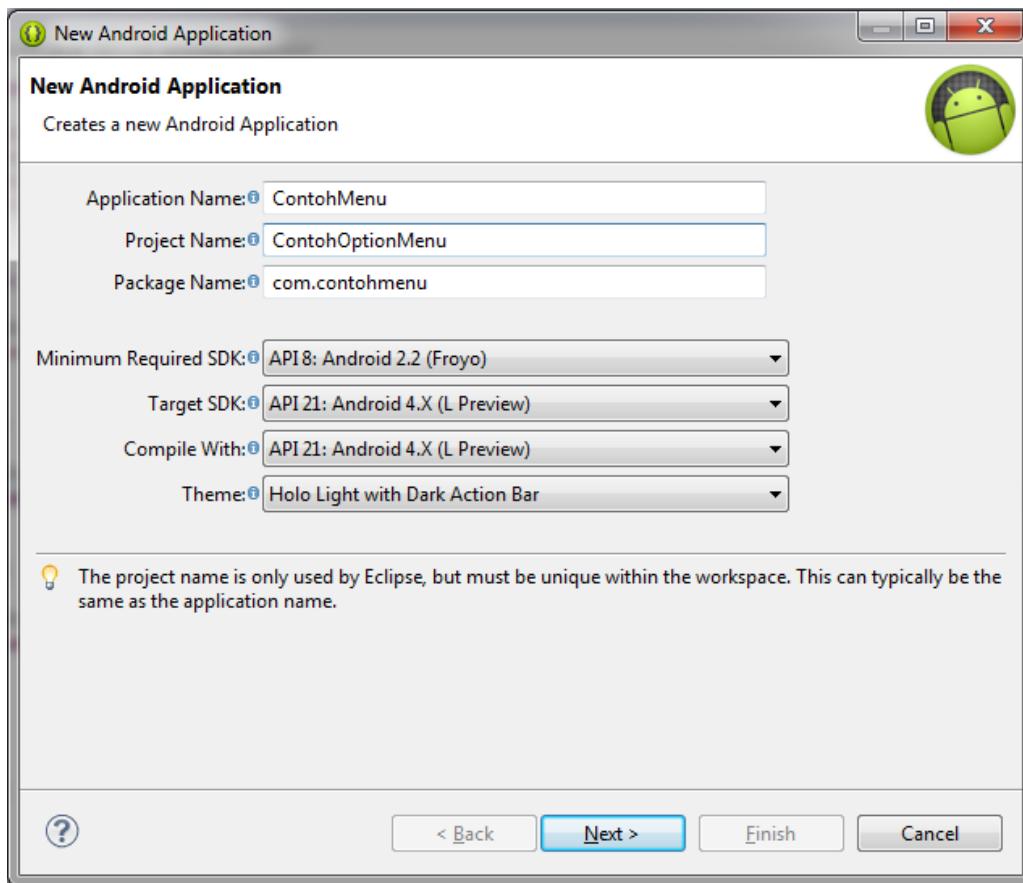
Tujuan

- Mahasiswa mampu membuat Menus.
- Mahasiswa mampu membuat Tabs.
- Mahasiswa mampu menerapkan sebuah Style dan Theme pada aplikasi android.
- Mahasiswa mampu mengkreasikan views.

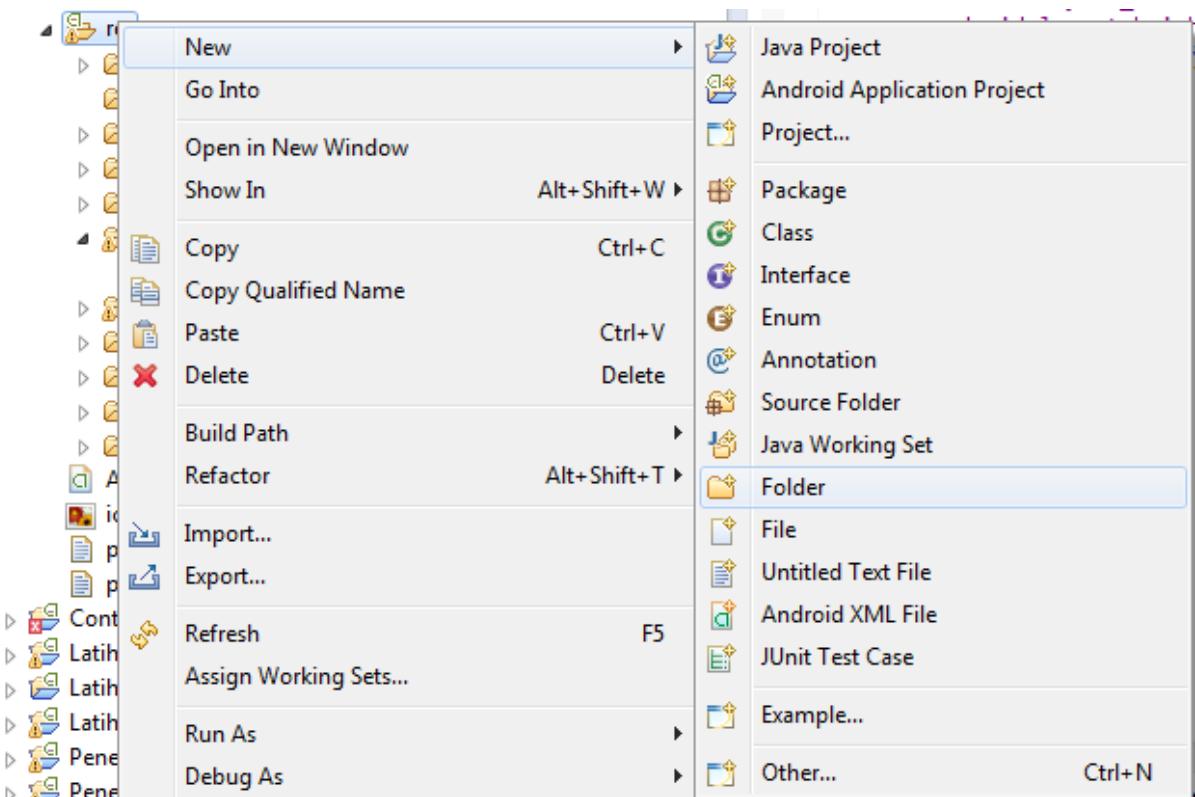
Membuat Menu

Menu adalah tampilan yang muncul ketika tombol menu pada perangkat android ditekan. Menu dipergunakan untuk berbagai kebutuhan sesuai dengan tujuan dari activity. Berikut adalah langkah-langkah untuk membuat menu di Android.

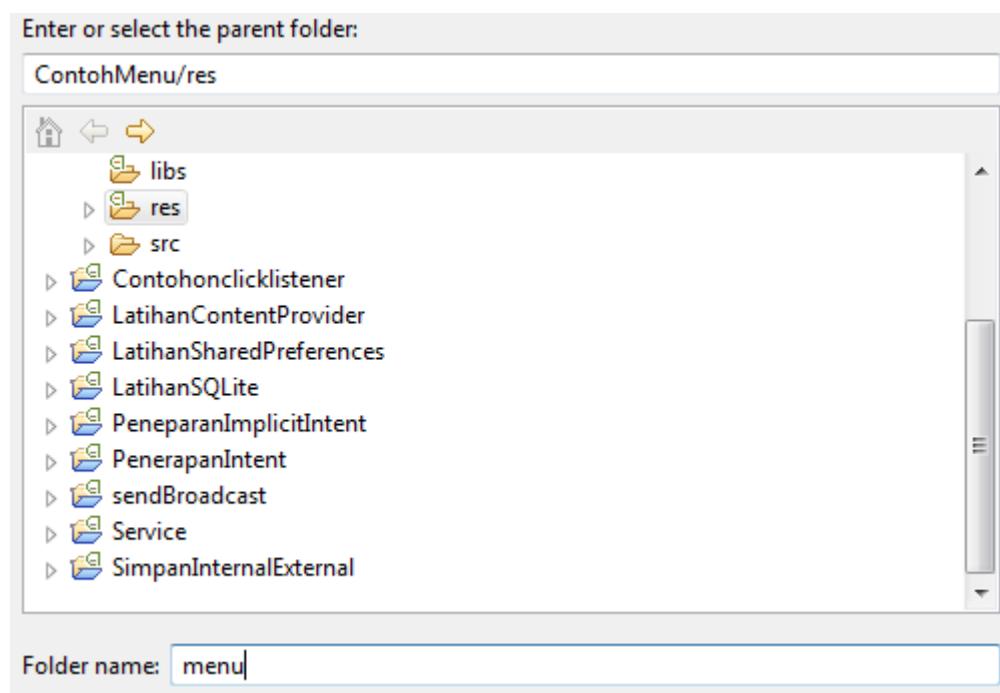
Buatlah project dengan nama “ContohMenu” dengan nama Activity adalah MainActivity dan Layout adalah activity_main



Langkah selanjutnya adalah membuat folder menu didalam folder res (jika kondisi di folder res tidak terdapat folder menu). Jika belum terdapat folder menu didalam folder res, untuk membuatnya dapat mengklik kanan pada folder res pilih new → folder.

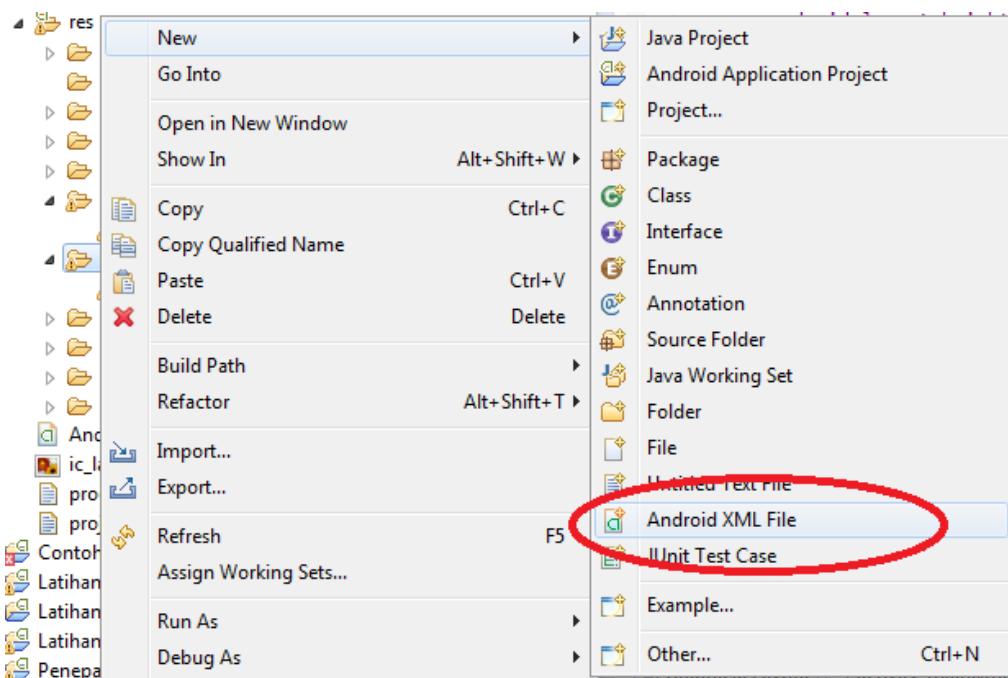


Berikan folder tersebut dengan nama menu. Seperti gambar berikut:

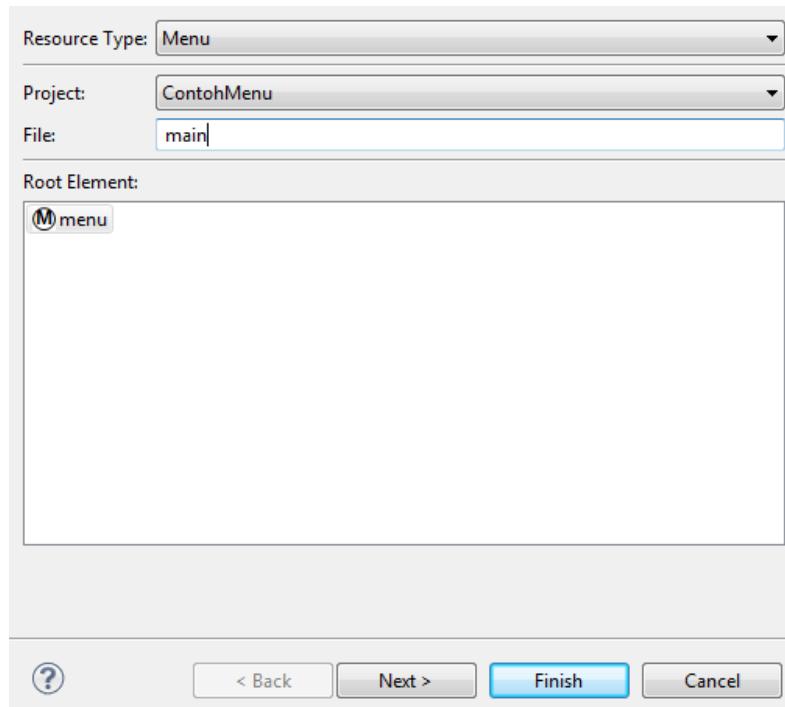


Pilih finish.

Selanjutnya dalam folder menu buat sebuah file xml dengan nama “main”. Caranya dengan mengklik kanan pada folder menu dan memilih new → AndroidXML File



Selanjutnya beri nama main pada file xml yang akan dibuat. Seperti gambar berikut:



Pilih finish

Pada main.xml yang baru dibuat, pada contoh kali ini akan membuat dua buah opsi pada menu yaitu kelola dan bantuan. Berikut kode untuk main.xml.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.contohmenu.MainActivity" >

    <item
        android:id="@+id/kelola"
        android:orderInCategory="100"
        android:title="Kelola"
        app:showAsAction="never"/>

    <item
        android:id="@+id/bantuan"
        android:orderInCategory="100"
        android:title="bantuan"
        app:showAsAction="never"/>

</menu>
```

Untuk memfungsikan menu, pada MainActvity tambahkan kode sebagai berikut:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.kelola) {
        Toast.makeText(getApplicationContext(), "menu kelola dipilih",
Toast.LENGTH_LONG).show();
    }else if(id == R.id.bantuan){
        Toast.makeText(getApplicationContext(), "menu bantuan dipilih",
Toast.LENGTH_LONG).show();
    }

    return super.onOptionsItemSelected(item);
}
```

Pada contoh berikut jika setiap opsi dipilih maka akan menampilkan pesan dari masing-masing opsi. Berikut kode lengkap dari MainActivity.

```
package com.contohmenu;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

public class MainActivity extends Activity {

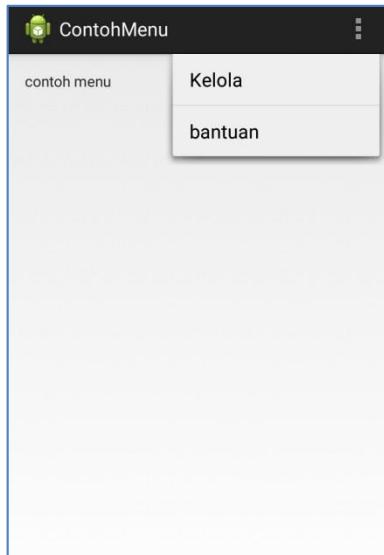
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.kelola) {
            Toast.makeText(getApplicationContext(), "menu kelola dipilih",
Toast.LENGTH_LONG).show();
        }else if(id == R.id.bantuan){
            Toast.makeText(getApplicationContext(), "menu bantuan dipilih",
Toast.LENGTH_LONG).show();
        }

        return super.onOptionsItemSelected(item);
    }
}
```

Ketika aplikasi di running. Menu akan tampil ketika menu pada pojok atas bar ditekan. Pilihan menu digambarkan dengan tiga titik yang linear secara vertikal. Seperti gambar berikut:



Ketika salah satu menu dipilih maka akan menampilkan toas dari opsi menu yang dipilih.

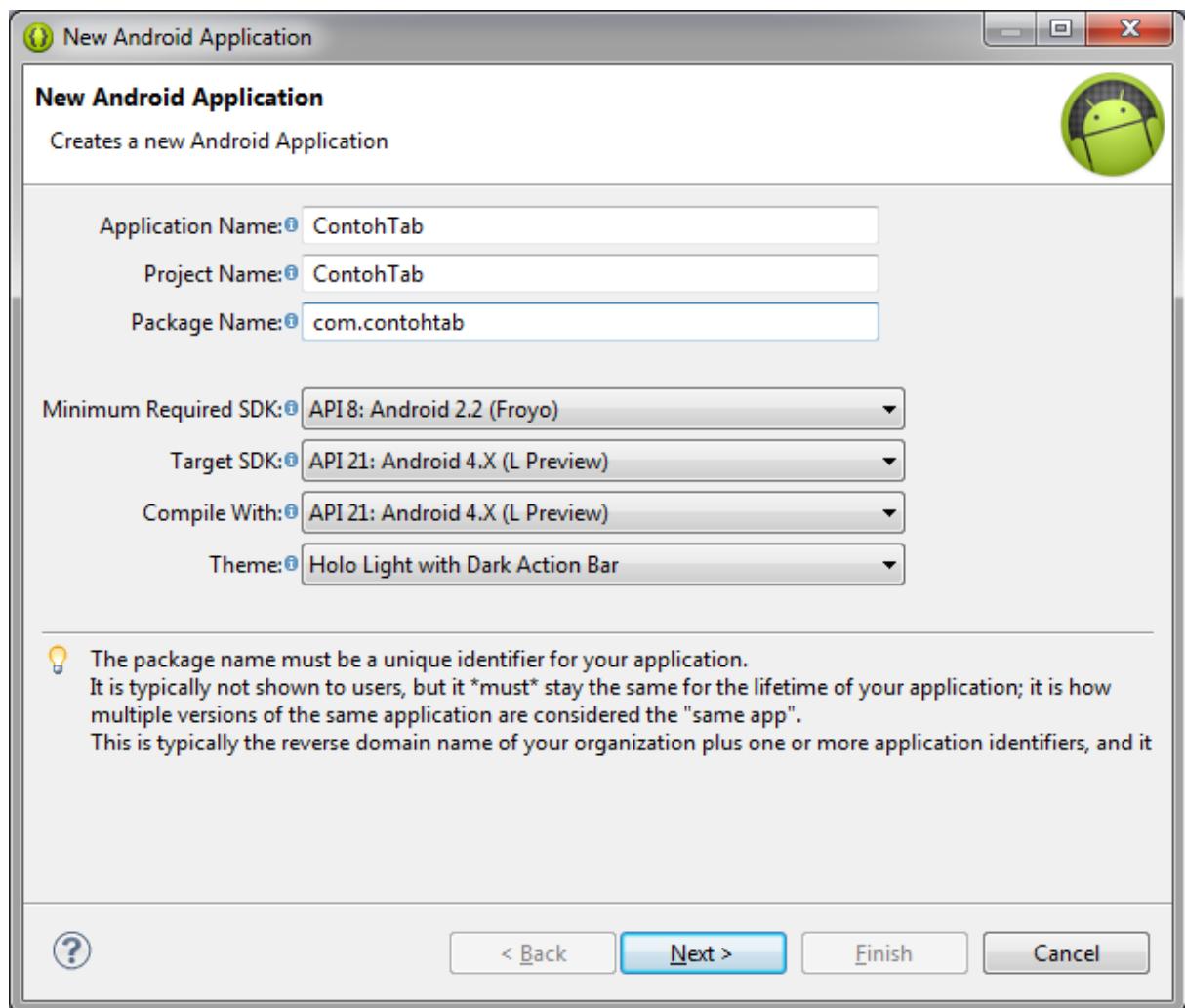


Membuat Tab

Menu Tab digunakan untuk menampilkan banyak tampilan dalam satu halaman. Untuk mengimplementasikan tab beberapa komponen diperlukan

- Tab host, dipergunakan sebagai container/penampung tombol tab dan isi dari tab.
- Tab widget, terdiri dari baris tombol, yang bisa mengandung icon atau label teks.
- Frame Layout, penampung dari isi tab.

Berikut adalah contoh implementasi dari tab. Buatlah sebuah project dengan nama ContohTab dengan Activity MainActivity dan Layout activity_main:



Selanjutnya buat tiga buah activity baru yaitu Activity1 dengan layoutnya activity_activity1.xml, Activity2 dengan layoutnya activity_activity2.xml dan Activity3 dengan layoutnya activity_activity3.xml.

Kode dari layout activity_activity1.xml

```
<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.contohtab.Activity1" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Activity 1" />

</RelativeLayout>
```

Kode dari layout activity_activity2.xml

```
<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.contohtab.Activity2" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Activity 2" />

</RelativeLayout>
```

Kode dari layout activity_activity3.xml

```
<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.contohtab.Activity3" >

    <TextView
```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Activity 3" />

</RelativeLayout>

```

Selanjutnya kode untuk layout activity_main.xml sebagai berikut, dimana terdapat tiga komponen wajib untuk membangun sebuah tab, yaitu TabHost, TabWidget dan FrameLayout.

```

<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tabhost"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >

        <TabWidget
            android:id="@+id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom" />

        <FrameLayout
            android:id="@+id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_gravity="bottom" />

    </LinearLayout>
</TabHost>

```

Selanjutnya kode dari Activity MainActivity sebagai berikut, dimana MainActivity merupakan subclass dari TabActivity.

```

package com.contohtab;

import android.app.TabActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TabHost;

@SuppressWarnings("deprecation")
public class MainActivity extends TabActivity {

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

        TabHost tabhost = getTabHost();

```

```
TabHost.TabSpec spec;
Intent intent;

intent = new Intent().setClass(this, Activity1.class); //content pada tab
yang akan kita buat
spec = tabhost.newTabSpec("Activity1").setIndicator("Activity
1", null).setContent(intent); //mengeset nama tab dan mengisi content pada menu tab
anda.
tabhost.addTab(spec); //untuk membuat tabbaru disini bisa diatur sesuai
keinginan anda

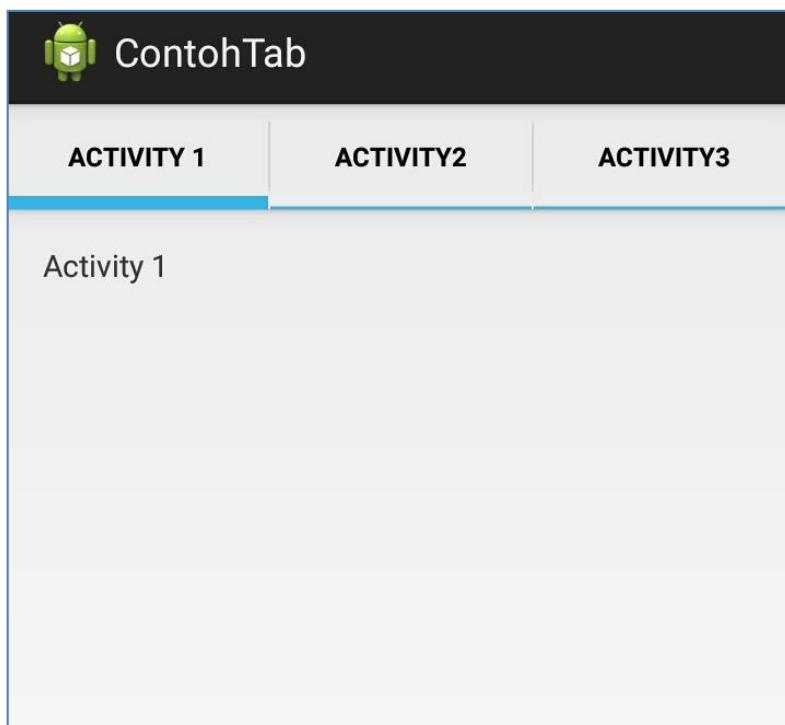
intent = new Intent().setClass(this, Activity2.class);
spec =
tabhost.newTabSpec("Activity2").setIndicator("Activity2", null).setContent(intent);
tabhost.addTab(spec);

intent = new Intent().setClass(this, Activity3.class);
spec =
tabhost.newTabSpec("Activity3").setIndicator("Activity3", null).setContent(intent);
tabhost.addTab(spec);

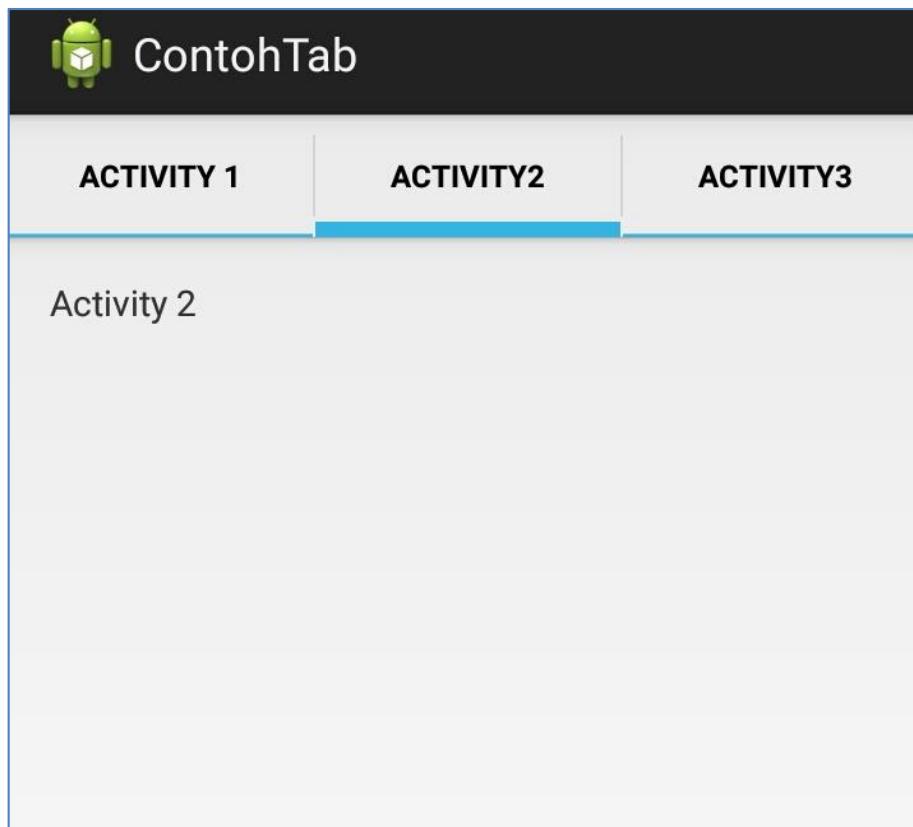
}
```

Setelah program di running maka akan tampil tampilan sebagai berikut. Ketika masing-masing tab ditekan maka akan memunculkan halaman activity yang sudah ditentukan pada MainActivity.

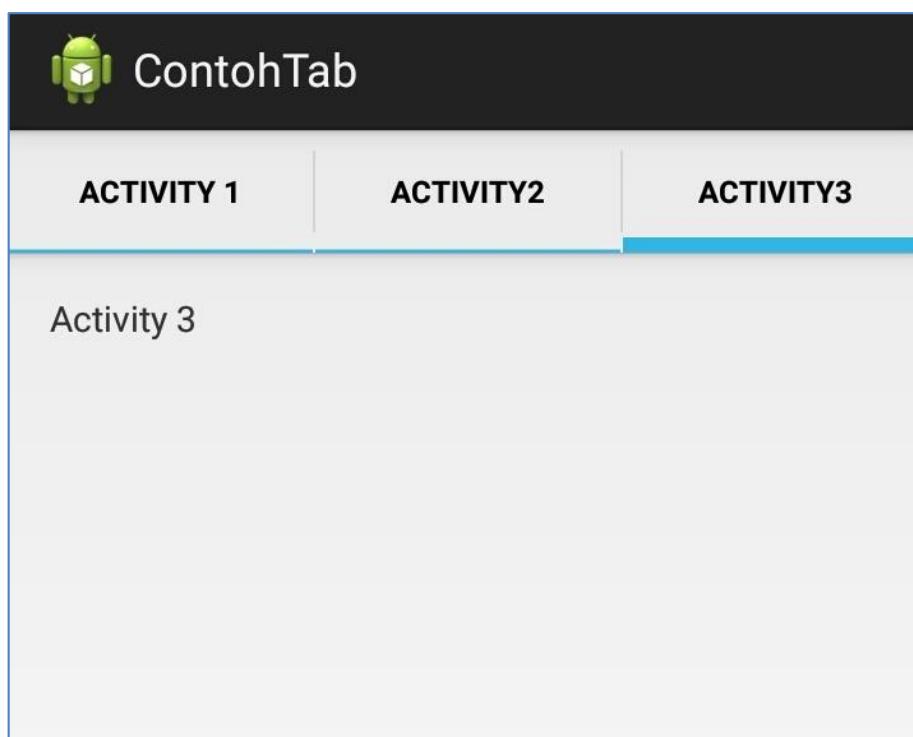
Ketika Tombol Activity 1 ditekan, maka akan memunculkan isi dari Activity1.



Ketika tombol Activity 2 ditekan, Maka akan memuncul isi dari Activity2.

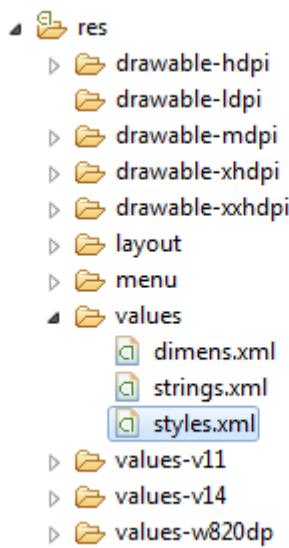


Ketika tombol Activity 3 ditekan, Maka akan memuncul isi dari Activity3.



Menerapkan Style dan Theme pada aplikasi Android

Style adalah kumpulan dari atribut yang digunakan secara spesifik untuk memberikan format dan tampilan pada view atau window. Dengan style dapat mengatur properti dari tampilan seperti tinggi, lebar, padding, warna font dan warna background. Untuk membuat style kustom, tambahkan koding pada file style.xml pada folder values.



Style menggunakan tag `<style>` `</style>` yang harus berada dalam tag `<resources>` seperti contoh berikut, yang membuat style dengan nama MyStyle yang berguna untuk mengatur warna, ukuran dan jenis tulisan.

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

  <style name="MyStyle" parent="@android:style/TextAppearance.Medium">
    <item name="android:textColor">#4682B4</item>
    <item name="android:textSize">20sp</item>
    <item name="android:typeface">serif</item>
  </style>

</resources>
```

Untuk menerapkan style pada widget seperti TextView tambahkan atribut style seperti contoh berikut.

```
<TextView
  style="@style/MyStyle"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/hello_world" />
```

Untuk menerapkan style diterapkan pada activity, tambahkan kode berikut di AndroidManifest.

```

<activity
    android:name=".MainActivity"
    android:theme="@style/MyStyle"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
  
```

Tambahkan antribut android:theme pada activity di manifest. Jika ingin diterapkan ke seluruh activity pada suatu aplikasi, ubah antribut android:theme pada tag application dengan style kustom yang kita buat. Seperti kode berikut.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.contohstyle"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="21" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/MyStyle" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
  
```

Latihan

Membuat aplikasi kreasi dengan memanfaatkan tab, style dan theme

MODUL 10

NOTIFIKASI PENGGUNA

Tujuan

- Mahasiswa mampu mengaplikasikan Notifikasi.
- Mahasiswa mampu mengaplikasikan Alarms.

Mengaplikasikan Notifikasi

Notifikasi adalah cara aplikasi untuk mendapatkan perhatian pengguna ketika terjadi suatu kejadian/event. Kejadian dapat berupa yang dibangkitkan oleh sistem, seperti pemberitahuan tentang baterai yang dayanya akan habis ataupun kejadian yang dibangkitkan oleh pengguna itu sendiri.

Terdapat beberapa teknik untuk menotifikasi pengguna diantaranya:

- Toast notification, digunakan untuk menampilkan pesan yang kecil.
- Dialog notification, digunakan untuk menampilkan progress bar atau pesan pendek yang memerlukan konfirmasi dari pengguna.

Membuat Toast Notification

Toast notification menampilkan pesan yang tidak memerlukan interaksi dengan pengguna. Toast notification tampil dengan animasi fade in dan fade out. Cara untuk membuat Toast notification.

```
android.widget.Toast.makeText(Context, text, duration);
```

pada kode tersebut metode `makeText` mempunyai parameter

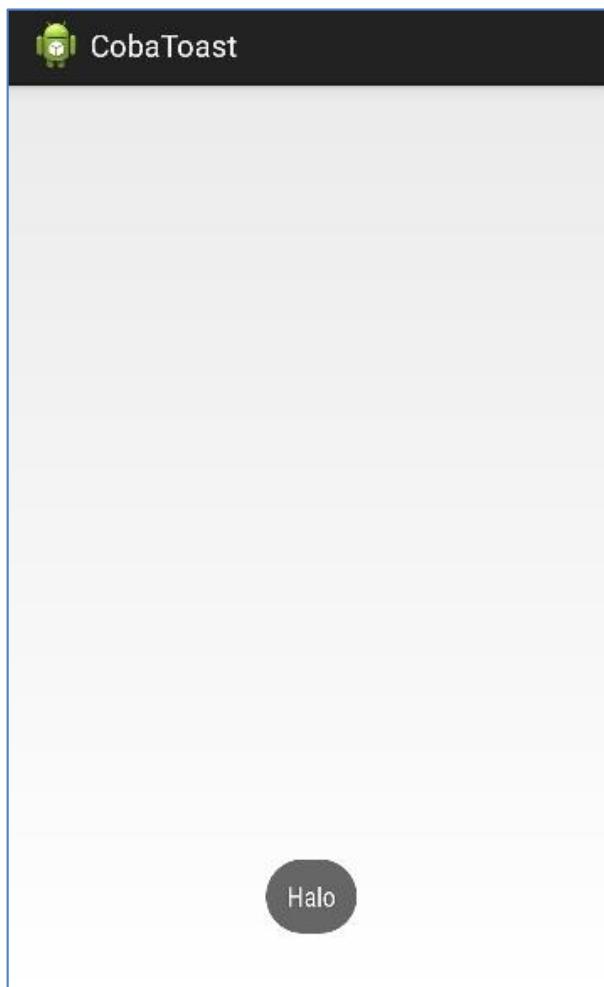
- `Context`, yaitu activity yang menampilkan toast.
- `text`, pesan yang ingin ditampilkan.
- `Duration`, lamanya pesan tampil. Untuk yang pendek pilih `LENGTH_SHORT` atau yang lebih panjang bisa pilih `LENGTH_LONG`.

Contoh Implementasi

Tambahkan kode berikut pada project.

```
android.widget.Toast.makeText(getApplicationContext(), "Halo", Toast.LENGTH_LONG).show();
```

Ketika aplikasi di running, akan tampil tampilan berikut.



Dialog adalah window kecil yang muncul di depan activity yang sedang aktif. Dialog menerima interaksi dengan pengguna. Dialog memang digunakan untuk menampilkan progress bar atau pesan yang memerlukan konfirmasi dari pengguna. Terdapat beberapa tipe Dialog, antara lain.

- **AlertDialog**, digunakan untuk membuat dialog peringatan yang dapat terdiri dari tiga tombol. Contoh implementasi

```
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

builder.setTitle("Delete");
builder.setMessage("Apakah anda akan menghapus?");

builder.setPositiveButton("delete",
new DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText(getApplicationContext(), "delete di tekan",
        Toast.LENGTH_SHORT).show();

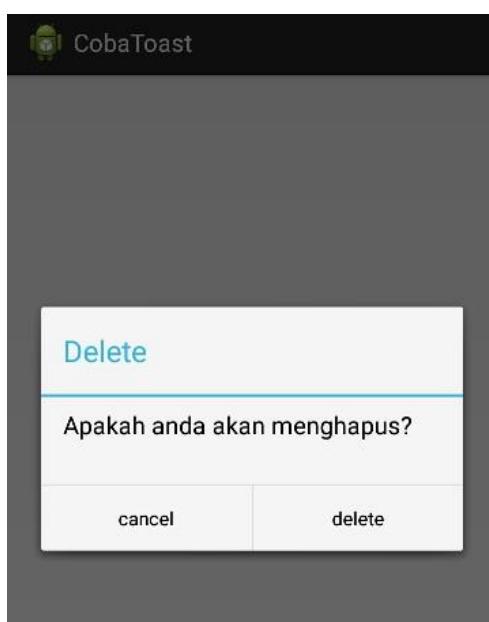
        dialog.dismiss();
    }
});

builder.setNegativeButton("cancel",
new DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface dialog,int which) {
        Toast.makeText(getApplicationContext(), "cancel di tekan",
        Toast.LENGTH_SHORT).show();
        dialog.dismiss();
    }
});

AlertDialog alert = builder.create();
alert.show();
```

Tampilan Running Programnya

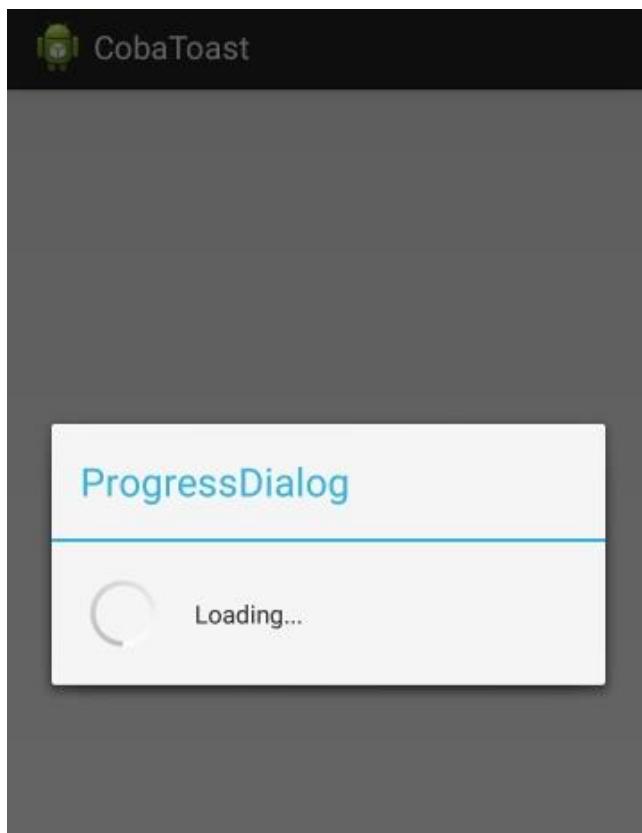


- **ProgressDialog**, digunakan untuk membuat progress/perkembangan. Misalnya untuk mengetahui progress dari download file. Terdapat dua buah style dari ProgressDialog.

Contoh implementasi style spinner

```
ProgressDialog progressDialog = new ProgressDialog(MainActivity.this);
progressDialog.setMessage("Loading..."); // Atur pesan
progressDialog.setTitle("ProgressDialog"); // Atur judul
progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER); // tipe Spinner
progressDialog.show(); // menampilkan Progress Dialog
```

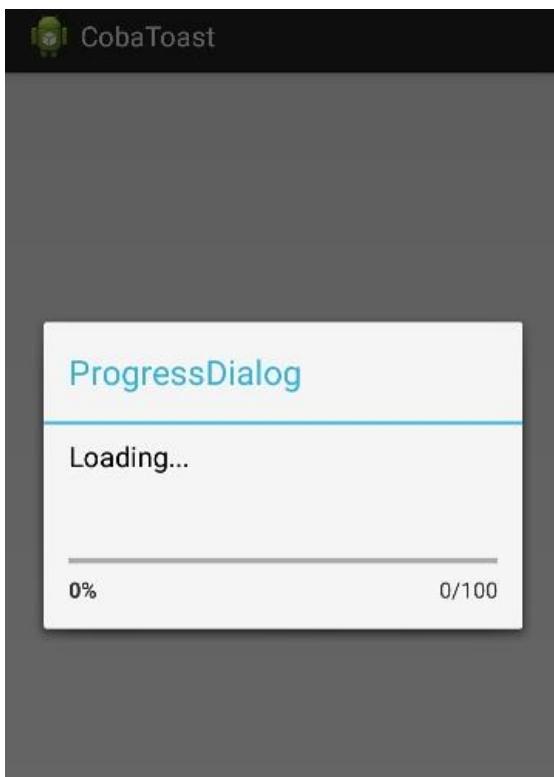
Tampilan Running Programnya



Contoh implementasi style spinner

```
ProgressDialog progressDialog = new ProgressDialog(MainActivity.this);
progressDialog.setMessage("Loading..."); // Atur pesan
progressDialog.setTitle("ProgressDialog"); // Atur judul
progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL); // tipe Horizontal
progressDialog.show(); // menampilkan Progress Dialog
```

Tampilan Running Programnya



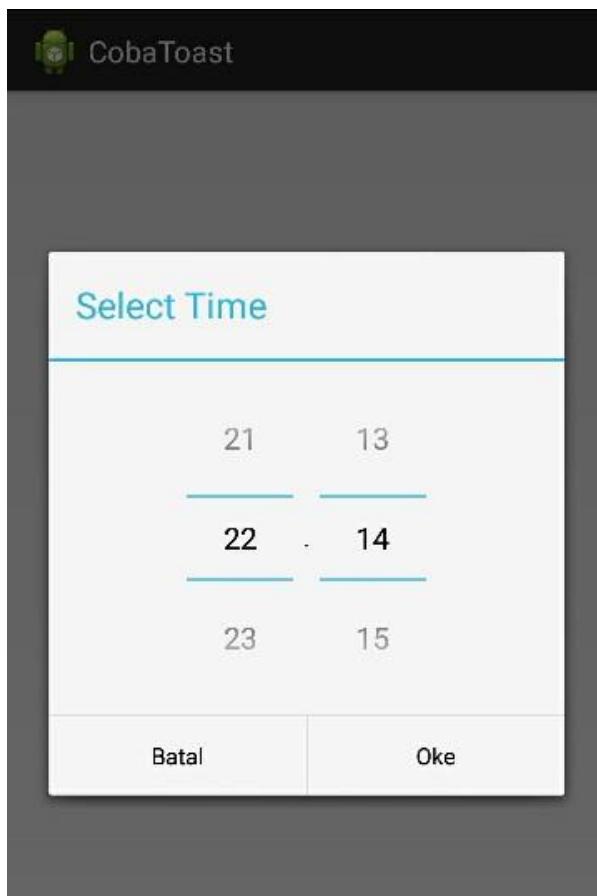
- **TimePickerDialog**, digunakan pengguna untuk memilih waktu berupa jam dan menit.

Contoh implementasi TimePickerDialog

```
Calendar mcurrentTime = Calendar.getInstance();
int hour = mcurrentTime.get(Calendar.HOUR_OF_DAY);
int minute = mcurrentTime.get(Calendar.MINUTE);
TimePickerDialog mTimePicker;
mTimePicker = new TimePickerDialog(MainActivity.this, new
TimePickerDialog.OnTimeSetListener() {

    @Override
    public void onTimeSet(TimePicker timePicker, int selectedHour, int
selectedMinute) {
        //jam dan waktu yang terpilih ditampilkan pada toast
        Toast.makeText(getApplicationContext(), selectedHour + ":" + selectedMinute,
        Toast.LENGTH_LONG).show();
    }
}, hour, minute, true);
mTimePicker.setTitle("Select Time");
mTimePicker.show();
```

Tampilan Running Programnya



- **DatePickerDialog**, digunakan pengguna untuk memilih tanggal.

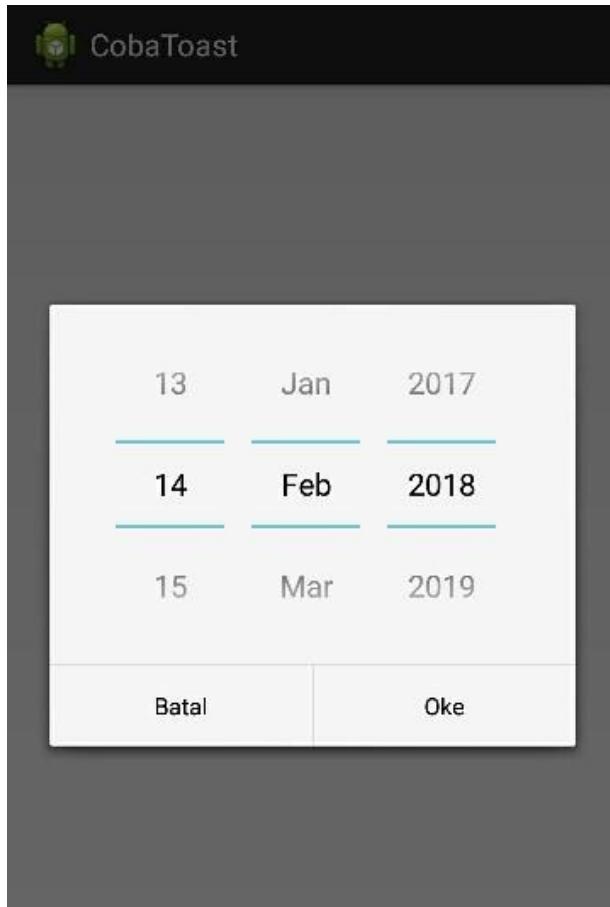
Contoh implementasi DatePickerDialog

```
final Calendar c = Calendar.getInstance();
int mYear = c.get(Calendar.YEAR); // tahun sekarang
int mMonth = c.get(Calendar.MONTH); // bulan sekarang
int mDay = c.get(Calendar.DAY_OF_MONTH); //tanggal sekarang

// date picker dialog
DatePickerDialog datePickerDialog = new DatePickerDialog(MainActivity.this,
new DatePickerDialog.OnDateSetListener() {

    @Override
    public void onDateSet(DatePicker view, int year,
    int monthOfYear, int dayOfMonth) {
        //hari bulan tahun ditampilkan dalam toast
        Toast.makeText(getApplicationContext(), dayOfMonth + "/"
        + (monthOfYear + 1) + "/" + year,
        Toast.LENGTH_LONG).show();
    }
}, mYear, mMonth, mDay);
datePickerDialog.show();
```

Tampilan Running Programnya



Mengaplikasikan alaram

Alarm digunakan dalam kehidupan sehari-hari. Alarm membantu untuk mengingatkan pengguna akan pekerjaan yang sudah direncanakan. Pada android untuk membuat Alarm digunakan class `AlarmManager`.

Contoh implementasi

Persiapan:

Dalam tutorial ini yang perlu anda persiapkan file audio (rekomendasi file audio dengan format mp3). Nama file audio diawali oleh huruf, tanpa spasi, symbol yang diperkenankan hanya underscore(_) dan lowercase (tidak ada huruf kapital).

Contoh :



Buatlah project baru dengan nama “Alarm”, dengan Activity : MainActivity dan Layout : activity_main.

kode pada layout activity_main.xml

Widget	Id	Text
TextView	textView1	My Alarm
AnalogClock	analogClock1	-
Button	startSetDialog	Set Alarm

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="My Alarm"
        android:textAppearance="?android:textAppearanceLarge" />

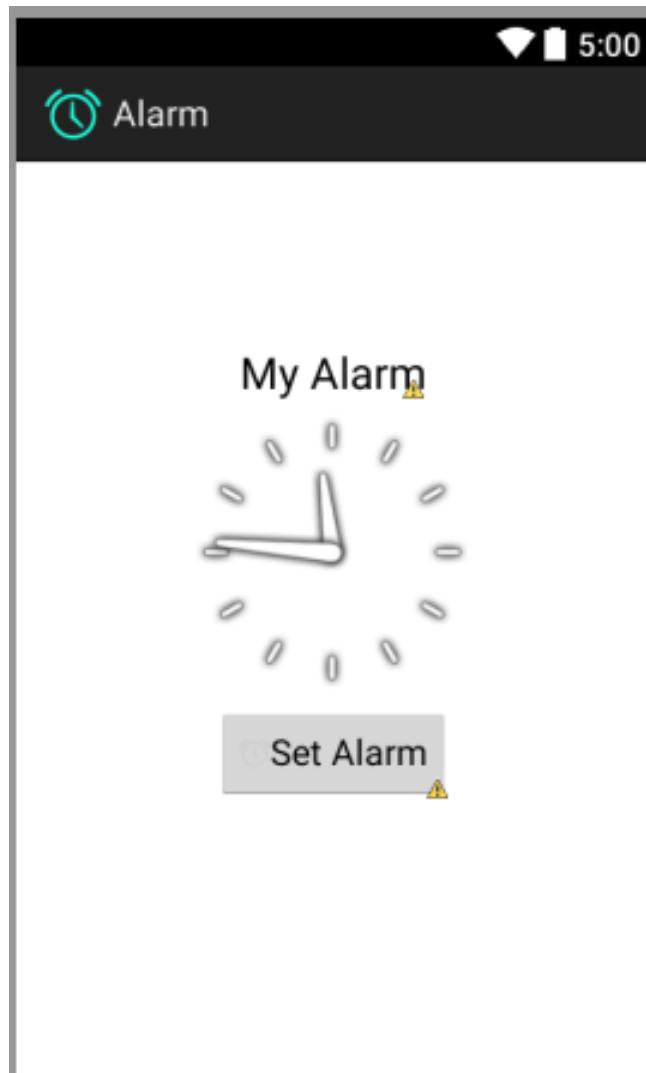
    <AnalogClock
        android:id="@+id/analogClock1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/startSetDialog"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:drawableLeft="@android:drawable/ic_lock_idle_alarm"
        android:text="Set Alarm" />

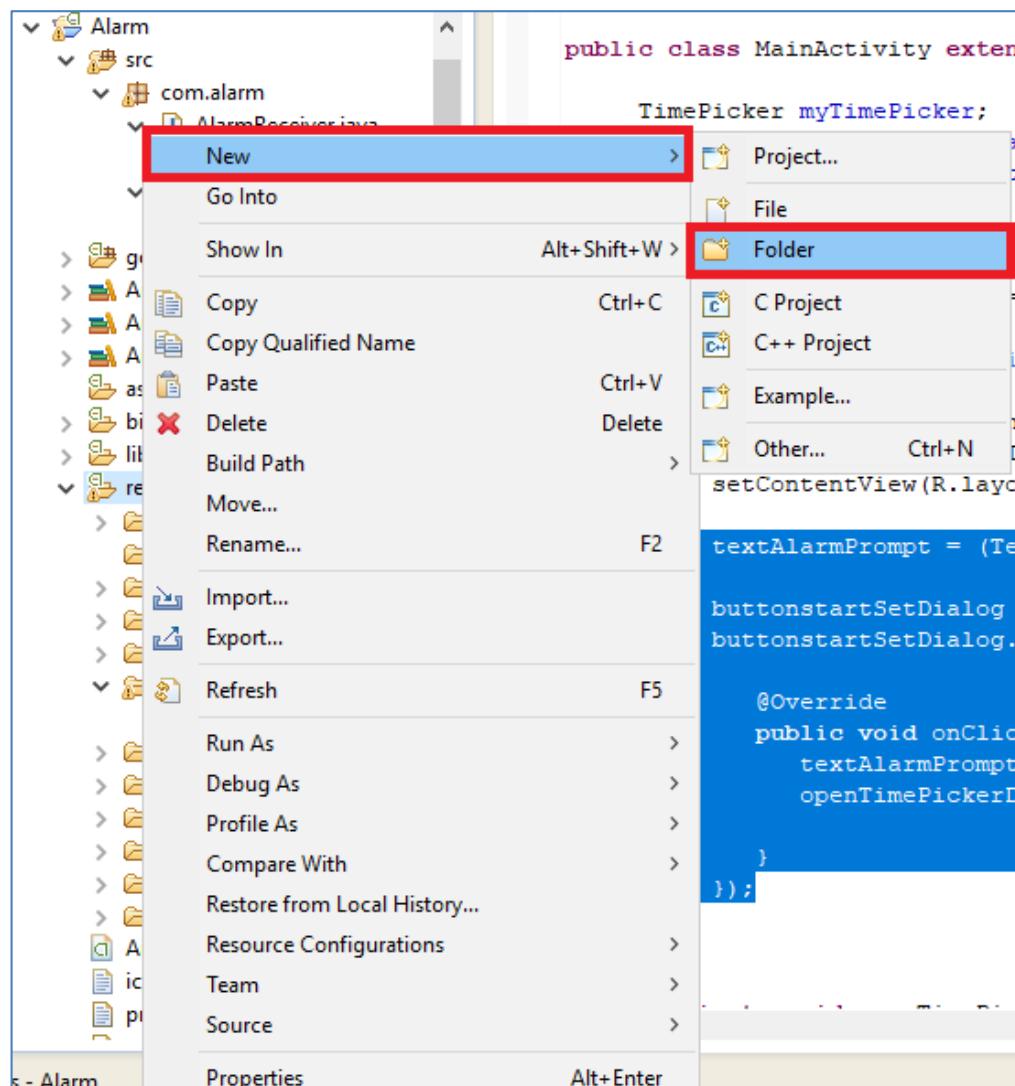
    <TextView
        android:id="@+id/alarmprompt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center" />

</LinearLayout>
```

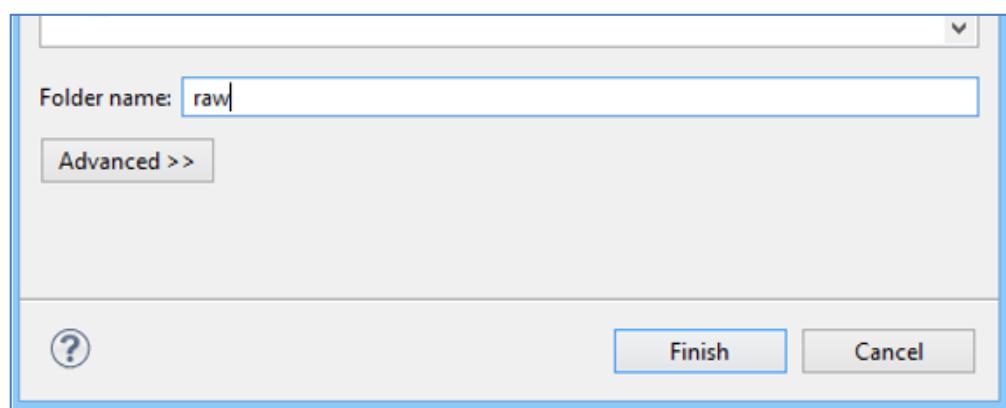
Jika sudah maka tampilan akan menjadi seperti ini



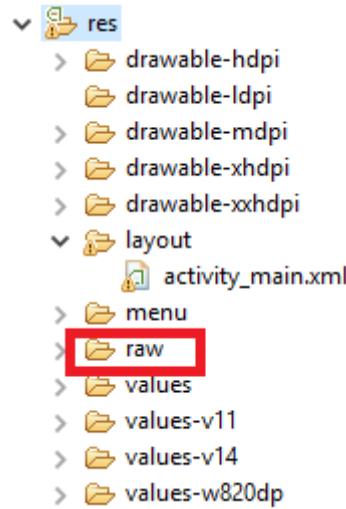
Pada activity_main.xml kita menggunakan View AnalogClock hanya untuk sekedar menampilkan waktu. Button Set Alarm digunakan untuk menentukan kapan alarm berbunyi. Langkah selanjutnya adalah membuat folder “raw” pada project. Untuk membuat folder, klik kanan pada folder res → pilih new → folder seperti gambar berikut:



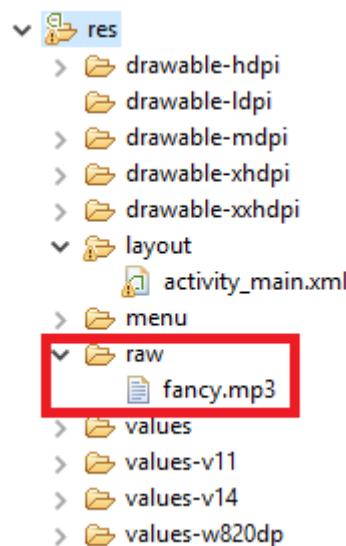
Beri nama folder “raw” kemudian klik finish.



Di folder res akan terdapat folder baru raw.



Copy file audio yang telah disiapkan ke folder raw dengan cara drag and drop.



Sekarang apabila pengguna menekan button maka akan ditampilkan TimePicker sehingga user bisa menentukan jam dan menit berapa alarm akan berbunyi. Tambahkan beberapa field berikut pada MainActivity. Field tersebut akan digunakan untuk view yang ada pada layout dan sebuah field untuk requestCode.

```
TimePicker myTimePicker;
Button buttonstartSetDialog;
TextView textAlarmPrompt;

TimePickerDialog timePickerDialog;

final static int RQS_1 = 1;
```

Pada fungsi onCreate tambahkan baris code ini

```
textAlarmPrompt = (TextView) findViewById(R.id.alarmprompt);

buttonstartSetDialog =
(Button) findViewById(R.id.startSetDialog);
buttonstartSetDialog.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
    textAlarmPrompt.setText("");
    openTimePickerDialog(false);

}
});
```

Pada saat pengguna menekan button Set Alarm maka aplikasi akan menampilkan TimePicker dengan fungsi berikut. Tambahkan baris code ini ke MainActivity.java.

```
private void openTimePickerDialog(boolean is24r) {
    Calendar calendar = Calendar.getInstance();

    timePickerDialog = new TimePickerDialog(MainActivity.this,
        onTimeSetListener, calendar.get(Calendar.HOUR_OF_DAY),
        calendar.get(Calendar.MINUTE), true);
    timePickerDialog.setTitle("Set Alarm Time");

    timePickerDialog.show();
}
```

Jika waktu sudah diset maka kita memerlukan suatu listener lagi yaitu onTimeSetListener. Beikut baris codenya.

```
OnTimeSetListener onTimeSetListener = new OnTimeSetListener() {

@Override
public void onTimeSet(TimePicker view, int hourOfDay, int
minute) {

    Calendar calNow = Calendar.getInstance();
    Calendar calSet = (Calendar) calNow.clone();

    calSet.set(Calendar.HOUR_OF_DAY, hourOfDay);
    calSet.set(Calendar.MINUTE, minute);
    calSet.set(Calendar.SECOND, 0);
    calSet.set(Calendar.MILLISECOND, 0);

    if (calSet.compareTo(calNow) <= 0) {
        // Today Set time passed, count to tomorrow
        calSet.add(Calendar.DATE, 1);
        Log.i("hasil", "=<0");
    } else if (calSet.compareTo(calNow) > 0) {
        Log.i("hasil", "> 0");
    } else {
        Log.i("hasil", "else ");
    }
}}
```

```

    }
    setAlarm(calSet);
}
;

```

Kode berikut untuk mengirim waktu yang sudah ditentukan ke receiver alarm.

```

private void setAlarm(Calendar targetCal) {

    textAlarmPrompt.setText("***\n" + "Alarm set on " +
targetCal.getTime()
        + "\n***");

    Intent intent = new
Intent(getApplicationContext(),AlarmReceiver.class);
    PendingIntent pendingIntent = PendingIntent.getBroadcast(
getBaseContext(), RQS_1, intent, 0);

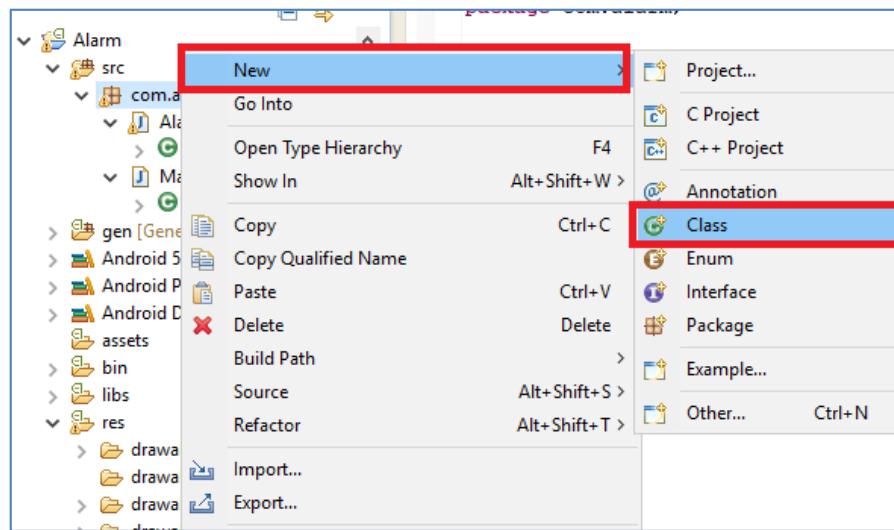
    AlarmManager alarmManager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);

    alarmManager.set(AlarmManager.RTC_WAKEUP,
targetCal.getTimeInMillis(),
pendingIntent);

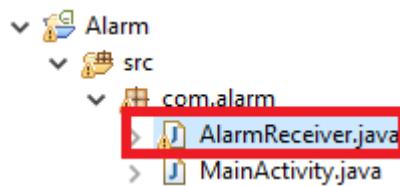
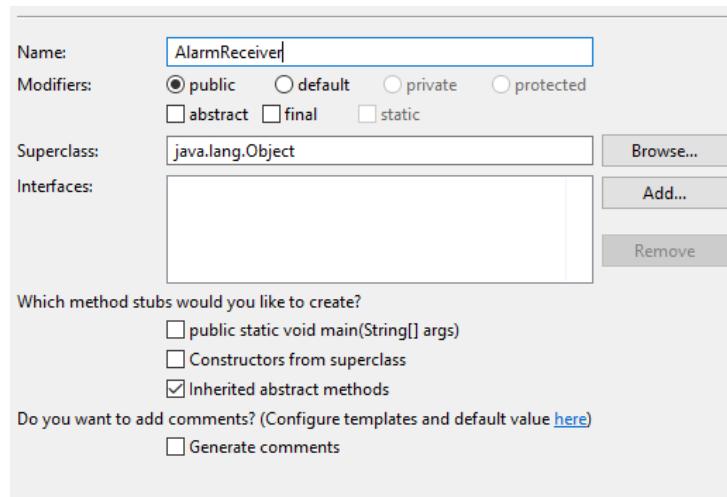
}

```

Selanjutnya membuat sebuah class baru bernama AlarmReceiver.java. Class ini akan bekerja sesuai namanya yaitu sebagai receiver. Caranya klik kanan pada package com.alarm New →Class.



Beri nama AlarmReceiver kemudian klik finish.



Tambahkan baris code berikut

```
package com.alarm;

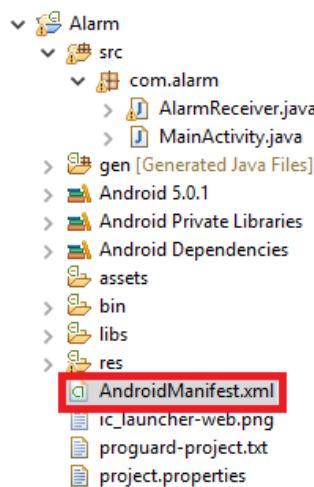
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.view.View;
import android.widget.Toast;

public class AlarmReceiver extends BroadcastReceiver {
    MediaPlayer player;

    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO Auto-generated method stub
        Toast.makeText(context, "Alarm aktif!",
        Toast.LENGTH_LONG).show();
        player = MediaPlayer.create(context, R.raw.fancy);
        player.start();

    }
}
```

Ubah AndroidManifest.xml dan tambahkan receiver pada application seperti baris code di bawah ini.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.alarm"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="21" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.alarm.MainActivity"
            android:label="Alarm" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
            />
            </intent-filter>
        </activity>

        <receiver
            android:name=".AlarmReceiver"
            android:process=":remote" />
    </application>

</manifest>
```

FULL SOURCE CODE

MainActivity.java

```

package com.alarm;

import java.util.Calendar;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.app.TimePickerDialog;
import android.app.TimePickerDialog.OnTimeSetListener;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.TimePicker;

public class MainActivity extends Activity {

    TimePicker myTimePicker;
    Button buttonstartSetDialog;
    TextView textAlarmPrompt;

    TimePickerDialog timePickerDialog;

    final static int RQS_1 = 1;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textAlarmPrompt = (TextView) findViewById(R.id.alarmprompt);

        buttonstartSetDialog =
        (Button) findViewById(R.id.startSetDialog);
        buttonstartSetDialog.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                textAlarmPrompt.setText("");
                openTimePickerDialog(false);
            }
        });
    }

    private void openTimePickerDialog(boolean is24r) {
        Calendar calendar = Calendar.getInstance();
    }
}

```

```
timePickerDialog = new TimePickerDialog(MainActivity.this,
    onTimeSetListener, calendar.get(Calendar.HOUR_OF_DAY),
    calendar.get(Calendar.MINUTE), true);
timePickerDialog.setTitle("Set Alarm Time");

timePickerDialog.show();

}

OnTimeSetListener onTimeSetListener = new OnTimeSetListener() {

    @Override
    public void onTimeSet(TimePicker view, int hourOfDay, int
minute) {

        Calendar calNow = Calendar.getInstance();
        Calendar calSet = (Calendar) calNow.clone();

        calSet.set(Calendar.HOUR_OF_DAY, hourOfDay);
        calSet.set(Calendar.MINUTE, minute);
        calSet.set(Calendar.SECOND, 0);
        calSet.set(Calendar.MILLISECOND, 0);

        if (calSet.compareTo(calNow) <= 0) {
            // Today Set time passed, count to tomorrow
            calSet.add(Calendar.DATE, 1);
            Log.i("hasil", " <=0");
        } else if (calSet.compareTo(calNow) > 0) {
            Log.i("hasil", " > 0");
        } else {
            Log.i("hasil", " else ");
        }

        setAlarm(calSet);
    }
};

private void setAlarm(Calendar targetCal) {

    textAlarmPrompt.setText("***\n" + "Alarm set on " +
targetCal.getTime()
        + "\n***");

    Intent intent = new Intent(getApplicationContext(),
    AlarmReceiver.class);

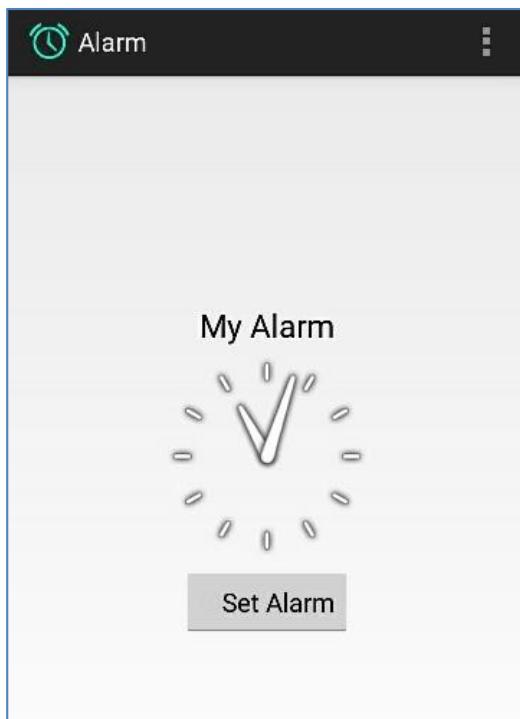
    PendingIntent pendingIntent = PendingIntent.getBroadcast(
        getBaseContext(), RQS_1, intent, 0);

    AlarmManager alarmManager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);

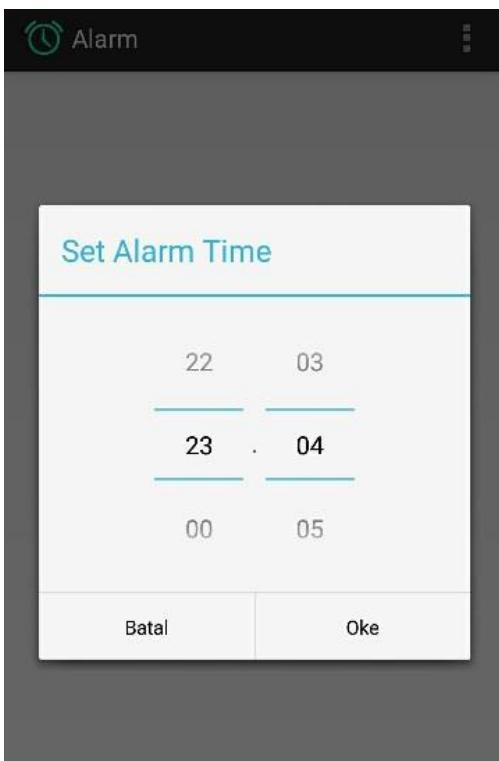
    alarmManager.set(AlarmManager.RTC_WAKEUP,
    targetCal.getTimeInMillis(),
    pendingIntent);

}
}
```

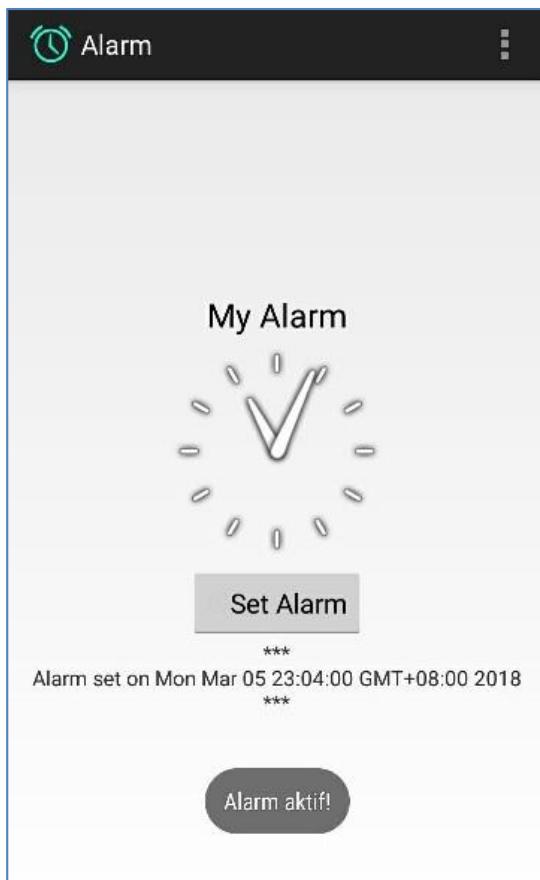
Tampilan saat program dirunning.



Tampilan program saat memilih waktu alarm



Tampilan program saat alarm aktif



Tutorial diatas untuk membuat alarm yang berbunyi sekali. Terkadang pengguna ingin alarm berbunyi berulang-ulang sesuai waktu interval yang ditentukan. Jika sebelumnya menggunakan kode berikut

```
alarmManager.set(AlarmManager.RTC_WAKEUP,  
targetCal.getTimeInMillis(),pendingIntent);
```

Maka pada alarm berulang menjadi

```
alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,  
targetCal.getTimeInMillis()+(5*1000), 10 * 1000,pendingIntent);
```

Latihan

Membuat aplikasi alarm.

MODUL 11

MENGGUNAKAN LOCATION-BASED SERVICES

Tujuan

- Mahasiswa mampu mengidentifikasi Location-based services.
- Mahasiswa mampu memilih lokasi dari provider.
- Mahasiswa mampu mengimplementasikan location-based services pada emulator.
- Mahasiswa mampu membuat Map-based App.

Mengidentifikasi Layanan Berbasis Lokasi

LBS adalah layanan informasi yang dapat diakses di perangkat bergerak dengan internet melalui layanan data jaringan, seperti *General Packet Radio Service* (GPRS), atau layanan WI-FI internet. *LBS* penyedia informasi tentang lokasi perangkat saat ini.

Untuk menyediakan LBS, perangkat harus menentukan lokasi secara akurat sehingga dapat membantu dengan informasi yang benar. Untuk mengidentifikasi lokasi, perangkat seluler dapat menggunakan berbagai penyedia lokasi, seperti berikut:

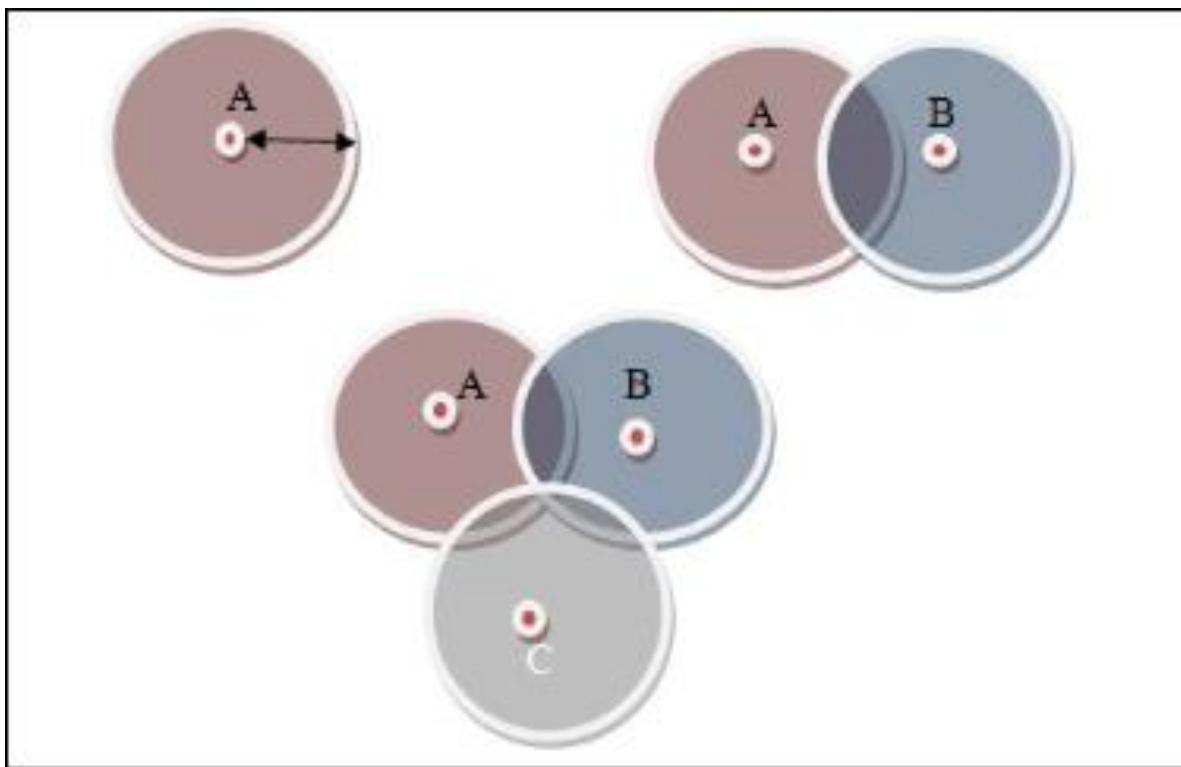
1. Global Positioning System (GPS)
2. Call tower triangulation
3. Public Wireless Fidelity (WI-FI) hotspots

GPS

GPS adalah sistem navigasi berbasis satelit dan menggunakan jaringan satelit untuk memberikan informasi yang berkaitan dengan lokasi. Menggunakan satelit untuk melacak posisi benda di Bumi. Penerima GPS atau perangkat berkemampuan GPS menempatkan tiga atau lebih satelit ini menentukan jarak ke masing-masing, dan menggunakan informasi ini untuk menyimpulkan lokasinya sendiri, orang akan menghitung posisi benda dalam ruang 2-D dan juga 3-D dengan menggunakan prinsip matematika disebut trilateration. Trilateration ada dua jenis, 2-D dan juga 3-D.

2-D Trilateration

Berikut adalah gambar proses 2-D Trilateration :



Proses 2-D Trilateration

3-D Trilateration

Berikut adalah gambar proses 3-D Trilateration :



Proses 3-D Trilateration

Beberapa penggunaan umum GPS adalah :

1. GPS dapat digunakan untuk menavigasi mobil Anda melalui traffic dan menemukan jalan ke tujuan manapun.
2. Penerima Laptop GPS dapat digunakan untuk pertemuan di dalam kendaraan, kantor berdasarkan lokasi yang dilacak.
3. GPS yang dilengkapi perangkat mobile dapat mengirimkan lokasi yang tepat ke penerima panggilan darurat. Tahan lokasi yang cepat dan akurat daripada mengandalkan deskripsi orang-orang yang tidak terbiasa dengan daerah tersebut atau terlalu bingung untuk menjelaskan lokasi mereka.
4. GPS sering digunakan oleh pejalan kaki, pemburu, mobil salju, pengendara sepeda gunung, dan pemain ski.
5. GPS di gunakan pada transportasi umum, seperti penerbangan, rel, bus, dan taksi, membantu komuter untuk melacak dan menjadwalkan secara real time.
6. GPS adalah sumber penting bagi unit layanan kesehatan polisi, pemadam kebakaran, dan darurat. Anda penerima untuk mengetahui apakah patroli polisi, sebuah ambulans dekat Respon cepat yang memungkinkan ini dalam situasi life-or-death.
7. GPS yang muncul dapat digunakan untuk melindungi orang yang Anda sayangi dan dekat dengan melacak keberadaan mereka. Anda juga dapat menggunakan.
8. GPS untuk perlindungan aset. Misalnya, GPS dapat digunakan di kendaraan.

Cell Tower Trilateration

Cell Tower Trilateration adalah teknik lain untuk mendapatkan posisi saat ini dari sebuah aplikasi mobile apakah stasioner atau bergerak. Jaringan adalah jaringan radio yang tersebar di wilayah darat pada sel heksagonal berbentuk sarang lebah. Setiap sel memiliki satu transceiver lokasi tetap yang dikenal sebagai menara sel atau base station, yang letak geografinya dan terlihat. Bersama-sama, sel-sel ini memberikan cakupan sinyal yang tersebar di wilayah geografis atau negara yang luas. Menara sel memungkinkan perangkat mobile seperti telepon seluler untuk berkomunikasi dengan transceiver atau jaringan telepon di manapun, bahkan jika beberapa transceiver bergerak lebih dari satu sel selama transmisi berlangsung.

Public Wi-Fi Hotspots

Merupakan lokasi penambahan untuk triangulasi menara umum dan triangulasi sinar matahari, hotspot wi-Fi publik dapat digunakan. Untuk mengetahui yang terlemah, misalnya di dalam ruangan Teknik ini perfo situation, perangkat bisa digunakan untuk menentukan lokasi lokasinya saat ini mungkin tidak seperti yang disediakan oleh GPS.

tentang hotspot Wi-Fi dipasang Data yang dikumpulkan oleh siaran televisi yang disiarkan secara publik -Fi dalam jangkauan kendaraan. penerima radio digunakan untuk mengkompilasi database LBS.

Informasi berikut digunakan untuk membangun database:

1. dua alamat Media Access Control (MAC) dari hotspot.
2. Koordinat GP kendaraan pada titik saat hotspot terlihat Untuk menentukan lokasi perangkat,

Proses berikut adalah alamat yang digunakan pada perangkat aplikasi :

1. Saat ini terlihat oleh Perangkat akan mengirimkan permintaan ke server lokasi dengan daftar perangkat MAC.
2. Server membandingkan daftar alamat dengan daftar alamat MAC yang diketahui Lokasi alamat lokasi geografis yang sesuai. hotspot untuk identitas yang cocok dengan MAC dan
3. Server lokasi menentukan perkiraan lokasi perangkat dengan menggunakan lokasi geografi dari alamat MAC yang terlihat untuk melakukan triangulasi perkiraan lokasi pengguna

Untuk mempertahankan database yang akurat, penyedia layanan harus terus memperbarui katalog hotspot yang melakukan penyapuan berkala di suatu daerah o mengumpulkan informasi tentang hotspot Wi-Fi. Jika tidak, akurasi teknologi mungkin akan terpengaruh karena dinamika perubahan hotspot yang bekerja.

Bekerja dengan Location-Based Service

SDK Android menyediakan API yang diperlukan untuk mengambil data lokasi berdasarkan beragam termasuk: sensor GP, triangulasi menara sel, dan Wi-Fi hotspot Setelah lokasi Anda menghalangi, kemudian dapat bekerja sama dengan API lain, seperti Google Maps API, untuk membuat peta yang intuitif dan berguna. yang dapat menampilkan peta secara otomatis dan menentukan lokasi Anda saat ini.

Ada dua kelas utama yang digunakan dalam LBS :

1. *LocationManager*: Menyediakan akses ke layanan lokasi perangkat. Dengan menggunakan pengelola lokasi, aplikasi yang dapat:
 - A. Dapatkan pembaruan berkala pada lokasi perangkat saat ini.
 - B. Tetapkan tanda kedekatan untuk mendeteksi pergerakan ke dan dari lokasi geografis yang ditentukan.
2. *LocationProvider*: Apakah kelas dasar untuk semua jenis penyedia lokasi. Penyedia Lokasi memberikan pembaruan berkala tentang lokasi perangkat.

Beberapa kelas lain yang digunakan untuk mengimplementasikan LBS dijelaskan pada tabel berikut.

Tabel Implementasi LBS

Class	Description
Address	Kelas yang mewakili alamat, yang merupakan sekumpulan string yang menggambarkan lokasi. kelas ini memungkinkan Anda untuk melampirkan informasi tambahan dengan alamat, seperti nomor telepon, lokalitas, dan kode pos.
Criteria	Kelas yang menunjukkan kriteria untuk memilih penyedia lokasi yang sesuai. Anda dapat menentukan kriteria, seperti akurasi, penggunaan daya, dan kemampuan untuk menentukan ketinggian, kecepatan, dan biaya yang terkait.
Geocoder	Sebuah kelas untuk menangani geo-coding. kelas ini memungkinkan Anda untuk mengubah alamat jalan menjadi lintang dan bujur melalui pengkodean geo dan mengubah garis lintang dan bujur menjadi alamat apah melalui geo-coding terbalik.
GpsSatellite	Kelas ini mewakili keadaan satelit GPS saat ini. kelas ini memungkinkan Anda mendapatkan rincian satelit seperti posisi, dan Pseudo-random Number (PRN) untuk satelit. satelit diidentifikasi oleh penerima GPS dengan menggunakan nomor PRN yang lucu atau unik.
Location	Kelas yang mewakili lokasi geografis terasa pada waktu tertentu.

Mengakses Location-Based Services

LBS Mengakses Layanan Berbasis Lokasi Aplikasi berkemampuan LBS mengekspos informasi pribadi dan lokasi pengguna dan dapat dikenai pelanggaran privasi. Selain itu, dengan menggunakan LBS mengkonsumsi sejumlah besar penyimpanan data jaringan dan baterai dari dev ke pengalaman pengguna, Anda dapat meminta izin yang diperlukan dari pengguna akhir untuk memastikan penggunaan LBS yang baik.

Untuk mengakses LBs, gunakan layanan sistem LocationManager dari platform Android. Untuk mengakses layanan Location Manager, Anda perlu mendapatkan instance dari layanan *LOCATION_SERVICE* dengan menggunakan *getSystemService()* seperti yang ditunjukkan pada potongan kode berikut:

```
String serviceString = Context.LOCATION_SERVICE;  
  
LocationManager LocationManager;  
  
LocationManager = (LocationManager) getSystemService (serviceString);
```

Aplikasi tidak dapat menggunakan LBS kecuali mereka menyertakan izin yang diperlukan dalam *AndroidManifest*.

Izin umum yang digunakan oleh aplikasi untuk mengakses LBS adalah:

- A. *android.permission.ACCESS_FINE_LOCATION*: Izin ini digunakan untuk menerima lokasi dari penyedia GPS.
- B. *Android.permission.ACCESS_COARSE_LOCATION*: Izin ini digunakan untuk menerima pengaturan lokasi dari operator jaringan bergerak.

GPS menggunakan izin bagus dan penyedia jaringan seluler hanya menggunakan aplikasi berikut bahwa izin memiliki izin kasar secara implisit. menggunakan code pada file AndroidManifest.xml:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Selain menyetel izin, Anda harus mengikuti dan menerapkan beberapa panduan tentang aplikasi berkemampuan LBS Anda, seperti:

- A. informasikan kepada pengguna sebelum mengumpulkan data pribadi mereka, karena informasi tentang lokasi mereka sekarang atau masa lalu mungkin sensitif.
- B. Izinkan pengguna untuk mengonfigurasi fitur nonaktifkan yang dapat mempengaruhi pengalaman mereka bila menggunakan aplikasi berbasis LBS, seperti opsi untuk menonaktifkan dan mengaktifkan penerima GPS atau opsi untuk membatasi pembagian informasi lokasi saat ini melalui jaringan.
- C. Tangani acara lainnya, seperti baterai rendah, atau peringatan lainnya di aplikasi Anda dan beri tahu pengguna tentang kejadian ini.

Memilih Lokasi Provider

Mencari Lokasi Providers

kelas *LocationManager* menyediakan akses ke layanan lokasi sistem, yang memungkinkan apl memperoleh pembaruan berkala dari lokasi geografis perangkat. kelas ini juga memungkinkan untuk *Intent* maksud yang ditentukan aplikasi saat perangkat memasuki kedekatan lokasi geografis tertentu. kelas *LocationManager* juga menyertakan konstanta *String* statis yang menyebutkan nama penyedia layanan commonly-used. tabel berikut mencantumkan dua penyedia commonly-used.

Tabel Commonly-Used

Name	Description
<i>LocationManager.GPS_PROVIDER</i>	Merupakan nama penyedia GPS. Penyedia ini menggunakan GPS untuk data lokasi operator. Penyedia ini membutuhkan izin <i>android.permission.ACCESS_FINE_LOCATION</i>
<i>LocationManager.NETWORK_PROVIDER</i>	Merupakan nama penyedia lokasi jaringan. Penyedia ini menggunakan menara panggilan dan WI-FI

	untuk menentukan lokasi Hasil diambil dengan menggunakan pencarian jaringan. Penyedia ini membutuhkan izin <i>android.permission.ACCESS_COARSE_LOCATION</i> Atau <i>android.permission.ACCESS_FINE_LOCATION</i>
--	--

Anda juga dapat menentukan penyedia lokasi yang tersedia dengan menggunakan metode *getProviders()*. Metode ini mengembalikan daftar nama penyedia yang diizinkan untuk diakses oleh aktivitas pemanggil. Metode ini hanya membutuhkan satu parameter boolean, yang jika benar, hanya mengembalikan penyedia yang saat ini diaktifkan.

Penggunaan metode ini ditunjukkan dalam cuplikan kode berikut:

```
Boolean enabledOnly = true;  
  
List<String> providers = LocationManager.getProviders (enabledOnly);
```

Cuplikan kode berikut membantu mengidentifikasi lokasi penyedia lokasi saat ini:

```
Private LocationManager locationManager;  
  
List<String> providers = locationManager.getProviders (true);  
  
For (String provider : provider) {  
  
    Location location = locationManager.getLastKnownLocation (provider) ;  
  
    If (location != Null ){  
  
        Toast.makeText (this, "Latidue: "  
                    +Location.getLatitude() + "\nLongitude: "  
                    +location.getLatitude(),Toast.LENGTH_SHORT);  
  
    }  
}
```

Mengimplementasikan Location-Based Services Pada Emulator.

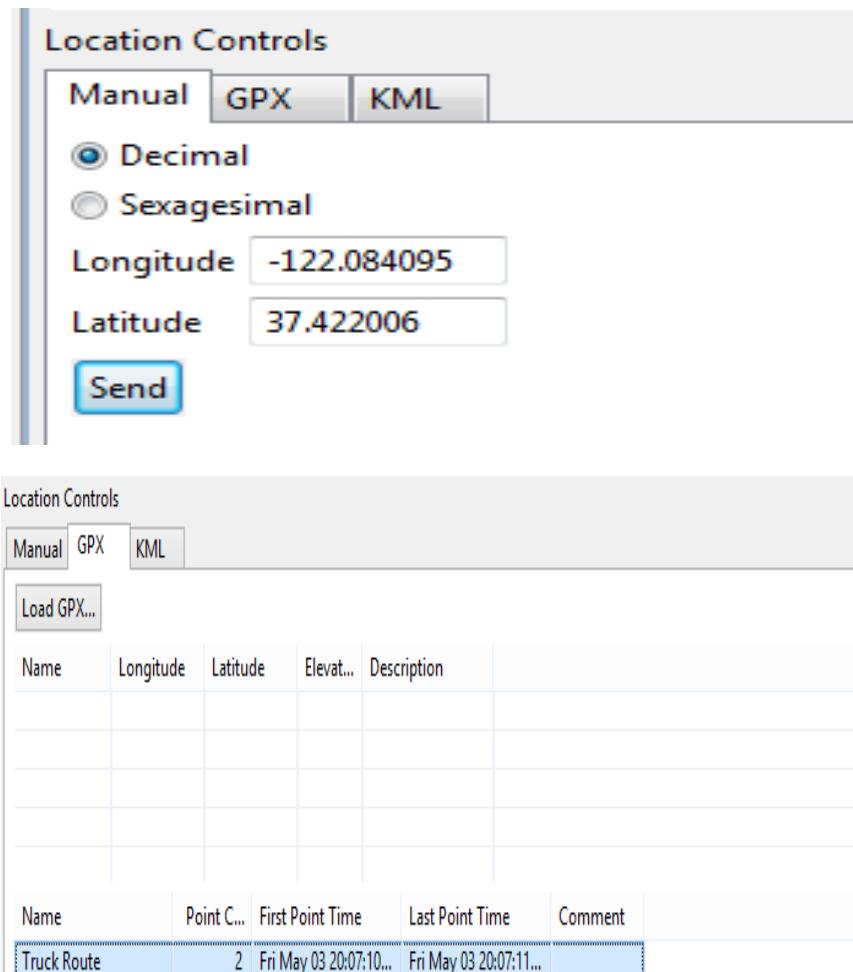
Menggunakan DDMS Untuk Simulasi Perangkat GPS

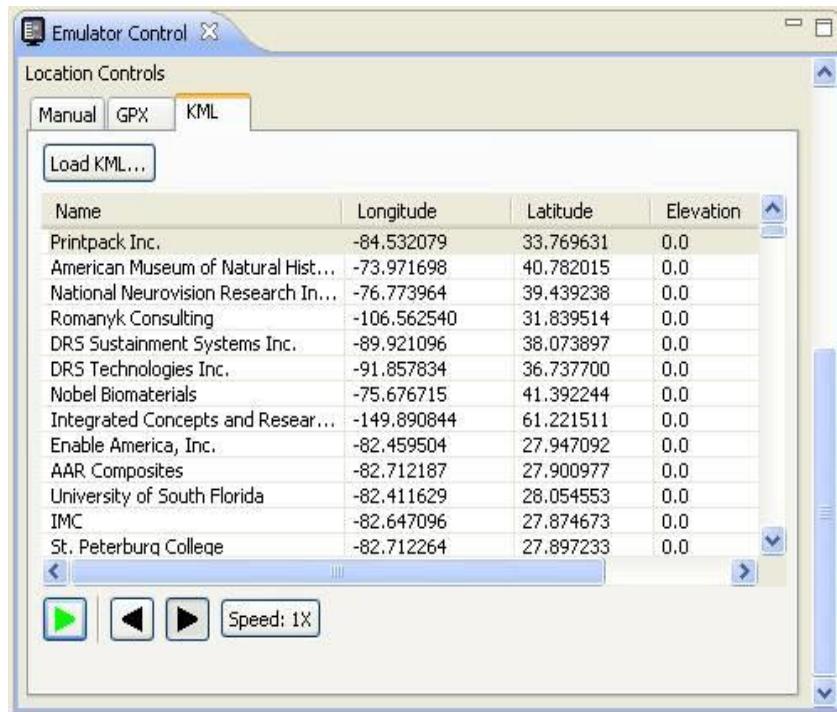
Menggunakan kontrol lokasi yang tersedia dalam perspektif DDMS untuk memasukkan perubahan lokasi. dir *GPS_PROVIDER* ke konstan untuk menguji berbagai aspek fitur spesifik lokasi aplikasi dengan bergerak secara fisik.

Ada tiga jenis kontrol lokasi yang tersedia di emulator adalah :

1. Manual
2. Marks Language Markup Language (GPX)
3. Keyhole Markup Language (KML)

Tiga jenis kontrol lokasi sebelumnya tersedia karena tiga tab di lokasi mengendalikan sektor jendela Kontrol Emulator dalam perspektif DDMS. Seperti yang ditunjukkan pada gambar berikut.



*Location Control Tabs*

Manual digunakan untuk menentukan pasangan lintang / bujur tertentu. Di sisi lain, laboratorium KML digunakan untuk memuat file KML dan GPX masing-masing. Setelah dimuat, kemudian dapat memindahkan lokasi tertentu atau memutar ulang setiap lokasi secara berurutan.

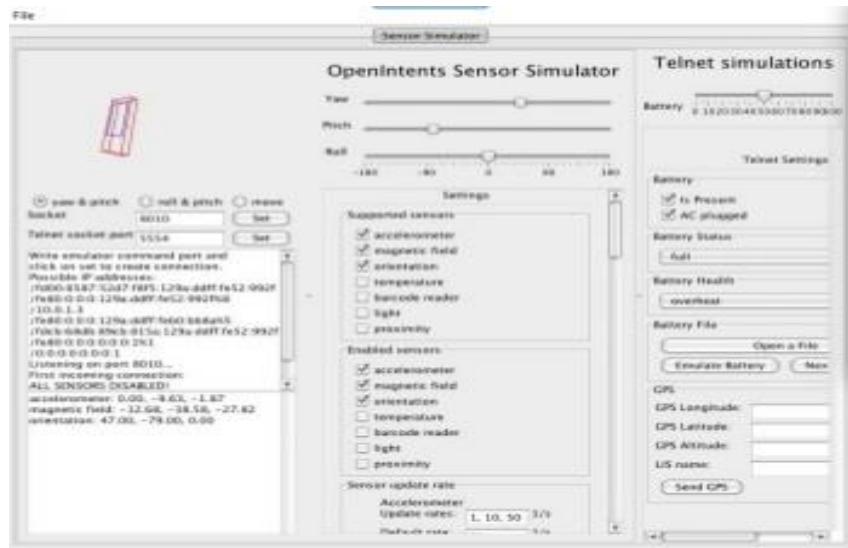
Sebagian besar sistem GPS menggunakan GPX untuk merekam file, sedangkan KML digunakan secara ekstensif secara online untuk menentukan informasi geografis. Bahkan bisa membuat file KML sendiri atau menghasilkan secara otomatis.

Menggunakan Sensor Simulator

Simulators mensimulasikan data sensor secara real time dan menyediakan model visual perangkat bersama dengan berbagai GPS, dan pengaturan sensor data orientasi. Ini mensimulasikan data sensor dari accelerometer, kompas, posisi dan Pengguna dapat melihat pembacaan sensor di bawah berbagai kondisi dengan memodifikasi data. kekuatan yang bekerja pada perangkat. dapat digunakan untuk memberikan aplikasi secara real-time.

Beberapa contoh populer dari sensor mulator yang dapat bekerja di lingkungan pengembangan Android simulator sensor Samsung dan simulator sensor OpenIntents.

Simulator ini dapat mensimulasikan informasi GPS umum secara real time. Simulator sensor seperti garis lintang, bujur, dan ketinggian dengan mana Anda dapat mensimulasikan lokasi di ap LBS yang ditempatkan pada emulator. Seperti yang ditunjukkan pada gambar berikut.

*OpenIntents Sensor Simulator*

Membuat Map-Based Apps

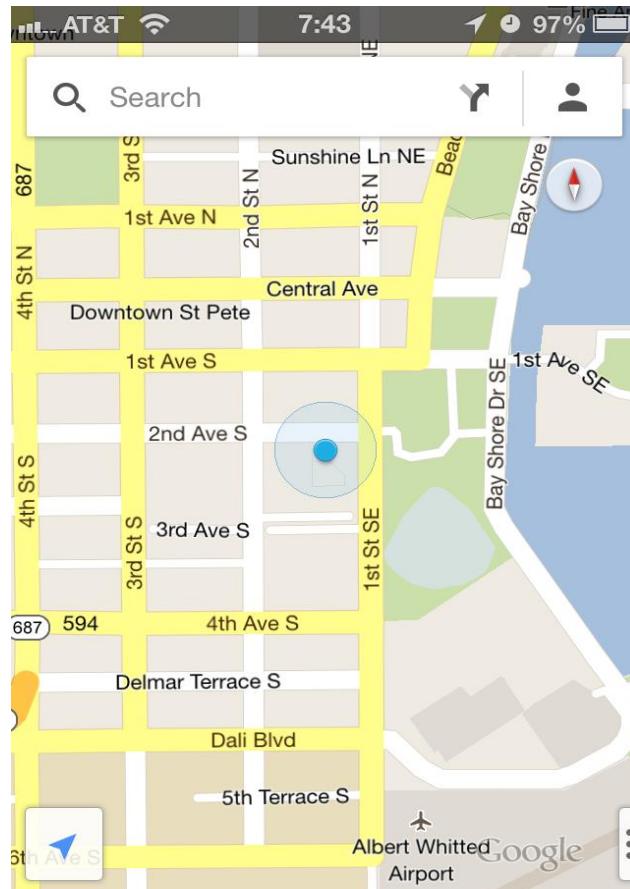
Menggunakan Class Map Fragment

Menambahkan peta ke aplikasi dengan menggunakan API Google Maps, yang didasarkan pada data Google maps. API ini menangani berbagai tugas, seperti akses ke server Google Maps, pengunduhan data, tampilan peta, dan gerakan isyarat pada peta.

Pada klas ini `com.google.android.gms.maps.MapFragment` menyediakan semua elemen UI yang diperlukan bagi pengguna untuk mengontrol peta. fragmen peta ditambahkan ke file tata letak aktivitas, seperti yang ditunjukkan pada cuplikan kode berikut:

```
<fragment
    android:id="@+id/map"
    android:layout_below="@+id/header"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
/>>
```

Ini adalah gambar simpel dari google map.



Simple Map View

Membuat Map-Based Activity

untuk menggunakan peta di aplikasi , Perlu membuat aktivitas baru yang memperluas kelas *Aktivity*. Perpustakaan peta Android bukanlah *package* standar dan oleh karena itu, perpustakaan tersebut harus disertakan secara eksplisit dalam file manifes aplikasi sebelum menggunakan.

Google Maps mendownload peta dari Internet dan juga memerlukan pengaturan izin untuk menggunakan file *AndroidManifest.xml* yang sama, seperti yang ditunjukkan pada cuplikan kode berikut.

```
<permission
    android:name="com.example.map.permission.MAPS_RECEIVE"
    android:protectionLevel="signature"/>
<uses-permission
    android:name="com.example.map.permission.MAPS_RECEIVE"/>
<uses-permission
    android:name="android.permission.INTERNET"/>
<uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission
```

```

<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  
```

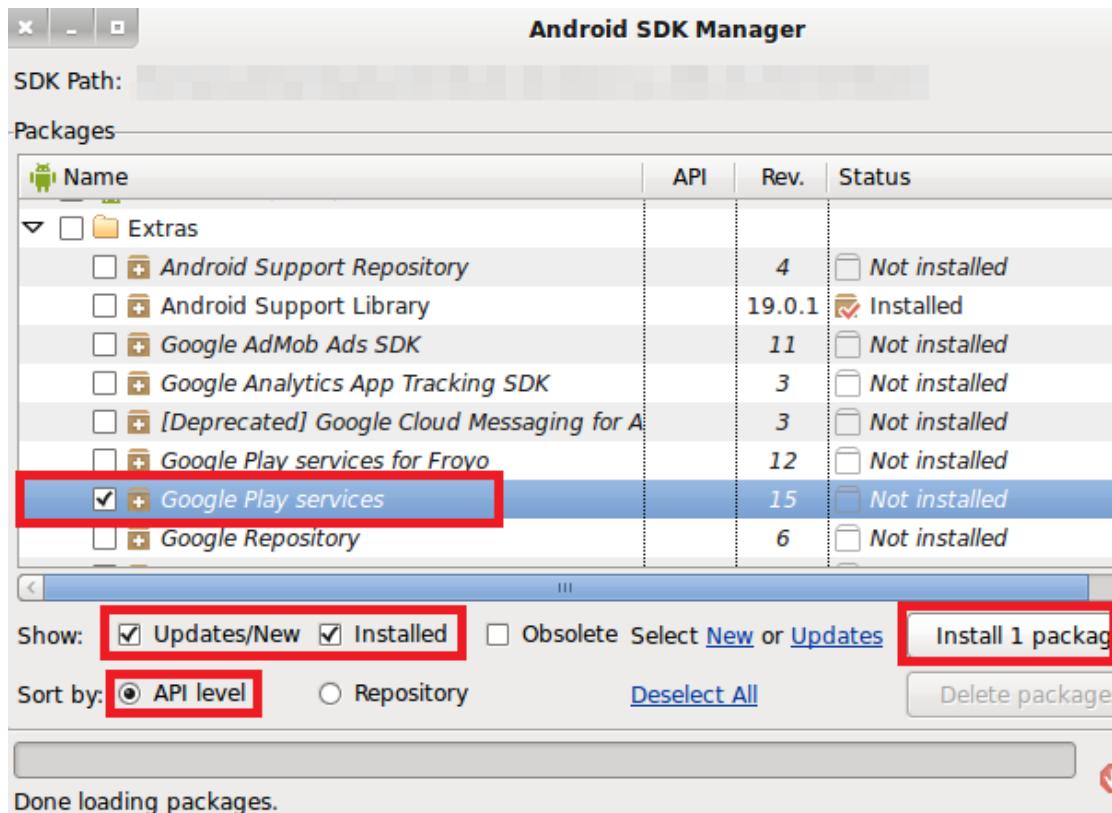
Setelah menambahkan perpustakaan dan mengkonfigurasi izin, Dapat mulai membuat activity berbasis peta baru.

Membuat Aplikasi Berbasis Peta

1. Download Layanan Google Play.
2. Buat proyek aplikasi Android baru.
3. Buat UI untuk aplikasi.
4. Buat kunci API untuk aplikasi.
5. Mengubah File Manifest Android.

Tahap 1: Mendownload Layanan Google Play

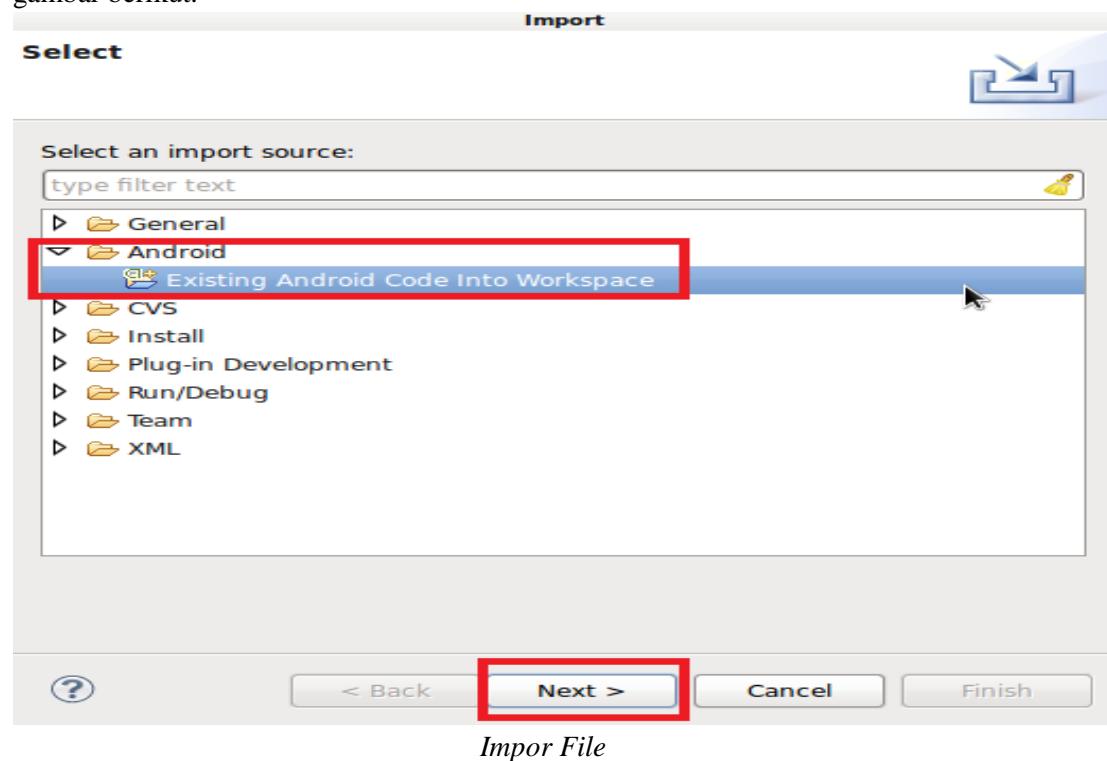
1. Buka *Eclipse IDE*.
2. Pilih *Window > Android SDK Manager*. Kemudian click tombol *Install 1 package*. Maka Download akan dimulai. Seperti yang ditunjukkan pada gambar berikut.



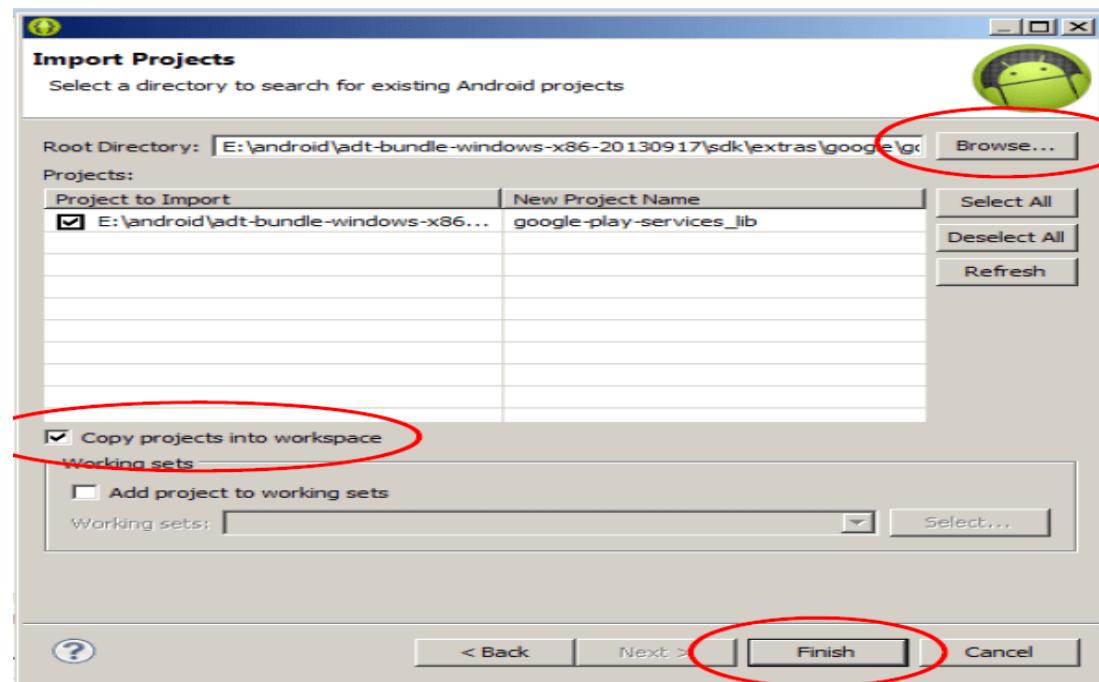
Android SDK Manager

3. Setelah proses download selesai dan layanan Google Play terinstal, tutup jendela Android SDK Manager

4. Kemudian kembali ke *Eclipse IDE*. Pilih *File > Impor*. Seperti yang ditunjukkan pada gambar berikut.



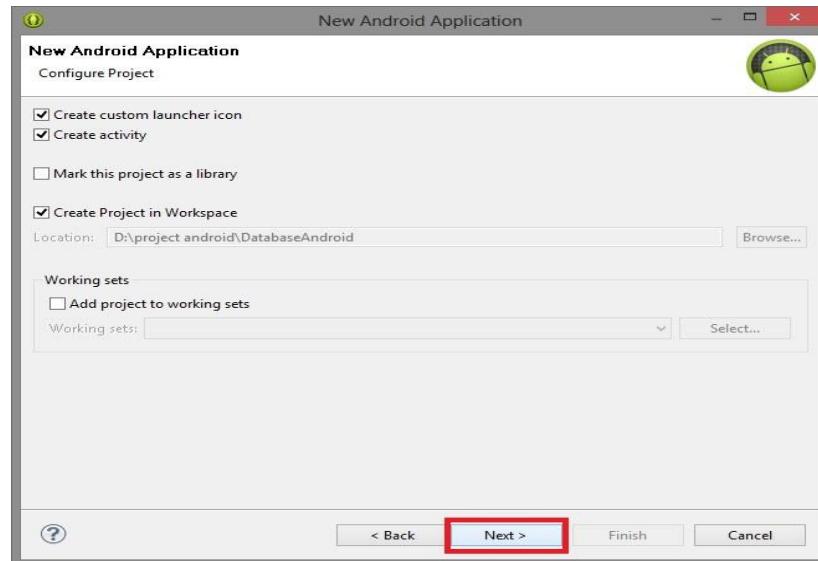
5. Click *Browse*. Kemudian Browse ke direktori file *adt-bundle-windows-x86_64-201305221\sdk\extras\google\google-play-services\libproject\google-play-services_lib* pada computer. Seperti yang ditunjukkan pada gambar berikut.



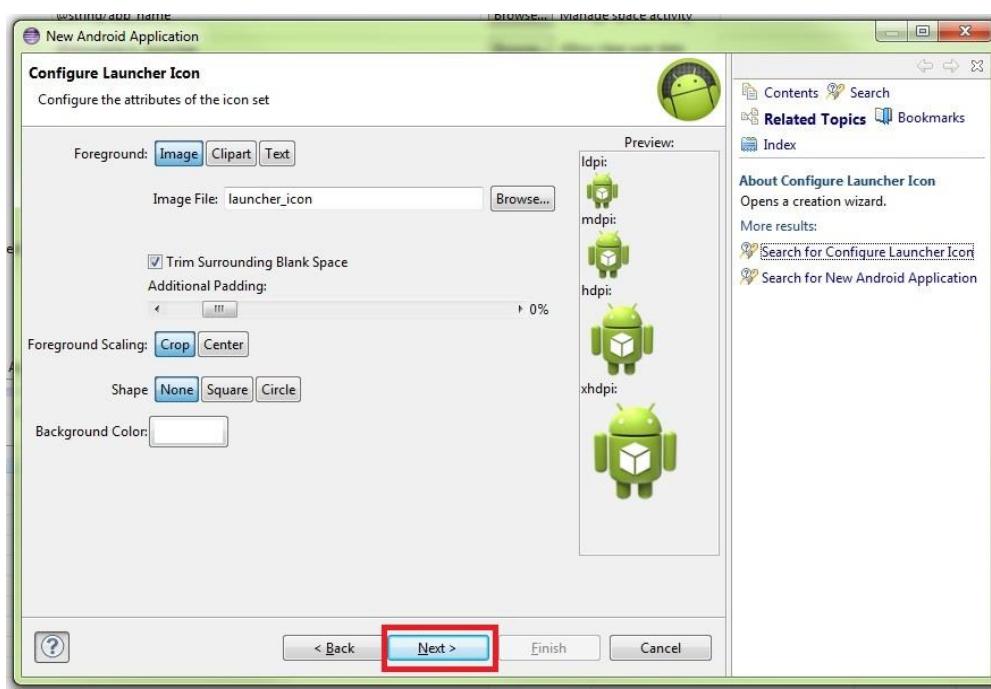
6. Setelah Finish Proyek lib ditambahkan ke bagian Proyek di layar Proyek Impor

Tahap 2: Buat proyek aplikasi Android baru

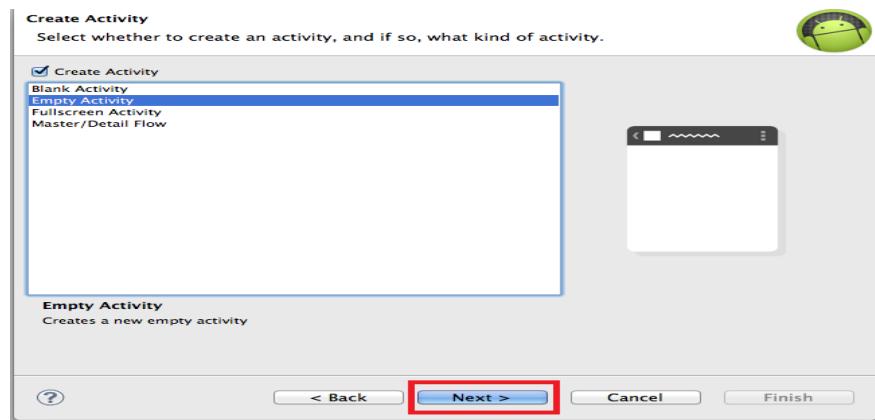
1. Pilih *File > New > Android Application Project*. Maka akan muncul tampilan *New Android Application* dan isi *Application Name* dengan *Map* Kemudian click *Next*.
2. Setelah itu akan muncul *Android Application Configure Project* langsung click next aja seperti yang ditunjukkan pada gambar berikut.

*Configure Project*

3. Setelah itu muncul tampilan seperti yang ditunjukkan pada gambar click next aja.

*Configure the attributes of the icon set*

4. Sehabis *Next* maka akan muncul tampilan dari *Create Activity* untuk memilih activity tampilan dari android, pilih *Blank Activity / Empty Activity* kemudian *click next / Finish* seperti yang ditunjukkan pada gambar berikut.

*Create Activity*

5. Setelah itu sebuah aplikasi baru dengan nama *Map* dibuat dalam *package explorer*.

Tahap 3: Buat UI untuk aplikasi

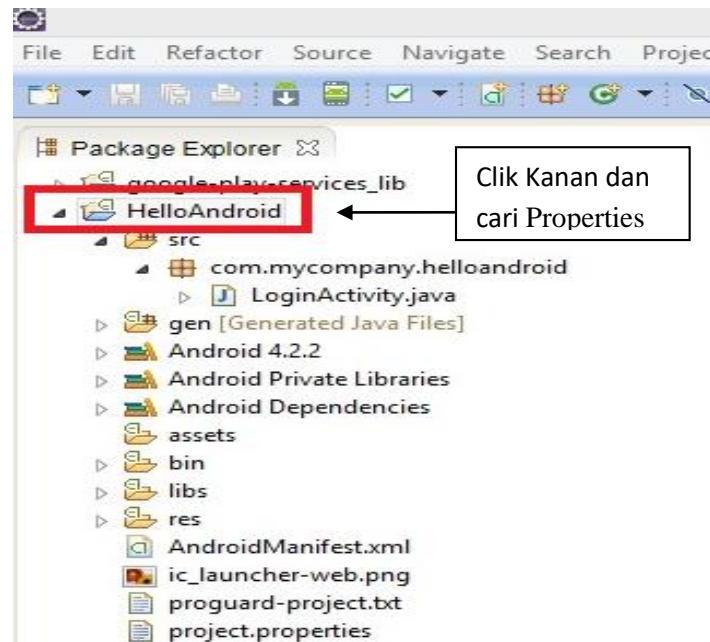
1. Click *activity_main.xml* pada *package*.
2. Kemudian gunakan code dibawah ini.

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    tools:context="com.kipli.myapplication.MainActivity">

    <TextView
        android:id="@+id/header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

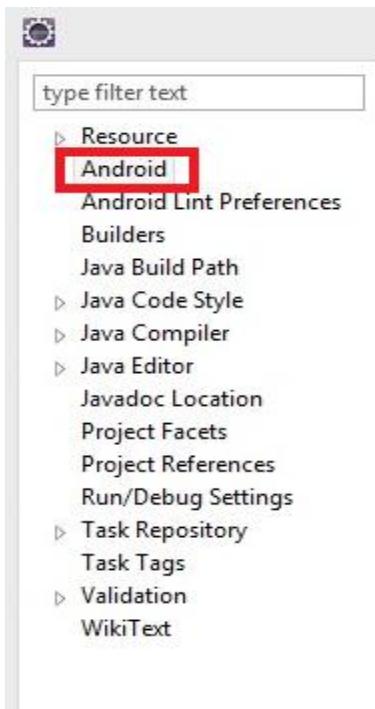
    <fragment
        android:id="@+id/mymap"
        android:layout_below="@+id/header"
        android:name="com.google.android.gms.maps.MapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</RelativeLayout>
```

3. Kemudian Pilih *File > Save All*.
4. Click *kanan* pada name=Map di Package Explorer, kemudian pilih Properties. seperti yang ditunjukkan pada gambar berikut.



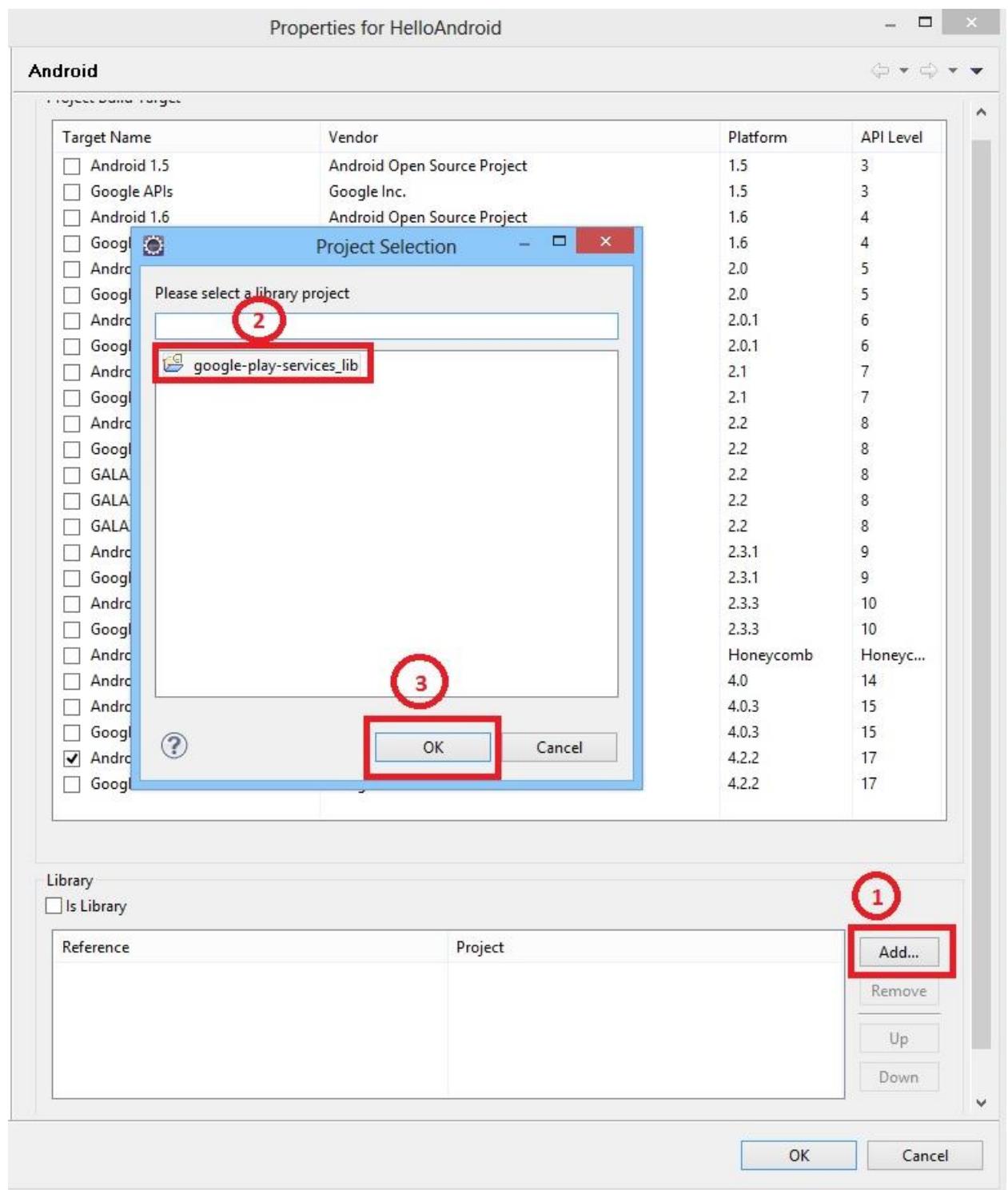
Folder Project

5. Kemudian pilih *Android* seperti yang ditunjukkan pada gambar berikut.



Android pada Properties

6. Click tombol *Add* di panel sebelah kanan. seperti yang ditunjukkan pada gambar berikut.

*Project Selection*

7. Kemudian click OK.

Tahap 4: Buat kunci API untuk aplikasi

1. Buka **CMD** (Command Prompt).
2. Lalu cari folder **bin** instalasi java.
3. Kemudian ketik code dibawah ini.

```
Keytool -list -v -keystore "C:\Users\<username>\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
```

4. Lalu Enter.
5. Maka akan keluar output pada cmd seperti yang ditunjukkan pada gambar berikut.



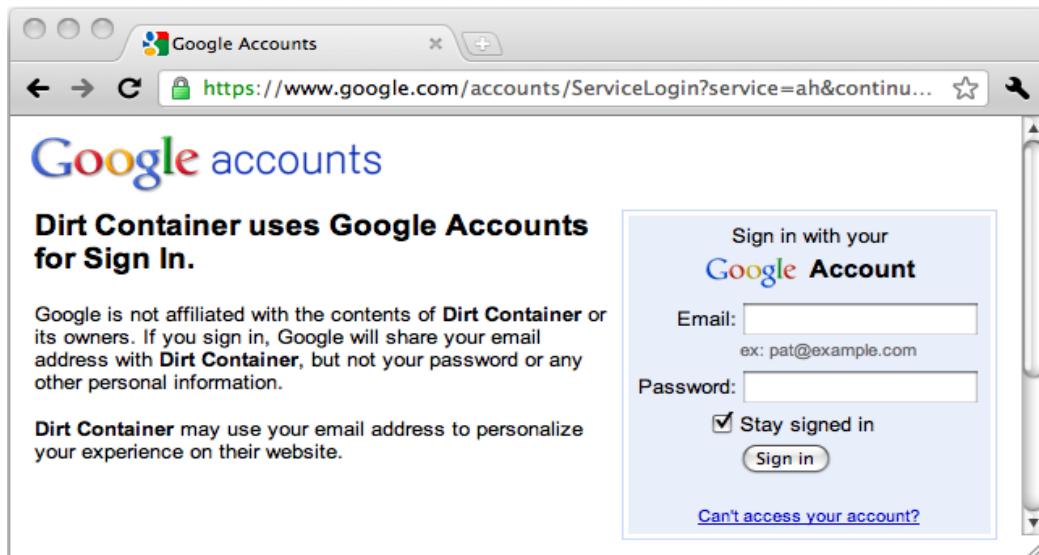
```
C:\WINDOWS\system32\cmd.exe
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 4afffbef
Valid from: Tue Jun 16 10:08:51 BDT 2015 until: Thu Jun 08 10:08:51 BDT 2045
Certificate fingerprints:
  MD5: 04:61:E0:E1:E1:4D:A7:0E:16:B3:0F:D1:FF:BC:73:1D:3A
  SHA1: A2:08:96:f0:88:57:98:C3:93:67:54:2F:73:88:7E:64:38:7A:25:52
  SHA256: 1B:E9:CC:AB:CC:5E:3A:44:A1:81:0A:0C:98:76:8E:88:71:86:66:50:84:FA:87:8C:02:D1:E0:9F:C7:7A:9A:0E
  Signature algorithm name: SHA256withRSA
  Version: 3

Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
  SubjectKeyIdentifier [
    KeyIdentifier [
      0000: 47 FA F8 60 B0 F5 1D E3   0C 3D 77 94 FF 50 C3 24  G,.,...,W,,P,$
      0010: 10 79 07 86               +...
    ]
  ]

C:\Program Files\Java\jdk1.8.0_45\bin>
```

CMD

6. Buka halaman <http://code.google.com/apis/console/#project:486217353208> maka gambar google Accounts ditampilkan, seperti yang ditunjukkan di halaman Akun Google berikut ini.



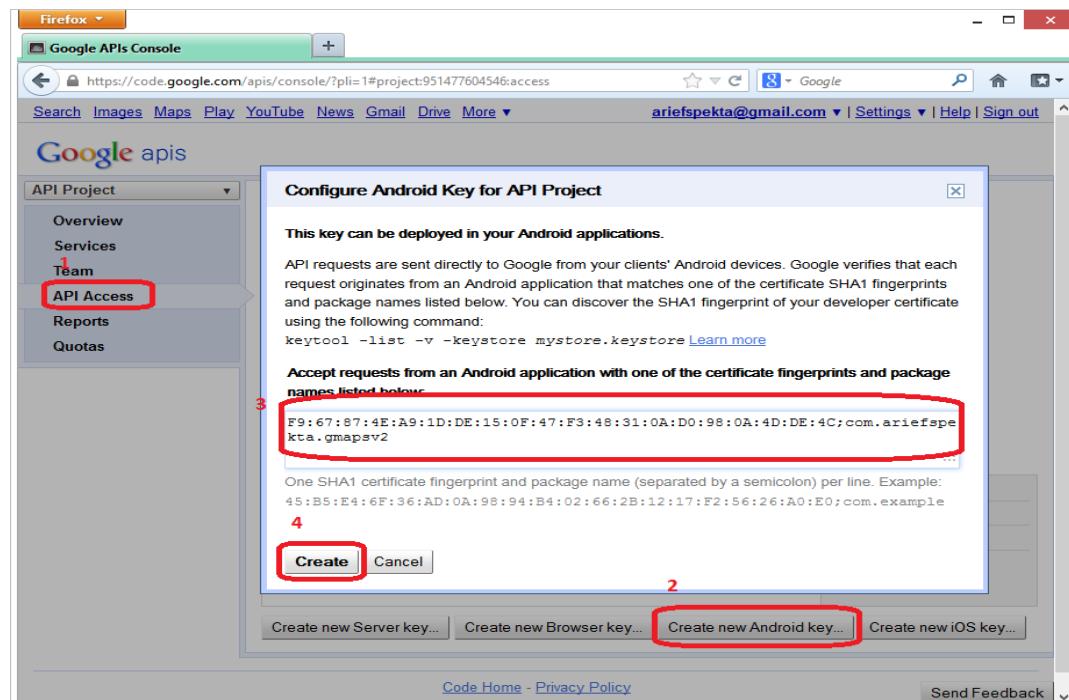
Accounts Google

7. Ketik nama pengguna Anda di kotak teks Email
8. Ketik kata sandi Anda di kotak teks Password.
9. Click tombol Masuk.
10. Click link *Services*. Semua halaman Layanan ditampilkan.
11. Gulir ke bawah halaman, dan kemudian pastikan status tombol yang berada di sebelah *opsi & Google Maps Android API v2 ON* seperti yang ditunjukkan pada gambar berikut.

Google Cloud SQL	OFF	Pricing
Google Cloud Storage	OFF	Pricing
Google Cloud Storage JSON API	Request access...	Courtesy limit: 100,000 re
Google Compute Engine	Request access...	Pricing
Google Maps Android API v2	ON	
Google Maps API v2	OFF	Courtesy limit: 25,000 req
Google Maps API v3	OFF	Courtesy limit: 25,000 req
Google Maps Coordinate API	OFF	Courtesy limit: 1,000 requ
Google Maps Geolocation API	OFF	Courtesy limit: 0 requests
Google Maps SDK for iOS	OFF	

Google Maps Android API v2

12. Gulir ke atas halaman dan kemudian Click *API Access* .
13. Lalu *Create Android key*. Maka akan keluar *Configure Android Key for API Project* seperti yang ditunjukkan pada gambar berikut.



Create Android key

14. Ketik sidik jari <*SIHA1 fingerprint*>; *com.example.gmap* terus ada **permintaan Terima dari Android** untuk mengajukan permohonan dengan salah satu **sidik jari sertifikat** dan nama **paket yang tercantum di bawah kotak teks** dan kemudian klik tombol Buat.
15. Kunci API akan dihasilkan. Anda perlu mencatat atau menyalin tombol API .

Tahap 5: Mengubah File Manifest Android

1. Click dua kali file *AndroidManifest.xml* di *Package Explorer* maka File manifes ditampilkan di area Editor.
2. Click tab *AndroidManifest.xml* yang terletak di bagian bawah area Editor.

3. Kemudian gunakan code dibawah ini Sebelum < **application** >.

```
<permission
    android:name="com.example.map.permission.MAPS_RECEIVE"
    android:protectionLevel="signature"/>
<uses-permission
    android:name="com.example.map.permission.MAPS_RECEIVE"/>
<uses-permission
    android:name="android.permission.INTERNET"/>
<uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission

    android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true"/>
```

4. Setelah itu Sebelum < **activity** > isi code dibawah ini.

```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="isi dengan API_KEY yang sudah di buat"/>
```

5. Dan di *Save All*.

Latihan

Membuat aplikasi Android untuk menampilkan peta menggunakan google map services.

MODUL 12

KOMUNIKASI MELALUI APLIKASI ANDROID

Tujuan

- Mahasiswa mampu mengirim dan menerima SMS melalui aplikasi android.
- Mahasiswa mampu mengaplikasikan Telephony.

Mengirim SMS dengan Aplikasi Android

Dengan perkembangan zaman yang sangat pesat saat ini sudah banyak aplikasi Android yang menggunakan fitur SMS atau pesan singkat untuk berbagai keperluan, salah satunya yang sering kita temui misalnya sebagai sarana verifikasi pengguna. OS Android sendiri, karena mempunyai tingkat modularitas yang tinggi, sudah menyediakan interface dan kelas tertentu yang bisa kita gunakan apabila kita ingin mengintegrasikan layaknya SMS ke dalam aplikasi yang kita buat.

Untuk dapat mengirim SMS melalui aplikasi Android disini kita ada tiga cara yang dapat dilakukan, yaitu cara yang pertama menggunakan `SMSManager`, cara yang kedua menggunakan `smsto:URI`, dan cara yang ketiga menggunakan `SMSViewIntent`.

1. Menggunakan `SMSManager`

Metode yang pertama adalah dengan menggunakan `SMSManager`, yaitu sebuah class khusus yang ada di OS Android yang di khususkan untuk mengakses fitur-fitur SMS di Android. Kita bisa menggunakan class tersebut untuk mengirim sebuah SMS ke nomor yang dituju. Ketika menggunakan `SMSManager`, SMS akan langsung dikirim dari dalam aplikasi kita, tanpa membutuhkan aplikasi SMS lain.

```
public void sendSmsByManager() {
    try {
        // Get the default instance of the SmsManager
        SmsManager smsManager = SmsManager.getDefault();

        smsManager.sendTextMessage(phoneNumber.getText().toString(),
            null,
            smsBody.getText().toString(),
            null,
            null);
        Toast.makeText(getApplicationContext(), "SMS Berhasil Dikirim!",
            Toast.LENGTH_LONG).show();
    } catch (Exception ex) {
        Toast.makeText(getApplicationContext(), "Pengiriman SMS
Gagal..."+ex,
            Toast.LENGTH_LONG).show();
        ex.printStackTrace();
    }
}
```

Bisa dilihat coding di atas memanggil satu method sendTextMessage(), dan kemudian memberikan parameter berupa nomor telepon tujuan, nomor telepon pengirim, isi SMS, intent yang akan dieksekusi apabila SMS berhasil terkirim/gagal, dan intent yang akan dieksekusi apabila SMS telah diterima. Namun pada tutorial ini kita hanya perlu mengirim nomor dan isi SMS saja tanpa tersambung aplikasi lainnya karna sifatnya berbayar atau menggunakan pulsa.

2. Menggunakan Intent SENDTO

Metode yang kedua yaitu Intent SENDTO, kita akan mengirim SMS menggunakan Aplikasi eksternal, apabila terdapat lebih dari 1 aplikasi SMS pada handphone (misalkan Hangouts,SMS), maka kita akan diberikan pilihan aplikasi yang digunakan untuk mengirim SMS. Namun apabila hanya ada Apikasi SMS di handphone anda, maka kita akan langsung masuk ke default SMS aplikasi di Android dan mengirim SMS dari sana.

```
public void sendSmsBySIntent() {
    // add the phone number in the data
    Uri uri = Uri.parse("smsto:" +
    phoneNumber.getText().toString());

    Intent smsSIntent = new Intent(Intent.ACTION_SENDTO, uri);
    // add the message at the sms_body extra field
    smsSIntent.putExtra("sms_body",
    smsBody.getText().toString());
    try{
        startActivity(smsSIntent);
    } catch (Exception ex) {
        Toast.makeText(MainActivity.this, "Pengiriman SMS Gagal...", Toast.LENGTH_LONG).show();
        ex.printStackTrace();
    }
}
```

3. Menggunakan VIEW Intent

Metode yang ketiga adalah menggunakan VIEW Intent, hampir sama dengan cara nomor 2. Hanya saja dengan menggunakan View Intent kita akan langsung di arahkan ke Aplikasi SMS default kepunyaan Android, dan kemudian mengirim SMS dari default Android.

```
public void sendSmsByVIntent() {

    Intent smsVIntent = new Intent(Intent.ACTION_VIEW);
    // hanya akan membuka aplikasi SMS/MMS default di Android
    smsVIntent.setType("vnd.android-dir/mms-sms");

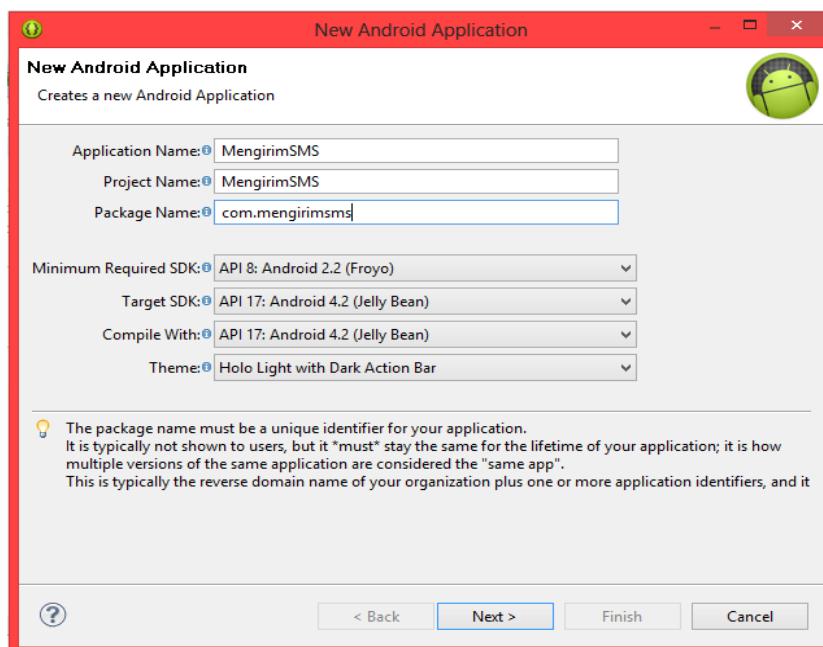
    // menambahkan nomor telepon dan isi SMS otomatis
    smsVIntent.putExtra("address", phoneNumber.getText().toString());
    smsVIntent.putExtra("sms_body", smsBody.getText().toString());
```

```

    try{
        startActivity(smsVIntent);
    } catch (Exception ex) {
        Toast.makeText(MainActivity.this, "Pengiriman SMS
Gagal...", Toast.LENGTH_LONG).show();
        ex.printStackTrace();
    }
}
  
```

Berikut Contoh implementasi .

Langkah pertama yaitu membuka aplikasi eclipse, selanjutnya pilih file – new – Android Application Project – Pada Application Name isi sesuai dengan nama project kalian (tampilan ada pada gambar dibawah) – next – next – Pada Configure Launcher Icon bisa dipilih sesuai dengan kengininan sendiri – next – Blank Activity – next – untuk Activity Name bisa diubah bisa juga didiamkan – finish.



Kemudian pada kelas MainActivity.java atau kelas utama di project kalian, berikut code untuk membuat Aplikasi SMS.

```

package com.mengirimsms;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
  
```

```
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {

    private EditText phoneNumber;
    private EditText smsBody;
    private Button smsManagerBtn;
    private Button smsSendToBtn;
    private Button smsViewBtn;

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

        phoneNumber = (EditText) findViewById(R.id.phoneNumber);
        smsBody = (EditText) findViewById(R.id.smsBody);
        smsManagerBtn = (Button) findViewById(R.id.smsManager);
        smsSendToBtn = (Button) findViewById(R.id.smsSIntent);
        smsViewBtn = (Button) findViewById(R.id.smsVIntent);

        smsManagerBtn.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {
                sendSmsByManager();
            }
        });

        smsSendToBtn.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {
                sendSmsBySIntent();
            }
        });

        smsViewBtn.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {
                sendSmsByVIntent();
            }
        });
    }

    public void sendSmsByManager() {
        try {
            // Get the default instance of the SmsManager
            SmsManager smsManager = SmsManager.getDefault();

            smsManager.sendTextMessage(phoneNumber.getText().toString(),
                    null,
                    smsBody.getText().toString(),
                    null,
                    null);
            Toast.makeText(getApplicationContext(), "SMS Berhasil Dikirim!",
                    Toast.LENGTH_LONG).show();
        } catch (Exception ex) {
            Toast.makeText(getApplicationContext(), "Pengiriman SMS
Gagal..." + ex,
                    Toast.LENGTH_LONG).show();
            ex.printStackTrace();
        }
    }
}
```

```
        }

    public void sendSmsBySIntent() {
        // add the phone number in the data
        Uri uri = Uri.parse("smsto:" +
phoneNumber.getText().toString());

        Intent smsSIntent = new Intent(Intent.ACTION_SENDTO, uri);
        // add the message at the sms_body extra field
        smsSIntent.putExtra("sms_body",
smsBody.getText().toString());
        try{
            startActivity(smsSIntent);
        } catch (Exception ex) {
            Toast.makeText(MainActivity.this, "Pengiriman SMS Gagal...", Toast.LENGTH_LONG).show();
            ex.printStackTrace();
        }
    }

    public void sendSmsByVIntent() {

        Intent smsVIntent = new Intent(Intent.ACTION_VIEW);
        // hanya akan membuka aplikasi SMS/MMS default di Android
        smsVIntent.setType("vnd.android-dir/mms-sms");

        // menambahkan nomor telepon dan isi SMS otomatis
        smsVIntent.putExtra("address", phoneNumber.getText().toString());
        smsVIntent.putExtra("sms_body", smsBody.getText().toString());
        try{
            startActivity(smsVIntent);
        } catch (Exception ex) {
            Toast.makeText(MainActivity.this, "Pengiriman SMS Gagal...", Toast.LENGTH_LONG).show();
            ex.printStackTrace();
        }
    }
}
```

Untuk Layout masukkan kode berikut pada file Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

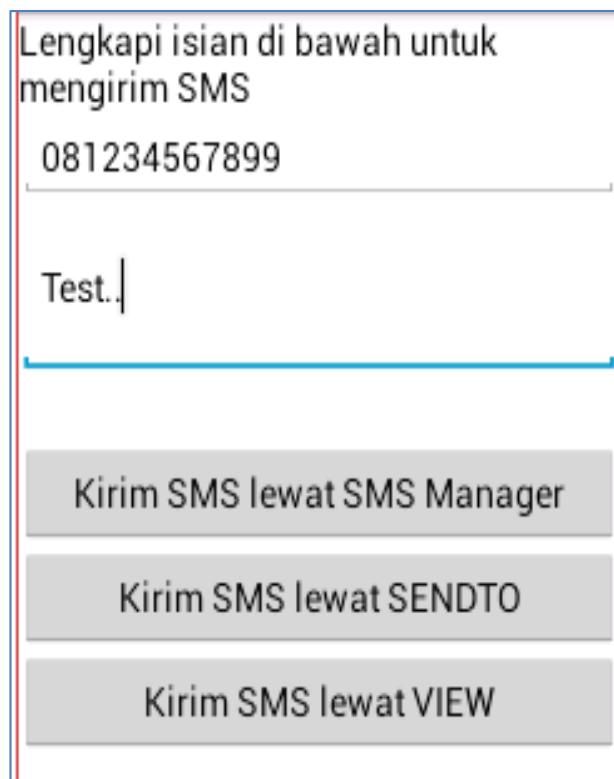
    <TextView
        android:id="@+id/text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Lengkapi isian di bawah untuk mengirim SMS"
        android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText  
    android:id="@+id/phoneNumber"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="isi nomor.." "  
    android:inputType="phone" />  
  
<EditText  
    android:id="@+id/smsBody"  
    android:layout_width="fill_parent"  
    android:layout_height="80dp"  
    android:layout_marginBottom="30dp"  
    android:hint="Isi pesan.." />  
  
<Button  
    android:id="@+id/smsManager"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Kirim SMS lewat SMS Manager" />  
  
<Button  
    android:id="@+id/smssIntent"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Kirim SMS lewat SENDTO" />  
  
<Button  
    android:id="@+id/smsVIntent"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Kirim SMS lewat VIEW" />  
  
</LinearLayout>
```

Tambahkan permission berikut pada AndroidManifest.xml

```
<uses-permission android:name="android.permission.SEND_SMS" />
```

Berikut tampilan jika program sudah dijalankan.



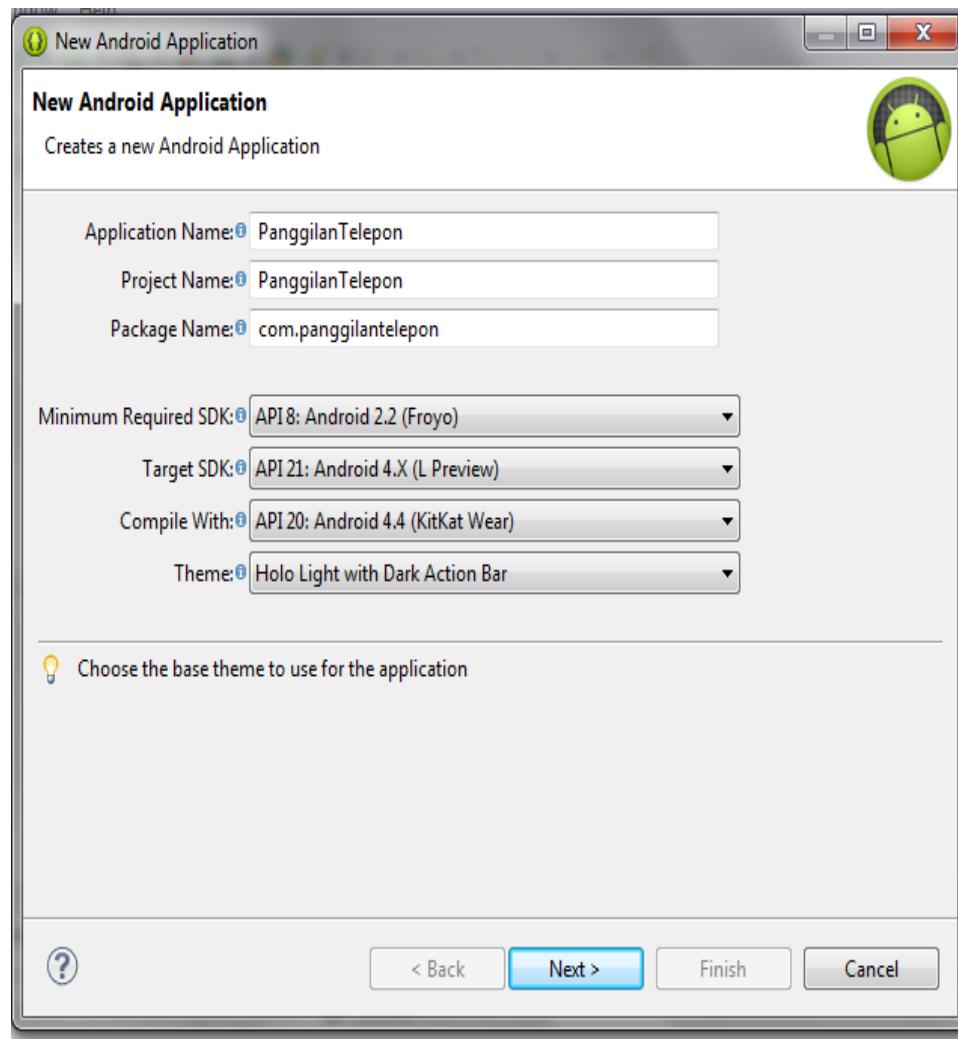
Jika ingin mengirim SMS lewat SMS Manager maka kita harus menggunakan pulsa untuk mengirim SMS. Jika mengirim SMS lewat SENDTO maka kita diberikan pilihan pada Aplikasi Android untuk menggunakan apa kita mengirim pesan. Dan jika mengirim SMS lewat VIEW maka kita akan langsung di arahkan ke Aplikasi SMS default kepunyaan Android, dan kemudian mengirim SMS dari default Android.

Mengaplikasikan Telephony

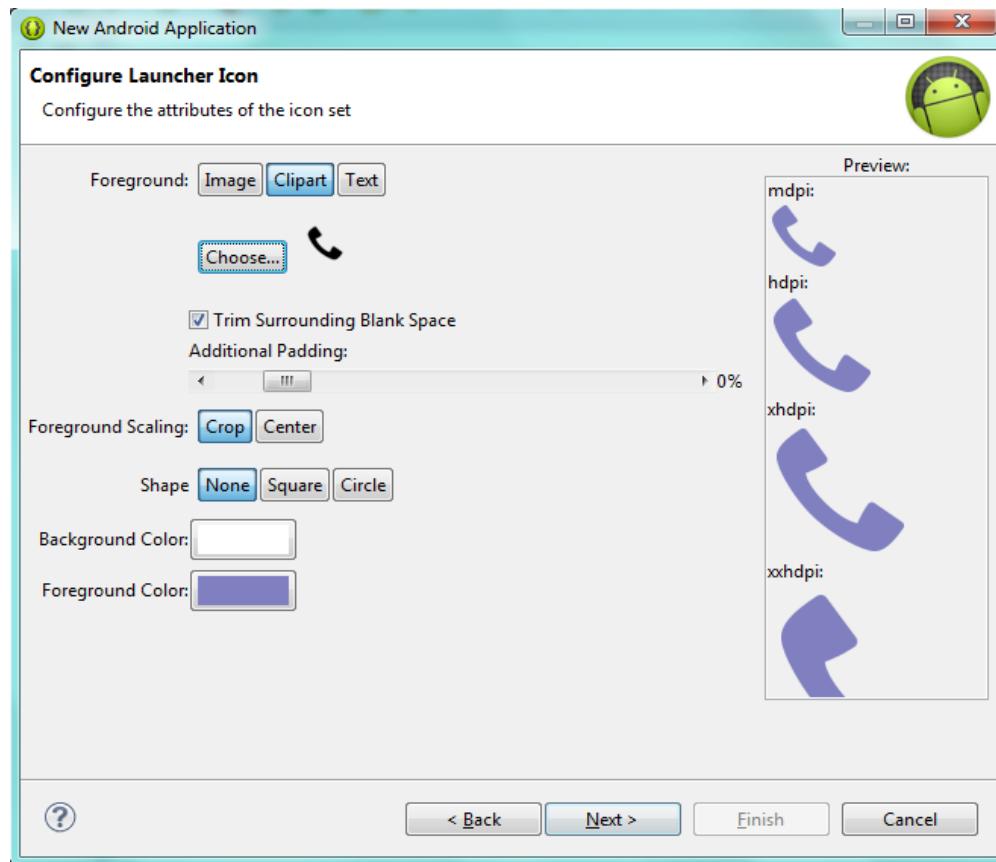
Platform Android menyediakan API untuk aplikasi mengakses fasilitas telepon untuk membuat panggilan telepon.

Contoh Implementasi

Buat Project Baru dengan nama “PanggilanTelepon” seperti gambar dibawah.

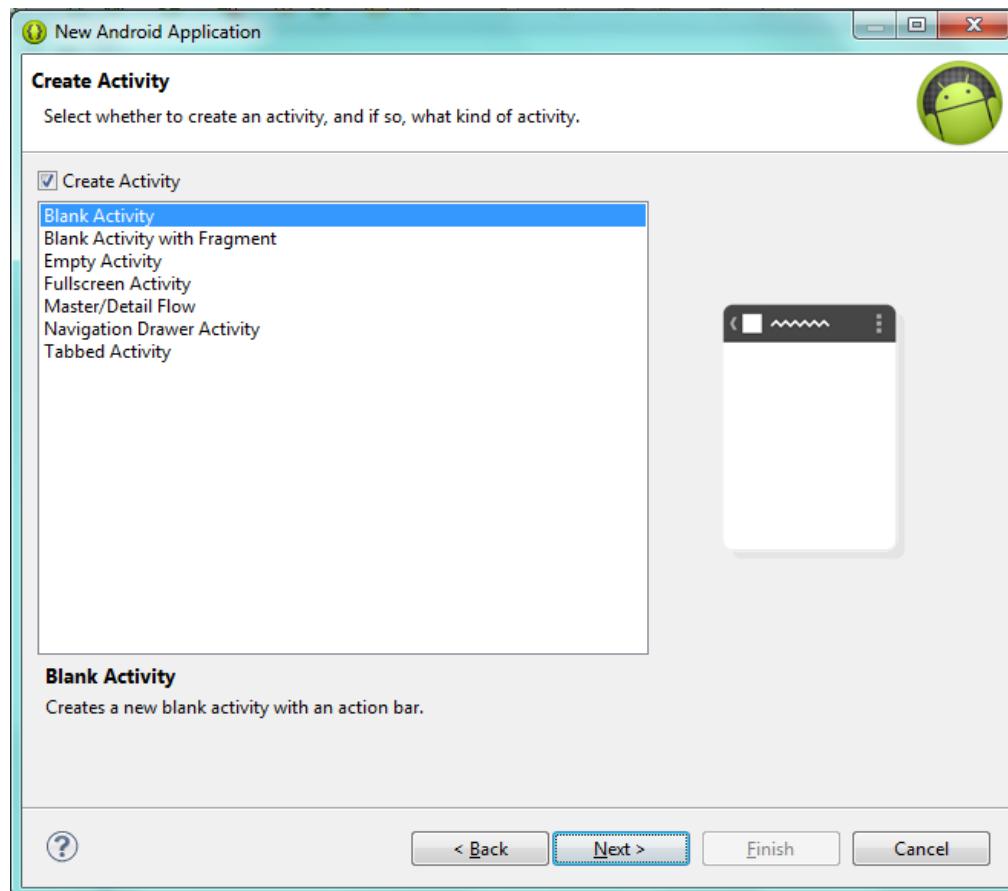


Kemudian akan muncul kotak dialog *configure launcher icon*. Pada *configure launcher icon* ini kita dapat mengganti *icon* yang nantinya akan digunakan untuk aplikasi kita. Jika ingin menggunakan gambar dapat memilih menu *image – browse*, jika ingin menggunakan *clipart* dapat memilih menu *clipart – choose* dan pada menu ini kita dapat memilih warna untuk *icon* aplikasi kita, kemudian jika ingin menggunakan *text* dapat memilih menu *text*. Pada tutorial ini penulis menggunakan *icon telephone* yang terdapat pada menu *clipart*.



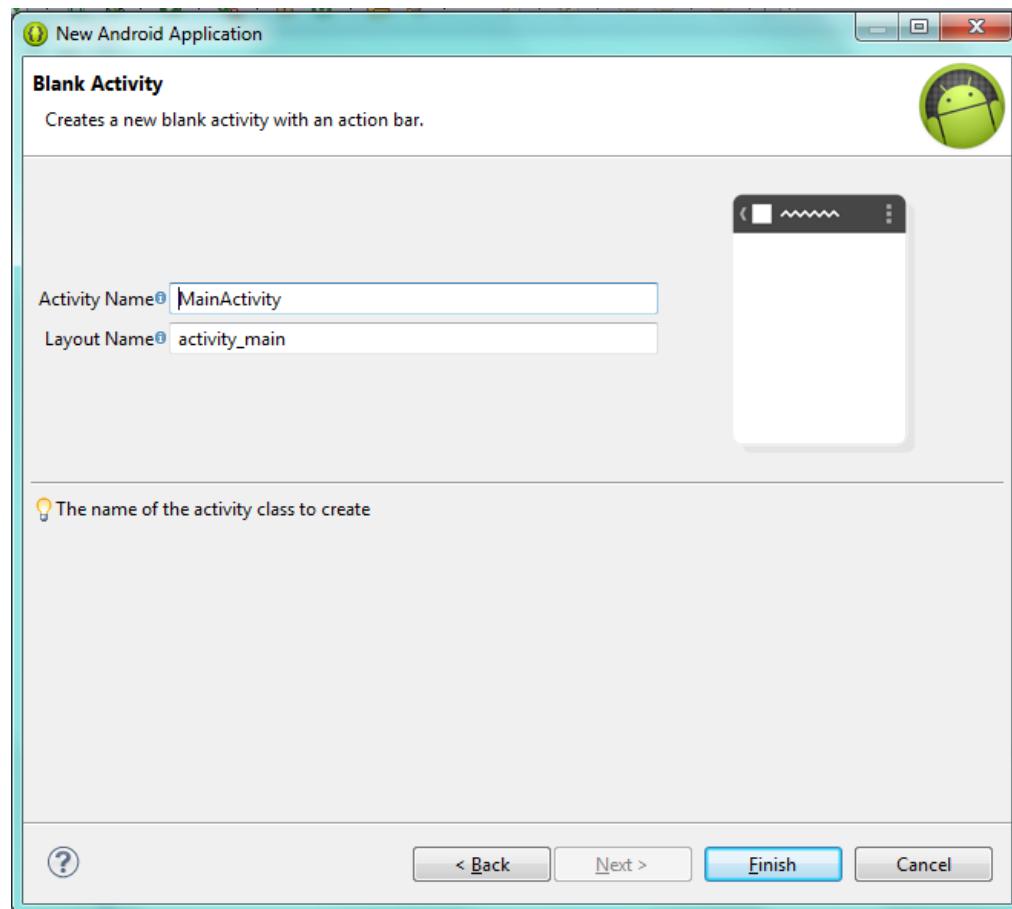
Kemudian pilih *Next*.

Langkah selanjutnya ialah *create activity*. Pada kotak dialog *create activity* pilih *blank activity*.



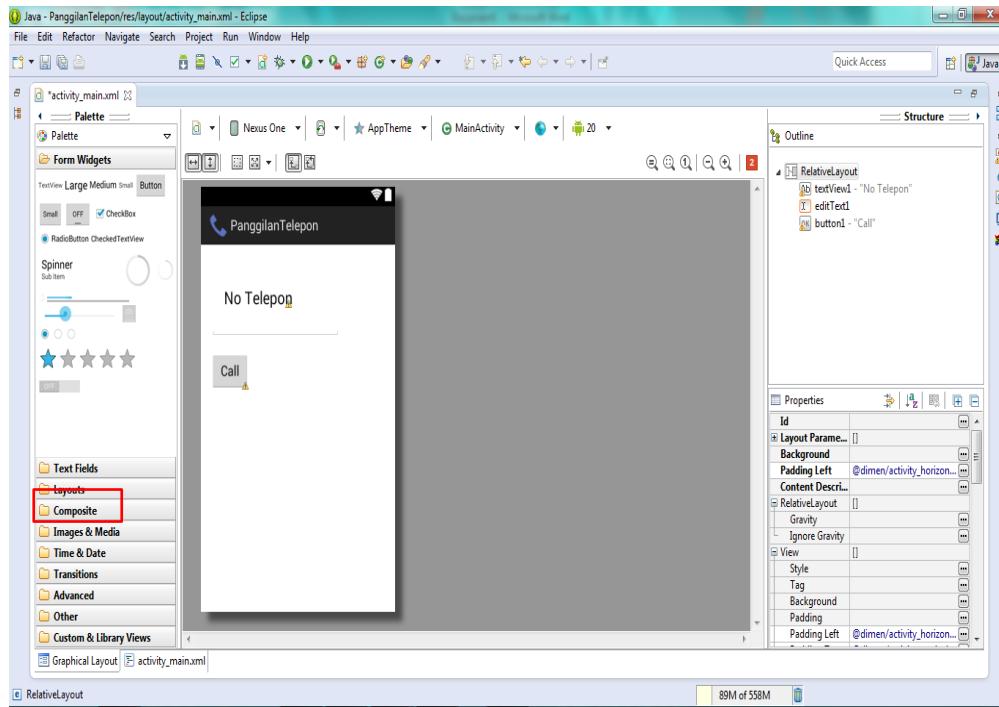
Kemudian pilih *Next*.

Selanjutnya akan muncul kotak dialog *blank activity*. Pada kotak dialog ini kita dapat mengatur nama *main activity* dan *layout activity* kita



Selanjutnya pilih *Finish*.

Langkah selanjutnya kita mulai membuat tampilan aplikasi panggilan telepon. Pada aplikasi ini kita membutuhkan *TextView* untuk memberikan *label*, *EditText* untuk memasukan nomor telepon, dan *Button* untuk tombol ketika melakukan telepon. Untuk mengatur tata letak *layout* kita menuju folder *res – layout – activity_main.xml – graphical layout*.



Selanjutnya kita menuju *activity_main.xml* yang terletak disebelah *Graphical Layout*.

Kemudian pada *activity_main.xml* tambahkan *coding* berikut :

```
<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.panggilantelepon.MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="22dp"
        android:layout_marginTop="38dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="No Telepon" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_alignParentStart="true"
        android:layout_alignLeft="@+id/textView1" />

```

```

    android:layout_below="@+id/textView1" />

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignLeft="@+id/editText1"
    android:layout_below="@+id/editText1"
    android:layout_marginTop="18dp"
    android:text="Call" />

</RelativeLayout>

```

Langkah selanjutnya tambahkan *coding* berikut pada *MainActivity.java*. Untuk mencari *MainActivity.java* pilih folder *src – com.panggilantelepon – MainActivity.java*.

```

package com.panggilantelepon;

import android.support.v7.app.ActionBarActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //EditText
        final EditText number=(EditText)findViewById(R.id.editText1);
        Button dial=(Button)findViewById(R.id.button1);
        //Ketika menekan tombol call maka akan melakukan dial
        dial.setOnClickListener(new OnClickListener(){
            public void onClick(View v){
                //number = inputan dari edittext
                String toDial="tel:"+number.getText().toString();

                startActivity(new
                    Intent(Intent.ACTION_DIAL,Uri.parse(toDial)));
            }
        });
    }
}

```

Selanjutnya memberikan *permission* pada aplikasi PanggilanTelepon agar dapat mengakses fitur untuk melakukan panggilan ke nomor telepon yang dituju. Untuk memberikan *permission* tambahkan *coding* pada *AndroidManifest.xml*.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.panggilantelepon"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="21" />

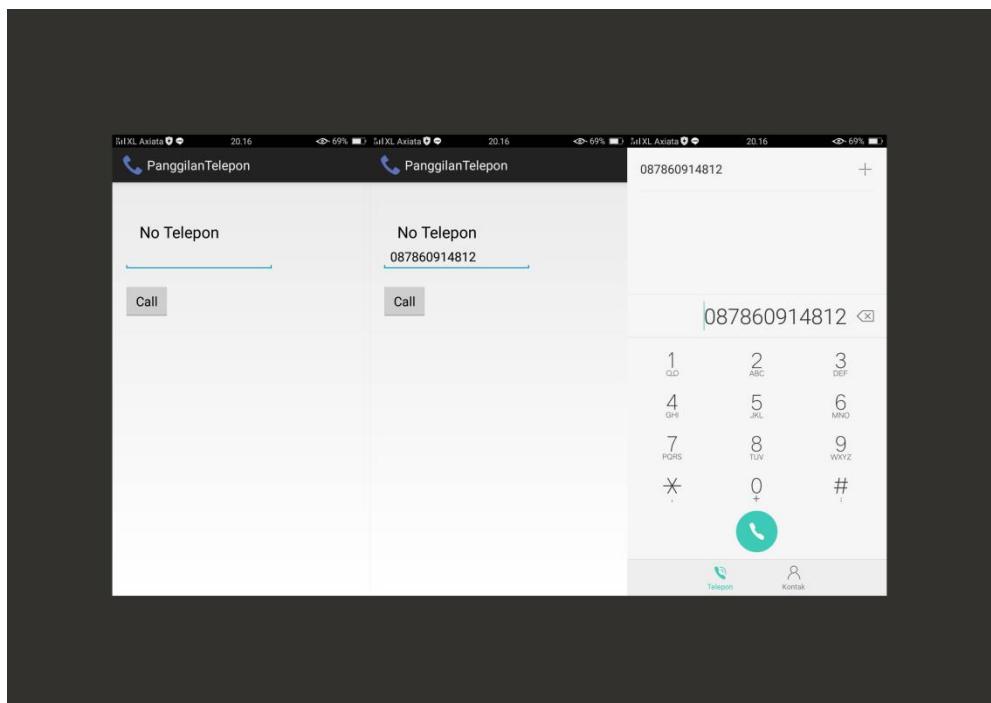
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Langkah selanjutnya adalah menjalankan aplikasi PanggilanTelepon melalui ponsel. Berikut adalah hasil dari aplikasi PanggilanTelepon.



Latihan

Membuat aplikasi Android untuk mengirim sms dan mengaplikasikan panggilan telepon.

MODUL 13

MENINGKATKAN APLIKASI MENGGUNAKAN GRAFIK DAN MULTIMEDIA

Tujuan

- Mahasiswa mampu mengaplikasikan Graphics.
- Mahasiswa mampu menambahkan element audio dan video.

Mengaplikasikan Graphics

Graphics merupakan salah satu komponen dari multimedia yang berguna untuk mendapatkan perhatian dari pengguna. Pada modul ini graphics yang akan dibahas adalah 2 Dimensi. Class yang akan digunakan adalah Class Canvas yang berguna sebagai permukaan tempat untuk menggambarkan grafik dan gambar. Serta Class Paint yang digunakan untuk mengatur style dan warna dari simbol, teks dan grafis yang akan digambar.

Berikut Contoh Implementasi

Buatlah project baru dengan nama Grafik2d. Selanjutnya tambahkan class baru yang bernama ViewGrafik. Class ViewGrafik adalah sebuah custom view yang akan digunakan menggambar grafik, simbol dan teks. Berikut adalah kode dari ViewGrafik.

```

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;
import android.graphics.Paint.Style;
import android.graphics.Typeface;
import android.view.View;

public class ViewGrafik extends View {

    // RectF batas Layar (kiri,atas,bawah,kanan) private
    RectF kotak1 = new RectF(100,100,300,200);//
    kiri,atas,kanan,bawah
  
```

```
private RectF kotak2 = new RectF(200,25,300,75);
private RectF kotak3 = new RectF(300,210,400,300);

private RectF oval = new RectF(100,200,200,300); //oval akan
mengikuti rect

private RectF oval2 = new RectF(100,320,200,400);

Paint cat= new Paint();

protected void onDraw(Canvas c) {

    //gambar kotak

    cat.setColor(Color.argb(200, 247, 4, 4)); //alpha
transparasi warna,red,green,blue
    c.drawRect(kotak1,cat);

    //gambar lingkaran

    cat.setColor(Color.BLUE);

    c.drawCircle(60,60, 40, cat); //kiri, atas,radius

    //gambar kotak dengan ujung tumpul

    c.drawRoundRect(kotak2, 10, 10, cat); //memanggil
lokasi kotak2,kiri-radius oval, kanan radius oval

    //ukuran sebuah lingkaran adalah 360 derajat

    // derajat dihitung dari derajat awal ditentukan ,lengkungan
akan mengikuti rect oval

    //paramater ke 4 jika true garis penutup melewati titik
pusat //berikut dimulai dari 30 derajat, sepanjang 120
derajat c.drawArc(oval, 30, 120, true,cat);

    cat.setColor(Color.YELLOW);
```

```
//jika parameter ke-4 false maka garis akan melewati
//ujung ke ujung tanpa harus ke titik pusat
//menyebabkan efek spt stengah lingkaran
c.drawArc(oval2, 180, 180, false,cat);

//gambar garis

cat.setColor(Color.RED);

c.drawLine(50,400,400,400,cat); //kiri,atas,kanan,bawah

//gambar kotak kosong

cat.setColor(Color.CYAN);
cat.setStyle(Style.STROKE);

cat.setStrokeWidth(5);

c.drawRect(kotak3,cat);

//gambar titik secara acak jika program direfresh maka posisi
akan berubah

cat.setColor(Color.BLACK);
for (int i=0;i<10;i++) {

c.drawPoint( (float) Math.random()*500,
(float)Math.random()*500, cat);

}

//Menggunakan tulisan

cat.setColor(Color.RED);

//diisi dalam maupun outline

cat.setStyle(Style.FILL_AND_STROKE);

cat.setStrokeWidth(2);

cat.setTextSize(35);
```

```
c.drawText("Hello World", 10, 500, cat);

//menggunakan font

cat.setTypeface(Typeface.create(Typeface.SERIF,Typeface.BOLD_ITALIC) );

c.drawText("font serif italic_bold", 10, 550, cat);

}

public ViewGrafik(Context context) {

super(context);

// TODO Auto-generated constructor stub

}

}
```

Pada MainActivity.java ubah setContentView dengan memanggil class ViewGrafik. Seperti berikut.

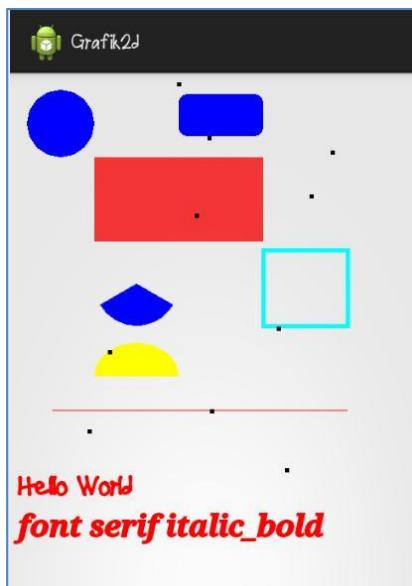
```
package com.grafik2d;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new ViewGrafik(this));
    }
}
```

Berikut adalah tampilan ketika program di running.



Menambahkan Elemen Audio dan Video.

Elemen audio dan video pada perangkat mobile menjadi hal yang penting dan menjadi kebutuhan kebanyakan pengguna. Hal ini seiring dengan perkembangan dari perangkat mobile yang semakin mendukung untuk penggunaan multimedia. Untuk dapat menggunakan elemen audio dan video pada aplikasi android yang akan dikembangkan digunakan class MediaPlayer. Berikut adalah contoh implementasinya.

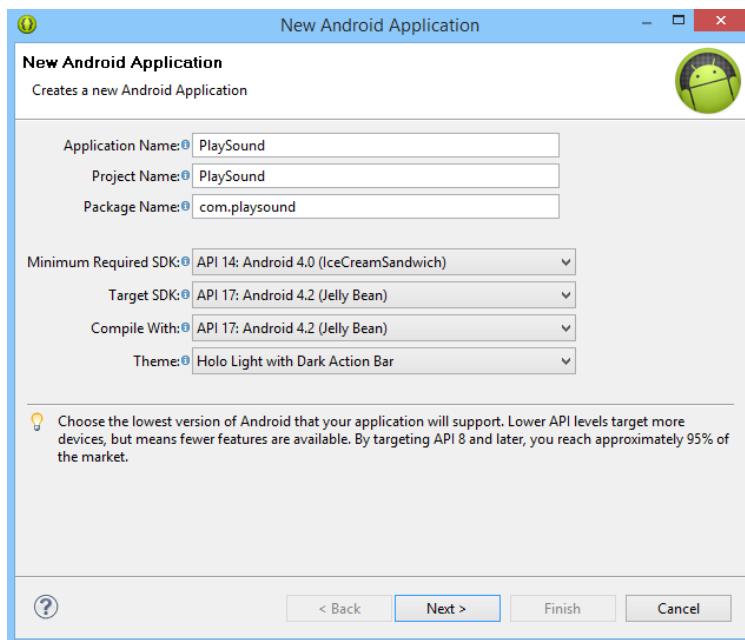
Contoh Implementasi MediaPlayer untuk memainkan audio

Persiapkan file audio (rekomendasi file audio dengan format mp3). Nama file audio diawali oleh huruf, tanpa spasi, symbol yang diperkenankan hanya underscore(_) dan *lowercase* (tidak ada huruf kapital).

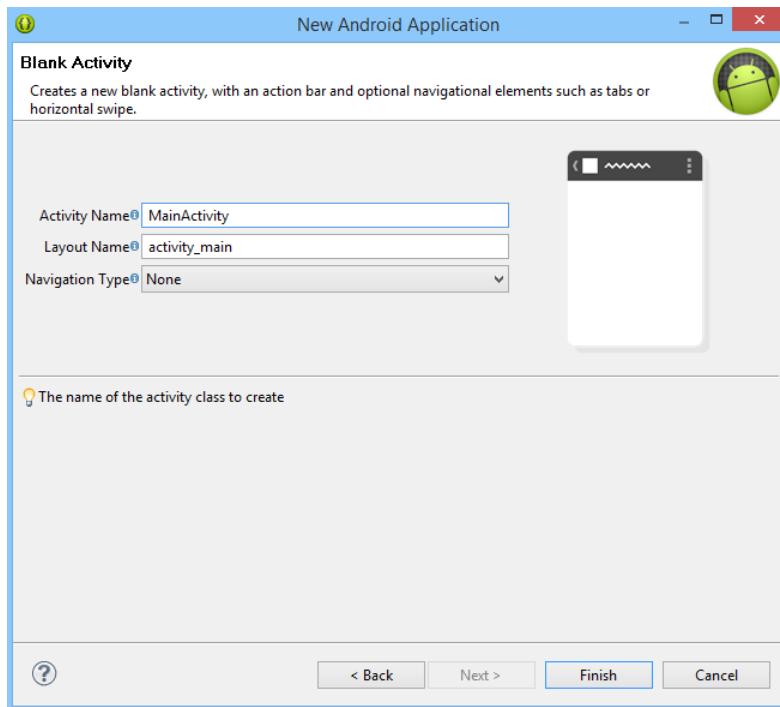
Contoh:



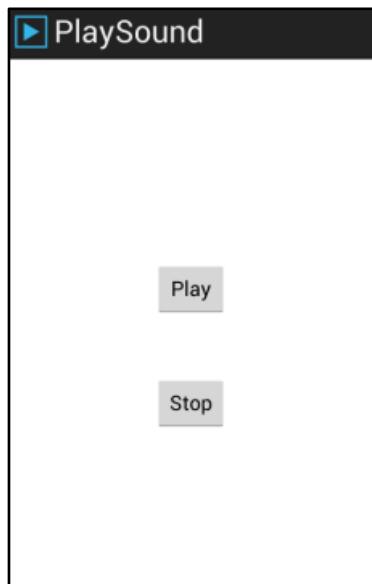
Buatlah project baru dengan nama “PlaySound”.



Untuk nama Activity dan layout saya biarkan default



Selanjutnya tambahkan dua buah button pada layout



Widget	id	Text
Button	btnplay	Play
Button	btnstop	Stop

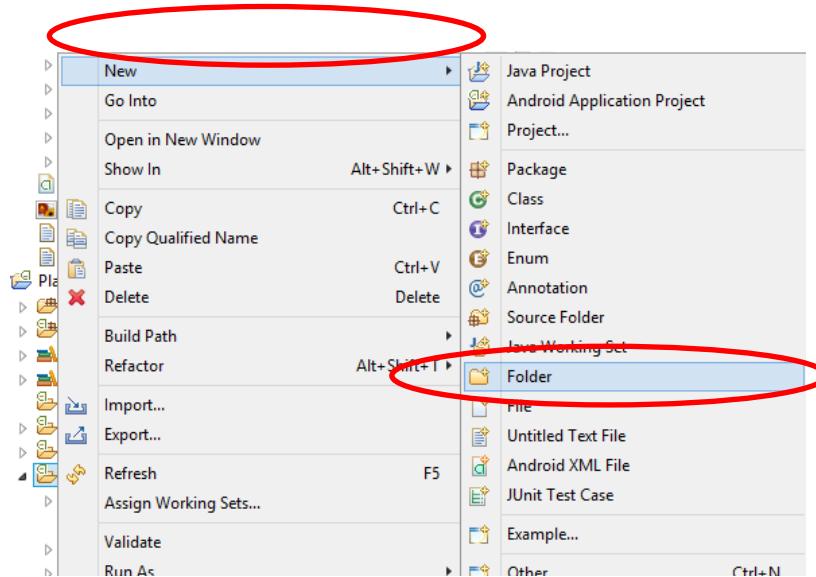
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"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

    <Button
        android:id="@+id	btnplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="161dp"
        android:text="Play"
    />

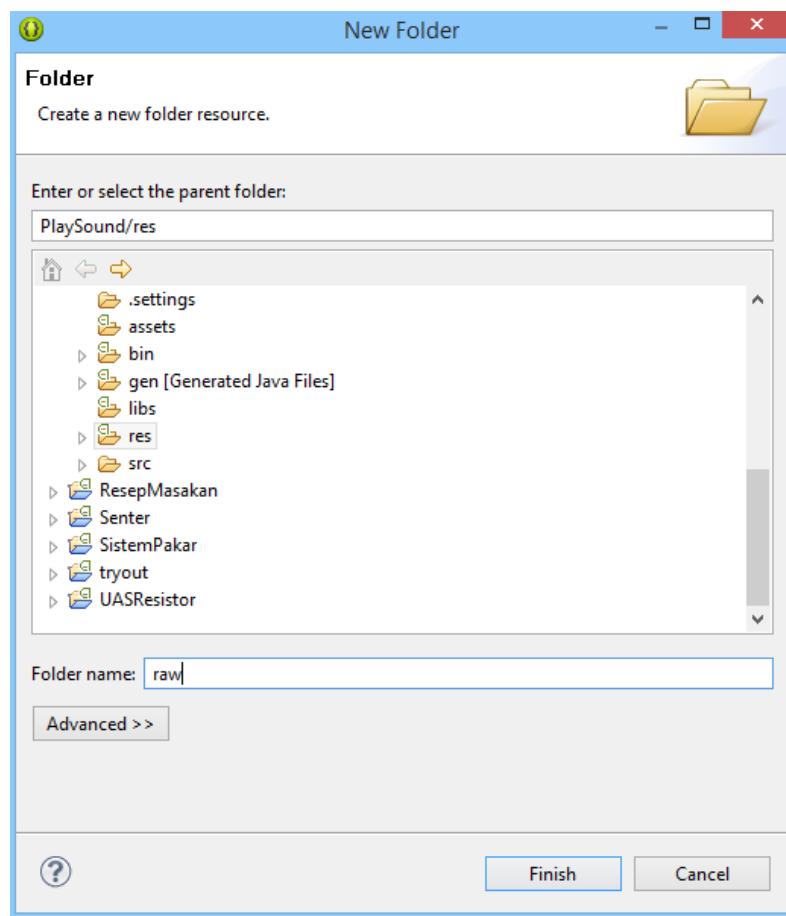
    <Button
        android:id="@+id	btnstop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id	btnplay"
        android:layout_below="@+id	btnplay"
        android:layout_marginTop="52dp"
        android:text="Stop"
    />

</RelativeLayout>
```

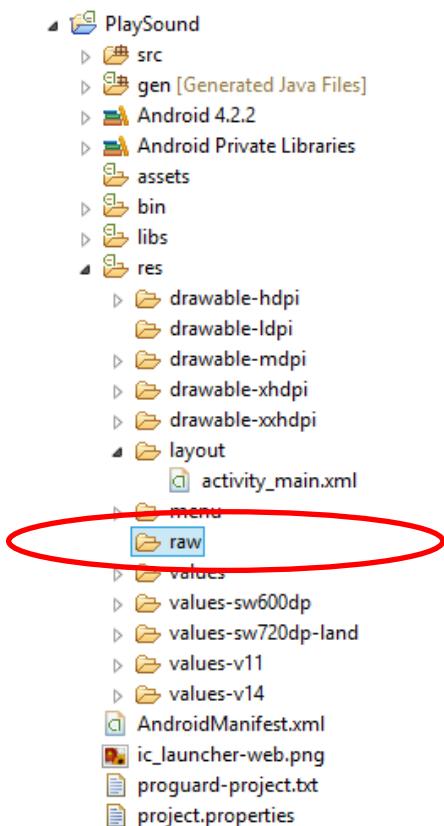
Langkah selanjutnya adalah membuat folder “raw” pada project. Untuk membuat folder, klik kanan pada folder **res** → pilih **new** → **folder** seperti gambar berikut:



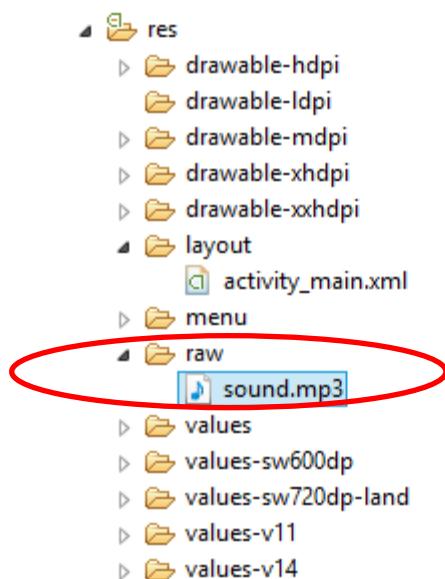
Beri nama folder “raw” kemudian klik finish.



Di folder res akan terdapat folder baru **raw**.



Copy file audio yang telah disiapkan ke folder raw dengan cara drag and drop.



Selanjutnya kode pada MainActivity nya., **Library** yang perlu diimport adalah sebagai berikut:

```
import android.media.MediaPlayer; → library untuk memainkan audio
import android.view.View; → library dari view untuk listener
```

Siapkan variable bernama mp

```
MediaPlayer mp;
```

membuat dua buah method yaitu, method untuk memainkan audio yang diberi nama “playsound”

```
public void playsound(View view) {
    try{
        if(mp.isPlaying()){
            mp.stop();
            mp.release();
        }
    }catch(Exception e){

    }
    //setup sound
    mp=MediaPlayer.create(this, R.raw.sound);

    //agar media tidak looping
    mp.setLooping(false);

    //memainkan suara
    mp.start();
}
```

method untuk menghentikan audio yang diberi nama “stopsound”

```
public void stopsound(View view) {
    //menghentikan suara
    try{
        mp.stop();
    }catch(Exception e){

    }
}
```

method tambahan untuk menghentikan audio ketika tombol back ditekan. Untuk mencegah audio dimainkan walupun aplikasi sudah dihentikan.

```
@Override  
public void onBackPressed() {  
    try{  
        mp.stop();  
    }catch(Exception e){  
  
    }  
    finish();  
}
```

Kode lengkap pada MainActivity

```
package com.playsound;  
  
import android.os.Bundle;  
import android.app.Activity;  
import android.view.Menu;  
  
//1 library  
import android.view.View;  
import android.media.MediaPlayer;  
  
public class MainActivity extends Activity {  
  
    //2. Variabel instance  
    MediaPlayer mp;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

```
//3 method playsound
public void playsound(View view) {
    try{
        if(mp.isPlaying()){
            mp.stop();
            mp.release();
        }
    }catch(Exception e){

    }
    //setup sound
    mp=MediaPlayer.create(this, R.raw.sound);

    //agar media tidak looping
    mp.setLooping(false);

    //memainkan suara
    mp.start();
}

-- 

//4 method stopsound
public void stopsound(View view) {
    //menghentikan suara
    try{
        mp.stop();
    }catch(Exception e){

    }
}

//method untuk menghentikan suara ketika tombol back ditekan
@Override
public void onBackPressed() {
    try{
        mp.stop();
    }catch(Exception e){

    }
    finish();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

}
```

Selanjutnya kembali ke layout untuk menambahkan property onClick pada masing-masing tombol. btnplay methodnya adalah playsound dan btnstop methodnya adalah stopsound.

```

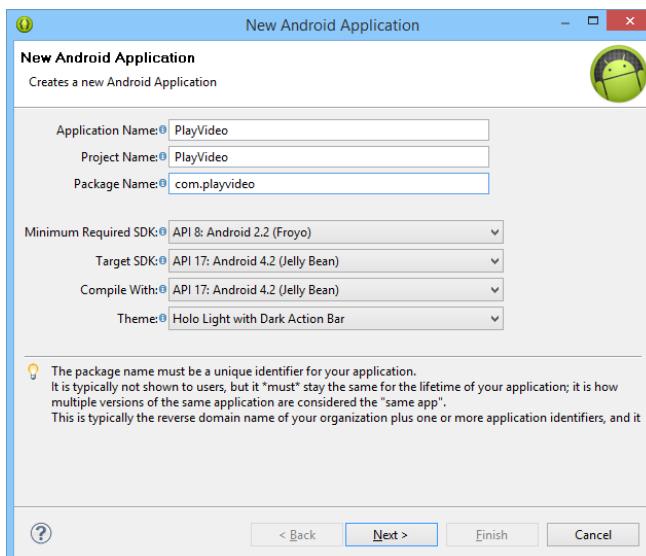
<Button
    android:id="@+id	btnplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="161dp"
    android:onClick="playsound"
    android:text="Play"
/>

<Button
    android:id="@+id	btnstop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id	btnplay"
    android:layout_below="@+id	btnplay"
    android:layout_marginTop="52dp"
    android:onClick="stopsound"
    android:text="Stop"
/>
  
```

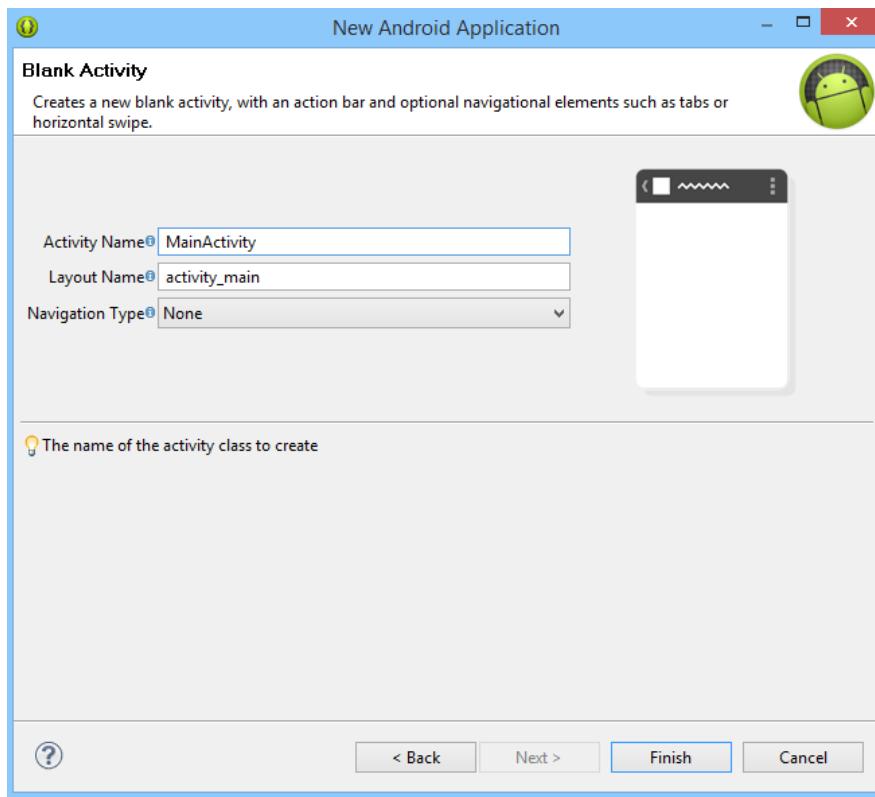
Contoh Implementasi MediaPlayer untuk memainkan video

anda persiapkan file video (rekomendasi file video dengan format 3gp). Nama file video diawali oleh huruf, tanpa spasi, symbol yang diperkenankan hanya underscore(_) dan *lowercase* (tidak ada huruf kapital).

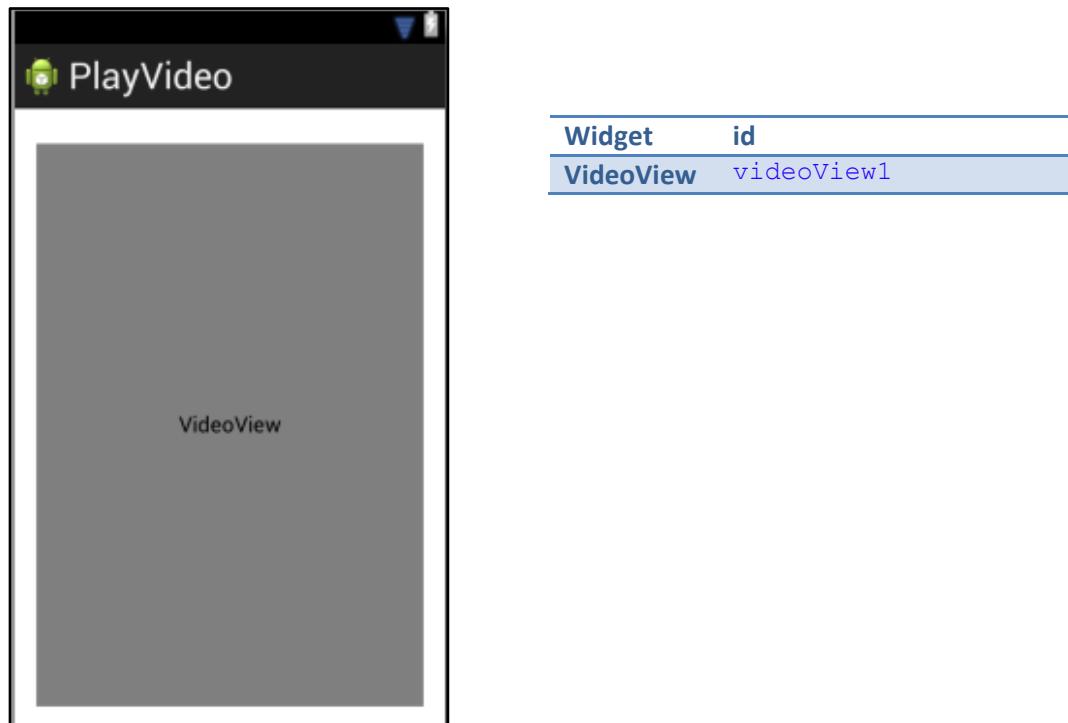
Buatlah project baru dengan nama “**PlayVideo**”.



Untuk nama Activity dan layout biarkan default



Selanjutnya tambahkan widget VideoView(dapat dicari pada pallete Images & Media) pada layout



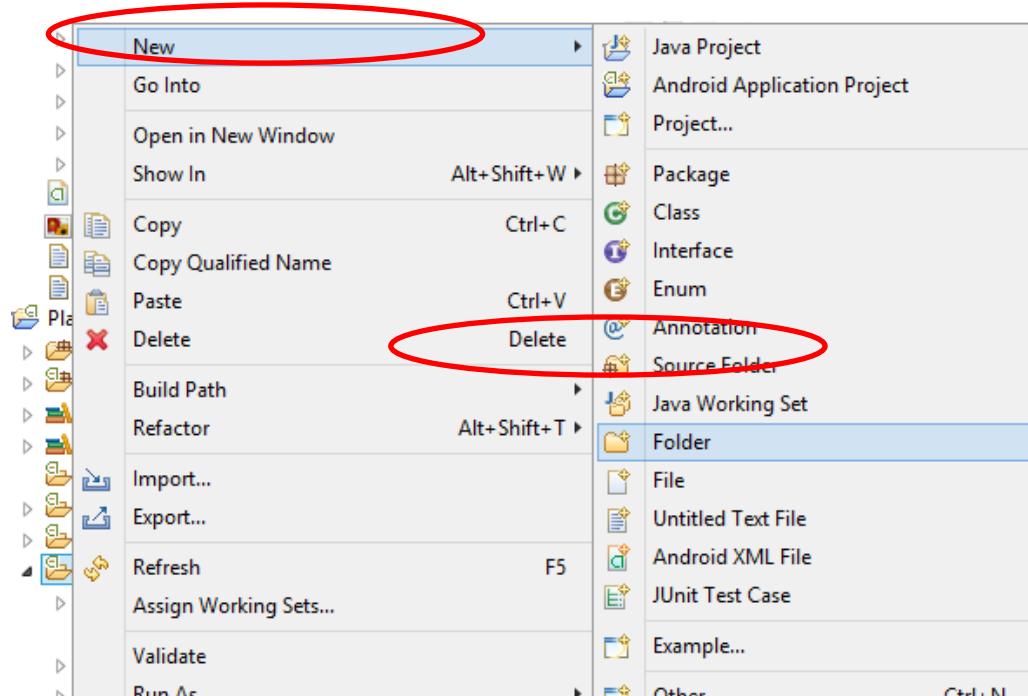
Kode pada layout

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_gravity="center"
    android:layout_height="fill_parent" >

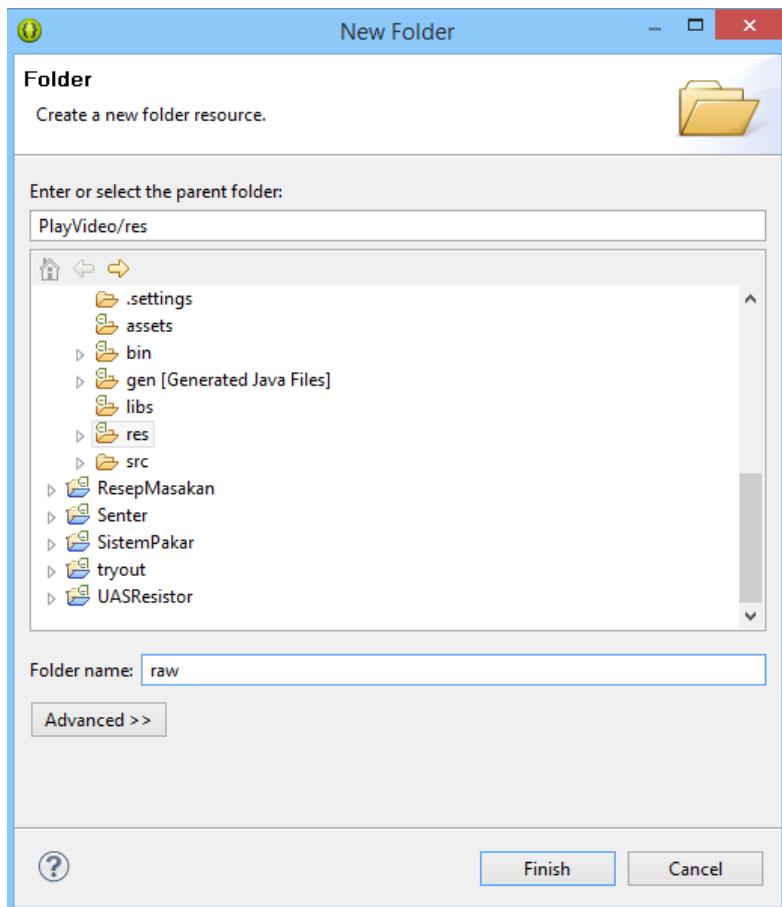
    <VideoView
        android:id="@+id/videoView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

</FrameLayout>
```

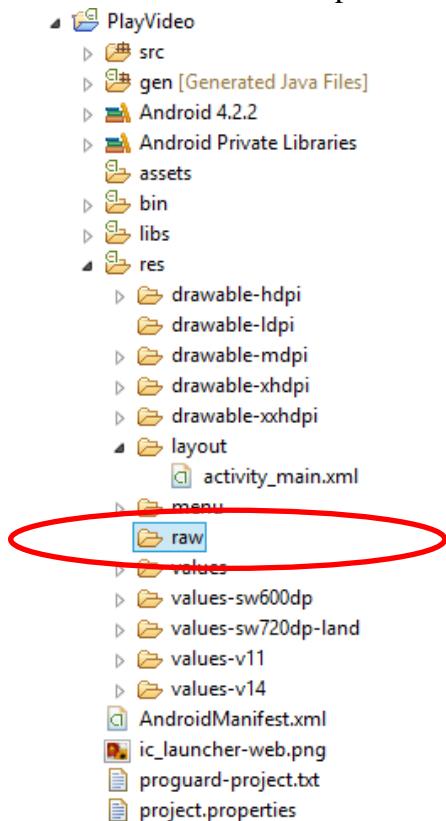
Langkah selanjutnya adalah membuat folder “raw” pada project. Untuk membuat folder, klik kanan pada folder **res** ➔ pilih **new** ➔ **folder** seperti gambar berikut:



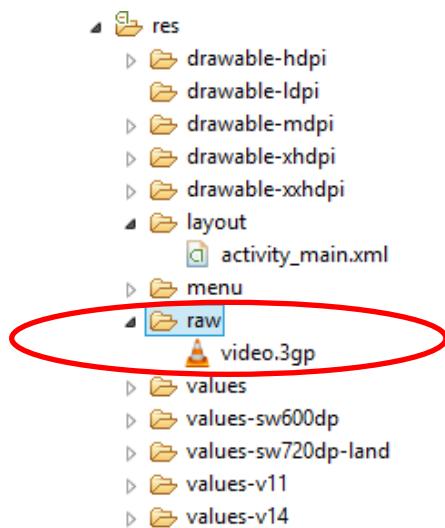
Beri nama folder “raw” kemudian klik finish.



Di folder res akan terdapat folder baru **raw**.



Copy file video yang telah disiapkan ke folder raw dengan cara drag and drop.



Kode Lengkap pada MainActivity

```
package com.playvideo;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

import android.app.AlertDialog;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnPreparedListener;
import android.net.Uri;
import android.widget.MediaController;
import android.widget.VideoView;

public class MainActivity extends Activity {

    private VideoView videoview;
    private int position = 0;
    private ProgressDialog progressDialog;
    private MediaController mediaControls;

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

        if (mediaControls == null) {
            mediaControls = new MediaController(this);
        }

        //inisialisasi VideoView pada layout
        videoview=(VideoView)findViewById(R.id.videoView1);

        // membuat progressbar
        progressDialog = new ProgressDialog(this);
    }
}
```

```

// Set progressbar title
progressDialog.setTitle("Memainkan video di android");
// Set progressbar message
progressDialog.setMessage("Loading...");

progressDialog.setCancelable(false);
// menampilkan progressbar
progressDialog.show();

try {
    videoview.setMediaController(mediaControls);
    videoview.setVideoURI(Uri.parse("android.resource://" +
getPackageName() + "/" + R.raw.video));

} catch (Exception e) {}

videoview.requestFocus();
videoview.setOnPreparedListener(new OnPreparedListener()
{

// Close the progress bar and play the video
public void onPrepared(MediaPlayer mp) {
progressDialog.dismiss();
    videoview.seekTo(position);
    if (position == 0) {
        videoview.start();
    } else {
        videoview.pause();
    }
}
});

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is
present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

//saat di pause
@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
    super.onSaveInstanceState(savedInstanceState);
    savedInstanceState.putInt("Position",
videoview.getCurrentPosition());
    videoview.pause();
}

//saat di restore
@Override
public void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    position = savedInstanceState.getInt("Position");
    videoview.seekTo(position);
}
}

```

Latihan

Membuat aplikasi multimedia.

MODUL 14

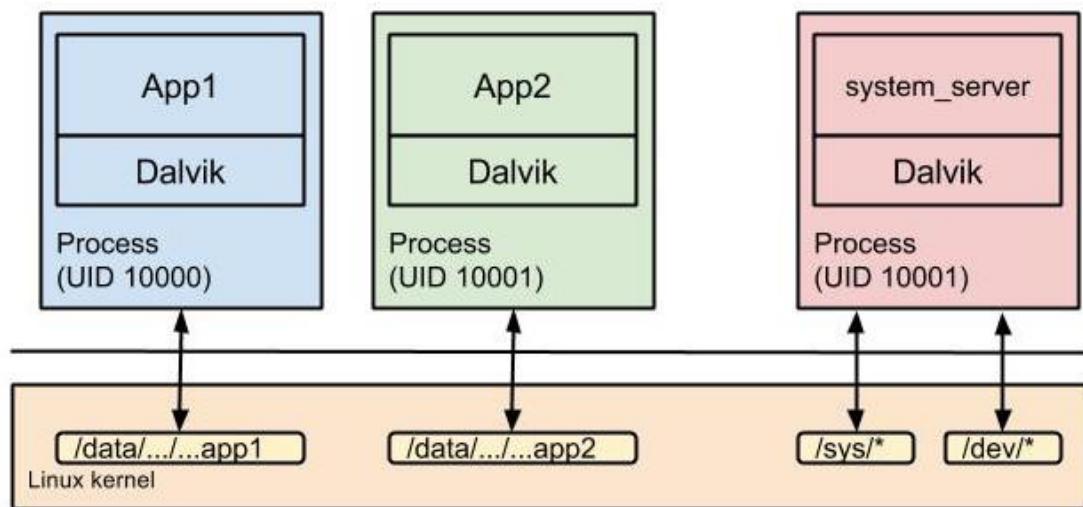
MENGAMANKAN APLIKASI ANDROID

Tujuan

- Mahasiswa mampu mengidentifikasi model keamanan untuk Platform Android.
- Mahasiswa mampu menggunakan permissions untuk operasi tertentu.

Mengidentifikasi Model Keamanan Platform Android

Untuk mencegah aplikasi mengakses user data atau data yang dimiliki oleh aplikasi lain, Android mengimplementasikan kernel-level security yang bertugas mengisolasi aplikasi dari aplikasi lain ataupun dari sistem. Arsitektur kemanan android menyertakan teknik linuk kernel dan sandboxing. Kernel mengisolasi (sandbox) aplikasi dari aplikasi lainnya dengan memberikan User ID (UID) kepada setiap aplikasi. Android memberikan Linux User ID (UID) ketika penginstalan aplikasi dan tetap bertahan selama aplikasi tersebut ada diperangkat. Setiap aplikasi akan mempunyai UID yang berbeda-beda.



Keuntungan dari Model Keamanan Android antara lain, proteksi komponen sehingga tidak bisa diakses dari aplikasi lain, Proteksi terhadap Application Programming Interface karena dalam penggunannya harus mendapatkan ijin dari pengguna, access control policy disederhanakan dengan mengijinkan pengembang untuk menentukan perijinan dari interface mereka, Pembatasan akses karena pada android, aplikasi tidak mendapatkan ijin untuk melakukan operasi yang dapat merusak sistem dan Unique Signature karena pada android semua aplikasi memerlukan signature yang unik untuk aplikasi tersebut. Kekurangan

dari model keamanan android adalah perijinan diandroid berdasarkan aturan ijinkan atau tolak. Segala perijinan tergantung dari pengguna.

Untuk dapat membuat dua aplikasi berjalan dalam satu proses dengan menambahkan `android:SharedUserId` pada AndroidManifest. Contoh penerapan.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.sample.myapp"
  android:sharedUserId="com.sample.myapp.share"
  android:versionCode="1"
  android:versionName="1.0" >
```

Menggunakan Permission Untuk Operasi Tertentu

Permission adalah mekanisme kemanan level applikasi (app-level security) yang mengijinkan pengguna untuk membatasi akses ke komponen aplikasi. Hal ini digunakan untuk mencegah aplikasi menggunakan aplikasi untuk mengkorupsi data ataupun mengakses data yang sensitif yang terdapat pada perangkat. Untuk menerapkan permission pada aplikasi menggunakan tag `<uses-permission/>`. Contoh penerapan.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Beberapa permission yang disediakan oleh class Manifest.permission pada package android.Manifest.

Permission	Kegunaan
ACCESS_NETWORK_STATE	Mengakses infomasi jaringan
ACCESS_WIFI_STATE	Mengakses informasi jaringan WIFI
BLUETOOTH	Menghubungkan dengan perangkat yang terhubung Bluetooth
BROADCAST_SMS	Menbroadcast notifikasi penerimaan SMS
CALL_PHONE	Memulai panggilan telepon tanpa melalui tampilan telepon
CAMERA	Mengakses perangkat kamera
CHANGE_WIFI_STATE	Mengganti status konektivitas WIFI
INSTALL_PACKAGES	Menginstall Packages
INTERNET	Membuka soket jaringan
READ_CONTACT	Membaca data kontak pengguna
READ_SMS	Membaca SMS

RECEIVE_SMS	Memonitor pesan SMS yang masuk
SEND_SMS	Mengirim pesan SMS
SET_ALARM	Membroadcast intent untuk menset alarm
WRITE_CONTACTS	Menulis data kontak tapi tidak membaca data kontak
WRITE_EXTERNAL_STORAGE	Menulis ke perangkat eksternal
WRITE_SMS	Menulis pesan SMS

MODUL 15

Testing dan Deploy Aplikasi Android

Tujuan

- Mahasiswa mampu mentesting aplikasi android.
- Mahasiswa mampu mendeploy aplikasi android.

Mentesting Aplikasi Android

Sebelum Aplikasi Android di Upload ke Google Playstore. Aplikasi harus ditesting untuk mengetahui kelayakan aplikasi digunakan oleh pengguna. Pertimbangan dalam melakukan tes untuk aplikasi android antara lain.

- Mengubah orientasi, android mempunyai kemampuan untuk mendeteksi perubahan orientasi perangkat. Hal ini dapat mengijinkan pengguna untuk mengubah orientasi dari portrait menjadi landscape. Secara umum, android akan mendestroy dan merestart Activity yang sedang berjalan ketika orientasi perangkat terdeteksi. Hal yang dipastikan dalam pengujian aplikasi terkait perubahan orientasi adalah memastikan tampilan di redraw secara benar dan tampilan antarmuka mampu menangani perubahan orientasi perangkat. Selain itu aplikasi harus mampu menangani kondisi untuk menangani perubahan orientasi sehingga pengguna tidak kehilangan inputan yang sudah dimasukkan pada antar muka.
- Perubahan dalam konfigurasi perangkat, secara umum Activity yang sedang berjalan akan destroy atau restart ketika terdapat perubahan dalam konfigurasi perangkat. Programmer harus mengecek apakah aplikasi mengupdate sendiri dalam rangka merespon perubahan konfigurasi secara benar.
- Daya hidup baterai, aplikasi android yang dikembangkan harus dioptimalisasi untuk menggunakan daya sedikit mungkin dari baterai. Aplikasi harus di tes penggunaan daya baterainya.
- Ketergantungan pada sumber daya eksternal. Aplikasi yang menggunakan sumber daya eksternal seperti internet, SMS dan GPS harus diuji tingkah laku dan performanya ketika sumber daya tersebut tidak ada. Sebagai contoh aplikasi menggunakan koneksi internet, tiba-tiba koneksi internet terputus. Dalam kondisi ini aplikasi harus mampu menampilkan notifikasi internet terputus pada pengguna.

Android SDK menyediakan berbagai tool yang powerfull seperti Junit, mongkeyrunner dan hierarchy AndroidApp. Tools tersebut membantu pengembang program android untuk melakukan test-driven development pada aplikasi android.

Contoh Testing Menggunakan Junit

Buatlah sebuah project dengan nama “**FindGreatest**” dengan nama Activity “**FindGreatestActivity**” dan nama layout “**main**”.

Berikut adalah kode untuk main.xml

```
<?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:orientation="vertical" >

    <EditText android:layout_width="match_parent"
        android:text="" android:layout_height="wrap_content"
        android:id="@+id/editText1" android:inputType="text">
    </EditText>
    <EditText android:layout_width="match_parent"
        android:text="" android:layout_height="wrap_content"
        android:id="@+id/editText2" android:inputType="text">
    </EditText>
    <EditText android:layout_width="match_parent"
        android:text="" android:layout_height="wrap_content"
        android:id="@+id/editText3" android:inputType="text">
    </EditText>
    <LinearLayout android:id="@+id/LinearLayout1" android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView android:layout_width="wrap_content" android:id="@+id/textView1"
            android:layout_height="wrap_content" android:text="Greatest number is:
        "></TextView>
        <TextView android:layout_width="wrap_content" android:id="@+id/textView2"
            android:layout_height="wrap_content" android:text=""></TextView>
    </LinearLayout>
    <Button android:layout_width="match_parent"
        android:text="Find greatest" android:layout_height="wrap_content"
        android:id="@+id/button1">
    </Button>
</LinearLayout>
```

Kode untuk FindGreatest.java

```
package com.example.findgreatest;
```

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class FindGreatestActivity extends Activity {
    private final static String TAG= "FIND GREATEST";
    private EditText mFirst;
    private EditText mSecond;
    private EditText mThird;
    private TextView mResult;
    private Button mFind;
    private int mGreatest=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        mFirst = (EditText)findViewById(R.id.editText1);
        mSecond = (EditText)findViewById(R.id.editText2);
        mThird = (EditText)findViewById(R.id.editText3);
        mResult = (TextView)findViewById(R.id.textView2);
        mFind = (Button)findViewById(R.id.button1);
        mFind.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                int num1 = Integer.parseInt(mFirst.getText().toString());
                Log.i(TAG, String.valueOf(num1));
                int num2 = Integer.parseInt(mSecond.getText().toString());
                Log.i(TAG, String.valueOf(num2));
                int num3 = Integer.parseInt(mThird.getText().toString());
                Log.i(TAG, String.valueOf(num3));
                if(num1>num2){
                    if(num1>num3){
                        mGreatest = num1;
                        Log.i(TAG, String.valueOf(mGreatest));
                    }
                    else{
                        mGreatest=num3;
                        Log.i(TAG, String.valueOf(mGreatest));
                    }
                }
                else if(num2>num1){
                    if(num2>num3){
                        mGreatest = num2;
                        Log.i(TAG, String.valueOf(mGreatest));
                    }
                    else{
                        mGreatest=num3;
                    }
                }
            }
        });
    }
}
```

```

        Log.i(TAG, String.valueOf(mGreatest));
    }

    mResult.setText(String.valueOf(mGreatest));
}

});

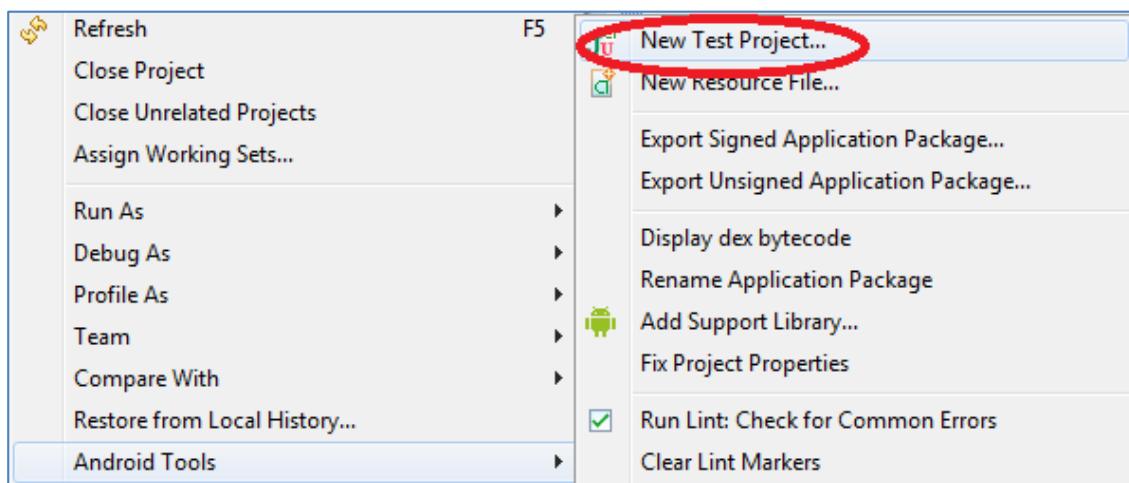
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is
    getMenuInflater().inflate(R.menu.find_greatest, menu);
    return true;
}
}

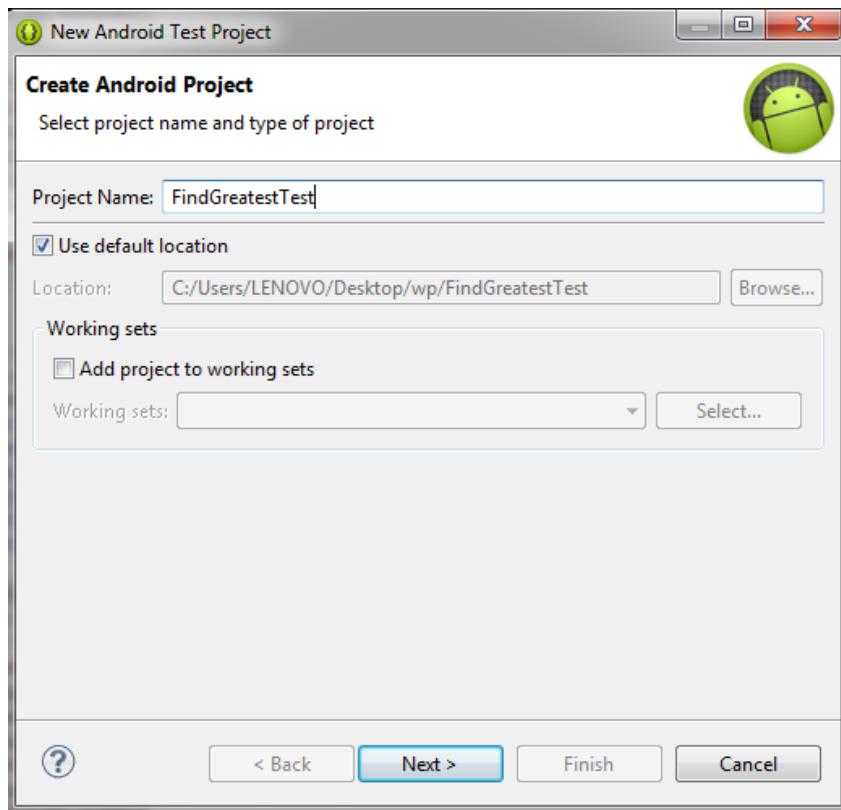
```

Selanjutnya membuat test project untuk aplikasi

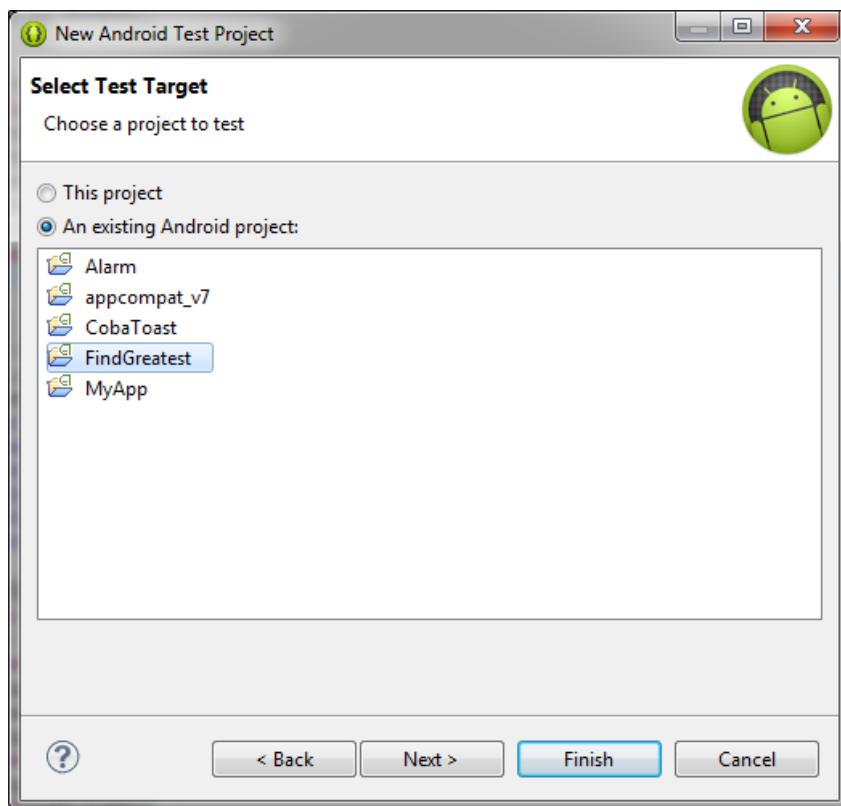
Pada **FindGreatest** folder di **Package Explorer**, klik kanan pilih **Android Tools → Test Project**.



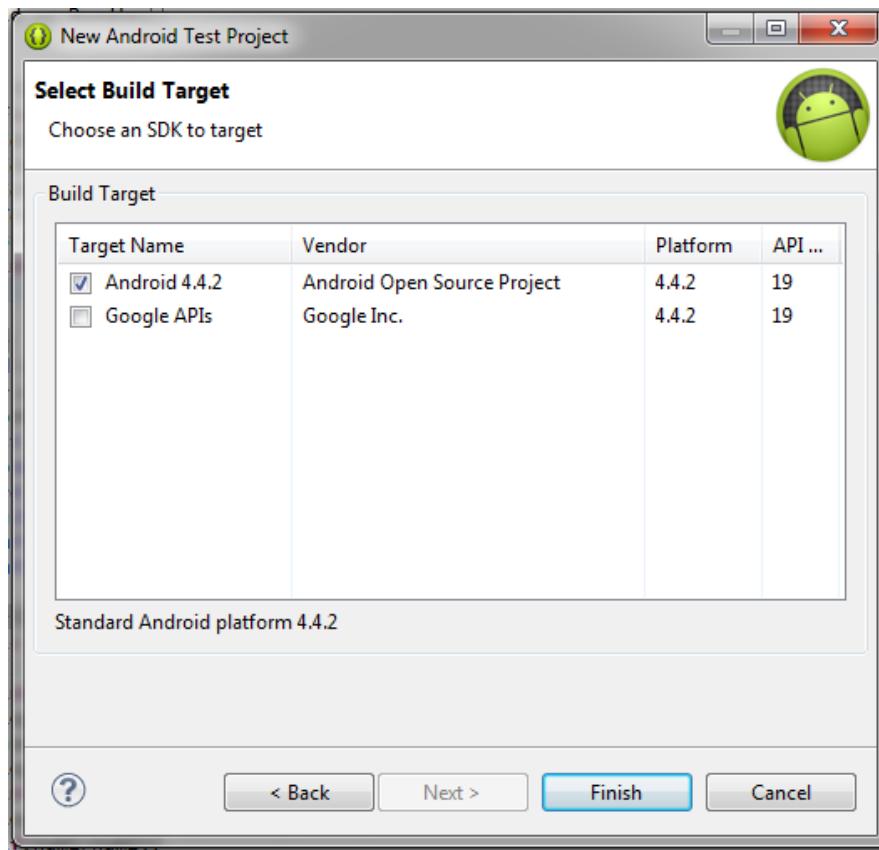
Selanjutnya muncul jendela “**New Android Test Project**”. Berinama project name dengan nama “**FindGreatestTest**”.



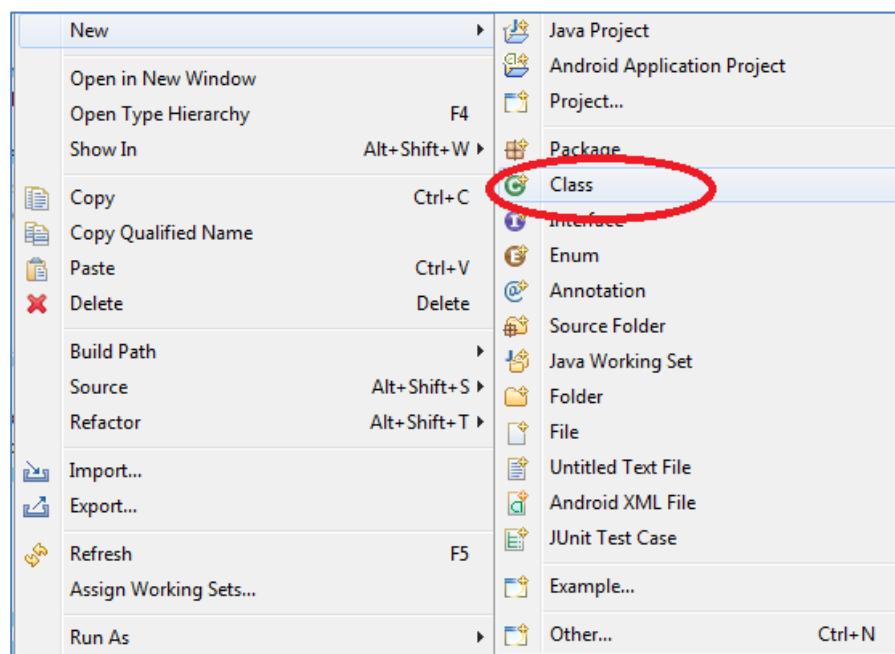
Pilih Next, kemudian pada Select Test Target Pilih Project **FindGreatest**.



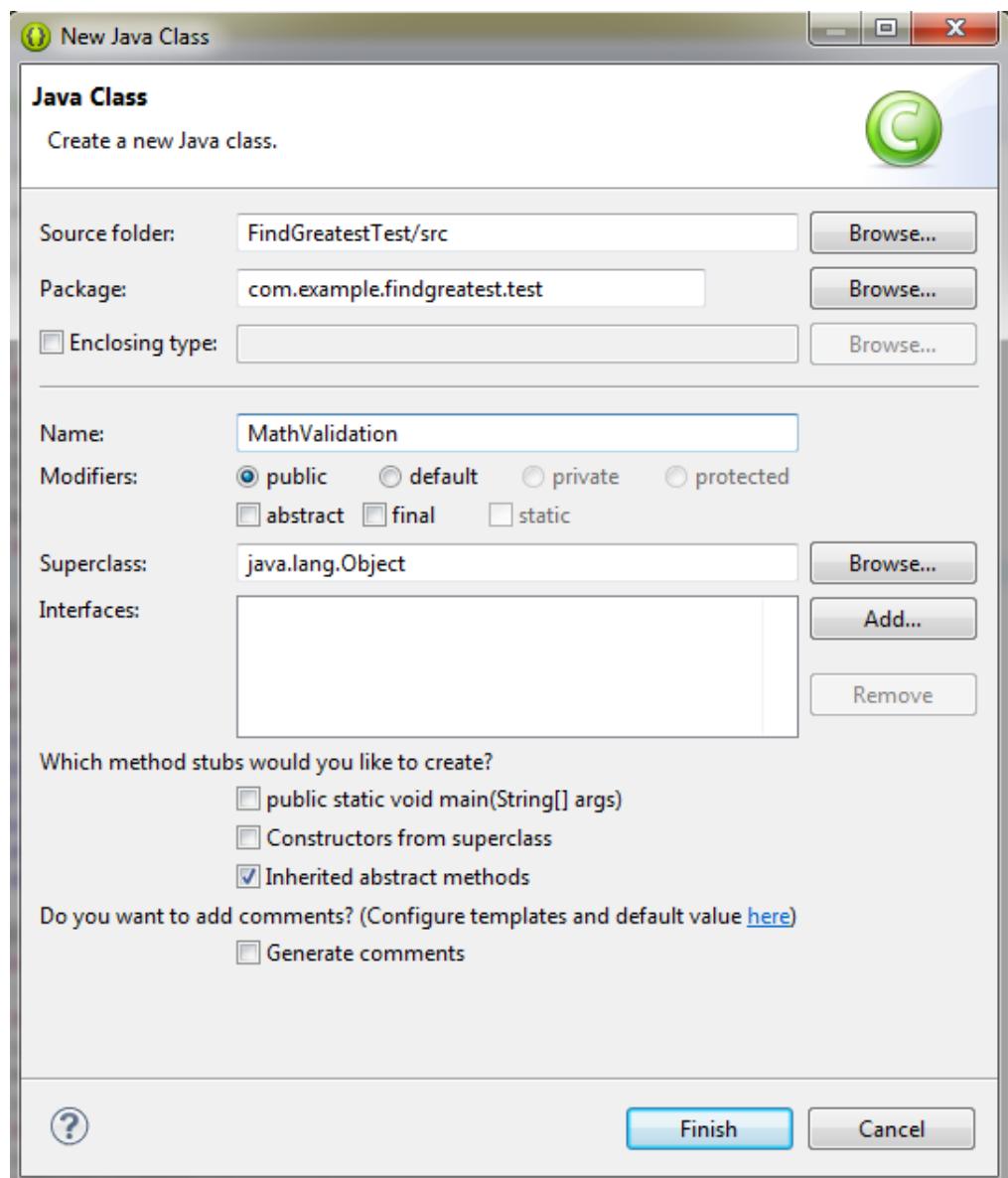
Pilih Next. Selanjutnya pada Select Build Target pilih Android 4.4.2 dan pilih finish.



Pada Package Explorer akan ada project baru dengan nama “**FindGreatestTest**”. Buka Folder tersebut dan cari **src**. Klik kanan folder **src** pilih **New → Class**.



Pada Jendela **New Java Class** beri nama “**MathValidation**”



Berikut Kode untuk MathValidation.java

```

package com.example.findgreatest.test;

import com.example.findgreatest.*;
import com.example.findgreatest.R;
import android.test.ActivityInstrumentationTestCase2;
import android.test.TouchUtils;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MathValidation extends
ActivityInstrumentationTestCase2<FindGreatestActivity> {
    private TextView mResult;
    private EditText mFirst;
    private EditText mSecond;
    private EditText mThird;
    private Button mFind;

    public MathValidation(String name) {
        super(FindGreatestActivity.class);
        setName(name);
    }
    protected void setUp() throws Exception {
        super.setUp();
        FindGreatestActivity findGreatest = getActivity();
        mResult = (TextView)findGreatest.findViewById(R.id.textView2);
        mFirst = (EditText)findGreatest.findViewById(R.id.editText1);
        mSecond = (EditText)findGreatest.findViewById(R.id.editText2);
        mThird = (EditText)findGreatest.findViewById(R.id.editText3);
        mFind = (Button)findGreatest.findViewById(R.id.button1);
    }
    @Override
    protected void tearDown() throws Exception {
        super.tearDown();
    }

    private static final String NUMBER_20 = "2 0";
    private static final String NUMBER_70 = "7 0";
    private static final String NUMBER_15 = "1 5";
    private static final String GREATEST_NUMBER = "70";

    public void testGreatestValue() {
        mFirst.clearComposingText();
        mSecond.clearComposingText();
        mThird.clearComposingText();
        TouchUtils.tapView(this, mFirst);
        sendKeys(NUMBER_20);
        TouchUtils.tapView(this, mSecond);
        sendKeys(NUMBER_70);
        TouchUtils.tapView(this, mThird);
        sendKeys(NUMBER_15);
        getActivity().runOnUiThread(
            new Runnable() {
                public void run() {
                    mFind.performClick();
                }
            });
    }
}

```

```

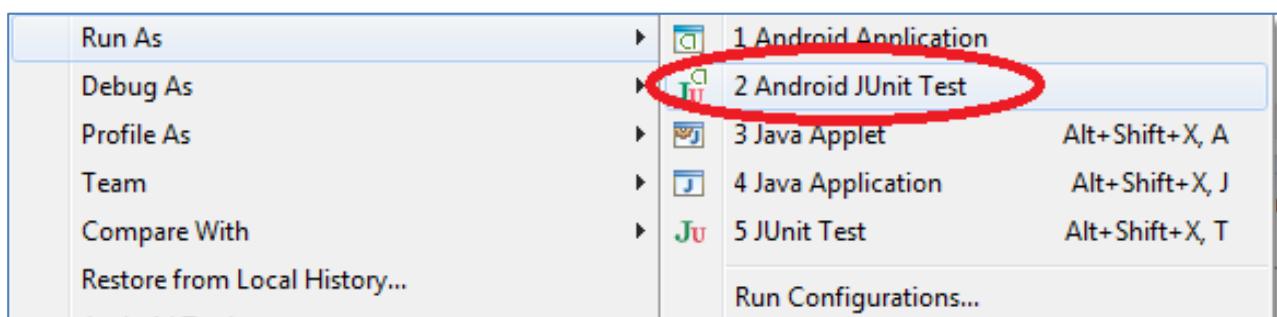
        getInstrumentation().waitForIdleSync();
        String result =mResult.getText().toString();
        assertTrue("Greatest number is: "+result,
result.equals(GREATEST_NUMBER));
    }

    private static final String NUMBER_2_1 = "2 1";
    private static final String NUMBER_40 = "4 0";
    private static final String NUMBER_5_5 = "5 5";
    private static final String GREATEST_DECIMAL_NUMBER = "55";

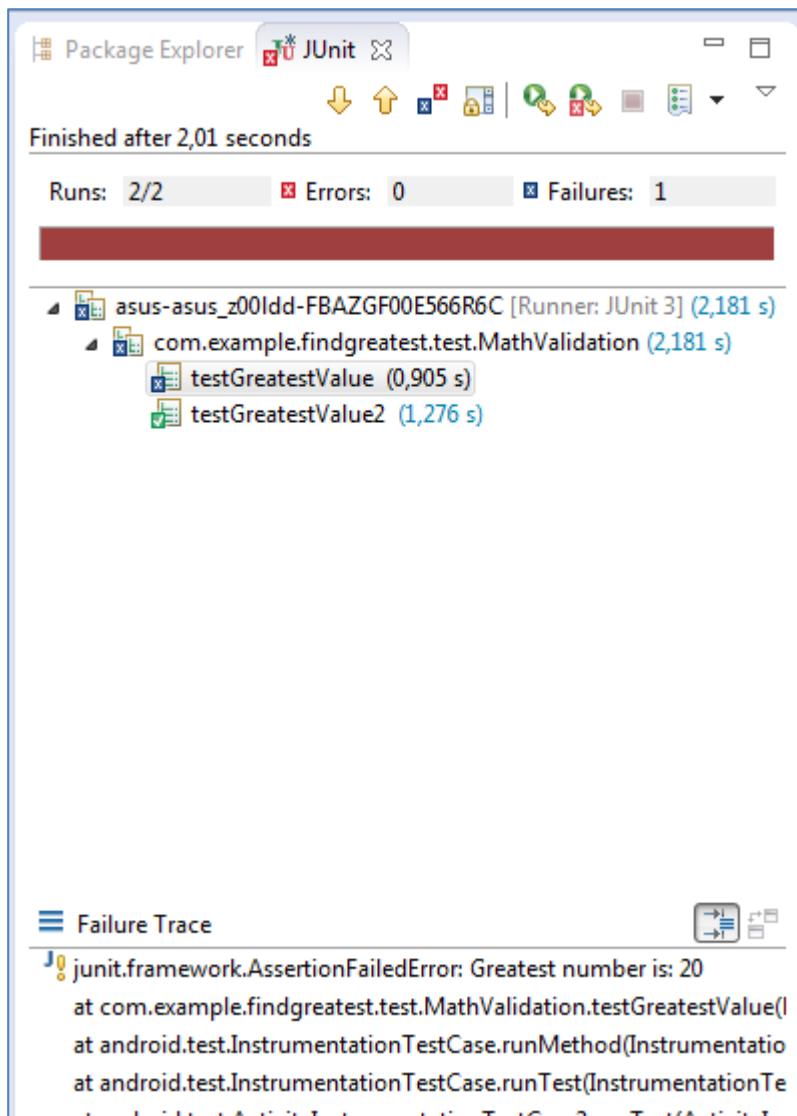
    public void testGreatestValue2(){
        mFirst.clearComposingText();
        mSecond.clearComposingText();
        mThird.clearComposingText();
        TouchUtils.tapView(this, mFirst);
        sendKeys(NUMBER_2_1);
        TouchUtils.tapView(this, mSecond);
        sendKeys(NUMBER_40);
        TouchUtils.tapView(this, mThird);
        sendKeys(NUMBER_5_5);
        getActivity().runOnUiThread(
            new Runnable() {
                public void run() {
                    mFind.performClick();
                }
            });
        getInstrumentation().waitForIdleSync();
        String result =mResult.getText().toString();
        assertTrue("Greatest number is "+result,
result.equals(GREATEST_DECIMAL_NUMBER));
    }

}
  
```

Selanjutnya mulai test dengan mengklik kanan project **FindGreatestTest** kemudian pilih **Run As→Android Junit Test**.



Setelah ditest ternyata muncul tampil pesan error pada tab JUnit disebelah tab Package Explorer.



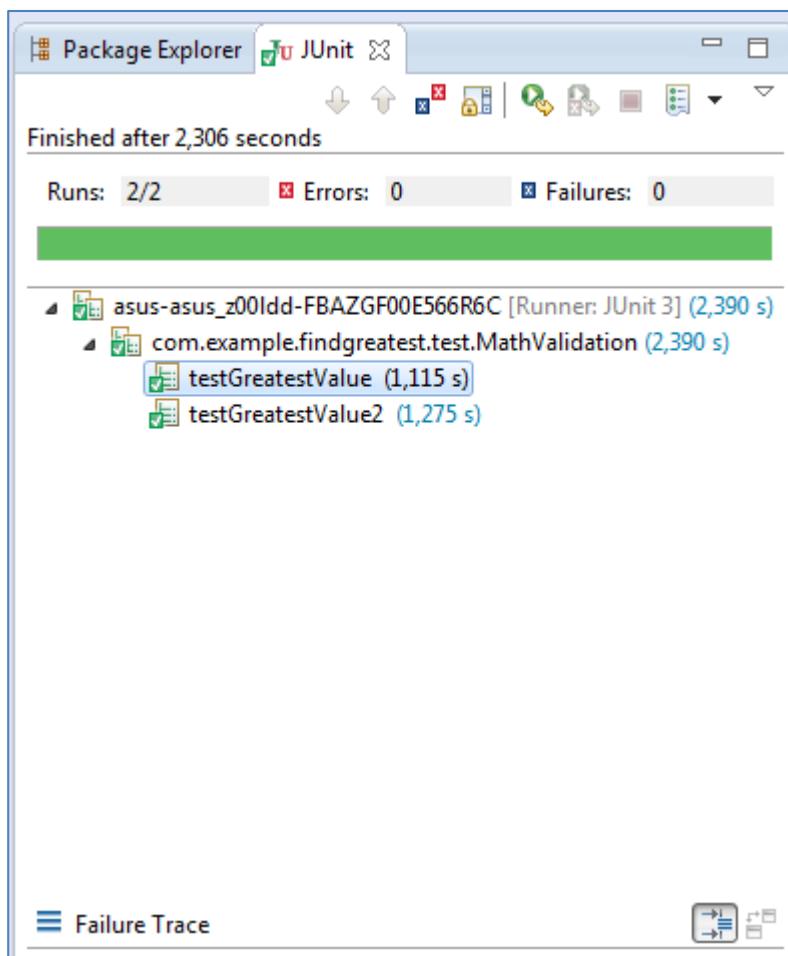
Perbaiki code di Activity **FindGreatestActivity** di project “**FindGreatest**”.

```
        else if(num2>num1){
            if(num2>num3){
                mGreatest = num1;
                Log.i(TAG, String.valueOf(mGreatest));
            }
            else{
                mGreatest=num3;
                Log.i(TAG, String.valueOf(mGreatest));
            }
        }
```

Menjadi

```
else if(num2>num1){  
    if(num2>num3){  
        mGreatest = num2;  
        Log.i(TAG, String.valueOf(mGreatest));  
    }  
    else{  
        mGreatest=num3;  
        Log.i(TAG, String.valueOf(mGreatest));  
    }  
}
```

Setelah di run ulang. Terlihat kode sudah benar



Mendeploy Aplikasi Android

Menginstall aplikasi pada perangkat nyata disebut dengan mendeploy Aplikasi Android. Dengan bermacam-macam perangkat serta konfigurasi masing-masing. Beberapa pertimbangan dalam mendeploy aplikasi antara lain.

- Ukuran Layar yang beragam, banyaknya pengembang perangkat yang menggunakan sistem operasi android tentu menjadi tantangan bagi pengembang aplikasi untuk membuat aplikasi tampil baik diberbagai ukuran layar. Android membagi ukuran layar menjadi 4 yaitu.
 - Small (426dp x 320dp)
 - Normal (470dp x 320dp)
 - Large (640dp x 480dp)
 - Xlarge(960dp x 720dp)

Dengan densitas

- **ldpi** (low)
- **mdpi**(medium)
- **hdpi**(high)
- **xhdpi**(extra high)

Disarankan dalam mengembangkan antar muka aplikasi android. Pengembang aplikasi harus memastikan berberapa hal

- Gunakan **wrap_content**, **fill_parent**, atau unit **dp** pada layout XML.
- Jangan gunakan hard-coded pixel untuk file java.
- Jangan gunakan Absolute layout.
- Sediakan gambar untuk densitas layar yang berbeda pada masing-masing folder di drawable.
- Kompabilitas Android, sistem operasi android banyak digunakan oleh berbagai tipe perangkat yang berbeda. Hal ini membuat pengembang harus membuat aplikasi mereka kompatibel di berbagai perangkat. Sebagai pengembang aplikasi harus memastikan aplikasi memenuhi fitur yang jelas pada manifest file.

Setelah mengembangkan dan mentesting aplikasi, selanjutnya aplikasi harus dipublish untuk dapat mencapai pengguna. Sebelum mempublish aplikasi, android mensyaratkan pengembang aplikasi untuk memberikan tandatangan digital pada paket aplikasi untuk mencegah spam atau aplikasi mencurigakan di publish di Playstore. Memberi tandatangan juga membuat Playstore mudah mengenali pengembang dari aplikasi.

Untuk menpublish aplikasi ke playstore. Pengembang memerlukan untuk mendaftar menggunakan akun Google pada alamat berikut <https://play.google.com/apps/publish/signup/>.