



disusun oleh:

Indah Puji Astuti, S.Kom., M.Kom
Ghulam Asrofi Buntoro, S.T., M.Eng

LABORATORIUM TEKNIK INFORMATIKA
FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH PONOROGO

KATA PENGANTAR

Alhamdulillah puji syukur kami panjatkan kehadirat Allah SWT yang telah memberikan semua nikmat dan hidayahnya sehingga penulis dapat menyelesaikan penulisan Modul Praktikum Pemrograman Bergerak ini dengan baik. Tujuan dari disusunnya modul ini adalah untuk pendukung perkuliahan Pemrograman Bergerak. Dicitak terbatas untuk Fakultas Teknik, Prodi teknik Informatika, Universitas Muhammadiyah ponorogo. Modul ini adalah edisi pertama yang terdiri dari 12 bab, yaitu Install Android Studio, Pengenalan Android, Menjalankan Aplikasi Android, Activity, Intent, Menampilkan Gambar dengan Java dan XML, Pembuatan View, Pembuatan ViewGroup, Pembuatan Style dan Themes, Membuat Efek Suara, Studi Kasus 1, dan Studi Kasus 2.

Dalam proses penyusunan modul ini tentunya tidak hanya usaha dari penulis saja, namun ada banyak pihak yang turut membantu dan memberikan dukungan baik pikiran, tenaga maupun support untuk keberhasilan penulisan modul ini. Untuk itu, dengan kerendahan hati penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan kepada penulis sehingga modul ini berhasil disusun dan diterbitkan.

Modul ini tentunya tidak luput dari banyak kekurangan, untuk itu kami sebagai penulis sangat terbuka akan adanya kritik, saran serta masukan dari pembaca sekalian.

Ponorogo, 7 Januari 2023

Penulis

DAFTAR ISI

Cover	i
Kata Pengantar	ii
Daftar Isi	iii
Tata Tertib Praktikum	iv
BAB 1 Install Android Studio	1
BAB 2 Pengenalan Android	11
BAB 3 Menjalankan Aplikasi Android	23
BAB 4 Activity	38
BAB 5 Intent	48
BAB 6 Menampilkan Gambar dengan Java dan XML	60
BAB 7 Pembuatan View	69
BAB 8 Pembuatan ViewGroup	77
BAB 9 Pembuatan Style dan Themes	86
BAB 10 Membuat Efek Suara	93
BAB 11 Studi Kasus 1.....	100
BAB 12 Studi Kasus 2.....	101
BIBLIOGRAPHY	102
Format Laporan Praktikum	103

Tata Tertib Praktikum

Tata Tertib berikut untuk kebaikan bersama

1. Praktikan hadir 5 menit sebelum praktikum dimulai.
2. Toleransi keterlambatan adalah 15 menit. Bila praktikan terlambat lebih dari 15 menit tidak diperkenankan mengikuti praktikum.
3. Praktikan berpakaian bebas rapi dan sopan. Praktikan tidak diperkenankan memakai kaos oblong, topi, sendal. Bagi cowok dilarang memakai anting – anting.
4. Praktikan Mengisi daftar hadir kemudian mengumpulkan kartu praktikum untuk diparaf asisten jaga.
5. Praktikan duduk sesuai dengan yang ditentukan dan menyiapkan modul praktikum serta buku penunjang praktikum.
6. Selama praktikum berlangsung, semua praktikan wajib menjaga ketertiban praktikum.
7. Praktikan tidak diperkenankan melakukan koneksi dengan internet selama praktikum berlangsung, kecuali memang dalam praktek tersebut membutuhkan koneksi internet.
8. Perangkat komunikasi dalam kondisi silent dimasukkan ke dalam tas.
9. Praktikan mengerjakan tugas di setiap akhir praktikum.
10. Praktikan Menjaga kebersihan peralatan laboratorium dan laboratorium.
11. Praktikan Tidak diperkenankan melakukan konfigurasi ulang terhadap hardware dan software di laboratorium.
12. Selesai praktikum, praktikan mematikan komputer dan merapikan peralatan.
13. Diberikan kesempatan inhal 2 kali pada akhir praktikum (praktikan berhalangan hadir dan memberi keterangan yang sah).
14. Praktikan wajib mengumpulkan laporan praktikum mingguan kepada instruktur sebelum praktikum dimulai.
14. Pada akhir praktikum, praktikan diwajibkan mengumpulkan laporan akhir praktikum.
15. Pada akhir praktikum akan diadakan responsi (di sesuaikan dengan jadwal ujian dan praktikan harus hadir 100% untuk dapat mengikuti responsi).
16. Bagi yang melanggar tata tertib akan dikenakan sangsi.

BAB 1 – INSTALL ANDROID STUDIO DAN SDK (ANDROID SOFTWARE DEVELOPMENT KIT)

1.1. Identitas

Kompetensi

1. Mengetahui syarat-syarat untuk menginstall Android Studio
2. Dapat menginstall tools yang dibutuhkan (Java, Android Studio, SDK)

Topik

1. Syarat atau spesifikasi minimum komputer yang mampu menampung Android Studio
2. Install Java, Android Studio dan SDK

1.2. Test Awal

Kerjakan test awal dibawah ini!

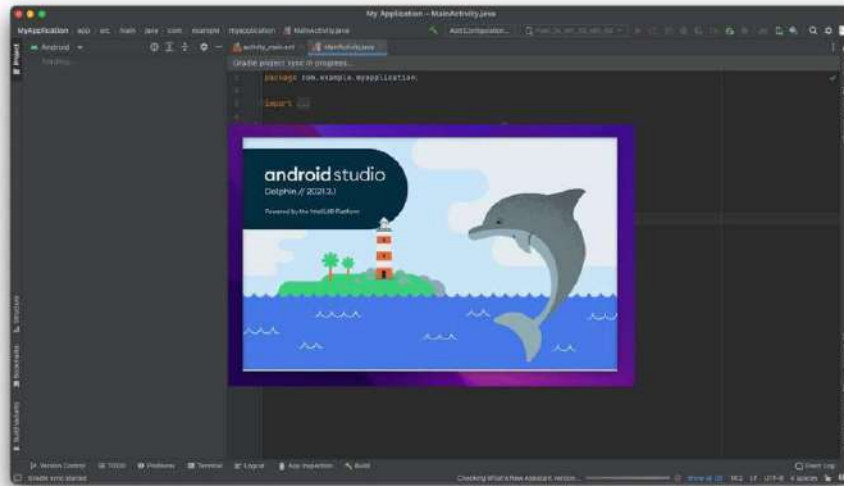
- a. Apa yang Anda ketahui tentang Java?

- b. Apa yang Anda ketahui tentang Android Studio?

- c. Sebutkan syarat atau spesifikasi minimum komputer yang mampu menampung Android Studio!

1.3. Materi

Syarat untuk Install Android Studio



Berikut ini syarat atau spesifikasi minimum komputer yang mampu menampung Android Studio.

1. RAM atau memori minimal 3 GB. Direkomendasikan RAM 8 GB.
2. Minimal 2 GB penyimpanan hardisk yang tersedia. Direkomendasikan 4 GB (500 MB untuk penyimpanan IDE dan 1,5 GB untuk penyimpanan Android SDK dan sistem emulator).
3. Resolusi layar minimum 1280 x 800.

Untuk install Android Studio kamu pastinya memerlukan software seperti:

Android Studio bisa kamu download di situs resmi <https://developer.android.com/studio/>

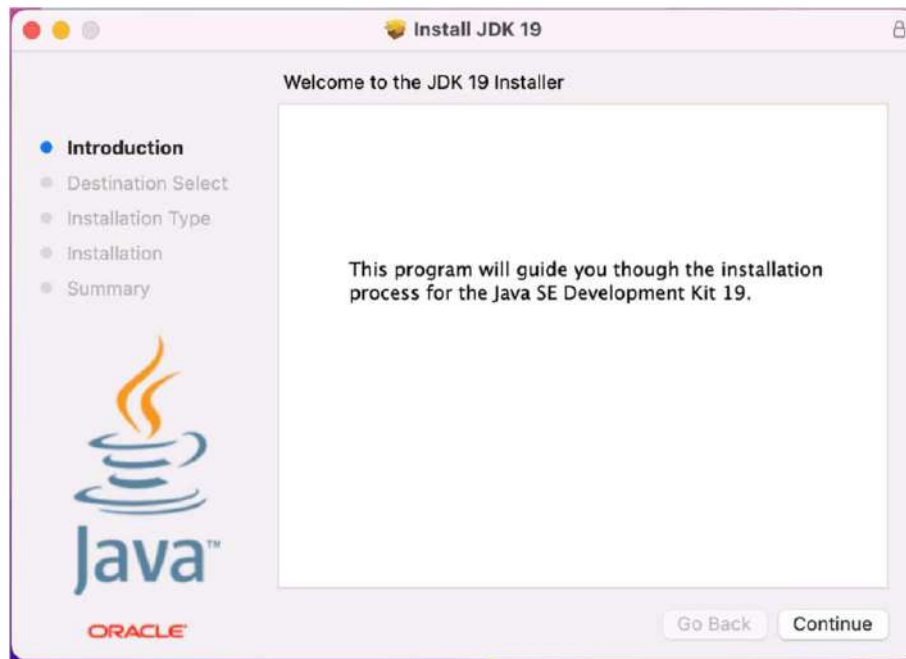
Untuk Java Development Kit (JDK) sendiri kamu bisa download dan kunjungi di <https://www.oracle.com/java/technologies/javase-downloads.html> direkomendasikan untuk install JDK terlebih dahulu.

Setelah download keduanya maka selanjutnya kita akan masuk ke tahap instalasi.

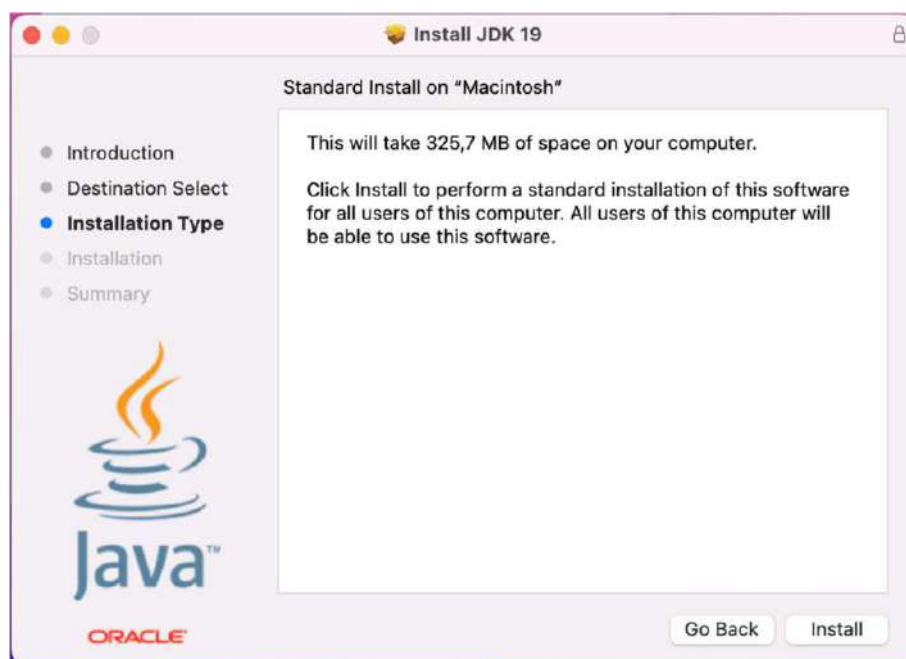
1.4. Studi Kasus

Instalasi Java Development Kit (JDK)

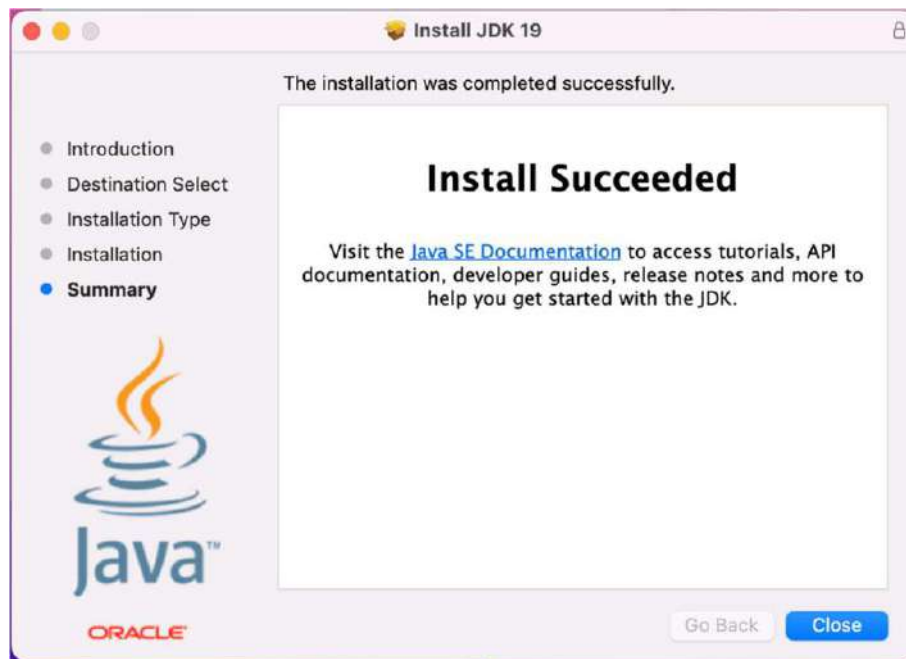
1. Buka (klik 2x) installer java yg sudah didownload -> klik **Continue**



2. Setelah masuk bagian Installation Type -> klik **Install**



3. Tunggu proses instalasi sampai selesai (Install Succeeded) -> selesai

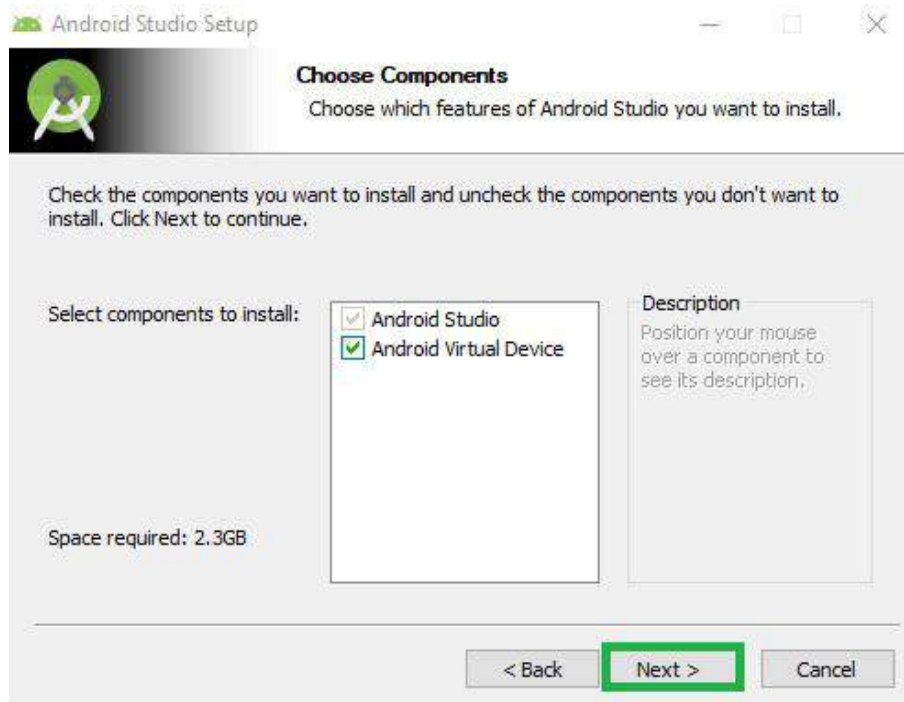


Install Android Studio

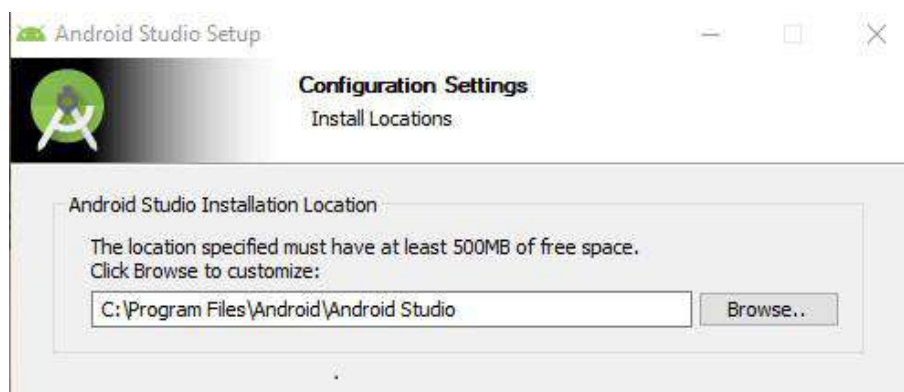
1. Buka (klik 2x) installer Android Studio yg sudah didownload -> klik **Next**



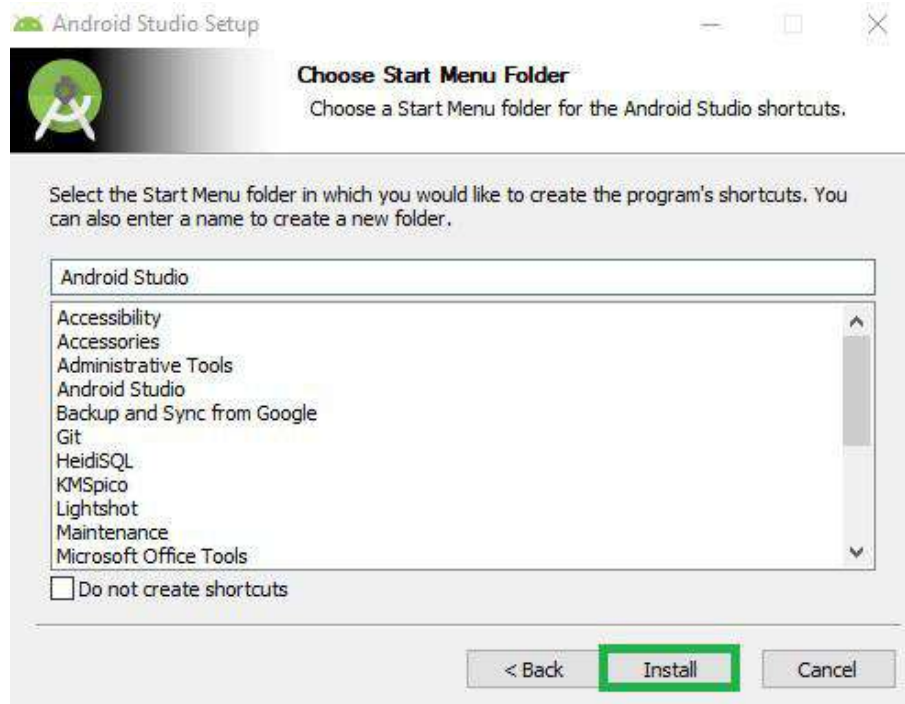
2. Centang “Android Virtual Device” jika kamu ingin memasang emulator Android.



3. Selanjutnya masuk ke dialog penyimpanan software. Bisa disesuaikan dengan keinginan kamu. Namun disarankan lokasi di C:\ atau sistem lalu klik -> **Next**.



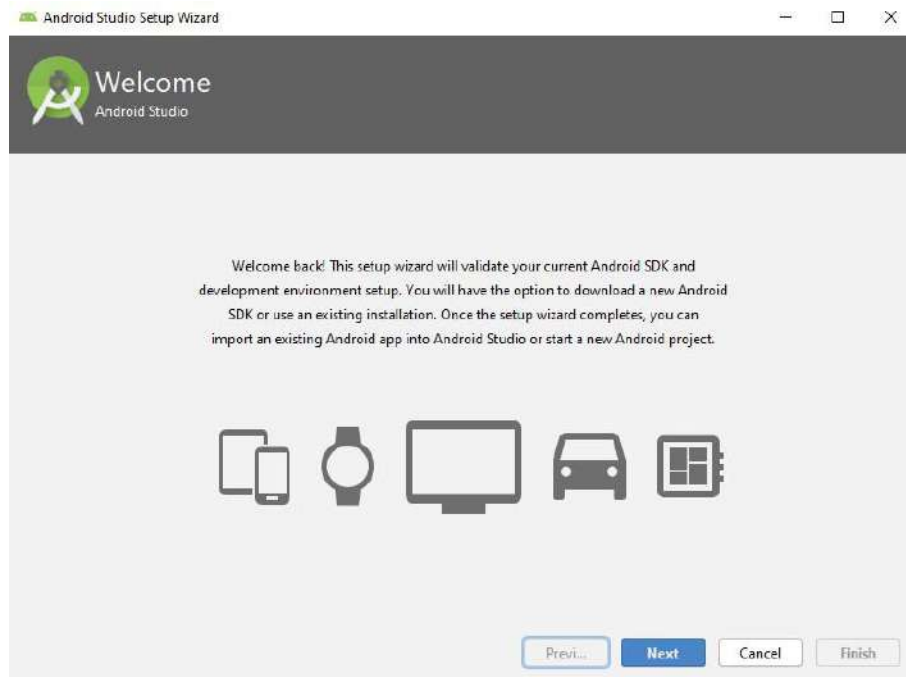
4. Selanjutnya Klik -> **install** untuk memulai proses instalasi. Tunggu proses hingga selesai.



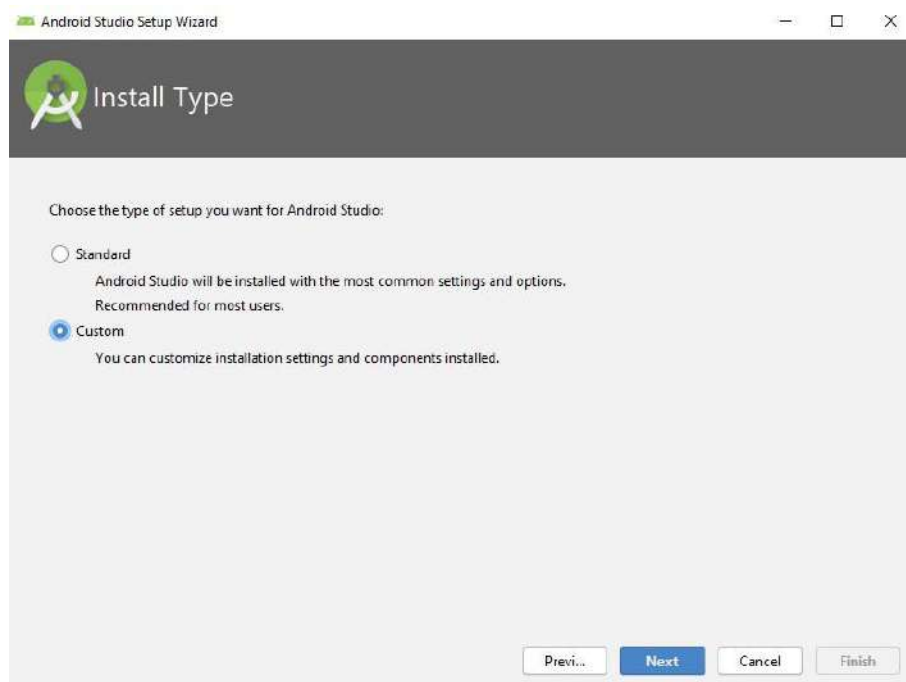
5. Setelah berhasil menginstall Android Studio. Buka aplikasi tersebut dan ikuti instruksi instalasi SDK (Android Software Development Kit). Pertama jalankan Android Studio.



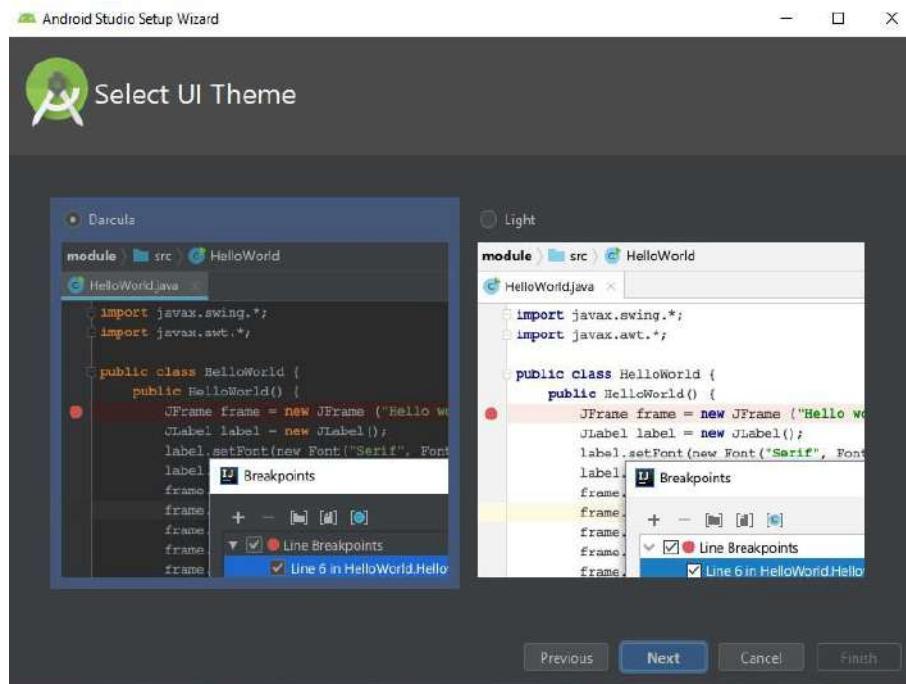
6. Kemudian muncul tampilan seperti di bawah ini. Klik -> **Next**



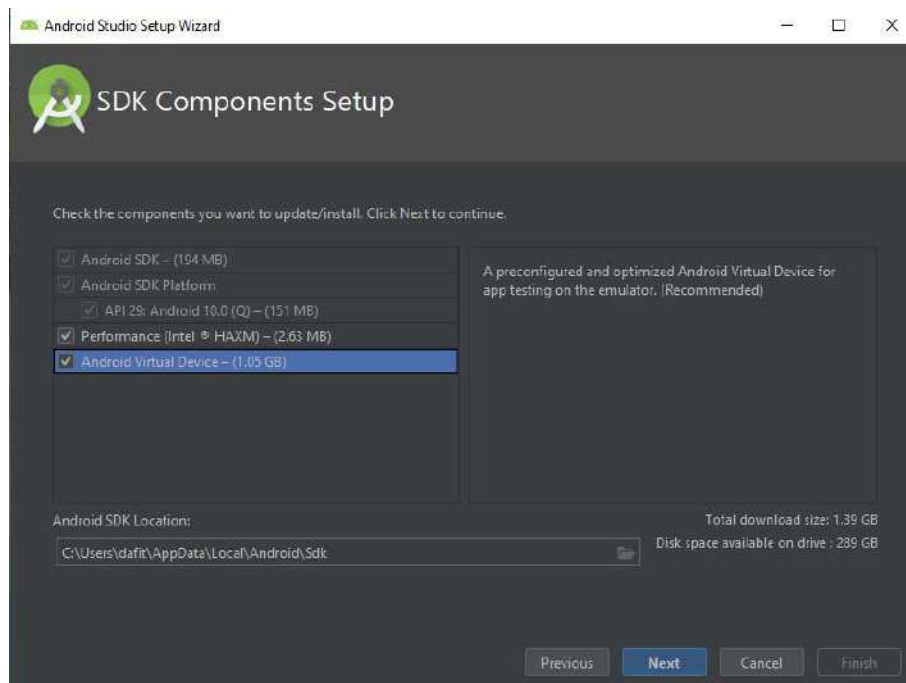
7. Pilihlah mode custom, kemudian atur paket mana saja yang akan di install. Kenapa pilih custom? Supaya terbiasa dengan settingan-settingan install yang ada pada Android Studio.



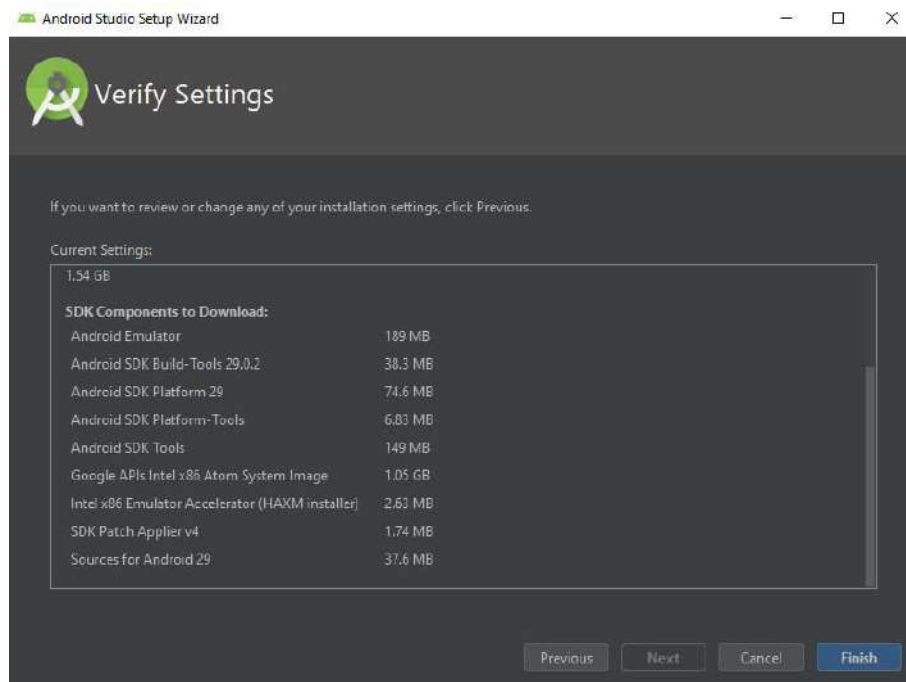
8. Ikuti saja instruksinya. Pilih tema sesuai dengan keinginan kamu lalu -> **Next**.



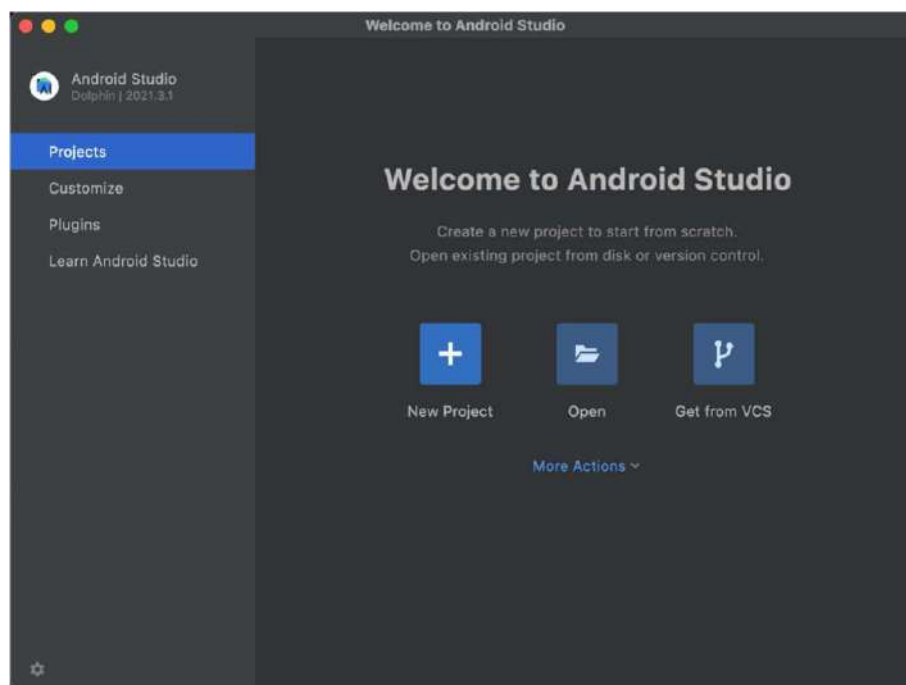
9. Setelah memilih mode tema, maka tampilan selanjutnya akan diminta untuk menginstall beberapa komponen. Komponen tersebut bisa disesuaikan dengan kebutuhan. Misal jika ingin menambahkan emulator maka wajib centang komponen seperti pada gambar di bawah ini.



10. Total data yang akan didownload cukup besar yaitu 1,39 GB jika kamu memasang virtual device. Lalu klik -> **Next**



11. Klik -> **Finish** untuk melanjutkan tahapan download. Lama atau tidak untuk proses downloading disini tergantung kecepatan koneksi internet. Setelah proses download dan install selesai. Kemudian akan muncul tampilan launcher Android Studio seperti pada gambar di bawah ini.



1.5. Test Akhir

Dari tutorial di atas, silahkan praktekkan dan tuliskan kesulitan-kesulitan selama proses instalasi.

1.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

Sumber :

<https://developer.android.com/studio/>

<https://www.oracle.com/java/technologies/javase-downloads.html>

BAB 2 – PENGENALAN ANDROID STUDIO

2.1. Identitas

Kompetensi

1. Memberikan pemahaman tentang Android Studio
2. Membuat Project Pertama dengan Android Studio

Topik

1. Pengantar Android Studio
2. Studi Kasus Membuat Hello Word di Android Studio

2.2. Test Awal

Kerjakan test awal dibawah ini!

- a. Apa yang Anda ketahui tentang Android?

- b. Dapatkan Anda menyebutkan perkembangan versi Android? Silahkan diurutkan dari versi pertama hingga terakhir

- c. Tools apa saja yang dibutuhkan ketika kita mau memulai membangun aplikasi dengan Android Studio? Sebutkan!

2.3. Materi



Mengenal Android

Google mengeluarkan sistem operasi yang diberi nama Android. Sistem Operasi ini khusus dibuat untuk *smartphone* dan *tablet*. Android telah digunakan banyak produsen peranti (*device*) sebagai sistem operasi dari device yang mereka produksi.

Android merupakan sistem operasi *smartphone* terpopuler. Android menjadi *operation system* (OS) terlaris pada *tablet* dan *smartphone* sejak tahun 2013. Fitur yang dimiliki Android sangat canggih dengan tampilan yang bagus seperti menerapkan konsep multimedia yaitu dapat memutar musik dan video.



Komunitas developer banyak yang tertarik dengan Android, karena source code dari Android bersifat open source. Nantinya para *developer* dapat mendistribusikan aplikasinya ke banyak pengguna dengan cara menggunakan *web*, *copy APK*, maupun melalui *Android store* (Google Play) yaitu *store* resmi Android yang dikelola oleh Google. Melalui store ini pengguna dapat menelusuri dan mendownload aplikasi yang dikembangkan menggunakan Android Software Development Kit (SDK).

Mengetahui Perkembangan Android

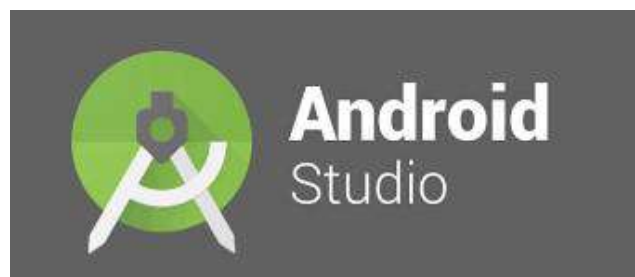
Untuk melihat perkembangan Android dari versi ke versi dapat dilihat pada Tabel di bawah ini.

Versi	Nama Kode	Tanggal Rilis	Level API
13	13	15 Agustus 2022	33
12L	12L	7 Maret 2022	32
12	12	4 Oktober 2021	31

11	11	8 September 2020	30
10	10	3 September 2019	29
9.0	Pie	6 Agustus 2018	28
8.0	Oreo	21 Agustus 2017	26
7.0	Nougat	22 Agustus 2016	24
6.0	Marshmallow	19 Agustus 2015	23
5.1	Lollipop	9 Maret 2015	22
5.0		15 Oktober 2014	21
4.4.x	KitKat	31 Oktober 2013	19
4.3.x	Jelly Bean	24 Juli 2013	18
4.2.x		13 November 2012	17
4.1.x		9 Juli 2012	16
4.0.3 – 4.0.4	Ice Cream Sandwich	16 Desember 2011	15
3.2	Honeycomb	15 Juli 2011	13
3.1		10 Mei 2011	12
2.3.3 – 2.3.7	Gingerbread	9 Februari 2011	10
2.3 – 2.3.2		6 Desember 2010	9
2.2	Rroyo	20 Mei 2010	8
2.0 – 2.1	Éclair	26 Oktober 2009	7
1.6	Donut	15 September 2009	4
1.5	Cupcake	30 April 2009	3

Android Software Development Kit (SDK)

Android Software Development Kit (SDK) merupakan *kit* yang dapat digunakan untuk mengembangkan aplikasi berbasis Android. Dalam kit tersebut terdapat beberapa *tools* seperti *sample code*, *tutorial*, *software libraries*, *debugger*, dokumentasi, dan *emulator*. Java adalah bahasa pemrograman yang sering digunakan untuk mengembangkan aplikasi Android, namun ada beberapa bahasa lainnya yang dapat digunakan, seperti C++ dan Kotlin.



Pada tahun 2014 Google mengeluarkan *Integrated Development Environment* (IDE) yang bernama Android Studio basis IntelliJ IDEA. *Developer* dapat membuat aplikasi dari nol hingga didaftarkan ke Google Play dengan menggunakan Android Studio. Fitur *built-in* yang dimiliki Android Studio sangat membantu para *developer* dalam memaksimalkan proses pembuatan aplikasi. Fitur-fitur tersebut diantaranya adalah Gradle, Code Completion, dan layanan dari Google, seperti Firebase.

Android Studio adalah *Integrated Development Environment* (IDE) atau lingkungan Pengembang Terpadu yang digunakan untuk mengembangkan aplikasi berbasis Mobile.

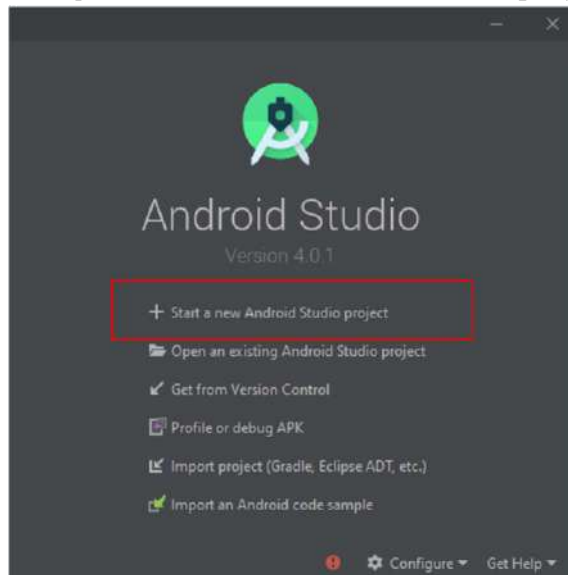
Menyiapkan Tools

Java adalah salah satu bahasa pemrograman yang dapat digunakan untuk membangun aplikasi Android. Untuk itu sebelum memulai untuk membuat coding, kita perlu menginstall beberapa software dibawah ini.

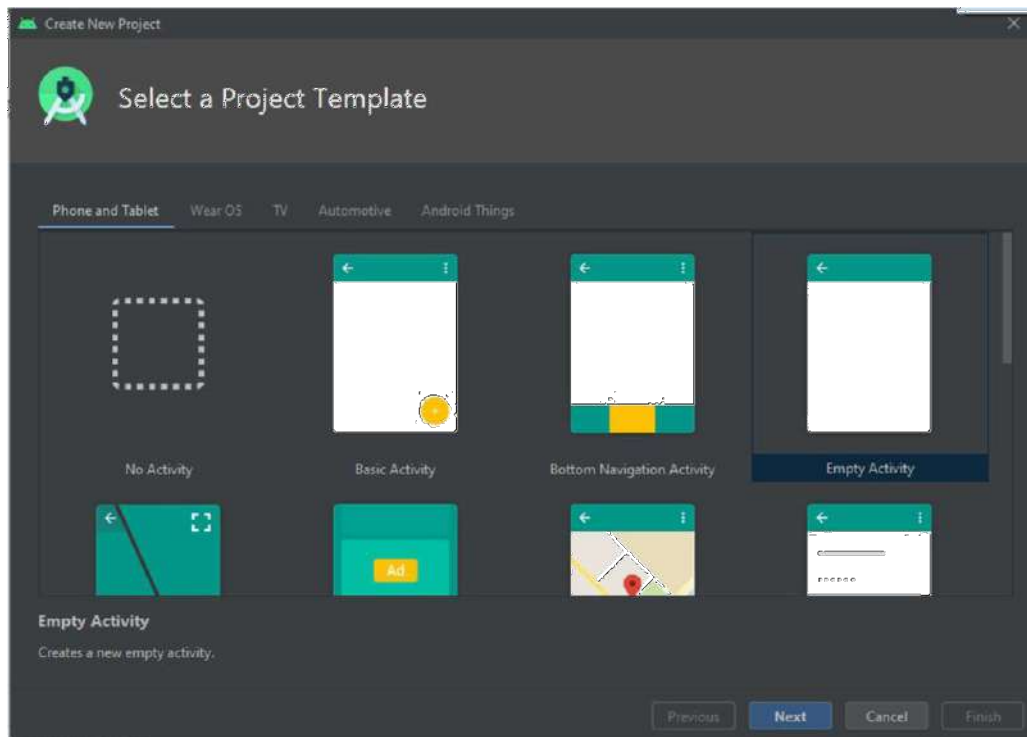
- Instal Java Development Kit (JDK), dapat di download di link berikut:
<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- Install Android Studio, dapat di download di link berikut:
<https://developer.android.com/studio/archive.html>

Membuat Project Android Pertama

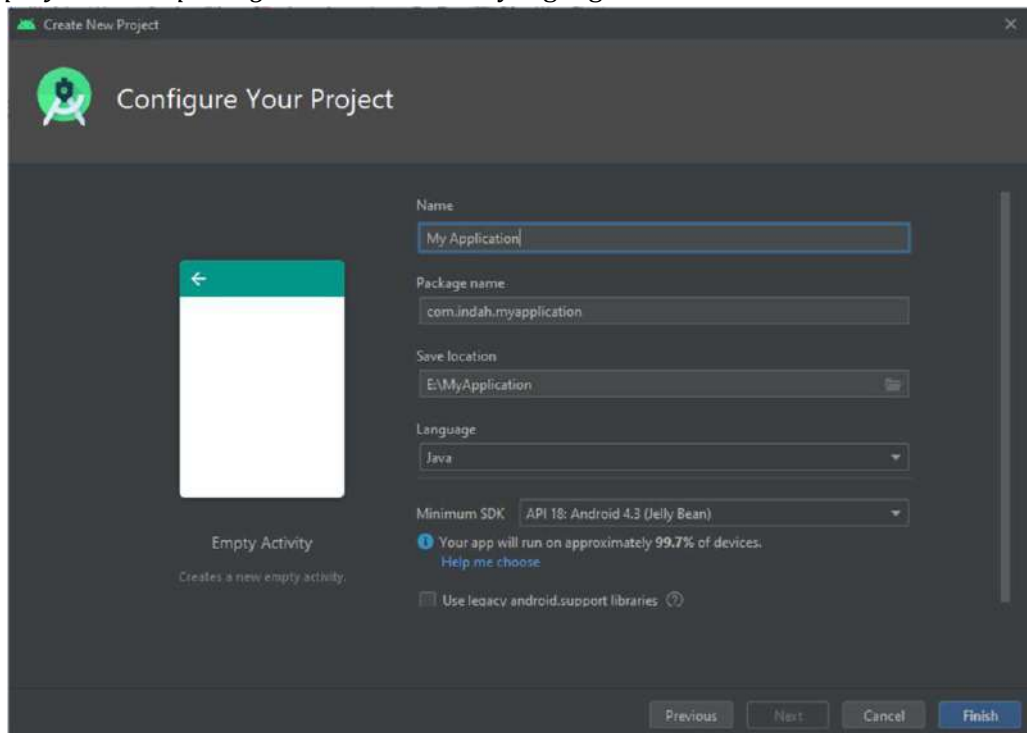
Jalankan Android Studio, setelah itu pilih **“Start a new Android Studio project”**.



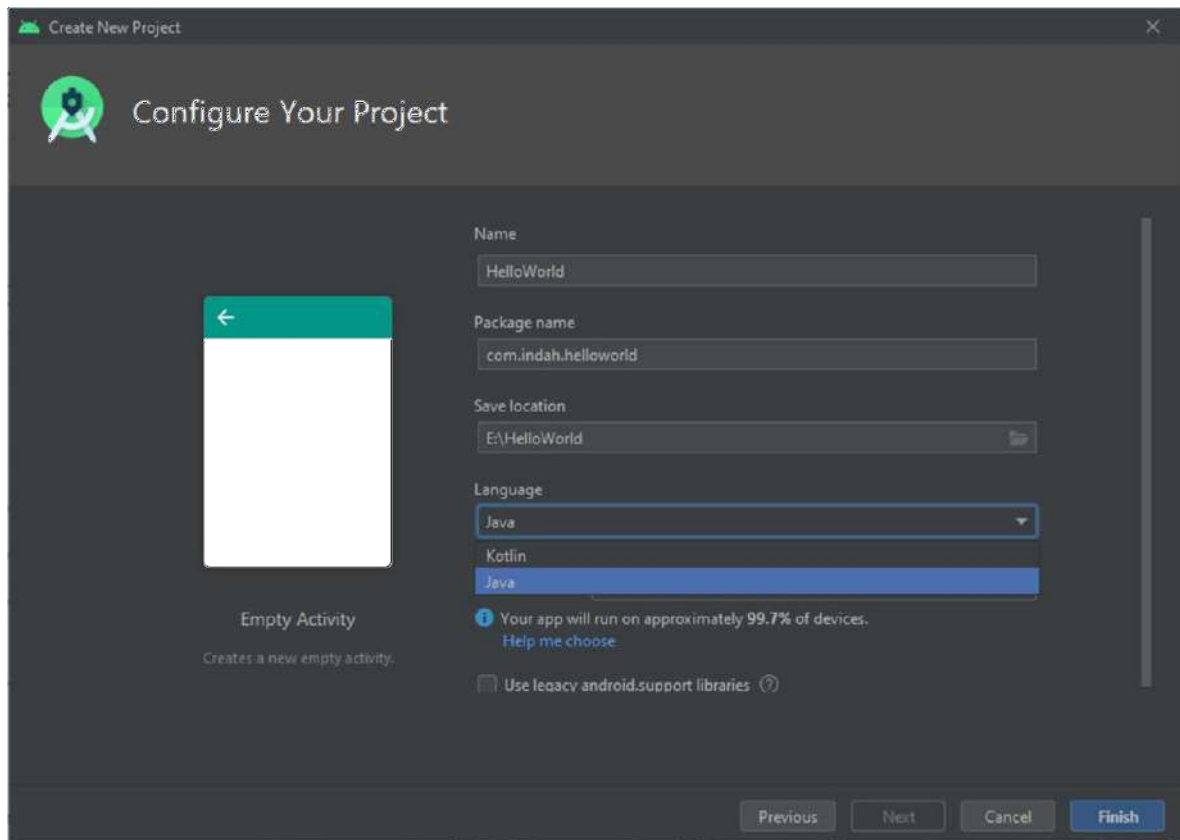
1. Setelah klik **“Start a new Android Studio project”**, maka tampilan akan berubah menjadi dialog baru. Silahkan melakukan konfigurasi untuk Create New Project. Terdapat beberapa pilihan *template* yang bisa dipilih seperti Empty Activity, Login Activity, Navigation Drawer Activity dan lain-lain.



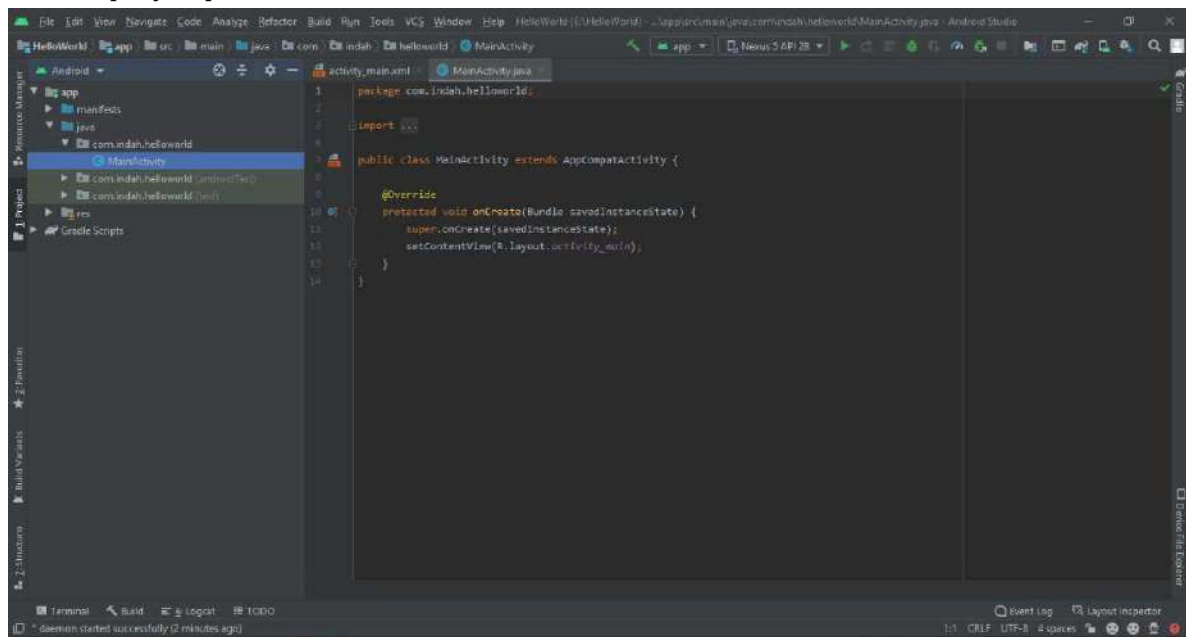
2. Setelah memilih template silahkan klik **“Next”**. Selanjutnya dapat memberi *nama aplikasi, lokasi proyek, nama package*, memilih bahasa yang digunakan, serta menentukan Minimum SDK



Contoh pengisian dapat dilihat pada gambar di bawah ini, kemudian klik **“Finish”**.



3. Selamat, proyek pertama di Android Studio berhasil dibuat.



Jika tampilan tidak sama persis dengan tampilan diatas tidak usah khawatir, hal ini bisa jadi karena versi Android studio yang berbeda ketika diinstall atau setting konfigurasi yang berbeda ketika Create New Project.

Mengenal antar muka Android Studio

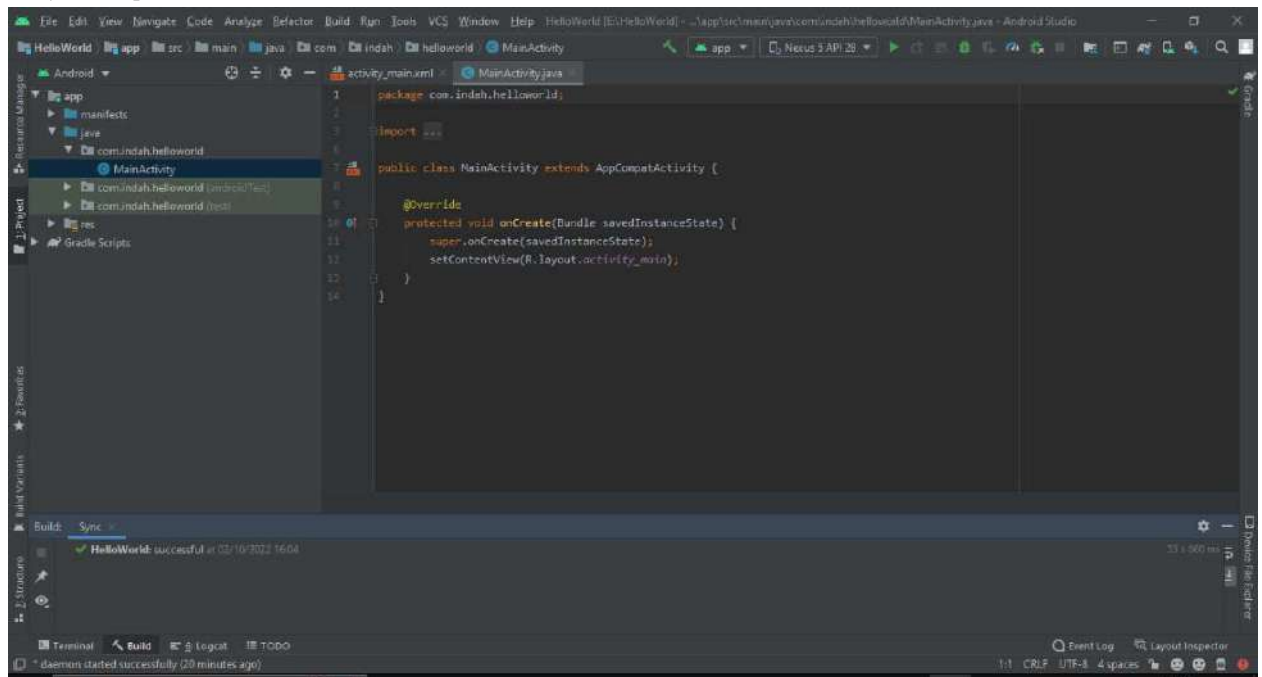
- Tools

Tools yang sering digunakan dalam membuat aplikasi mobile seperti Debug 'app', menjalankan aplikasi, hingga menjalankan emulator.

- Navigasi

Digunakan untuk melihat struktur dari kedalaman dan posisi proyek yang sedang dibuka saat ini.

- Project Explorer dan Editor



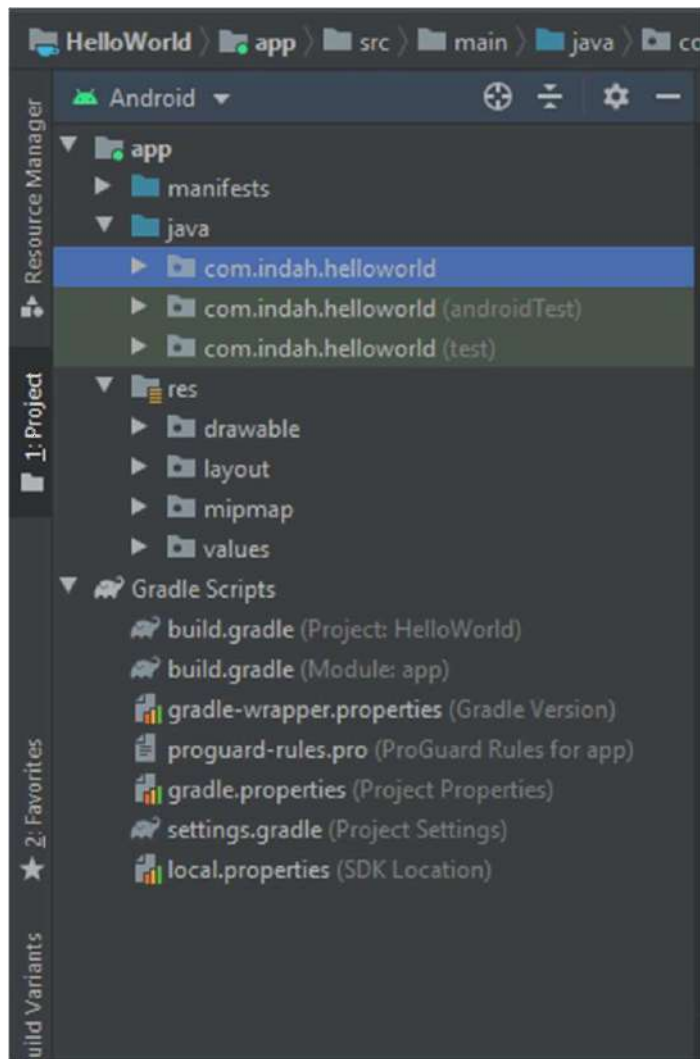
adalah bagian utama dari IDE Android Studio, tempat di mana developer akan menuliskan coding. Pada tampilan di atas, sebelah kiri adalah struktur proyek yang dibuat dan sebelah kanan adalah editor.

Struktur Proyek

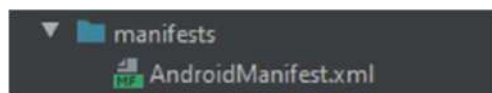
Setiap proyek di Android Studio berisi beberapa modul diantaranya:

- ❑ Modul Aplikasi Android,
- ❑ Modul Pustaka, dan
- ❑ Modul Google App Engine.

Android Studio secara default akan menampilkan berkas proyek yang kita buat dalam tampilan seperti di bawah ini. Tampilan akan disusun berdasarkan modul untuk memberikan akses cepat ke berkas sumber utama proyek yang kita buat.



Android Manifest



adalah salah satu berkas penting yang ada di dalam proyek Android. Di dalamnya terdapat beragam informasi penting seperti Informasi apa saja yang akan digunakan oleh aplikasi sebelum dijalankan. Beberapa Informasi yang ada dalam Manifest adalah sebagai berikut:

❏ Nama Package

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.indah.helloworld">
```

Nama dari sebuah package bertindak sebagai identitas unik dari sebuah aplikasi.

❏ Komponen Aplikasi

```

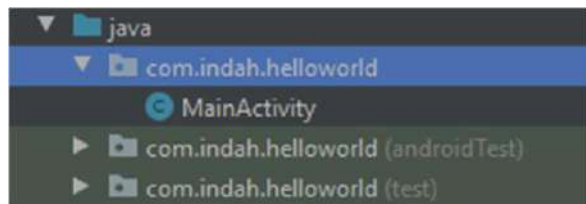
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="HelloWorld"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

```

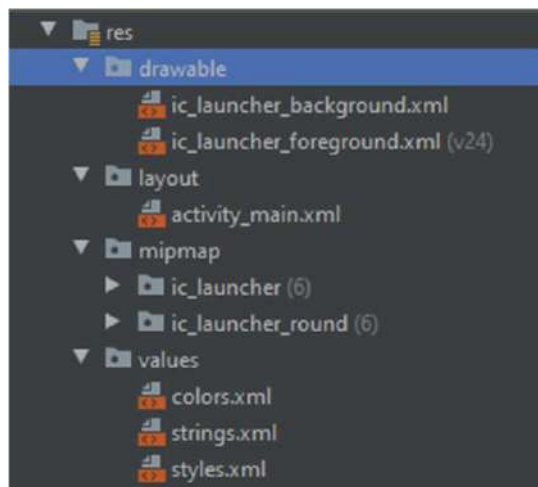
Komponen aplikasi berada di antara tag **<application>**. Berfungsi sebagai penamaan kelas yang mengimplementasi komponen dan mendeskripsikan kemampuannya seperti intent-filter, di mana fungsinya adalah untuk mendeskripsikan bahwa komponen tersebut dijalankan pertama kali.

Java



Adalah salah satu folder yang berisi berkas *source code* yang kita tulis menggunakan bahasa pemrograman Java atau Kotlin. Dalam folder ini terdapat folder yang berisi kode Unit Test dan androidTest (Instrumentation Test).

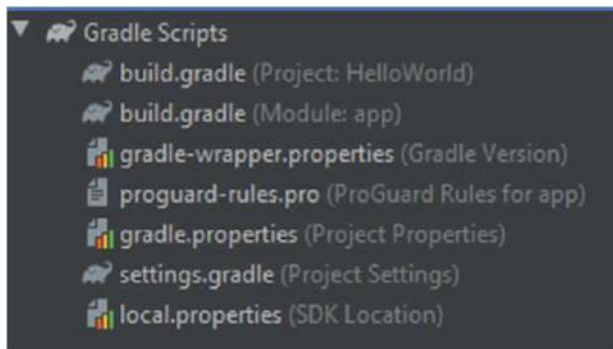
Res



Adalah folder yang digunakan untuk menyimpan resource pada aplikasi yang dibangun. Di dalamnya terdiri dari layout aplikasi, sumber gambar, ikon, hingga style. Di dalam folder res terdiri dari beberapa folder diantaranya:

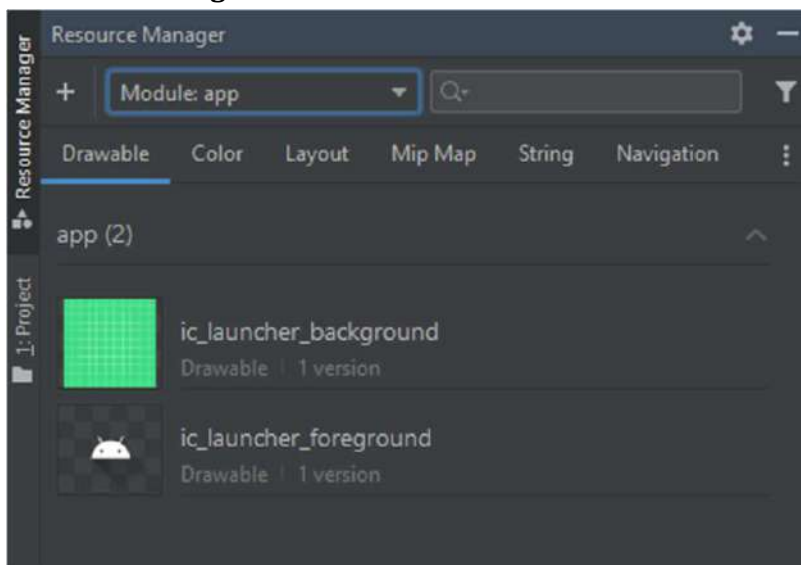
- ❑ Drawable
Digunakan untuk menyimpan berkas gambar maupun ikon.
- ❑ Layout
Digunakan untuk menyimpan berkas desain aplikasi ber ekstensi .xml.
- ❑ Mipmap
Digunakan untuk menyimpan logo dalam berbagai dimensi.
- ❑ Values
Digunakan untuk menyimpan berbagai macam sumber data, seperti colors.xml untuk warna, strings.xml untuk teks, dimens.xml untuk ukuran, dan themes.xml untuk membuat theme dan style.

Gradle



Adalah *build automation system* yang bersifat *open source*. Digunakan untuk mengotomatisasi proses pembuatan dari *software build* dan proses-proses terkait lainnya. Penggunaan gradle termasuk digunakan *compile source code* menjadi *binary code*, *packaging binary code*, dan menjalankan *automated test*.

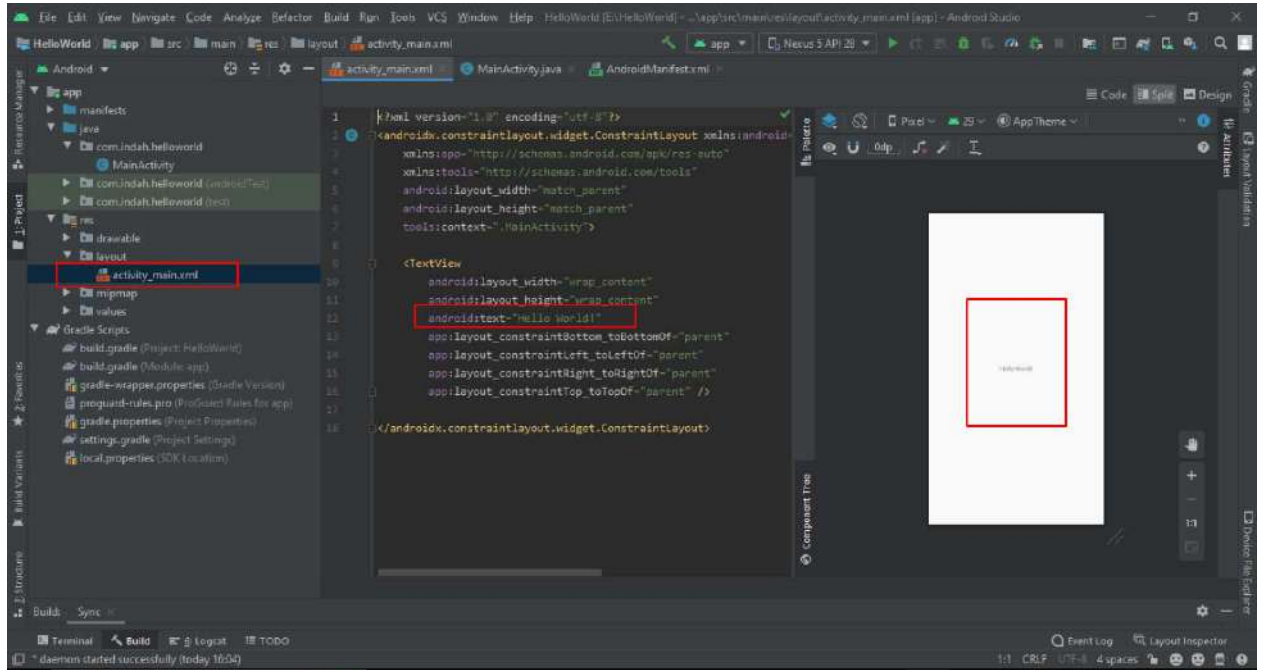
Resource Manager



Adalah untuk manajemen segala *resource* yang ada di proyek yang sedang dibangun seperti gambar, warna, layout dst.

2.4. Studi Kasus

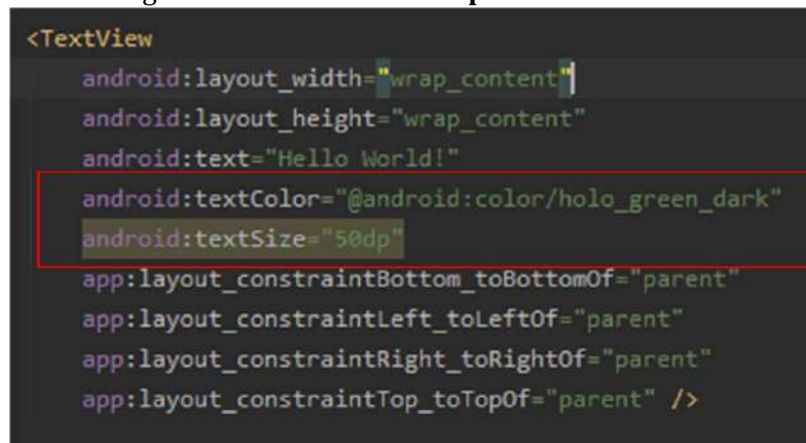
Berikut ini contoh membuat proyek pertama di Android Studio dengan menampilkan kata **Hello World**.



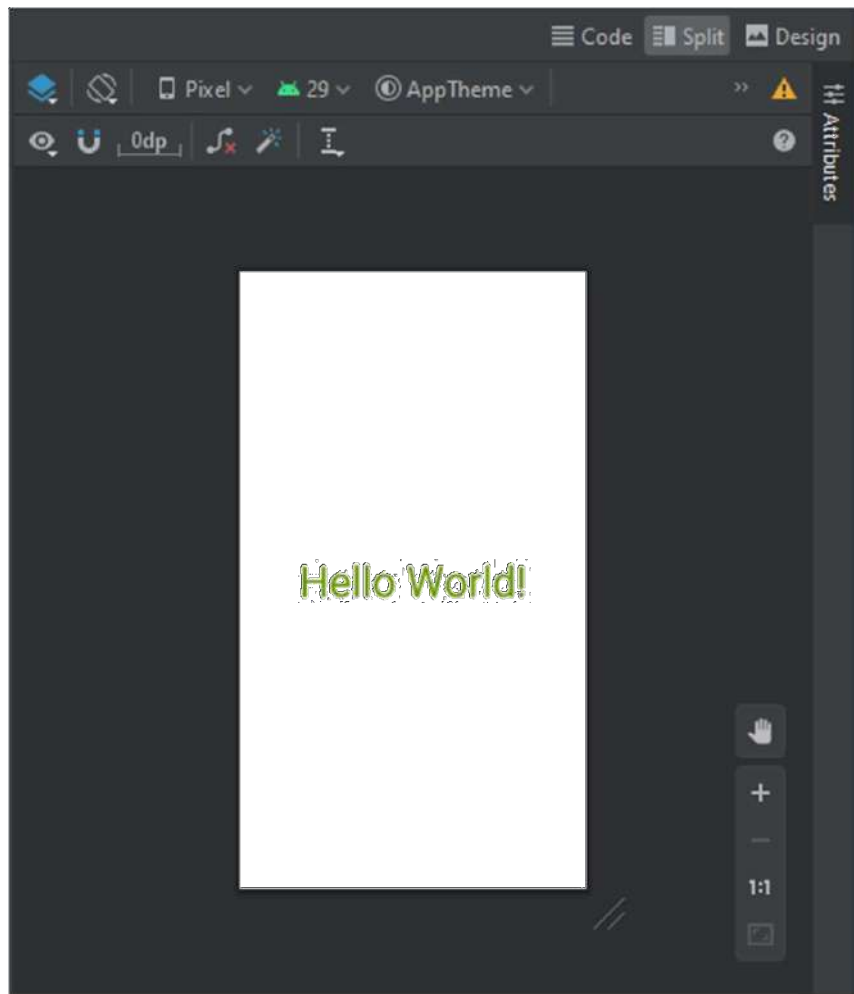
Buka folder `res - layout - activity_main.xml`, maka di layar bagian editor (layar bagian tengah) akan secara default muncul **`android:text="Hello World!"`**. Kita dapat melihat posisi teks di layar bagian paling kanan.

Dalam praktik kali ini kita akan mencoba memodifikasi tulisan Hello World tersebut dengan memperbesar ukuran dan memberi warna pada huruf.

Pertama, tuliskan coding berikut **`android:textColor="@android:color/holo_green_dark"`** dan dibawahnya tuliskan coding **`android:textSize="50dp"`**.



Maka desain akan berubah seperti tampilan di bawah ini:



2.5. Test Akhir

Dari contoh studi kasus di atas, silahkan modifikasi file `activity_main.xml` untuk menuliskan :

Nama :

NIM :

Kelas :

No. Hp :

Silahkan diisi sesuai dengan identitas masing-masing, dan silahkan di csreenshoot hasil desain nya.

2.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

Sumber :

<https://developer.android.com/about/versions/>

[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

BAB 3 – MENJALANKAN DAN BUILD APLIKASI ANDROID

3.1. Identitas

Kompetensi

1. Mampu menjalankan Aplikasi Android di Device dan Emulator
2. Dapat mem-build berkas executable dalam format APK (Android Application Package)

Topik

1. Syarat atau spesifikasi minimum komputer yang mampu install Emulator Android Studio
2. Membuat executable Aplikasi Android format APK (Android Application Package)

3.2. Test Awal

Kerjakan test awal dibawah ini!

- a. Apa yang Anda ketahui tentang Emulator Android?

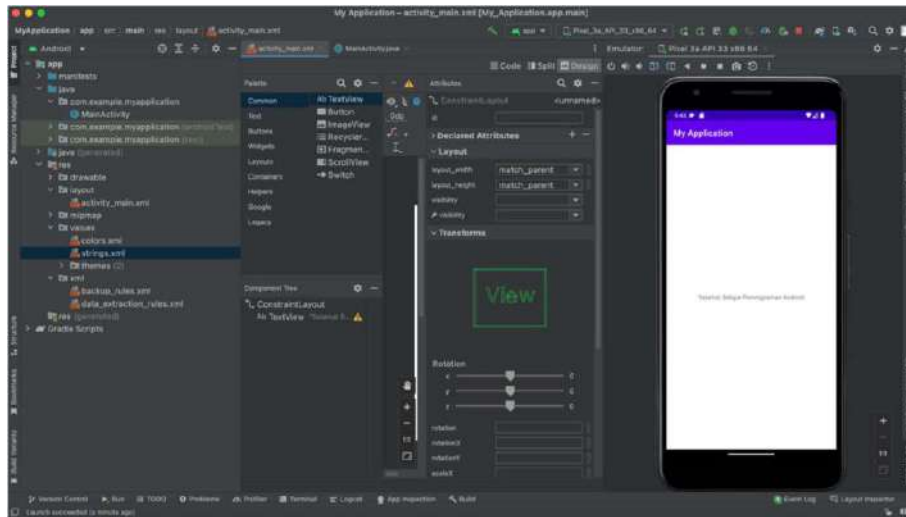
- b. Apa yang Anda ketahui tentang format executable Aplikasi Android?

- c. Sebutkan syarat atau spesifikasi minimum komputer yang mampu install Emulator Android Studio!

3.3. Materi

Menjalankan Aplikasi di Device & Emulator

Uji coba aplikasi wajib dilakukan seorang developer. Proses running atau debugging bisa dilakukan dengan dua cara, yaitu running dengan Device Smartphone Android atau Emulator.



3.4. Studi Kasus

1. Menjalankan aplikasi menggunakan Emulator

Sebelum menggunakan emulator, perhatikan beberapa syarat berikut ini:

1) Virtualization

Untuk menjalankan emulator di dalam Android Studio, perhatikan pentingnya aspek virtualization. Sistem Anda harus memenuhi persyaratannya, yakni ketentuan prosesor dan sistem operasi dari laptop / PC yang Anda gunakan sebagai berikut:

Processor

Prosesor Intel: Jika laptop/pc Anda menggunakan prosesor Intel, maka pastikan ia mendukung Intel VT-x, Intel EM64T (Intel 64), dan Execute Disable (XD) Bit functionality.

Prosesor AMD: Jika laptop/pc Anda menggunakan AMD, maka pastikan bahwa ia support dengan AMD Virtualization (AMD-V) dan Supplemental Streaming SIMD

Extensions 3 (SSSE3).

Sistem Operasi

Intel: Jika menggunakan processor Intel maka Anda dapat menjalankannya di sistem operasi Windows, Linux, maupun Mac.

AMD: Untuk prosesor AMD maka hanya bisa menjalankannya di sistem operasi Linux.

2) Konfigurasi Hardware Accelerated Execution Manager (HAXM)

Dalam beberapa device yang tidak mendukung Emulator, Anda perlu instal HAXM. HAXM adalah hardware-assisted virtualization engine yang menggunakan teknologi VT dari Intel untuk mempercepat aplikasi Android yang diemulasi di mesin host. HAXM diperlukan untuk menjalankan emulator di Android Studio.

HAXM diperlukan jika sistem operasi yang Anda gunakan adalah Windows atau Mac. Untuk menginstalnya, ikuti petunjuk berikut ini.

- 1) Buka SDK Manager.
- 2) Pilih SDK Update Sites, kemudian hidupkan Intel HAXM.
- 3) Tekan OK.
- 4) Cari berkas installer-nya di directory folder sdk komputer Anda,
~sdk\extras\intel\Hardware_Accelerated_Execution_Manager\intelhaxm-android.exe.
- 5) Jalankan installer dan ikuti petunjuknya sampai selesai.

Catatan:

Jika HAXM sudah otomatis ter-install, maka Anda dapat melewati langkah di atas.

3) Menginstal Kernel-based Virtual Machine (KVM) untuk Pengguna Linux

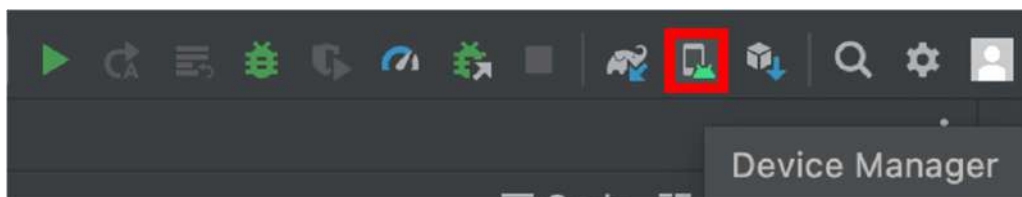
Karena HAXM hanya untuk Windows dan Mac, bagaimana dengan sistem operasi Linux? Untuk Linux, Anda harus instal KVM. Sistem operasi Linux dapat support accelerated virtual machine dengan menggunakan KVM. Untuk instal KVM, Anda bisa menggunakan perintah berikut ini.

```
$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils  
ia32-libs-multiarch
```

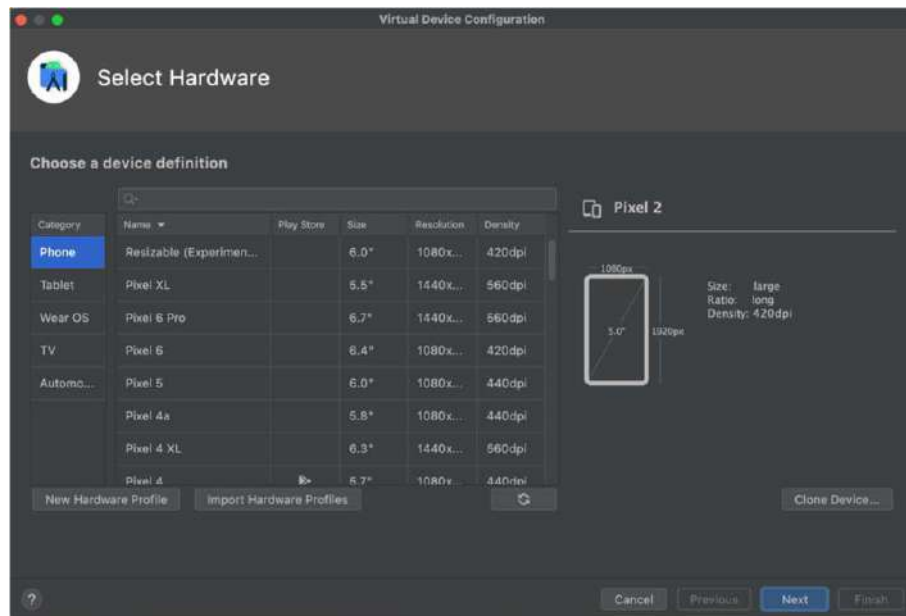
Menggunakan Emulator di Android Studio

Setelah sukses dalam proses instalasi dan laptop/PC memiliki spesifikasi hardware yang lumayan tinggi sehingga mampu menjalankan emulator, ikuti tahapan menggunakan emulator berikut ini:

1. Buka AVD Manager dengan klik icon berikut:

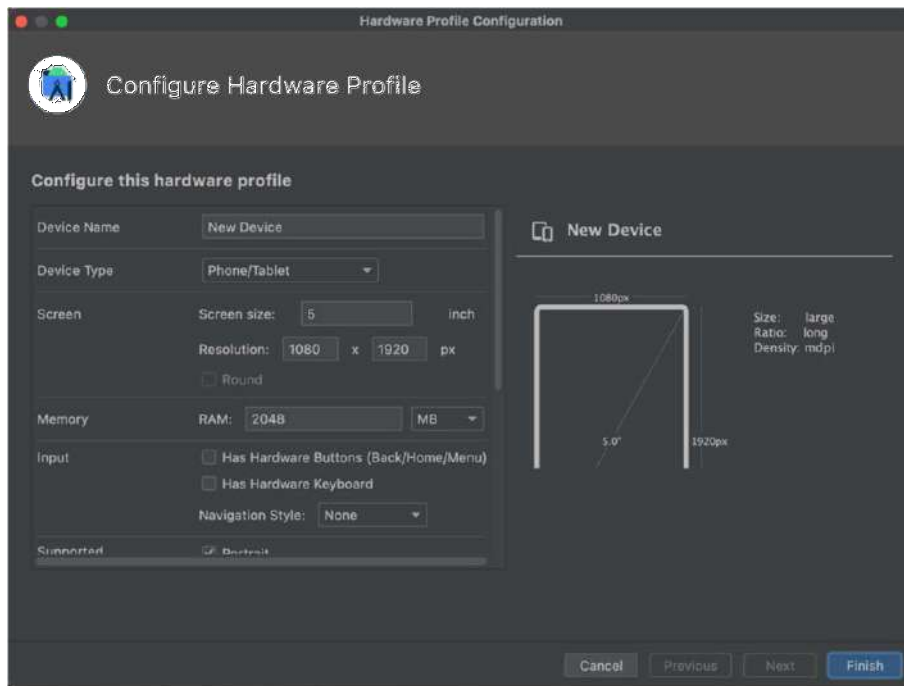


2. Kemudian pilih -> **Create New Virtual Device**. Akan muncul dialog dengan pilihan beberapa emulator yang bisa Anda pilih. Pilih salah satu dan klik -> **Next**.

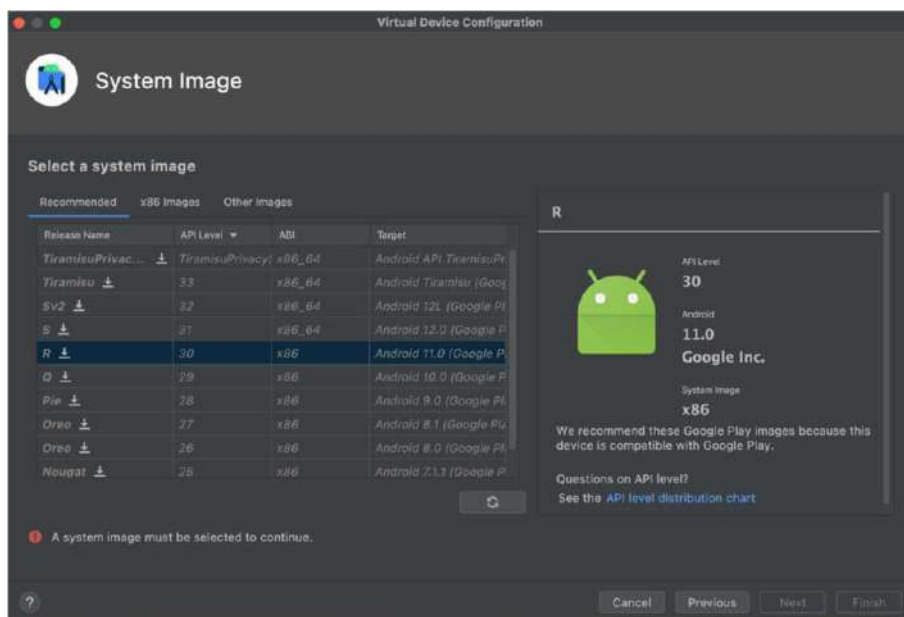


3. Kemudian pilih satu system image, usahakan pilih versi Android yang terbaru dan klik -> **Next**.
4. Jika anda ingin membuat spesifikasi hardware (perangkat keras) sendiri, Anda bisa memilihnya pada pilihan -> **New Hardware Profile**. Akan muncul dialog seperti di bawah ini.

Anda dapat menentukan konfigurasi hardware sesuai dengan kebutuhan Anda. Yang perlu diingat adalah untuk menggunakan konfigurasi emulator yang sesuai dengan kemampuan laptop atau komputer yang Anda gunakan.

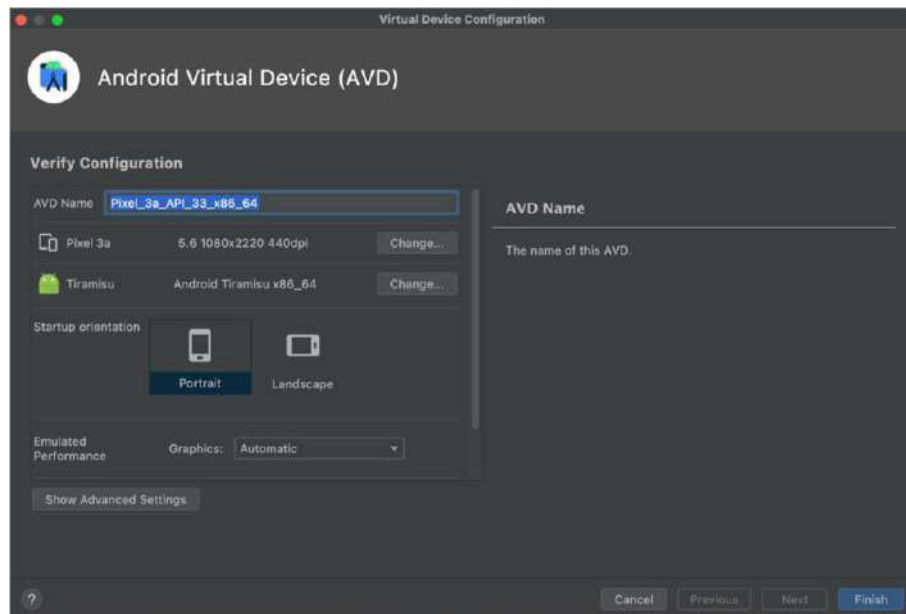


5. Anda dapat membuat hardware emulator baru atau memilih hardware emulator yang sudah ada. Setelah memilih hardware emulator, akan muncul dialog seperti ini.

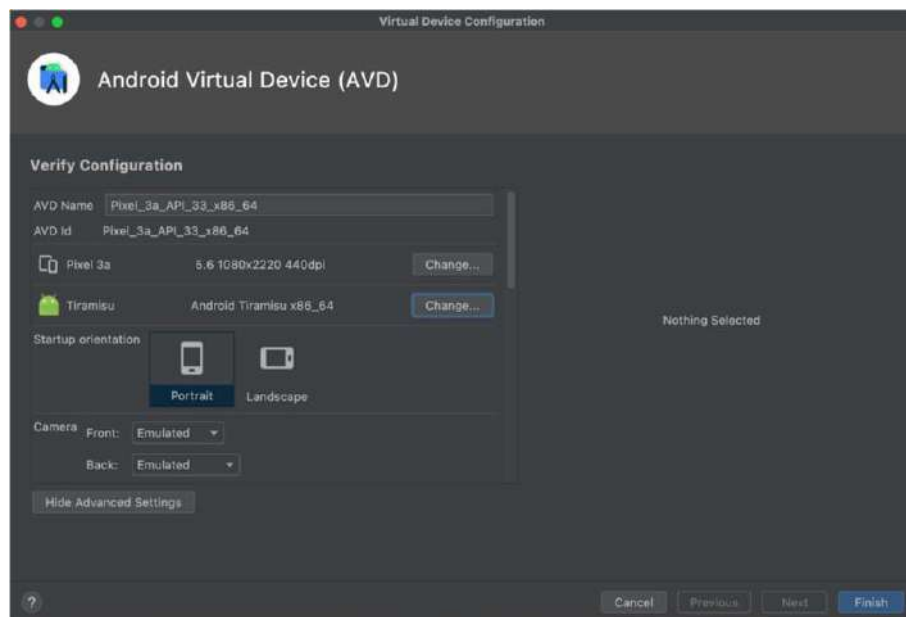


Pada dialog ini Anda akan memilih versi android dari emulator yang akan anda buat. Pada dialog tersebut, Anda perlu memilih versi yang sudah diunduh yaitu Nougat. Tombol/icon **download** di sebelah kanan versi menunjukkan bahwa Anda perlu mengunduhnya terlebih dahulu jika ingin menggunakannya.

6. Selanjutnya klik -> **Next**. Akan muncul dialog verify configuration. Pada dialog ini, Anda bisa memeriksa konfigurasi dari emulator yang Anda pilih.

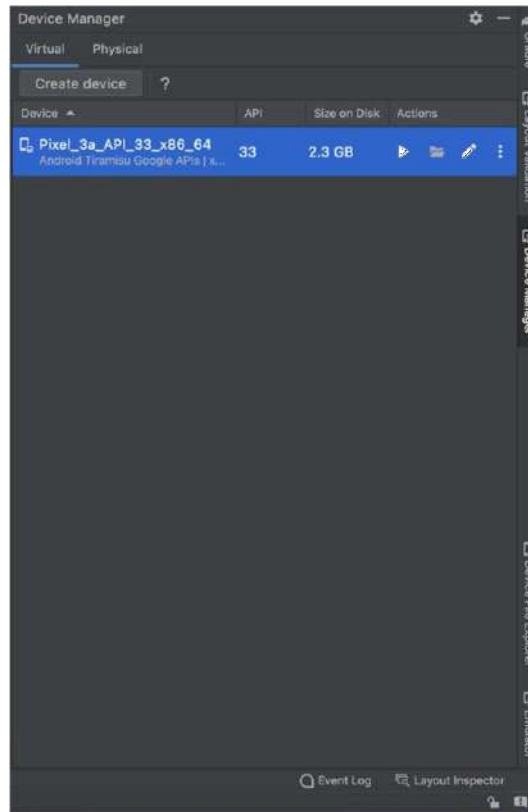


Pada bagian kiri bawah, terdapat tombol -> **Show Advanced Settings**. Bila Anda menekan tombol ini, akan muncul tampilan dialog baru seperti gambar di bawah ini.



Pada bagian **Advanced Setting**, Anda bisa mengubah konfigurasi hardware yang telah ditentukan sebelumnya.

7. Jika sudah selesai, Anda dapat menekan tombol -> **Finish**. Anda dapat membuka emulatorenya dengan menekan tombol **hijau** yang ada di sebelah kanan.



8. Pengaturan emulator sudah selesai dan bisa langsung dijalankan.

Catatan :

Menggunakan emulator dari Android Studio kadang terasa berat untuk beberapa device dengan spesifikasi rendah. Untuk mengatasinya Anda bisa menjalankan langsung pada device smartphone Android dengan mengikuti langkah selanjutnya atau menggunakan emulator pihak ketiga yang lebih ringan seperti :

- Nox
- Player
- Memu
- Play
- Bluestack
- KoPlayer
- Andy

Pilih salah satu emulator di atas, kemudian install dan jalankan aplikasinya. Maka emulator tersebut akan otomatis muncul di List Virtual Devices pada Android Studio.

2. Menjalankan Aplikasi dengan Device/Smartphone Android

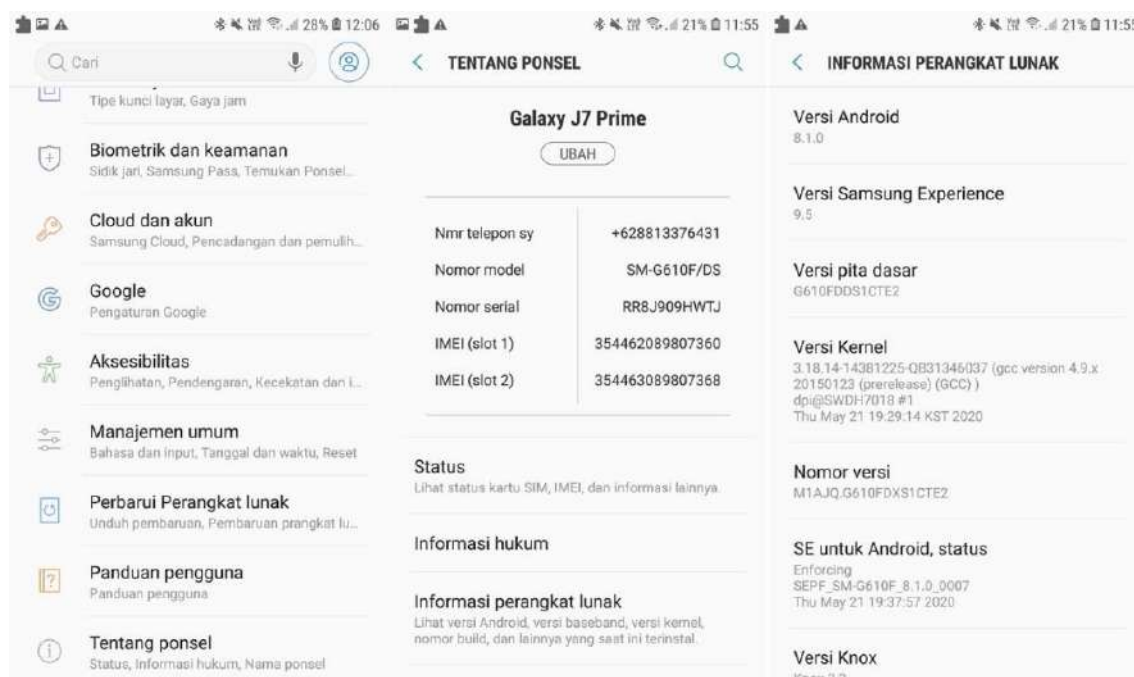
Bila Anda hendak melakukan run atau debugging, lebih baik Anda menjalankannya pada peranti smartphone android asli. Running dengan menggunakan smartphone android memiliki beberapa kelebihan jika dibandingkan dengan emulator yaitu:

Lebih cepat dan ringan,

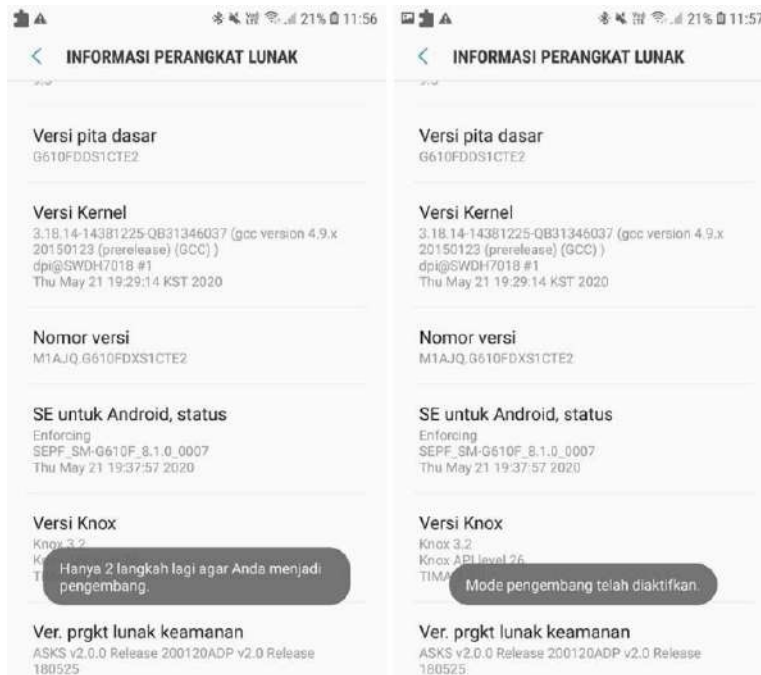
Fitur seperti geo-location, push notif bisa digunakan, Bisa mengetahui daya serap baterai terhadap aplikasi, Lebih mudah.

Dengan menggunakan peranti smartphone android asli, kita dapat memastikan bahwa aplikasi kita berjalan dengan wajar ketika sudah sampai di tangan pengguna. Kekurangannya adalah beragamnya model peranti smartphone android yang ada di pasaran membuat sedikit berbeda cara menjalankan proses *run* atau *debugging*. Langkah-langkah berikut bisa mungkin akan berbeda dengan peranti smartphone android yang Anda pakai. Akan tetapi secara garis besar Langkah-langkahnya sama.

1. Pastikan smartphone android yang akan dipakai sesuai dengan target SDK atau paling tidak mendukung versi SDK terendah yang digunakan aplikasi.
2. Buka **setting/pengaturan** smartphone android dan masuk ke dalam menu **About/Tentang Ponsel**. Pada halaman menu ini, Anda perlu menemukan informasi tentang **Build number/Nomor versi**.



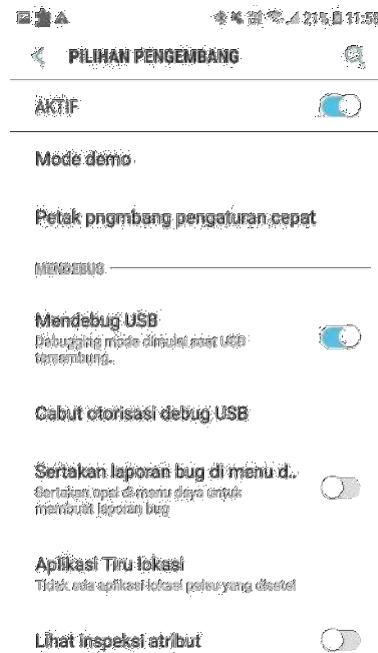
3. Kemudian tekan **Build number/Nomor versi** sebanyak 7 kali.



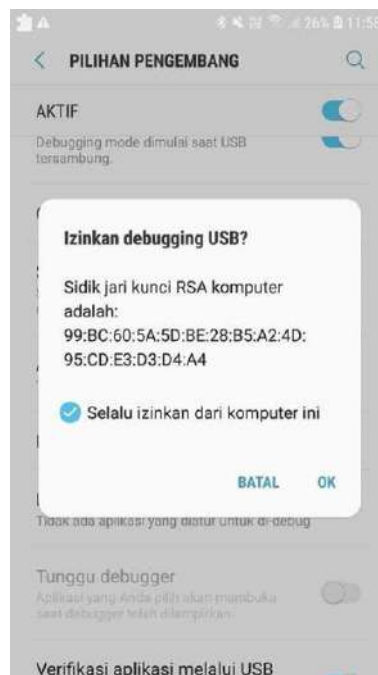
4. Kembali ke **menu setting/pengaturan** di awal dan akan muncul menu baru di bawah **About/Tentang Ponsel** yaitu **Developer Options/Pilihan Pengembang**.



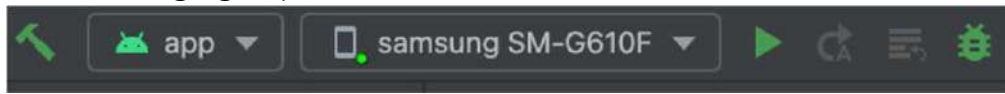
5. Masuk ke dalam menu **Developer Options/Pilihan Pengembang** dan pastikan opsi **USB Debugging/Mendebug USB Mode** sudah on.



6. Setelah menyelesaikan pengaturan pada peranti smartphone android, peranti smartphone android pun dapat dihubungkan dengan laptop atau komputer dengan menggunakan kabel data. Klik **Izinkan/OK** dan centang untuk **Selalu izinkan dari Komputer ini** apabila muncul pop up **Izinkan debugging USB**.



7. Jika device smartphone android sudah terdeteksi pada Android Studio, maka akan otomatis muncul pada daftar device dan Anda bisa menjalankan aplikasi dengan klik ikon segitiga hijau berikut:



Catatan :

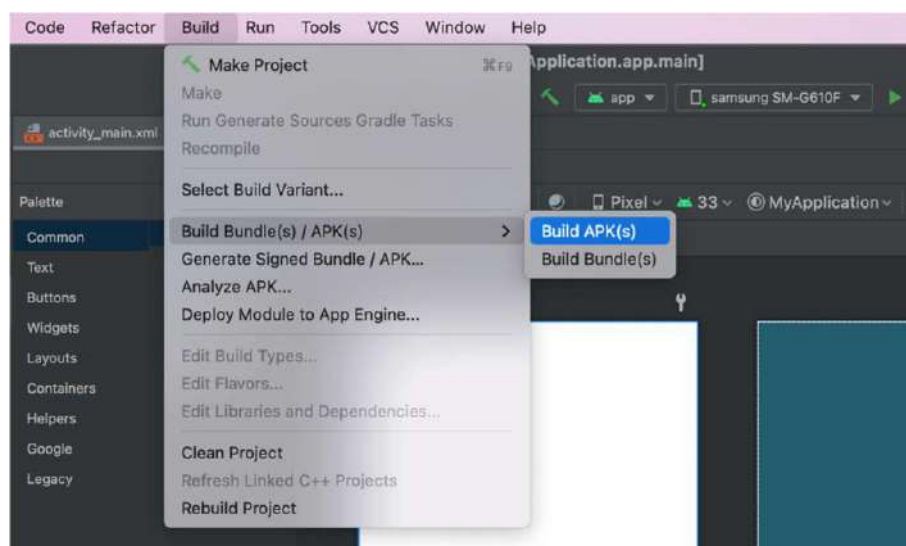
Beberapa merk smartphone android memerlukan cara tersendiri selain yang sudah dibahas diatas, mungkin ada yang memerlukan install software tertentu milik merk smartphone android tersebut untuk bisa terhubung dengan android studio di PC/Laptop. Untuk lebih jelas silahkan mengunjungi website resmi dari merk smartphone android tersebut.

3. Build APK (Android Application Package)

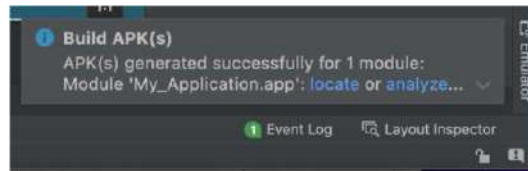
Setelah selesai proses pengembangan aplikasi android adalah build executable Aplikasi Android/Build APK (Android Application Package). Membuat APK dapat dibagi menjadi 2:

1. Dengan menggunakan *default keystore*
Cara ini digunakan bisa digunakan saat kita ingin menguji menginstall APK aplikasi yang telah kita buat dengan distribusi lewat website/flashdisk/perangkat lain. Cara build APKnya sebagai berikut:

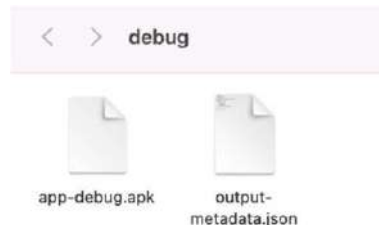
- 1) Buka menu **Build -> Build Bundle(s) / APK(s)** klik -> **Build APK(s)**



- 2) Tunggu sampai proses **Build APK(s)** selesai, jika sukses akan ada pesan notifikasi seperti berikut



- 3) Selanjutnya klik **locate** pada pesan notifikasi maka akan menuju lokasi di mana berkas APK disimpan.

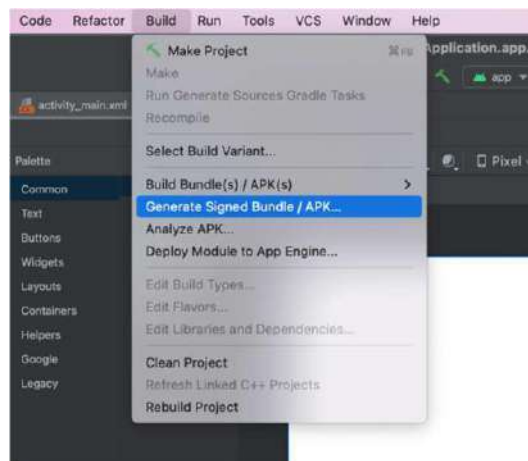


- 4) Sekarang coba pindahkan berkas APK yang baru dibuat ke dalam smartphone Android. Buka lokasi berkas tersebut menggunakan file explorer pada smartphone android tersebut. Lakukan instalasi seperti biasa, jika sukses berarti telah berhasil membuat aplikasi android. Selamat!

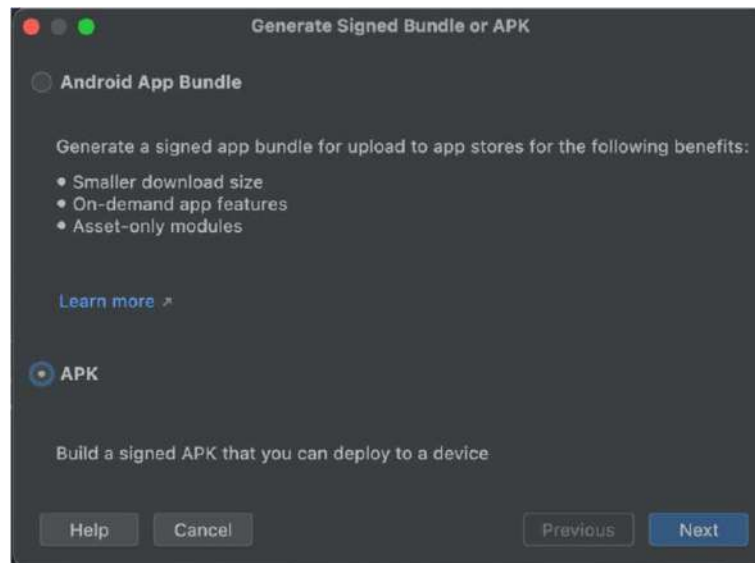
2. Dengan menggunakan *custom keystore*

Membuild APK dengan cara ini digunakan saat akan mempublish atau memasukkan aplikasi android kita **Google Play Store**. Berikut ini tahapan pembuatan APKnya:

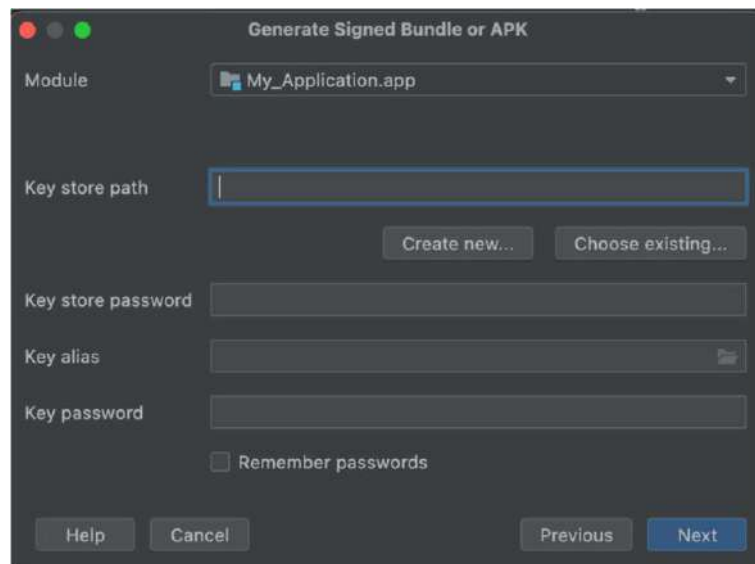
- 1) Buka menu **Build** klik -> **Generate Signed Bundle / APK**



- 2) Selanjutnya pilih **APK**



3) Kemudian pilih **Create new**



4) Pada form yang tampil isikan seperti berikut ini. Setelah selesai klik **OK**

New Key Store

Key store path: /Users/gab/key.jks

Password: Confirm:

Key

Alias: key0

Password: Confirm:

Validity (years): 25

Certificate

First and Last Name: Umpo Devid

Organizational Unit: Dev

Organization: Umpo

City or Locality: Ponorego

State or Province: Jawa Timur

Country Code (XX): 63471

Cancel OK

- 5) Kemudian akan Kembali ke form sebelumnya dan form sebelumnya akan secara otomatis terisi. Klik **Next** untuk melanjutkan.

Generate Signed Bundle or APK

Module: My_Application.app

Key store path: /Users/gab/key.jks

Create new... Choose existing...

Key store password:

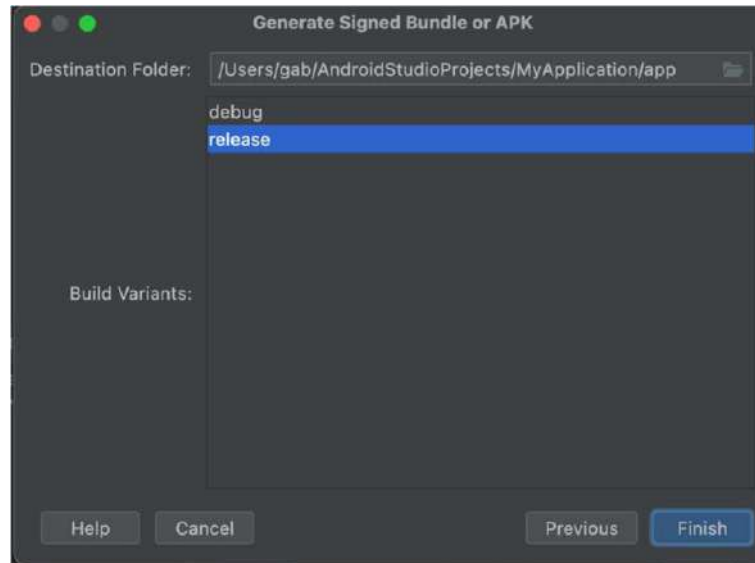
Key alias: key0

Key password:

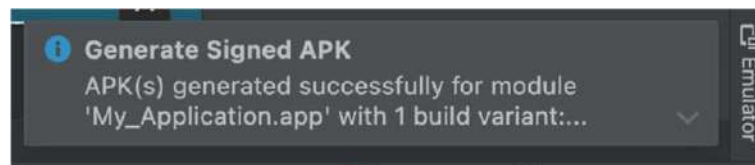
☐ Remember passwords

Help Cancel Previous Next

- 6) Selanjutnya tentukan di mana Anda menyimpan APK yang dihasilkan. Di sini kami membiarkan secara default. Klik **Finish** untuk memulai generate signed APK.



7) Ketika berhasil sukses, akan ada pesan notifikasi seperti ini.



8) Selamat APK aplikasi android versi release sudah berhasil dibuat.

3.5. Test Akhir

Dari tutorial di atas, silahkan praktekkan dan tuliskan kesulitan-kesulitan selama proses instalasi emulator, menghubungkan smartphone dengan Android Studio dan kesulitan saat build executable Aplikasi Android/Build APK (Android Application Package). Kemudian buat APK dari aplikasi sederhana yang sudah dibuat pada BAB 2.

3.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

Sumber :

<https://developer.android.com/studio/>
<https://developer.android.com/studio/run/device>
<https://developer.android.com/studio/run/emulator>

BAB 4 – ACTIVITY

4.1. IDENTITAS

Kompetensi

1. Memberikan pemahaman tentang Activity
2. Dapat membuat layout .xml

Topik

1. Activity di Android Studio
2. Studi Kasus Membuat layout untuk menghitung luas

4.2. Test Awal

Kerjakan test awal dibawah ini!

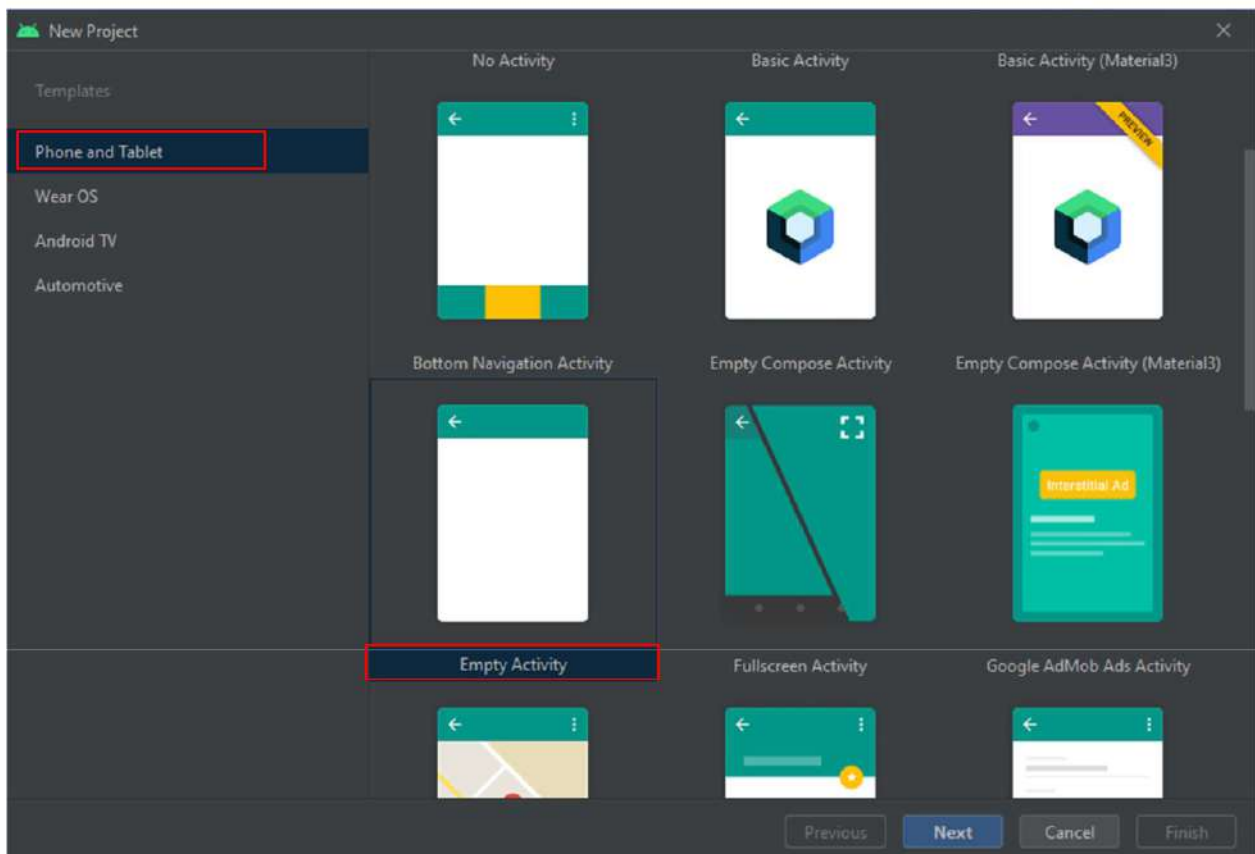
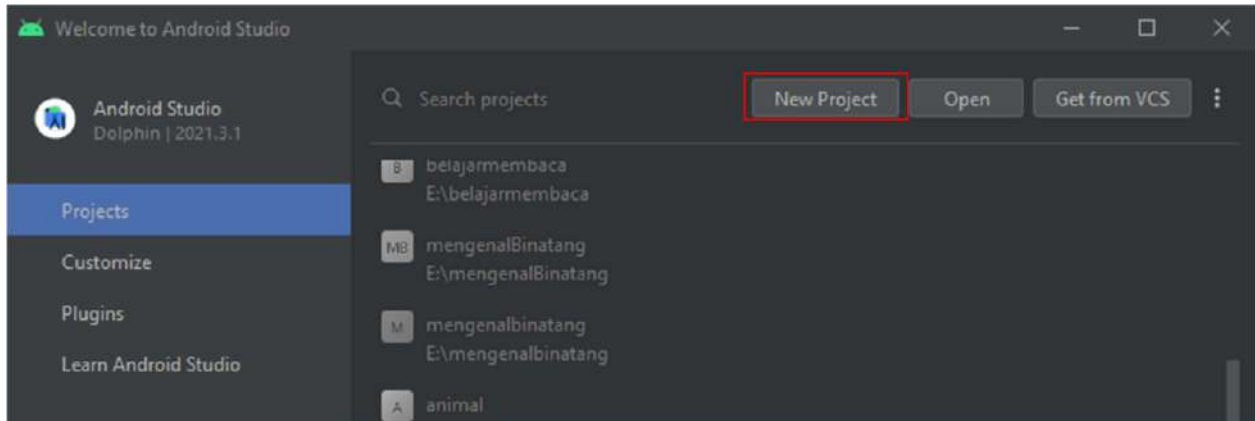
- a. Apa yang Anda ketahui tentang activity di Android Studio?

- b. Apa fungsi utama dari adanya activity di project yang menggunakan android studio?

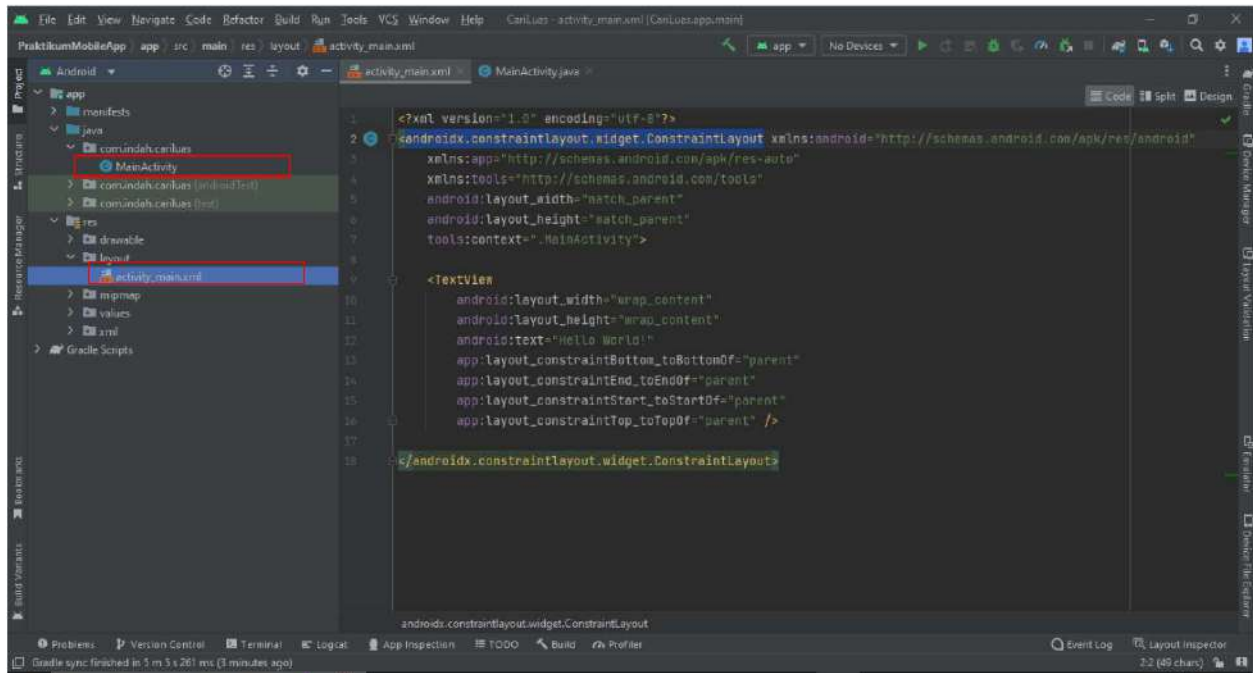
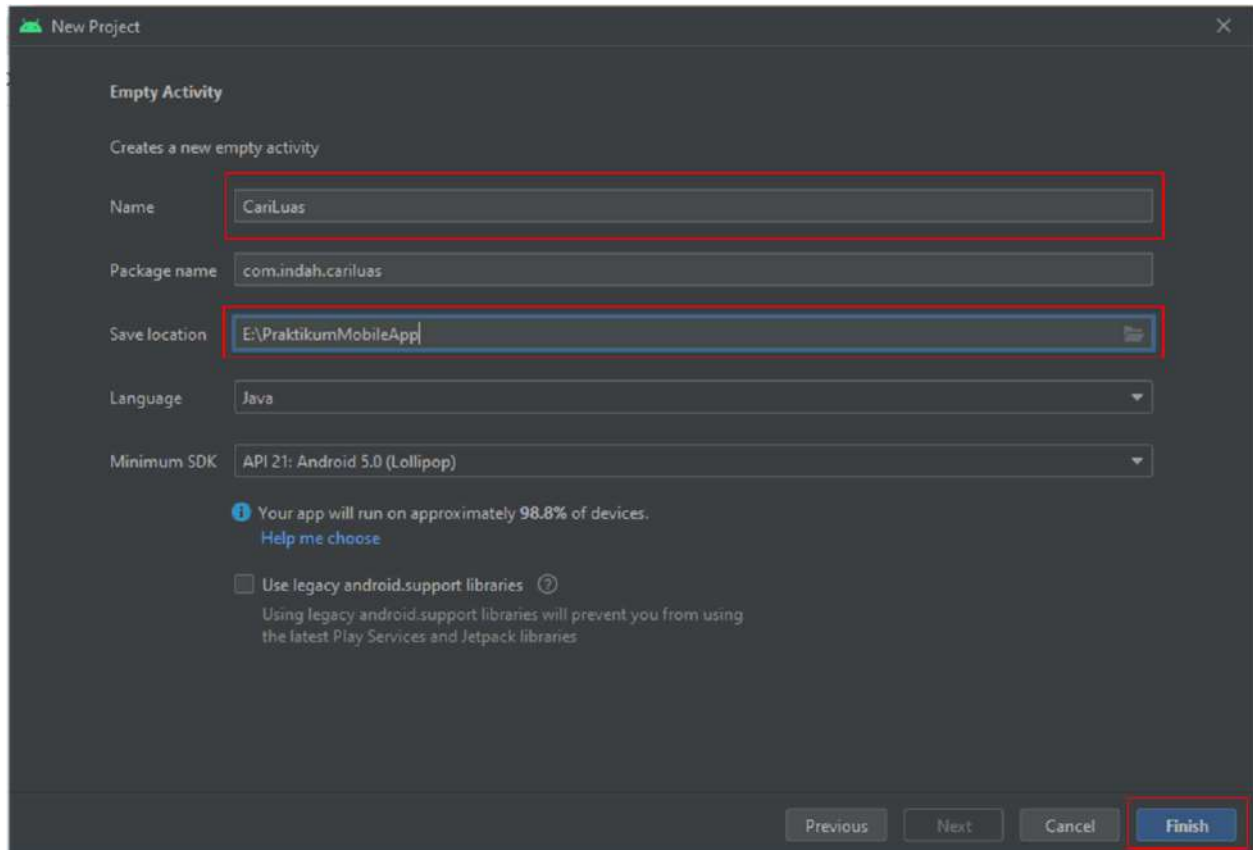
- c. Template apa saja yang ada di Activity? Sebutkan!

4.3. Materi

Activity yang biasa disebut layout adalah komponen yang menghubungkan antara pengguna dengan aplikasi. Activity nantinya digunakan untuk menampilkan user interface ke pengguna, seperti menampilkan list maupun tombol/button. Activity dapat berupa teks maupun gambar. Ketika membuat project baru di Android Studio, biasanya akan ada dua berkas yang sudah default tercipta, yaitu **MainActivity** dan **activity_main.xml**. MainActivity ini disebut sebagai class Activity. Digunakan untuk menampilkan layout **activity_main.xml**.

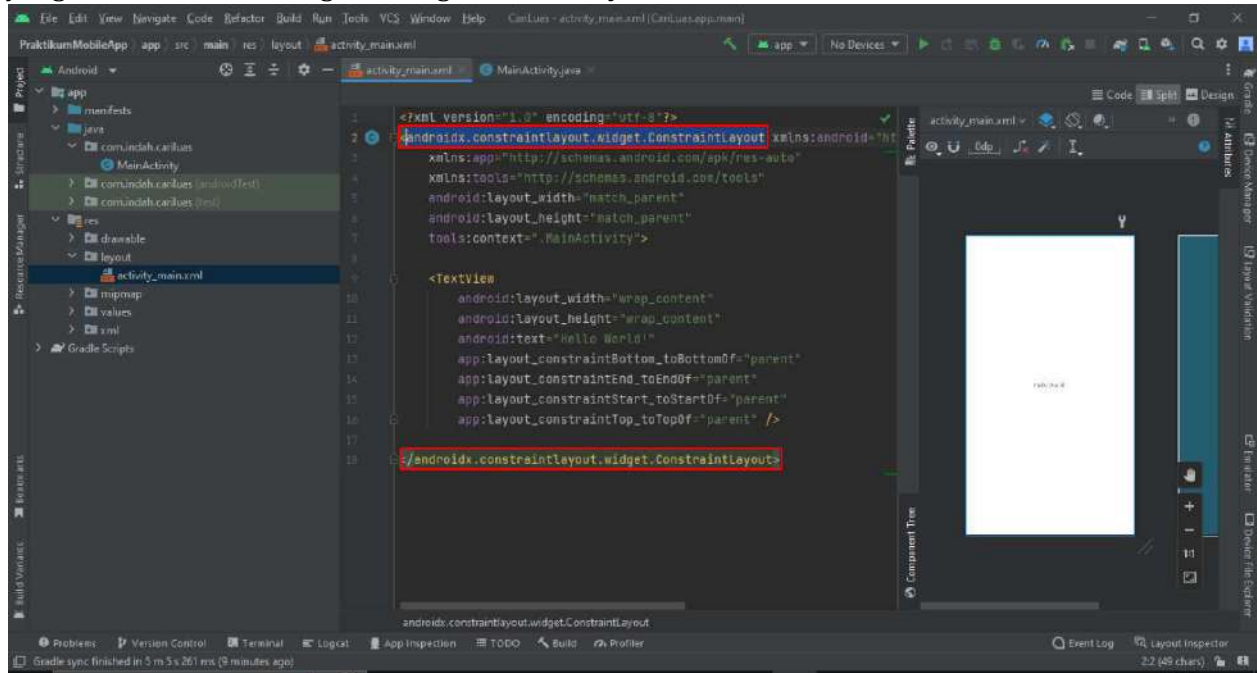


Beri nama pada **Name** → **Finish**

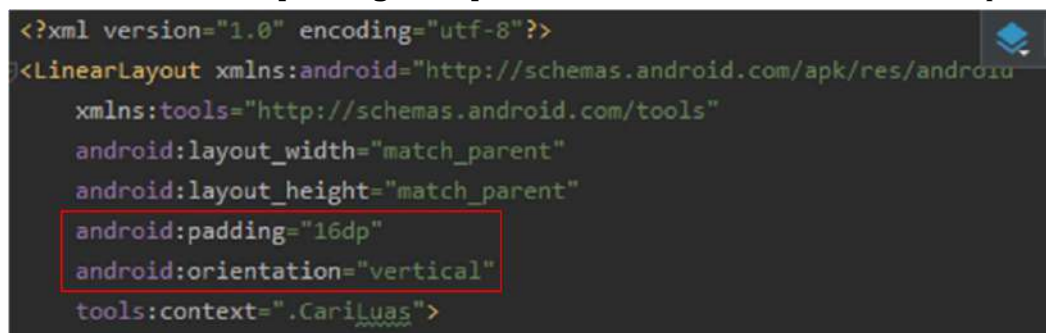


Selanjutnya kita akan mencoba mendesain tampilan untuk mencari luas persegi panjang. Silahkan dilihat tahap-tahapannya pada gambar-gambar di bawah ini.

Klik **Split** untuk dapat menampilkan halaman coding bersamaan dengan design. Kemudian blog code yang diberi tanda merah ganti dengan **LinearLayout**.



Tambahkan **android:padding="16dp"** dan **android:orientation="vertical"** pada code.



Root layout (tag layout terluar) yang dipakai di sini adalah **LinearLayout**. Jika Android Studio yang diinstall adalah versi 3 ke atas, secara *default* root yang dipakai adalah **ConstraintLayout**. Dalam praktikum ini dapat disesuaikan dengan mengubahnya menjadi LinearLayout.

Selanjutnya tuliskan code seperti di bawah ini.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:padding="16dp"
    android:orientation="vertical"
    tools:context=".Cariluas">

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

    <EditText
        android:id="@+id/isi_panjang"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:lines="1" />

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

    <EditText
        android:id="@+id/isi_lebar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:lines="1" />

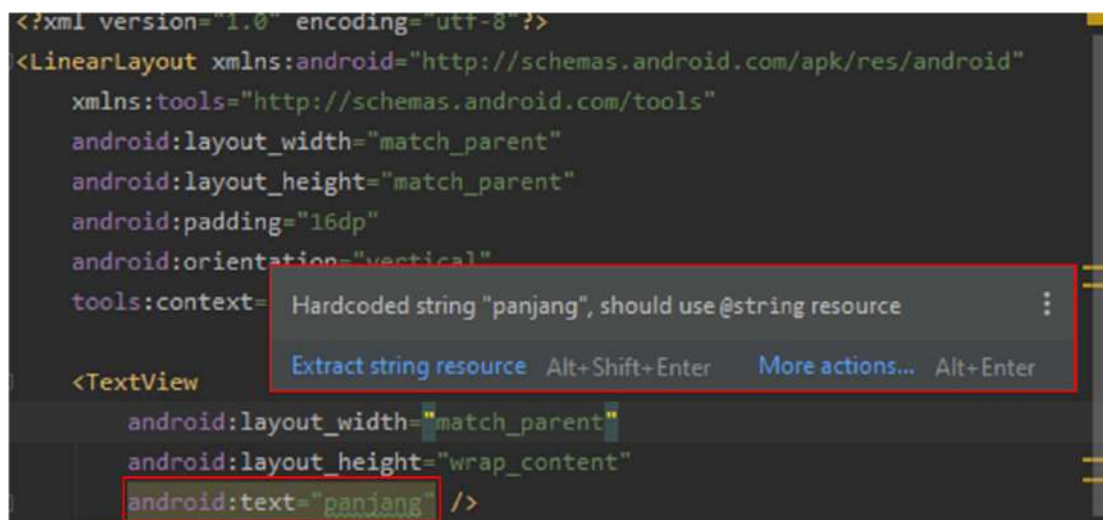
    <Button
        android:id="@+id/btn_hitung"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hitung" />

    <TextView
        android:id="@+id/tv_hasil"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Hasil"
        android:textSize="24sp"
        android:textStyle="bold" />

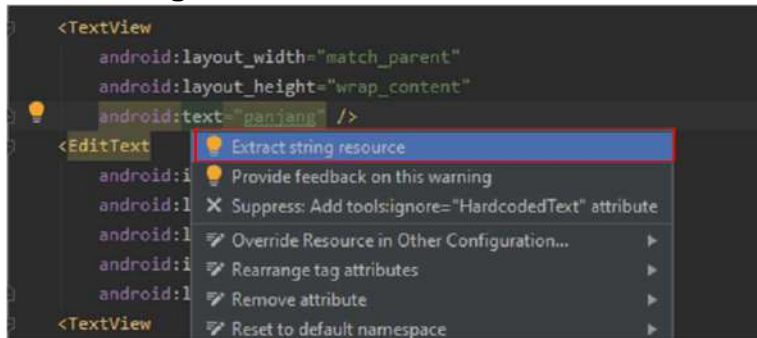
</LinearLayout>

```

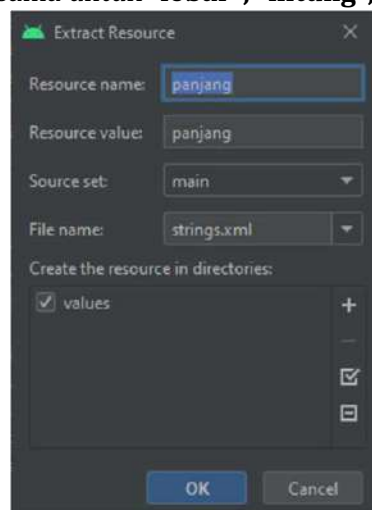
Jika kita perhatikan akan muncul tanda warning dibagian **android:text**, dikatakan bahwa kita telah melakukan *Hardcoding* (menulis teks secara langsung). Hal ini dikatakan kurang baik karena seharusnya semua teks kita tuliskan di **res/values/strings.xml** dan kemudian kita panggil dalam coding.



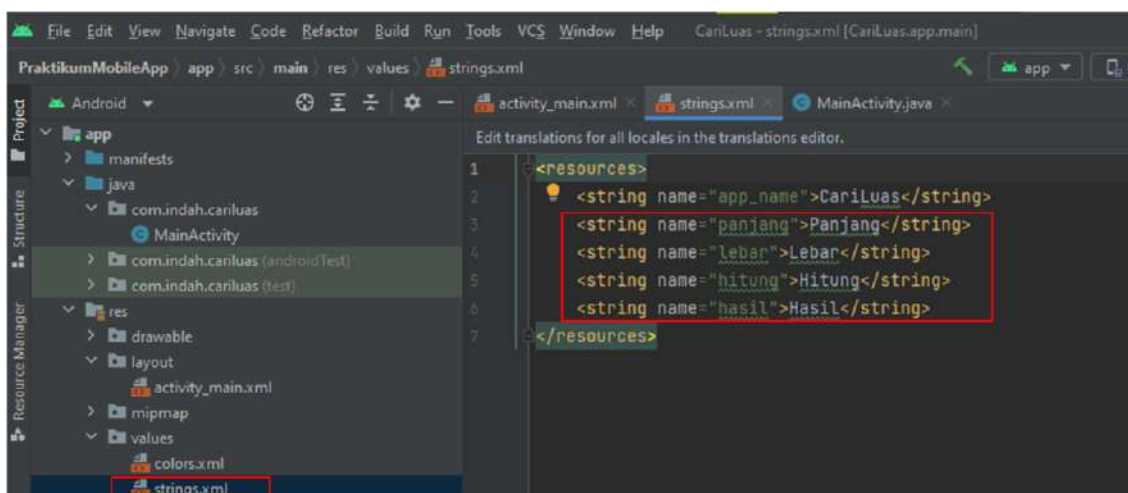
Kita dapat hilangkan *warning* tersebut dengan menekan **Alt+Enter** atau klik lampu kuning yang muncul pada sebelah kiri android:text. Ketika tampil seperti gambar di bawah ini kemudian pilih **extract string resource**.



Maka akan muncul tampilan dialog seperti pada gambar di bawah ini. Silahkan diisi sesuai dengan nama yang ada. Lakukan hal yang sama untuk **“lebar”**, **“hitung”**, dan **“hasil”**.



Fungsi **extract string resource** akan secara otomatis menambahkan nilai ke dalam berkas **strings.xml** seperti pada gambar di bawah ini.



Untuk itu codingan di berkas **activity_cari_luas.xml** akan berubah seperti pada gambar di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:orientation="vertical"
    tools:context=".Cariluas">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/panjang" />
    <EditText
        android:id="@+id/isi_panjang"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:lines="1" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/lebar" />
    <EditText
        android:id="@+id/isi_lebar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:lines="1" />

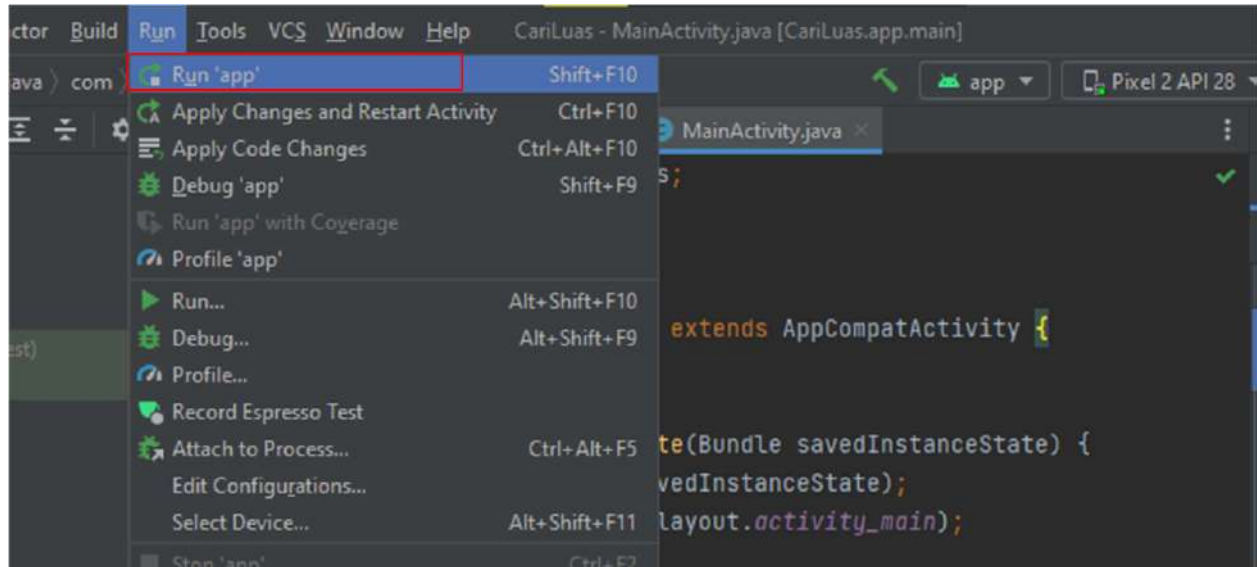
    <Button
        android:id="@+id/btn_hitung"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/hitung" />
    <TextView
        android:id="@+id/tv_hasil"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/hasil"
        android:textSize="24sp"
        android:textStyle="bold" />

</LinearLayout>
```

Berikut adalah tampilan desain dari berkas **activity_cari_luas.xml**

The image shows a mobile application interface for calculating area. It consists of a vertical stack of elements: a text input field labeled 'Panjang', another text input field labeled 'Lebar', a blue button with the text 'HITUNG', and a text label 'Hasil'.

Kita dapat menjalankan aplikasi dengan cara klik **Run** → **Run 'app'**



Maka akan terhubung dengan emulator untuk menampilkan desain layout dari **activity_cari_luas.xml**



4.4. Studi Kasus

Melanjutkan materi yang disampaikan di 4.3, mari kita lanjutkan untuk mencari luas persegi panjang dengan cara membuat logikanya di **MainActivity.java**. Tuliskan code sebagai berikut:

```

package com.indah.cariluas;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    private EditText edtPanjang, edtLebar;
    private Button btnHitung;
    private TextView tvHasil;
    private static final String STATE_HASIL = "state_hasil";

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

        edtPanjang = findViewById(R.id.edt_panjang);
        edtLebar = findViewById(R.id.edt_lebar);
        btnHitung = findViewById(R.id.btn_hitung);
        tvHasil = findViewById(R.id.tv_hasil);

        btnHitung.setOnClickListener(this);

        if (savedInstanceState != null) {
            String hasil = savedInstanceState.getString(STATE_HASIL);
            tvHasil.setText(hasil);
        }

    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_hitung) {
            String inputPanjang = edtPanjang.getText().toString().trim();
            String inputLebar = edtLebar.getText().toString().trim();

            boolean isEmptyFields = false;

            if (TextUtils.isEmpty(inputPanjang)) {
                isEmptyFields = true;
                edtPanjang.setError("Isikan Panjang Persegi Panjang");
            }

            if (TextUtils.isEmpty(inputLebar)) {
                isEmptyFields = true;
                edtLebar.setError("Isikan Lebar Persegi Panjang");
            }

            if (!isEmptyFields) {
                double luas = Double.valueOf(inputPanjang) *
Double.valueOf(inputLebar);
                tvHasil.setText(String.valueOf(luas));
            }
        }
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putString(STATE_HASIL, tvHasil.getText().toString());
    }
}

```

Jalankan kembali aplikasi dengan klik **Run → Run 'App'**, maka hasilnya adalah seperti pada gambar di bawah ini.



4.5. Test Akhir

Dari contoh studi kasus di 4.4, silahkan membuat layout di berkas **xml** untuk mencari luas segitiga. Selanjutnya buat logika perhitungan luas segitiga di **MainActivity.java**, dan jalankan di emulator atau di device secara langsung dan di screen shoot hasilnya.

4.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

BAB 5 – INTENT

5.1. IDENTITAS

Kompetensi

1. Memberikan pemahaman tentang Intent
2. Dapat membuat intent

Topik

1. Intent di Android Studio
2. Studi Kasus Membuat intent untuk menampilkan detail mata kuliah

5.2. Test Awal

Kerjakan test awal dibawah ini!

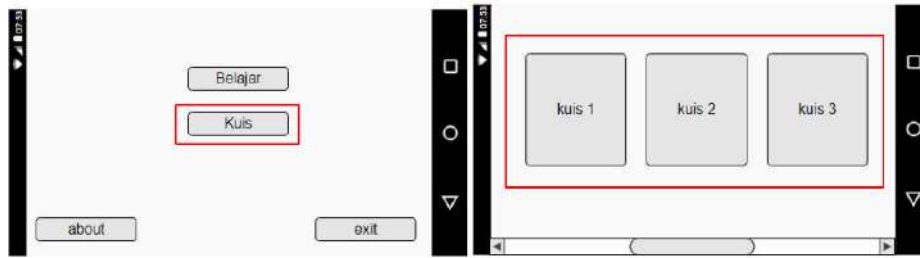
- a. Apa yang Anda ketahui tentang intent di Android Studio?

- b. Ada berapa jenis intent? Sebutkan!

- c. Apa perbedaan dari tiap-tiap intent yang ada di Soal 5.2.b?

5.3. Materi

Intent digunakan untuk mengendalikan activity yang memiliki proses lebih dari satu, yaitu antara activity satu dengan activity lain pada suatu aplikasi. Ilustrasi intent dapat dilihat pada gambar di bawah ini.



Pada gambar di atas terlihat bahwa ketika kita memilih menu kuis maka akan ditampilkan halaman baru yang mana akan ada pilihan-pilihan level pada kuis, yaitu **Kuis 1, Kuis 2, Kuis 3, dst.** Dan ketika di klik **Kuis 1** akan menampilkan halaman baru lagi untuk detail Kuis 1, dst.

Terdapat 2 macam intent:

❑ Intent eksplisit

Kita bisa menggunakan intent eksplisit untuk memulai sebuah komponen pada aplikasi kita sendiri, karena kita sudah mengetahui nama class dari aktivitas atau layanan yang ingin dimulai.

Misalnya, memulai aktivitas baru pada aplikasi sebagai respons terhadap suatu tindakan

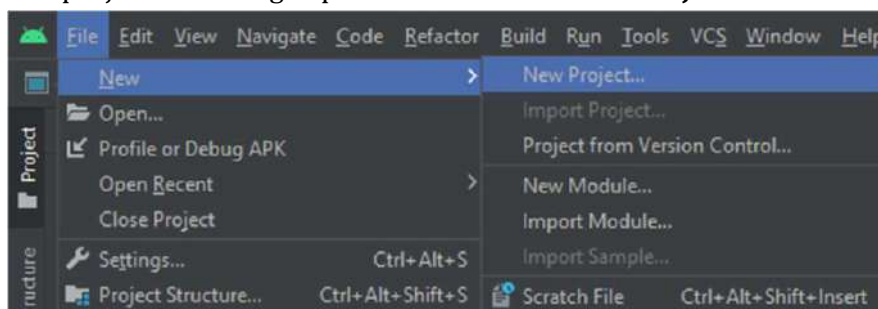
❑ Intent implisit

Tidak menetapkan komponen tertentu, melainkan mendeklarasikan tindakan umum yang akan dilakukan, yang memungkinkan komponen aplikasi lain untuk menanganinya.

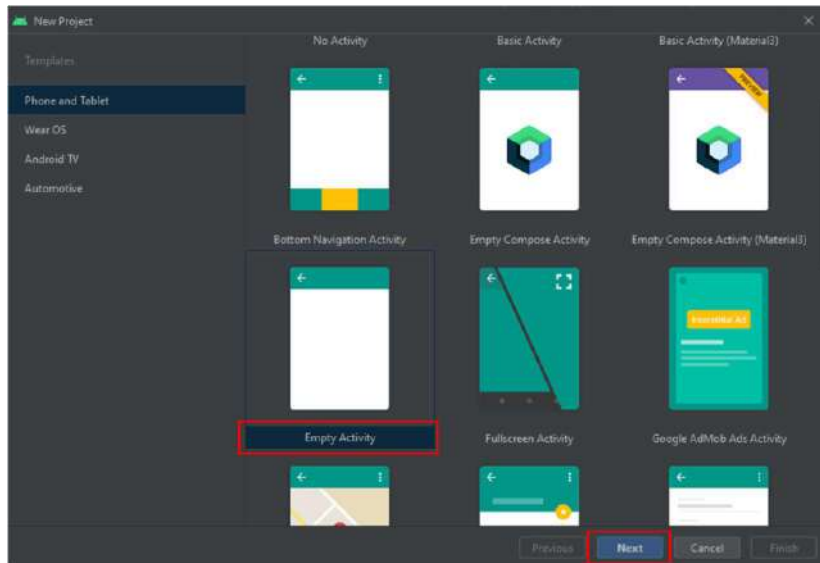
Misalnya, jika ingin membuat button yang dapat digunakan untuk menghubungi seseorang, maka dapat menggunakan intent implisit untuk langsung terhubung ke layanan **dial up** sehingga dapat langsung *men-dial* sebuah nomor telepon melalui aplikasi bawaan dari peranti bawaan.

Berikut ini adalah cara untuk membuat intent yang dapat mengendalikan proses pada activity yang lebih dari 1 pada 1 aplikasi yang sama.

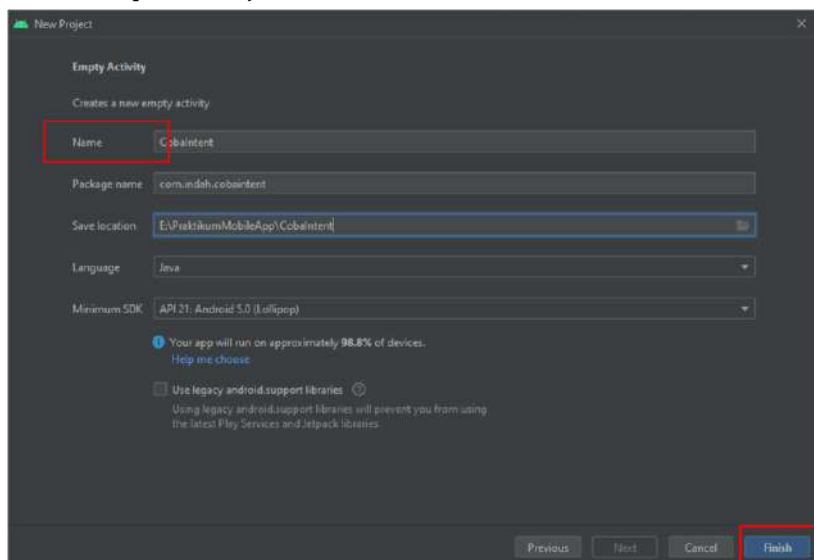
1. Buat project baru dengan pilih **File → New → New Project...**



2. Kemudian pilih template activity (**Empty Activity**)



3. Beri nama pada **Project Name** kemudian klik **Finish**.



4. Tambahkan sintaks di bawah ini pada berkas **MainActivity.java**

```

package com.indah.cobaintent;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener{

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

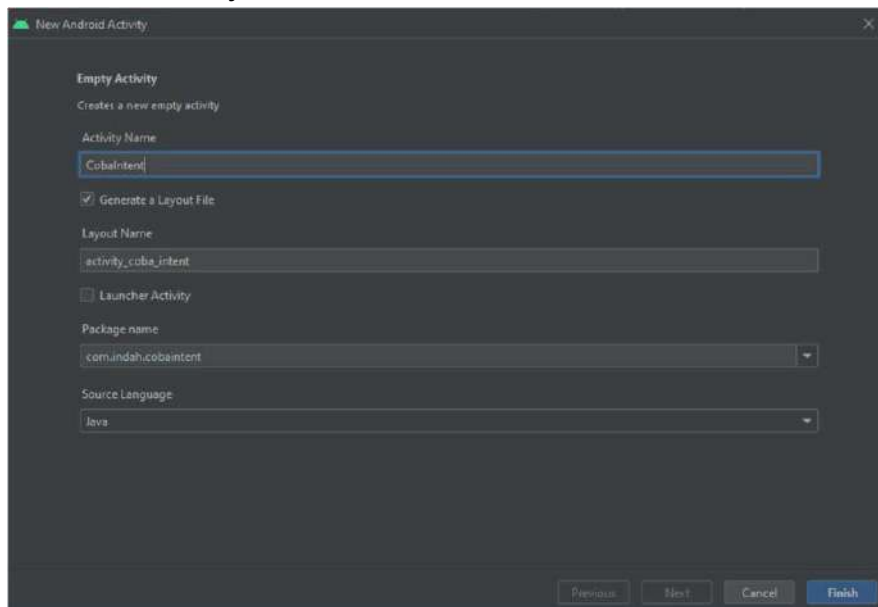
        Button btnPindahActivity = findViewById(R.id.btn_pindah_activity);
        btnPindahActivity.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_pindah_activity) {

        }
    }
}

```

5. Langkah selanjutnya adalah membuat class baru dengan cara pilih **File → New → Activity → Empty Activity**
6. Beri nama **Activity Name**



7. Buka **activity_main.xml** dan buat sintaks seperti di bawah ini

```

<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:padding="16dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btn_pindah_activity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text="@string/pindah_activity" />

</LinearLayout>

```

8. Jangan lupa menambahkan sintaks di **res** → **values** → **strings.xml**

```
<resources>
    <string name="app_name">CobaIntent</string>
    <string name="pindah_activity">Coba Intent</string>
</resources>
```

9. Tampilan layout akan Berubah menjadi seperti ini



10. Selanjutnya buka berkas **activity_coba_intent.xml** dan ubah codingan menjadi seperti pada gambar di bawah ini

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".CobaIntent">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/purple_500"
        android:text="Coba Intent Berhasil" />

</LinearLayout>
```

11. Maka tampilan layout berkas **activity_coba_intent.xml** akan berubah menjadi seperti pada gambar di bawah ini



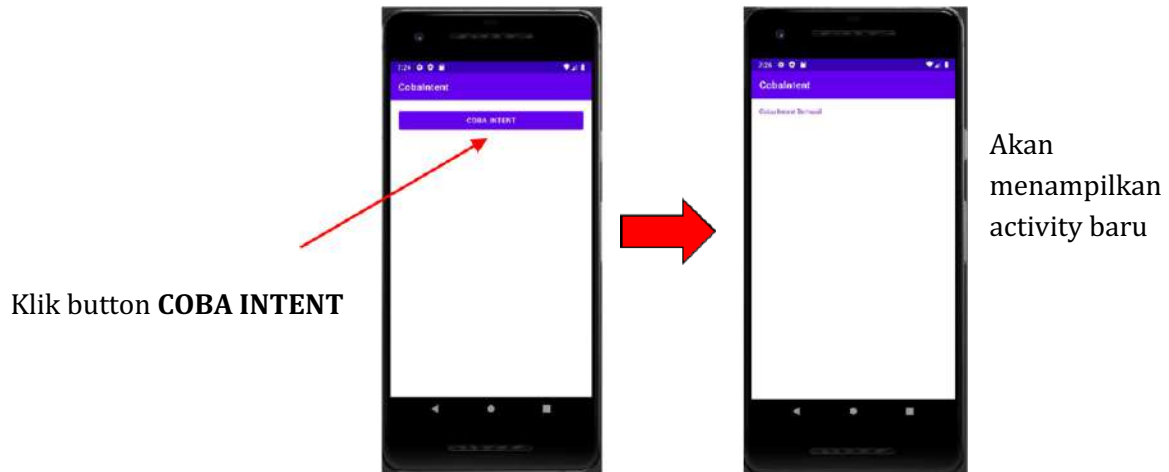
12. Selanjutnya tambahkan Intent pada method **onClick()** di **MainActivity.java** menjadi sebagai pada gambar di bawah ini


```

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_pindah_activity) {
        Intent pindahIntent = new Intent(MainActivity.this,
        CobaIntent.class);
        startActivity(pindahIntent);
    }
}

```

13. Jalankan program dengan klik **Run**, maka tampilan akan seperti pada gambar di bawah ini



5.4. Studi Kasus

Pada studi kasus kali ini kita akan mencoba membuat intent untuk menampilkan biodata dengan memodifikasi codingan di project **CobaIntent**.

1. Buka berkas **activity_main.xml** dan buat codingan menjadi seperti ini

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btn_Rumaisha"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text="@string/pindah_activity1" />

    <Button
        android:id="@+id/btn_Indah"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text="@string/pindah_activity2" />

</LinearLayout>

```

2. Jangan lupa menambahkan string di **res** → **values** → **String.xml**

```

<resources>
  <string name="app_name">CobaIntent</string>
  <string name="pindah_activity1">Biodata 1</string>
  <string name="pindah_activity2">Biodata 2</string>
  <string name="nama1">Nama : Rumaisha Naura Salsabila</string>
  <string name="alamat1">Alamat : Magetan</string>
  <string name="email1">Email : rumaisha@xxxx.com</string>
</resources>

```

3. Maka tampilan layout akan berubah menjadi seperti pada gambar di bawah ini



4. Langkah selanjutnya adalah membuat tampilan Biodata 1 (**activity_coba_intent.xml**)

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="16dp"
  tools:context=".CobaIntent">

  <TextView
    android:id="@+id/nama1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/nama1"
    android:textColor="@color/purple_500" />

  <TextView
    android:id="@+id/alamat1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/nama1"
    android:text="@string/alamat1"
    android:textColor="@color/purple_500" />

  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/alamat1"
    android:text="@string/email1"
    android:textColor="@color/purple_500" />

</RelativeLayout>

```

5. Tampilan Layout dari berkas **activity_coba_intent.xml** akan berubah menjadi seperti gambar di bawah ini

Nama : Rumaisha Naura Salsabila
Alamat : Magetan
Email: rumaisha@xxxx.com

6. Selanjutnya kita buat logikanya di **MainActivity.java**

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener{

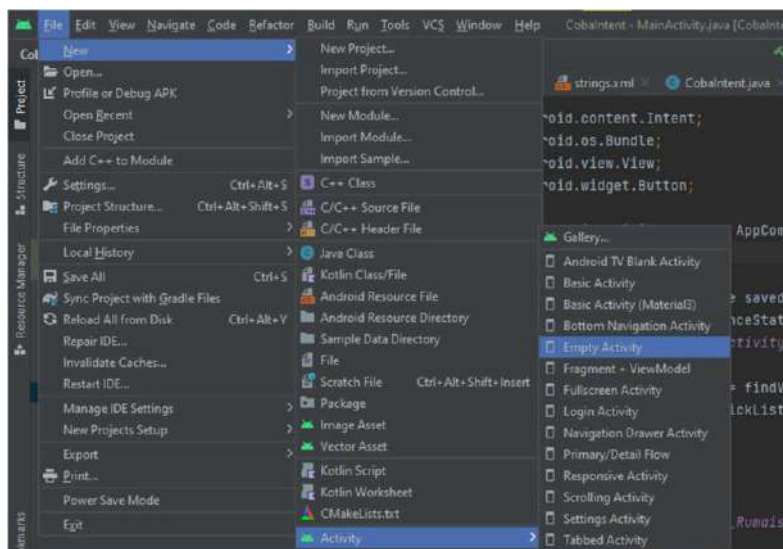
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btnPindahActivity = findViewById(R.id.btn_Rumaisha);
        btnPindahActivity.setOnClickListener(this);
    }

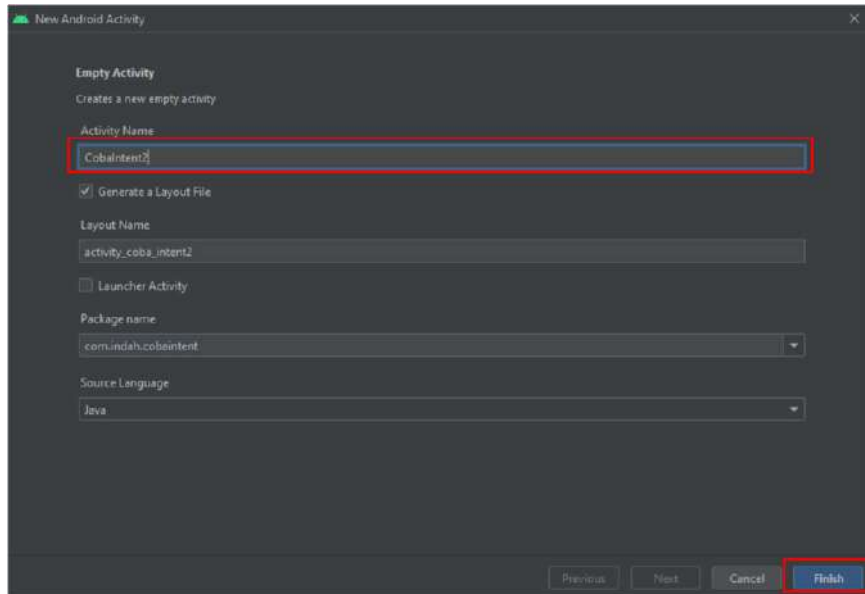
    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_Rumaisha) {

            Intent pindahIntent = new Intent(MainActivity.this,
CobaIntent.class);
            startActivity(pindahIntent);
        }
    }
}
```

7. Di project CobaIntent terdapat 2 button biodata, untuk itu kita perlu menambahkan 1 activity. Klik



8. Jangan lupa memberi **nama activity**



9. Kemudian kita buat layout yang kedua dimana langkah-langkahnya sama dengan membuat biodata dengan nama Rumaisha.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".CobaIntent2">

    <TextView
        android:id="@+id/nama2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/nama2"
        android:textColor="@color/purple_500" />

    <TextView
        android:id="@+id/alamat2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/nama2"
        android:text="@string/alamat2"
        android:textColor="@color/purple_500" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/alamat2"
        android:text="@string/email2"
        android:textColor="@color/purple_500" />
</RelativeLayout>
```

10. Jangan lupa menambahkan string di **res** → **values** → **String.xml**

```

<resources>
  <string name="app_name">CobaIntent</string>
  <string name="pindah_activity1">Biodata 1</string>
  <string name="pindah_activity2">Biodata 2</string>
  <string name="nama1">Nama : Rumaisha Naura Salsabila</string>
  <string name="alamat1">Alamat : Magetan</string>
  <string name="email1">Email : rumaisha@xxxx.com</string>
  <string name="nama2">Nama : Indah Puji Astuti </string>
  <string name="alamat2">Alamat : Ponorogo</string>
  <string name="email2">Email : indah@xxxx.com</string>
</resources>

```

11. Tampilan layout dari berkas CobaIntent2.xml akan menjadi seperti pada gambar di bawah ini

```

Nama : Indah Puji Astuti
Alamat : Ponorogo
Email : indah@xxxx.com

```

12. Selanjutnya kita tambahkan logika di **MainActivity.java** seperti pada gambar di bawah ini

```

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

    Button btnPindahActivity = findViewById(R.id.btn_Rumaisha);
    btnPindahActivity.setOnClickListener(this);

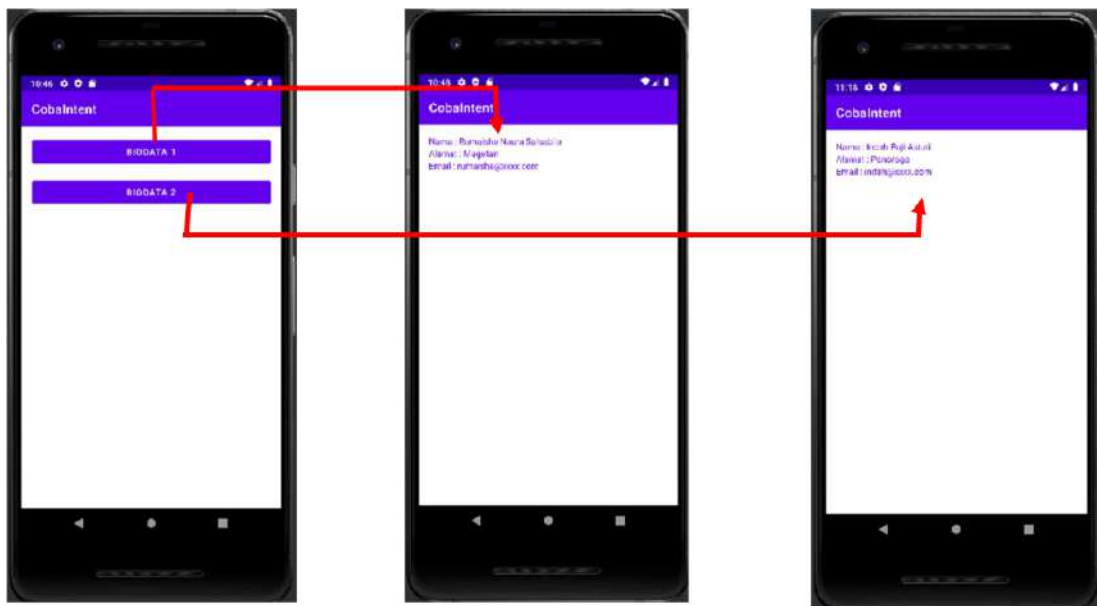
    Button btnPindahActivity2 = findViewById(R.id.btn_Indah);
    btnPindahActivity2.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_Rumaisha) {
        Intent pindahIntent = new Intent(MainActivity.this,
CobaIntent.class);
        startActivity(pindahIntent);
    }

    else if (v.getId() == R.id.btn_Indah) {
        Intent pindahIntent = new Intent(MainActivity.this,
CobaIntent2.class);
        startActivity(pindahIntent);
    }
}
}

```

13. Jalankan program dengan klik **Run**, maka tampilan akan tampak seperti pada gambar di bawah ini



5.5. Test Akhir

Dari contoh studi kasus di 5.4, silahkan membuat intent untuk menampilkan detail mata kuliah, jika data mata kuliah adalah sebagai berikut:

No.	Nama Mata Kuliah	Pengampu	Kode Mata Kuliah	Deskripsi
1.	Analisis dan Desain Sistem	Indah Puji Astuti	MKW202201	Pada mata kuliah ini mahasiswa belajar untuk menjadi seorang analis sistem. Mahasiswa belajar berbagai metode pengembangan sistem, cara-cara menganalisis permasalahan dan membuat solusi alternative tentang system yang sedang berjalan. Setelah Memahami konsep analisis mahasiswa Belajar tentang bagaimana membuat perancangan sistem. Dalam mata kuliah ini mahasiswa diminta untuk menyelesaikan studi kasus kemudian menyusun laporan yang berisi dokumen analisis dan dokumen perancangan sert dapa mempresentasikannya secara berkelompok.
2.	Data Mining	Ghulam Asrofi Buntoro	MKW202202	Pada mata kuliah ini mahasiswa belajar untuk memahami data, mulai dari Teknik preprosesing sampai dengan Teknik mengolah data dengan beberapa algorima klasifikasi seperti Decision Tree, Algoritma K-nearest Neighbor, dan juga Bayesian sehingga diperoleh pola-pola tertentu yang dapat menjadi informasi yang berguna. Dalam mata kuliah ini mahasiswa diminta untuk menyelesaikan studi kasus kemudian menyelesaikannya dengan menerapkan algoritma data mining.

Jalankan aplikasi menggunakan emulator atau device secara langsung dan silahkan di screenshot hasilnya.

5.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

Sumber :

<https://developer.android.com/>

<https://developer.android.com/reference/android/content/Intent?hl=en>

BAB 6 – MENAMPILKAN GAMBAR DENGAN JAVA DAN XML

6.1. IDENTITAS

Kompetensi

1. Memberikan pemahaman drawable
2. Dapat menampilkan gambar

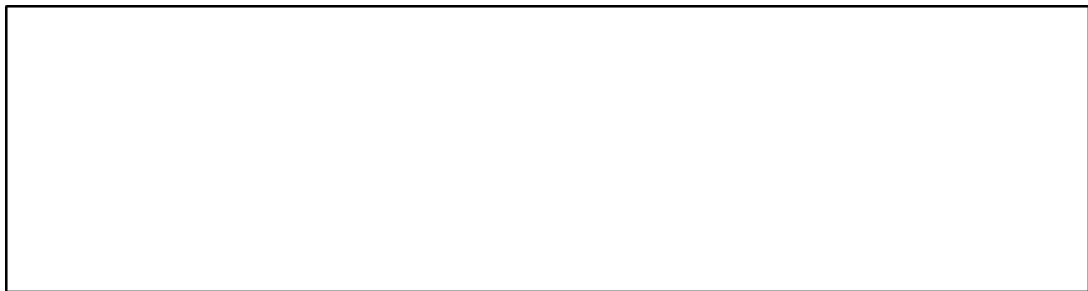
Topik

1. Drawable di android studio
2. Studi Kasus menampilkan nama-nama buah dengan gambar

6.2. Test Awal

Kerjakan test awal dibawah ini!

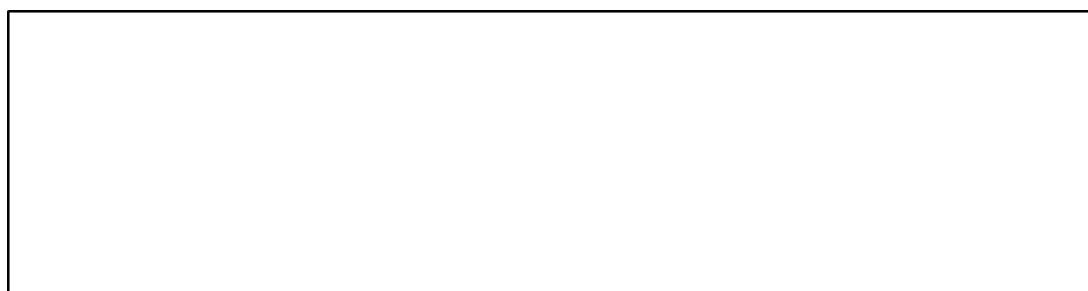
- a. Apa yang Anda ketahui tentang drawable di Android Studio?



- b. Apa fungsi utama dari adanya drawable android studio?



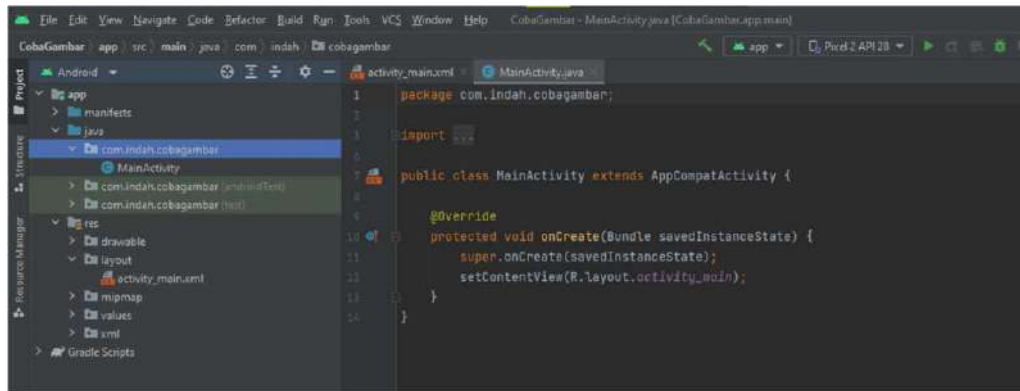
- c. Ada berapa macam cara menampilkan gambar dengan pemrograman android?



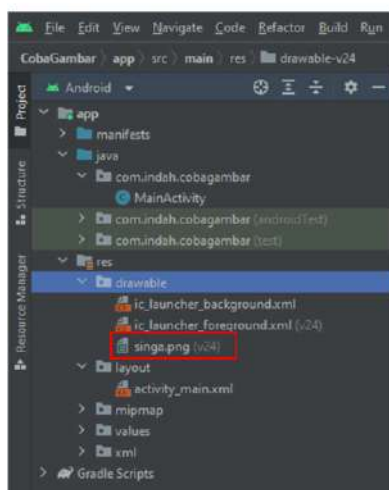
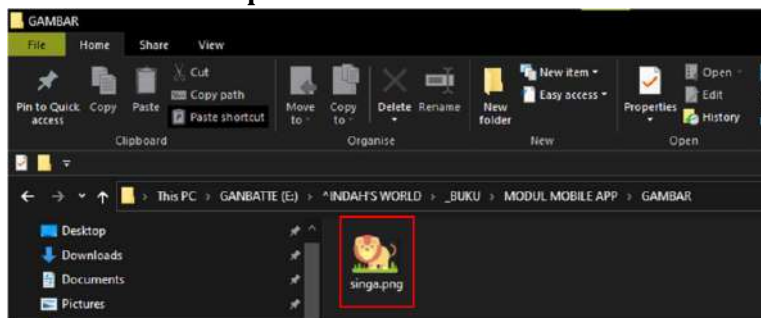
6.3. Materi

Program Android dapat menampilkan program image ataupun gambar selain dapat menampilkan teks. Untuk dapat menampilkan gambar dapat menggunakan berkas **.java** maupun **.xml**. pada praktikum di bab 6 ini akan dibahas terlebih dahulu tentang menampilkan gambar atau image dengan di berkas **.java**. ikuti langkah-langkah seperti urutan di bawah ini.

1. Buat project baru di Android Studio dengan klik **File → New → New Project...**
2. Jangan lupa memberi nama pada **Project Name (CobaGambar)**
3. Tunggu proses nya sampai selesai sehingga tampilan seperti pada gambar di bawah ini



4. Sebelum memulai untuk coding, siapkan terlebih dahulu gambar atau image yang nantinya akan diletakkan di folder **res → drawable**. Caranya klik copy pada gambar kemudian **klik kanan drawable** dan klik **paste**.



5. Selanjutnya kita buat logika di berkas **MainActivity.java**. Tuliskan coding seperti pada gambar di bawah ini

```
package com.indah.cobagambar;

import androidx.appcompat.app.AppCompatActivity;
import android.widget.ImageView;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

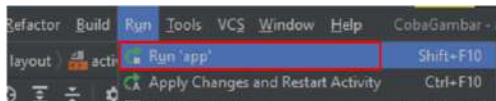
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ImageView TampilGambar = new ImageView(this);
        TampilGambar.setImageResource(R.drawable.singa);
        // setContentView(R.layout.activity_main);
        setContentView(TampilGambar);
    }
}
```

6. Kemudian buka berkas **activity_main.xml** yang ada dalam folder **res** → **layout**. Lakukan modifikasi pada sintaks sehingga menjadi seperti pada gambar di bawah ini

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

</LinearLayout>
```

7. Coba run aplikasi dengan klik tombol **Run 'App'**



8. Maka emulator akan menampilkan tampilan seperti pada gambar di bawah ini. Selamat, berarti sekarang kita sudah berhasil membuat aplikasi untuk menampilkan gambar melalui berkas **.java**

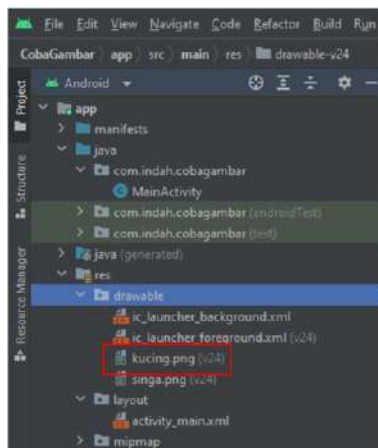
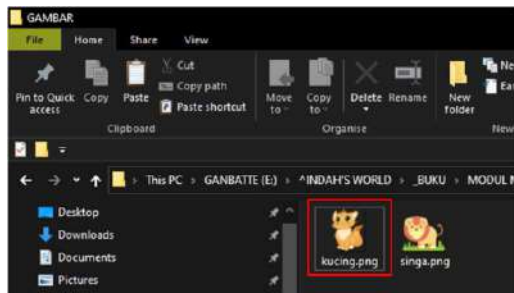


Kita memerlukan perintah **import android.widget.ImageView;** untuk menampilkan gambar atau image di berkas **.java**. Widget pada Android ditampilkan dengan konsep view, yang biasanya pada aplikasi Android menggunakan widget sebagai layout **.xml**. Namun dengan menggunakan sintaks atau perintah di berkas **.java** dapat diimplementasikan sebagai aksi dari widget. Hal ini dapat dilakukan dengan membuat sintaks di berkas **.java** yang memanggil semua nilai yang ada di layout **.xml** yang sudah didefinisikan sebelumnya.

Selanjutnya kita akan Mencoba untuk menampilkan gambar atau image menggunakan berkas **.xml**.

Ikuti langkah-langkah di bawah ini.

1. Sebelumnya tambahkan gambar yang berbeda di folder **drawable**



2. Buka kembali project CobaGambar yang dibuat sebelumnya, kemudian buka **MainActivity.java**. Nonaktifkan beberapa sintaks agar tidak bisa dijalankan pada waktu aplikasi di run, diantaranya adalah

```
package com.indah.cobagambar;

import androidx.appcompat.app.AppCompatActivity;
//import android.widget.ImageView;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //ImageView TampilGambar = new ImageView(this);
        //TampilGambar.setImageResource(R.drawable.singa);
        setContentView(R.layout.activity_main);
        //setContentView(TampilGambar);
    }
}
```

Pada berkas **.java** kali ini kita tidak memerlukan **import android.widget.ImageView**; karena di cara ke 2 ini kita akan menampilkan gambar dengan **.xml**

3. Langkah selanjutnya buka berkas **activity_main.xml**. Modifikasi sintaks sehingga tampil seperti pada gambar di bawah ini

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/kucing" />

</LinearLayout>
```

4. Jalankan project dengan klik **Run 'App'**. Maka akan tampil seperti pada gambar di bawah ini

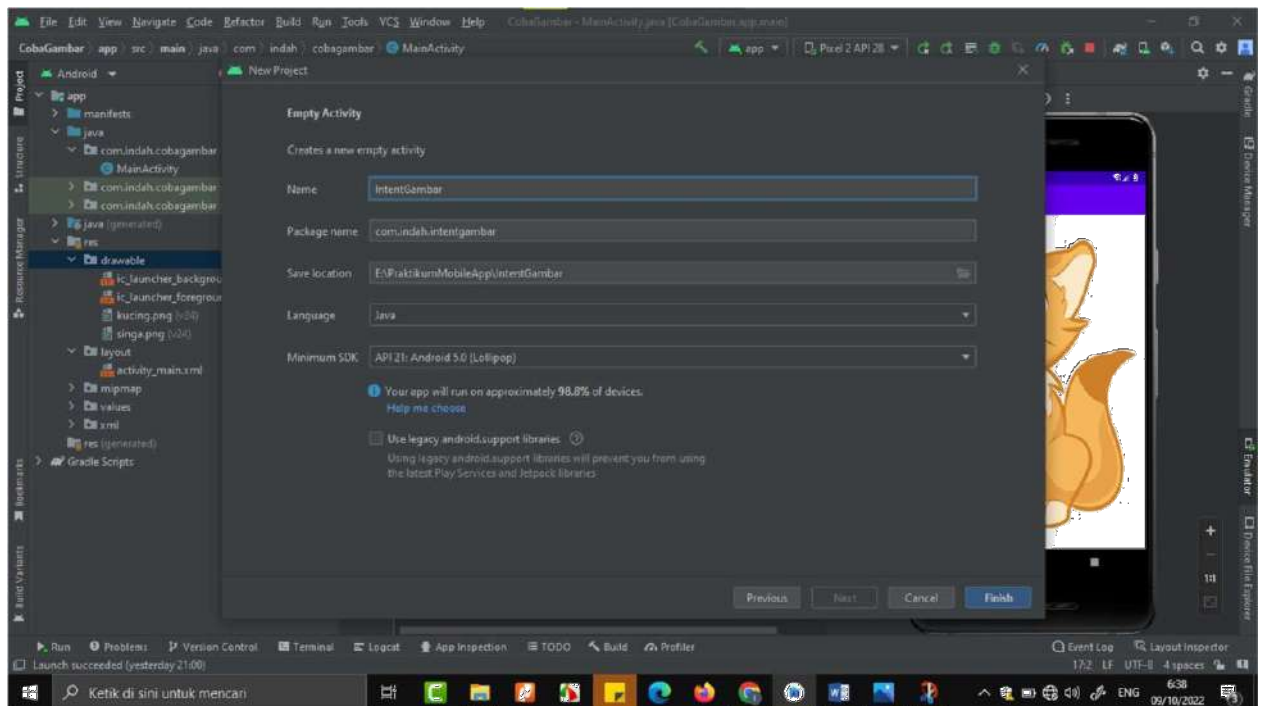


Folder drawable adalah folder yang berisi sumber semua gambar atau image yang digunakan dalam pembuatan aplikasi Android yang dibangun. Gambar-gambar tersebut nantinya akan dipanggil untuk ditampilkan pada sistem Android.

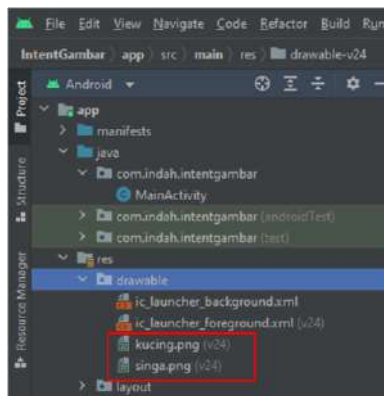
6.4. Studi Kasus

Pada studi kasus ini kita akan coba menampilkan lebih dari satu gambar dengan menggunakan button dan membutuhkan intent. Dimana ketika button di klik akan menampilkan halaman lain yaitu berupa gambar. Untuk langkah-langkahnya silahkan ikuti seperti tahapan di bawah ini

1. Membuat project baru



2. Menyiapkan gambar dan meletakkannya di folder drawable



3. Buka berkas **activity_main.xml** dan modifikasi sintaks seperti pada gambar di bawah ini

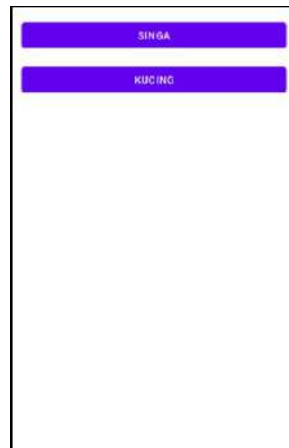
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btn_singa"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text="@string/singa" />

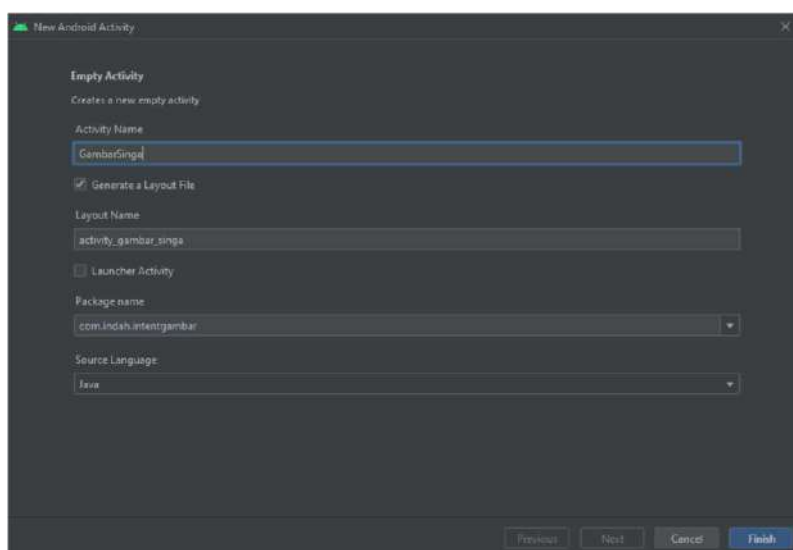
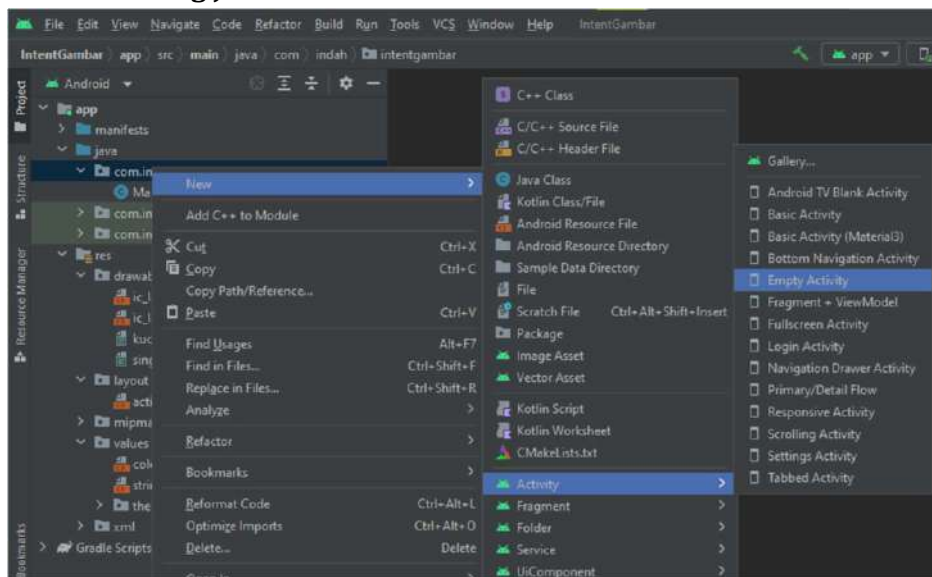
    <Button
        android:id="@+id/btn_kucing"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text="@string/kucing" />

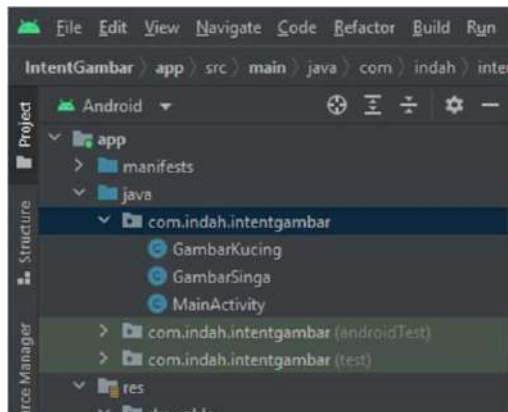
</LinearLayout>
```

Sehingga tampilan layout akan menjadi seperti gambar di baeah ini



4. Langkah selanjutnya kita membuat 2 aktivitas baru beri nama **GambarSinga.java** dan **GambarKucing.java**





5. Selanjutnya kita membuat logika di berkas **MainActivity.java**. sama seperti di materi intent. Silahkan modifikasi sintaks seperti pada gambar di bawah ini

```
package com.indah.intentgambar;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity implements
View.OnClickListener{

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

        Button btnPindahActivity = findViewById(R.id.btn_singa);
        btnPindahActivity.setOnClickListener(this);

        Button btnPindahActivity2 = findViewById(R.id.btn_kucing);
        btnPindahActivity2.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_singa) {
            Intent pindahIntent = new Intent(MainActivity.this,
GambarSinga.class);
            startActivity(pindahIntent);
        }

        else if (v.getId() == R.id.btn_kucing) {
            Intent pindahIntent = new Intent(MainActivity.this,
GambarKucing.class);
            startActivity(pindahIntent);
        }
    }
}
```

6. Pada studi kasus kali ini, kita mencoba menampilkan gambar dengan xml. Caranya yaitu kita buka berkas **activity_gambar_singa.xml** terlebih dahulu. Dan silahkan sintaks dimodifikasi seperti pada gambar di bawah ini

```

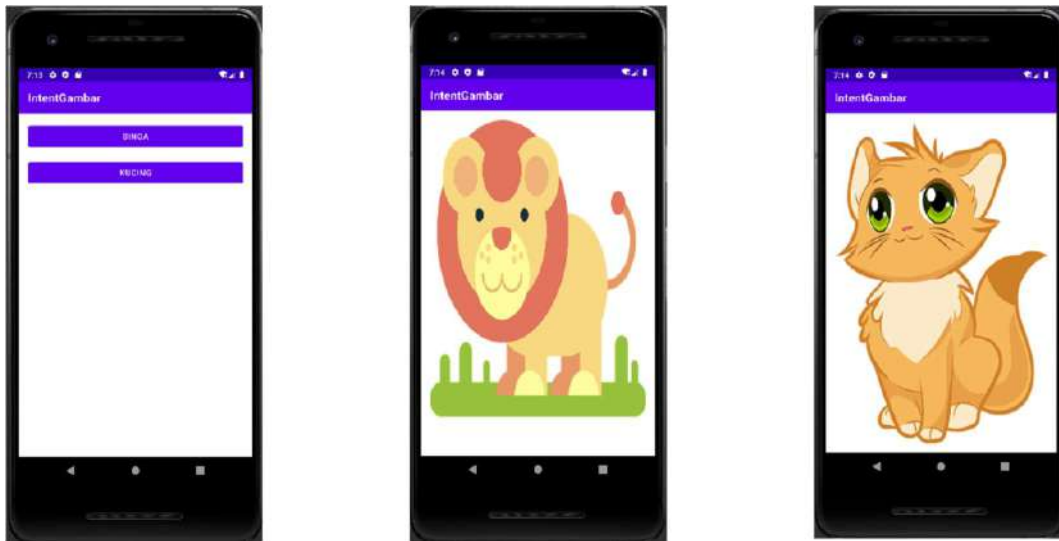
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".GambarSinga">

    <ImageView
        android:id="@+id/imageview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/singa" />

</LinearLayout>

```

7. Lakukan hal yang sama untuk berkas **activity_gambar_kucing.xml**
8. Jalankan aplikasi dengan klik **Run 'App'**. Maka tampilan aplikasi adalah seperti pada gambar di bawah ini



6.5. Test Akhir

Dari contoh studi kasus di 6.4, silahkan membuat program Android untuk menampilkan nama-nama buah beserta gambarnya (minimal 5 nama buah).

6.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

BAB 7 – PEMBUATAN VIEW

7.1. Identitas

Kompetensi

1. Mengetahui komponen-komponen view pada Android Studio
2. Dapat menggunakan komponen-komponen view pada Android Studio

Topik

1. View pada aplikasi Android
2. Komponen-komponen view pada Android Studio

7.2. Test Awal

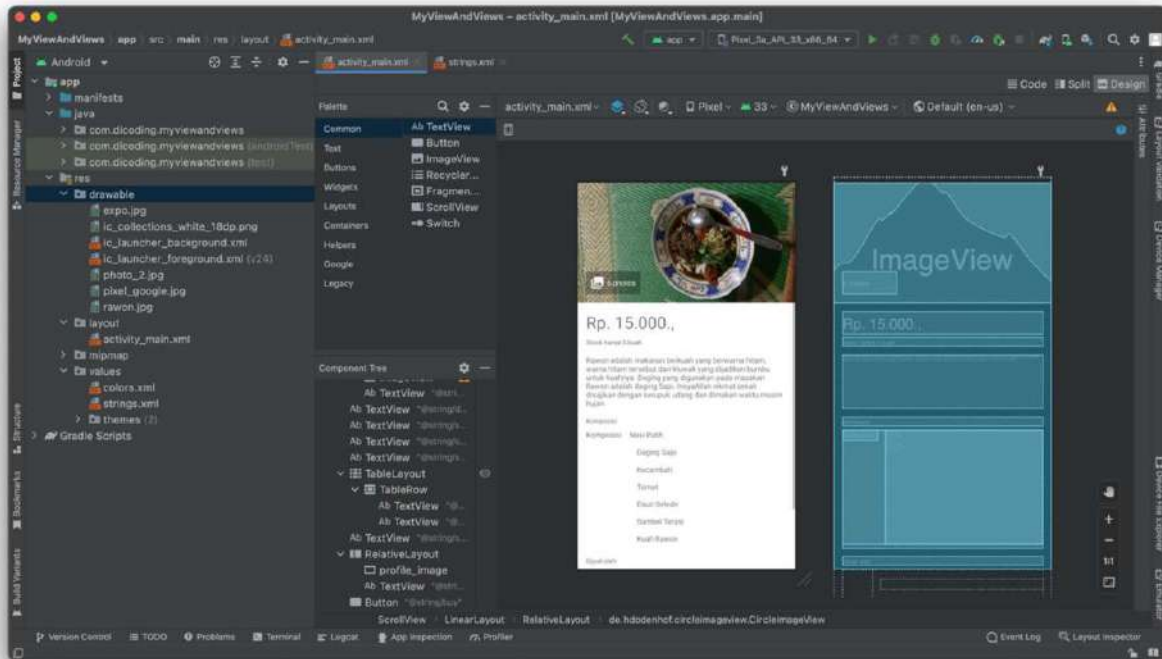
Kerjakan test awal dibawah ini!

- a. Apa yang Anda ketahui tentang view?

- b. Apa yang Anda ketahui tentang view pada Android?

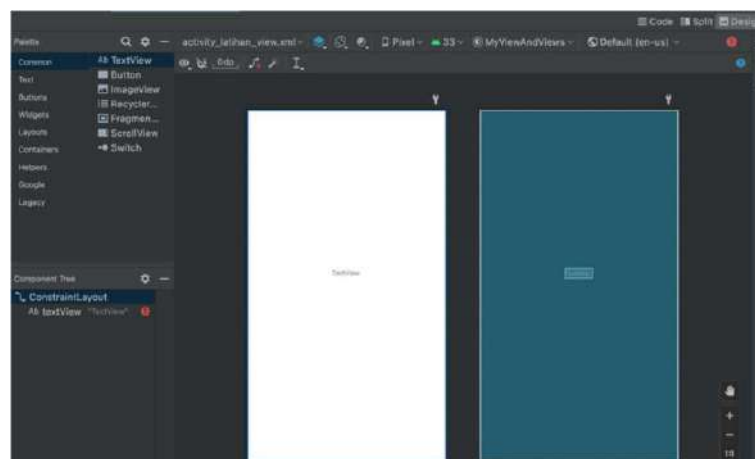
- c. Sebutkan menggunakan komponen-komponen view pada Android Studio!

7.3. Materi

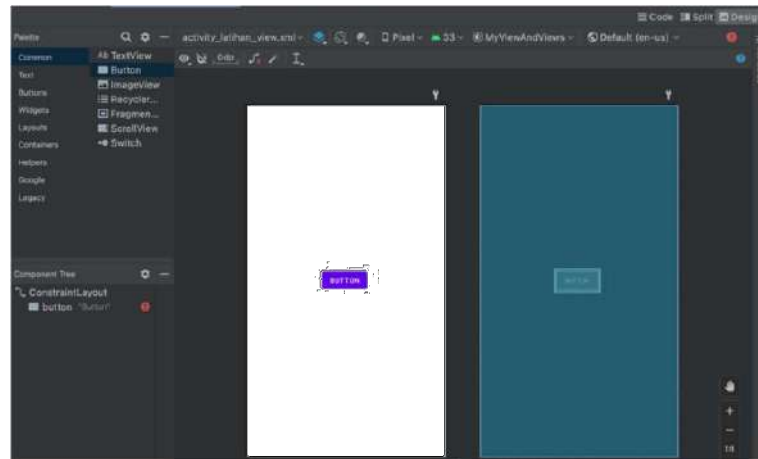


Sebuah view adalah obyek yang menggambar komponen tampilan ke layar yang mana pengguna dapat melihat dan berinteraksi langsung. Contoh komponen turunan dari view seperti :

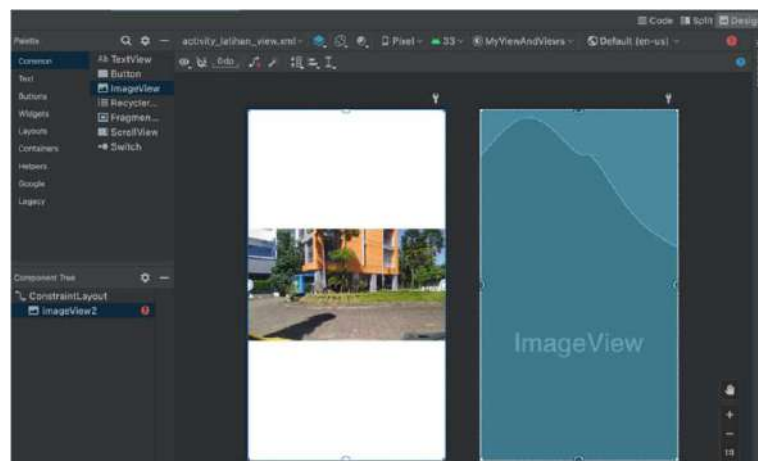
TextView, komponen yang berguna untuk menampilkan teks ke layar.



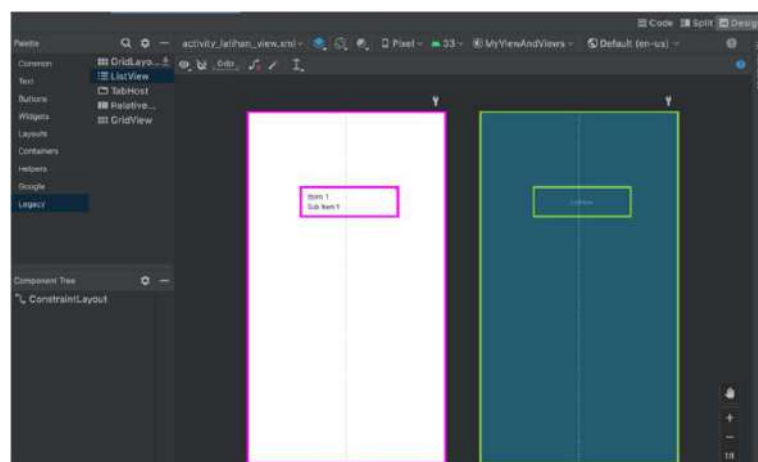
Button, komponen yang membuat pengguna dapat berinteraksi dengan cara ditekan untuk melakukan sesuatu.



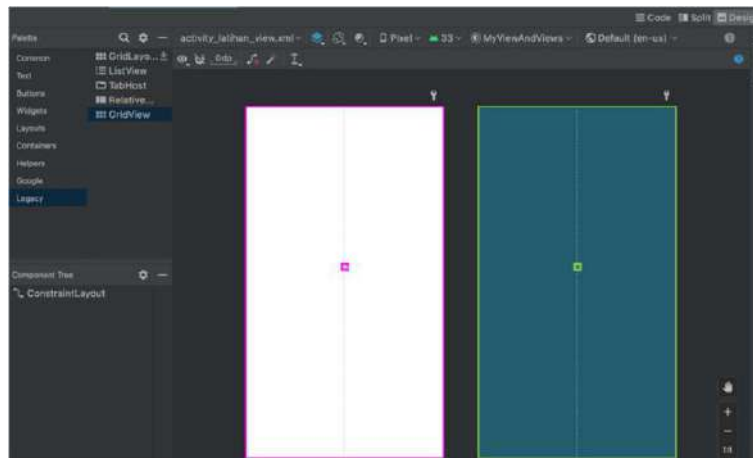
ImageView, Komponen untuk menampilkan gambar.



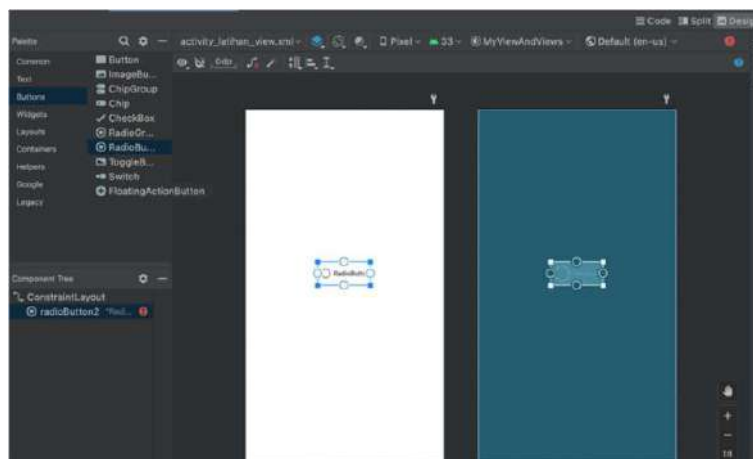
ListView, komponen untuk menampilkan informasi dalam bentuk list.



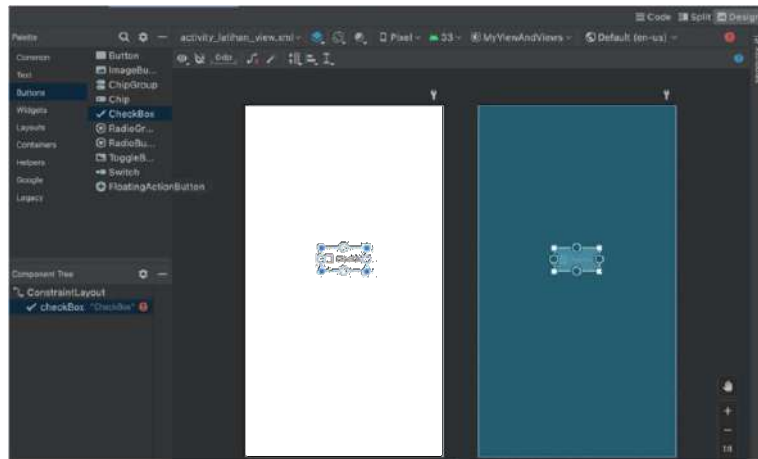
GridView, komponen untuk menampilkan informasi dalam bentuk grid.



RadioButton, komponen yang memungkinkan pengguna dapat memilih satu pilihan dari berbagai pilihan yang disediakan.



Checkbox, komponen yang memungkinkan pengguna dapat memilih lebih dari satu dari pilihan yang ada.

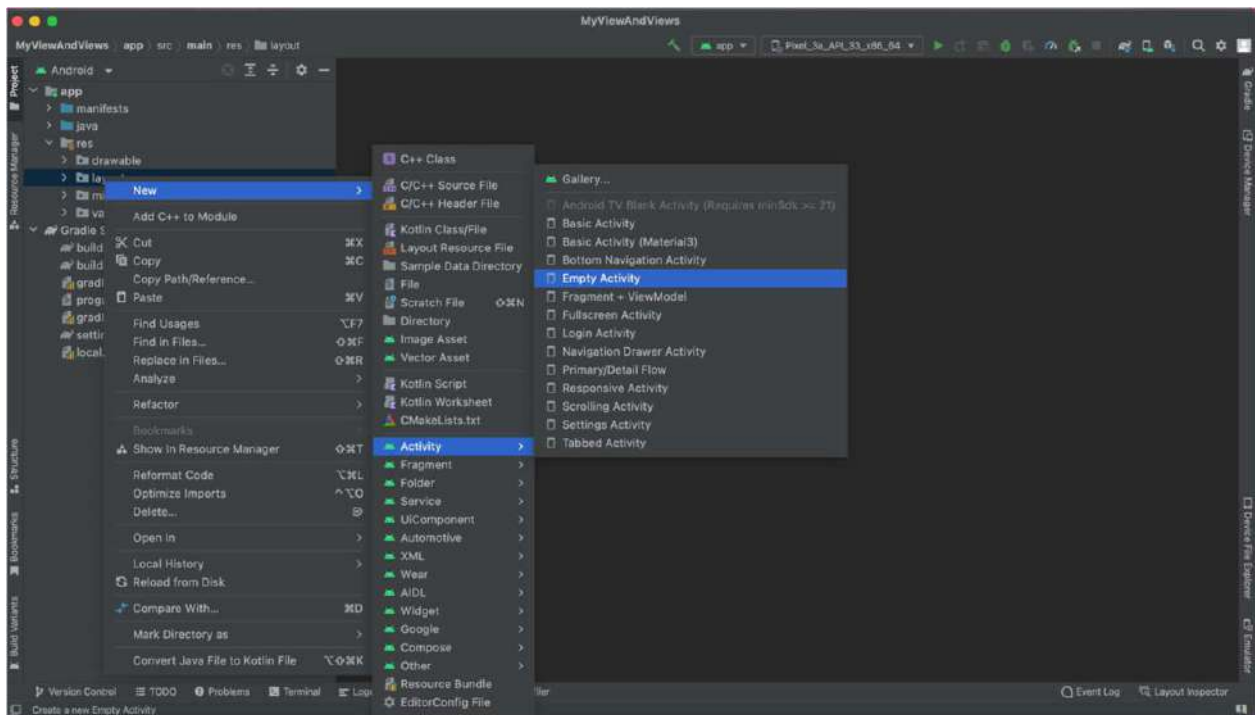


Yang dijelaskan di atas adalah contoh dan masih banyak komponen turunan dari view yang ada di dalam Android Studio.

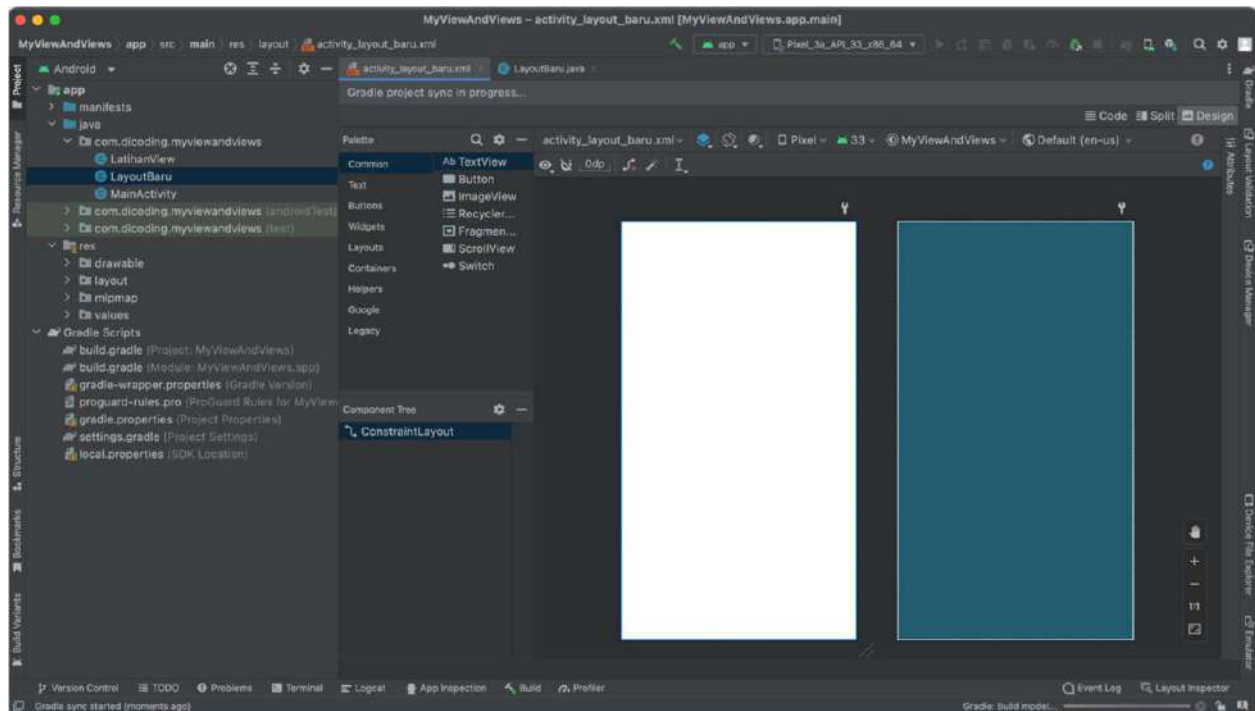
7.4. Studi Kasus

Berikut ini contoh cara membuat layout dan menggunakan komponen-komponen view di Android Studio.

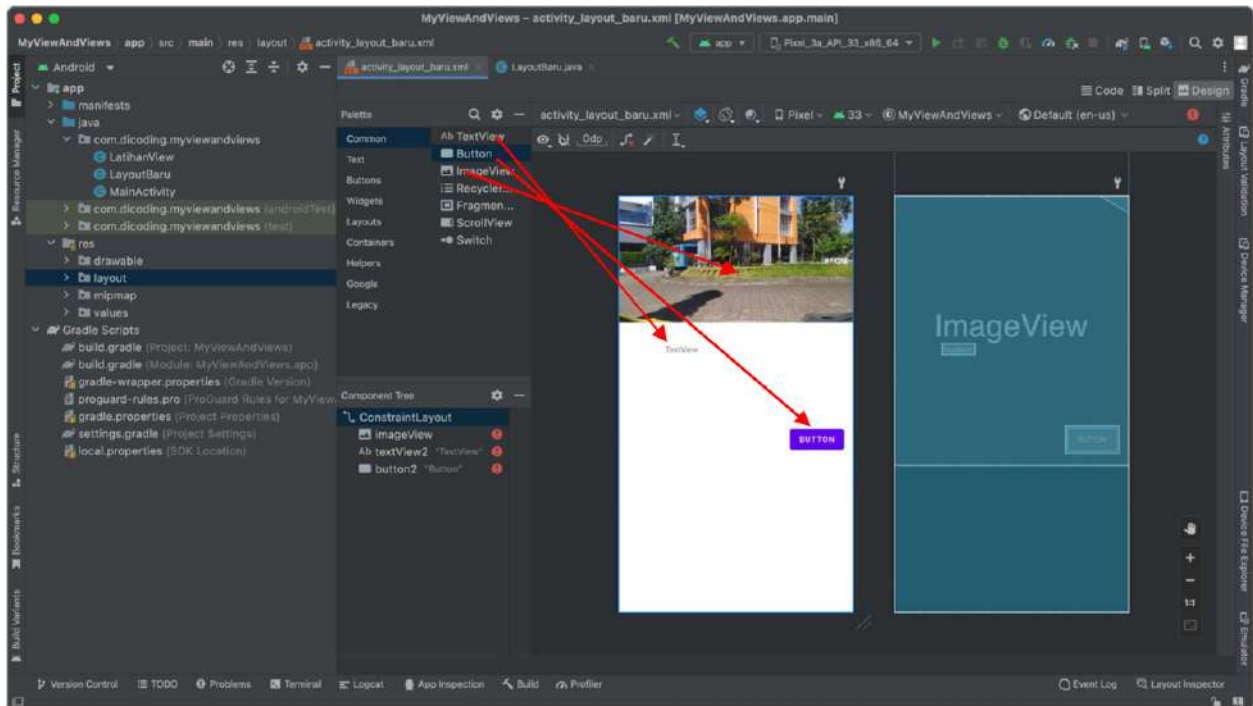
1. Membuat layout baru dengan klik kanan folder layout-> New -> Activity -> Empty Activity



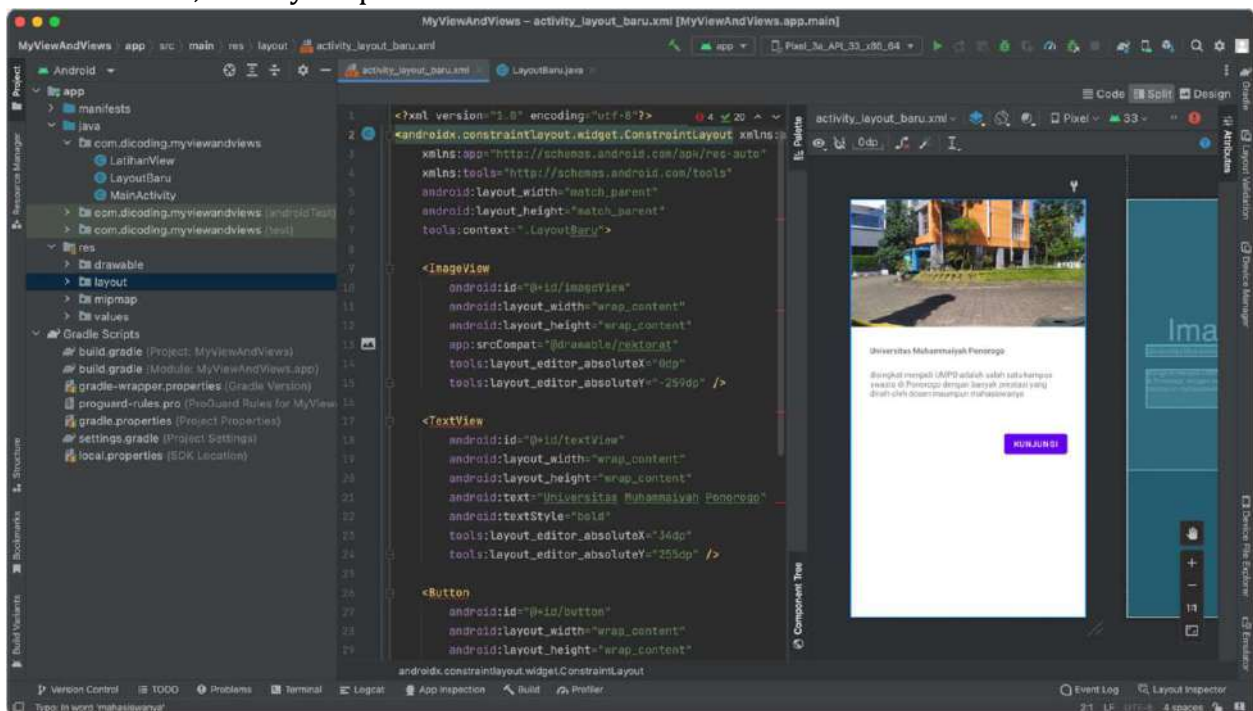
2. Setelah memberi nama layout kemudian klik -> Finish, akan masuk tampilan untuk mendesain layout seperti ini



3. Untuk menggunakan komponen-komponen view, Tarik komponen-komponen view yang akan digunakan pada bagian sebelah kiri ke halaman desain layout di android studio seperti berikut.



4. Pada contoh di atas menggunakan komponen view : ImageView, TextView dan Button, hasilnya seperti berikut



7.5. Test Akhir

Dari tutorial di atas, silahkan praktekkan dengan membuat antar muka aplikasi Android menggunakan komponen-komponen view pada Android Studio minimal 3 komponen view.

7.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

Sumber :

<https://developer.android.com/studio/>

<https://developer.android.com/reference/android/view/View>

BAB 8 – PEMBUATAN VIEWGROUP

8.1. Identitas

Kompetensi

1. Mengetahui komponen viewgroup Aplikasi Android
2. Dapat merancang antarmuka Aplikasi Android dengan viewgroup

Topik

1. Viewgroup pada aplikasi Android
2. Komponen-komponen viewgroup pada Android Studio

8.2. Test Awal

Kerjakan test awal dibawah ini!

- a. Apa yang Anda ketahui tentang viewgroup?

- b. Apa yang Anda ketahui tentang viewgroup pada Android?

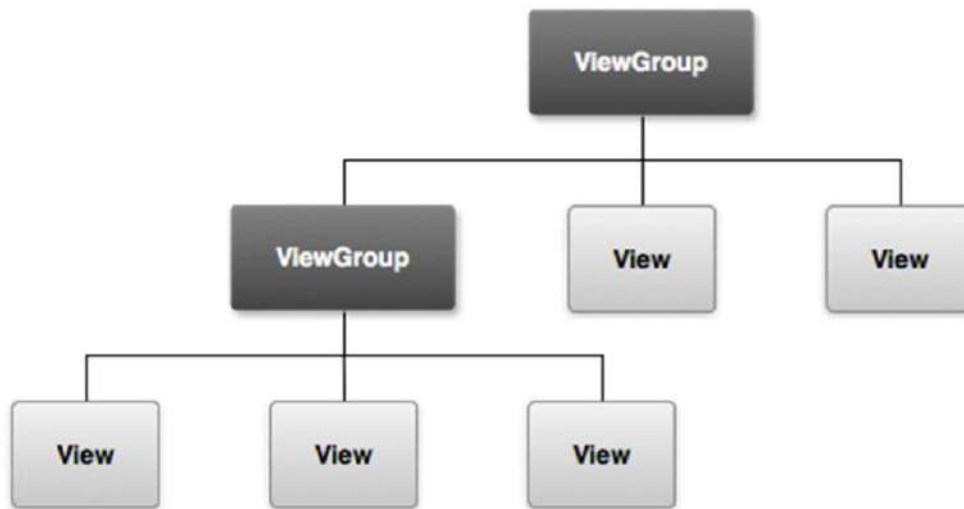
- c. Sebutkan menggunakan komponen-komponen viewgroup pada Android Studio!

8.3. Materi

Viewgroup adalah objek yang menampung objek tampilan dan grup tampilan itu sendiri untuk membentuk tampilan terpadu dari aplikasi yang lengkap. Berikut adalah contoh komponen viewgroup:

LinearLayout
FrameLayout
RelativeLayout
t TableLayout

Hierarki komponen view dan viewgroup dapat digambarkan dengan diagram berikut:



Mengonversi ke viewgroup memerlukan dua komponen view dan komponen viewgroup yang terdiri dari tiga komponen tampilan.

Berikut ini contoh tampilan layout.xml yang menunjukkan kolaborasi komponen view dan viewgroup

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    tools:context=".LatihanView">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        tools:layout_editor_absoluteX="177dp"
        tools:layout_editor_absoluteY="245dp" />
```

```

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    tools:layout_editor_absoluteX="169dp"
    tools:layout_editor_absoluteY="369dp" />
</LinearLayout>

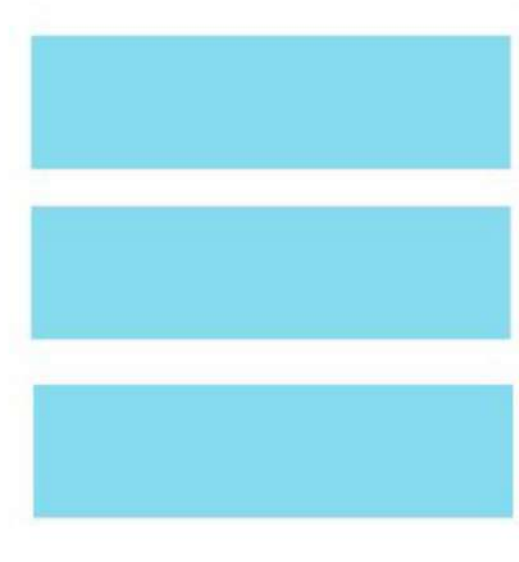
```

Turunan dari viewgroup LinearLayout menjadi wadah untuk objek anak View, Button, dan TextView. Komponen viewgroup seperti linearlayout, relativelayout, framelayout dan tablelayout adalah komponen yang paling umum digunakan sebagai induk/akar komponen tampilan.

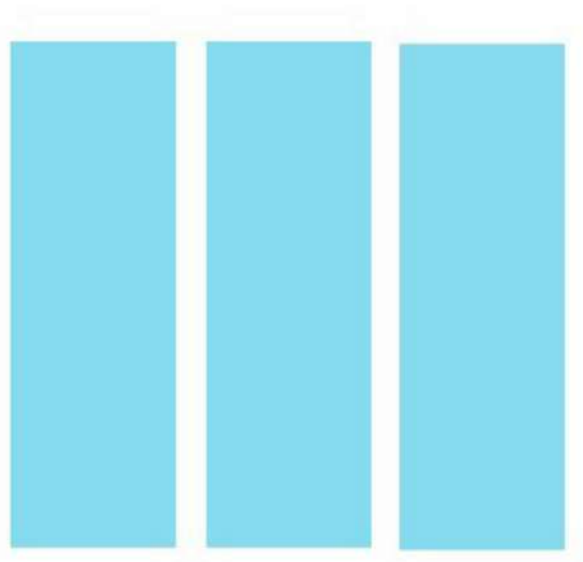
Di bawah ini adalah definisi dasar sederhana dari komponen-komponen di atas yang berkaitan dengan penempatan komponen view.

LinearLayout

Atur komponen secara horizontal atau vertikal. Tata letak linier memiliki atribut bobot untuk setiap tampilan turunan yang membantu menentukan ukuran pecahan tampilan di ruang yang tersedia.



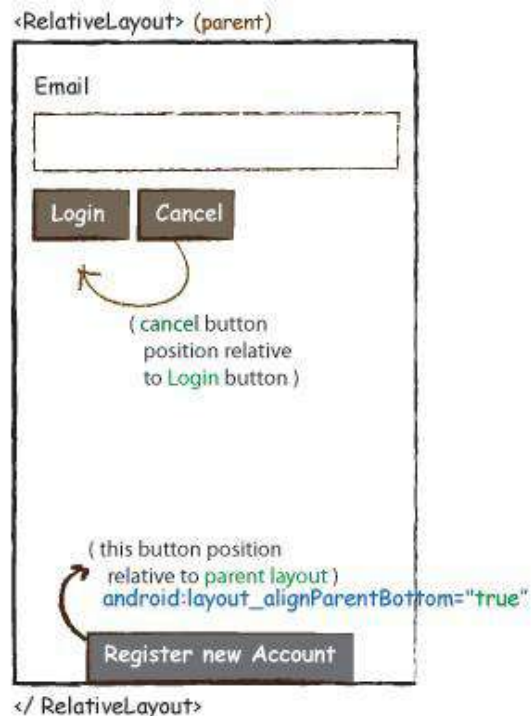
android:orientation="vertical"



android:orientation="horizontal"

RelativeLayout

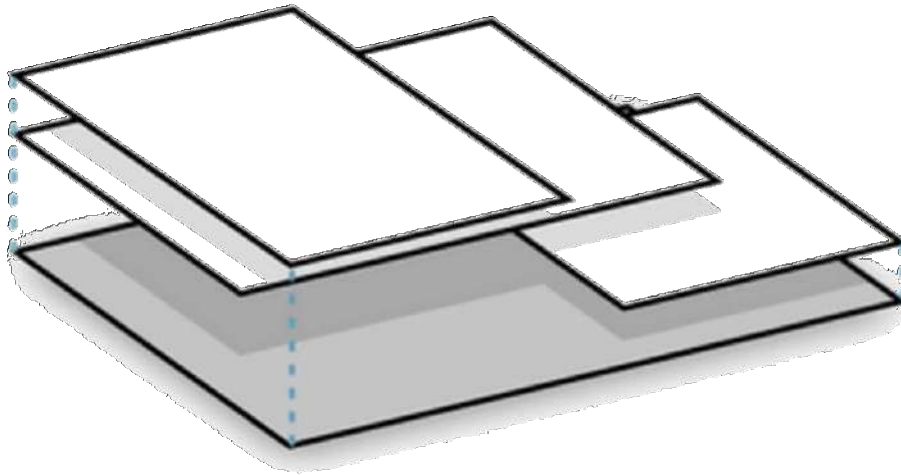
Layout yang paling fleksible dikarenakan posisi dari masing-masing komponen di dalamnya dapat mengacu secara relatif pada komponen yang lainnya dan juga dapat mengacu secara relatif ke batas layar.



FrameLayout

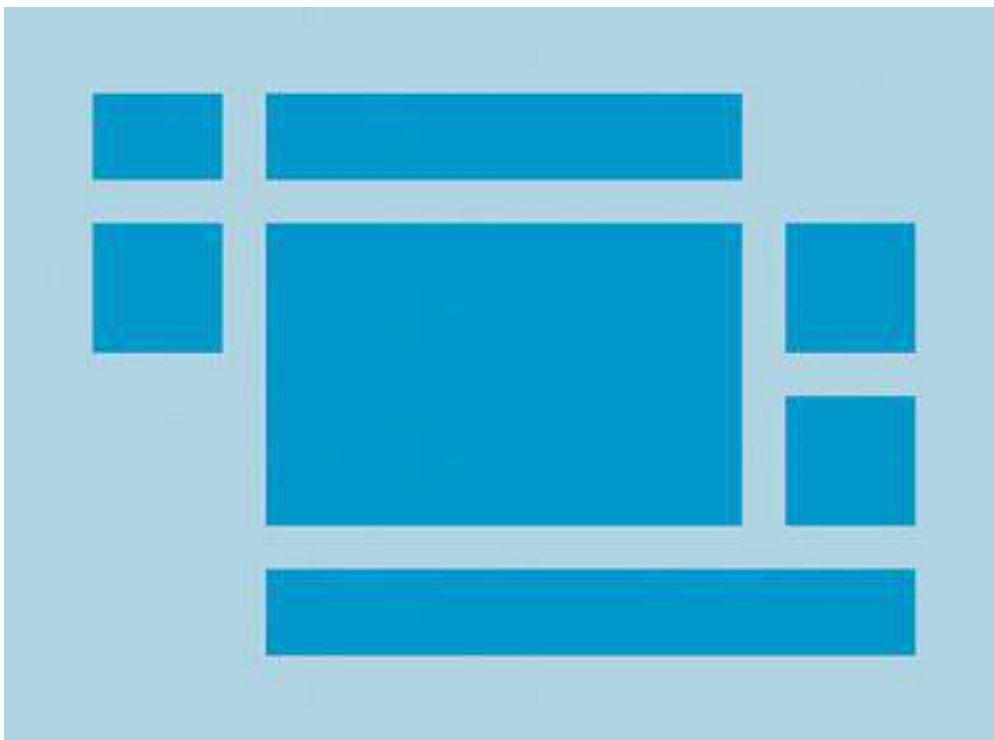
Layout ini adalah layout yang paling sederhana. Layout ini akan membuat komponen yang ada di dalamnya menjadi menumpuk atau saling menutupi satu dengan yang lainnya.

Komponen yang paling pertama pada layout ini akan menjadi alas bagi komponen-komponen di atasnya. Pada materi penggunaan fragment di materi sebelumnya, framelayout memiliki kemampuan untuk menjadi kontainer untuk fragment- fragment di dalam sebuah activity. Berikut ilustrasi penggunaan framelayout terhadap child view yang dimilikinya:



TableLayout

Susunan komponen di dalam tablelayout akan berada dalam baris dan kolom. Namun layout jenis ini tidak akan menampilkan garis pembatas untuk baris, kolom atau cell-nya.

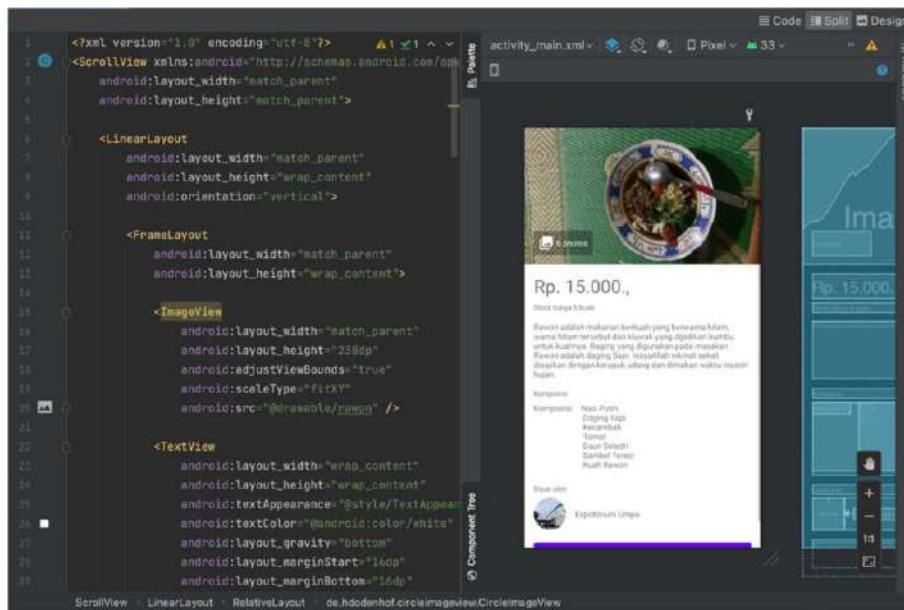


Salah satu perbedaan mendasar dari keempat jenis layout di atas adalah bagaimana komponen view di dalamnya tersusun.

Kapan saatnya menggunakan linearlayout, relativelayout, framelayout dan tablelayout? Jawabannya tergantung pada kebutuhan dan performa.

Salah satu tips performa untuk aplikasi Android adalah dengan menyusun hierarki layout dalam sebuah file layout xml secara sederhana dan flat. Artinya, kita tidak membuat layout dengan jumlah kedalaman yang tinggi. Selain itu, penting juga bagi kita untuk memahami dasar pembangunan antar muka, pengalaman pengguna ketika menggunakannya, dan selalu mencari pendekatan terbaik dalam penggunaan komponen antar muka.

Semua tergantung latihan dan seberapa sering kita mengubah mockup menjadi tampilan sebenarnya, terutama dengan memaksimalkan penulisan kode antara muka di berkas xml.



ScrollView

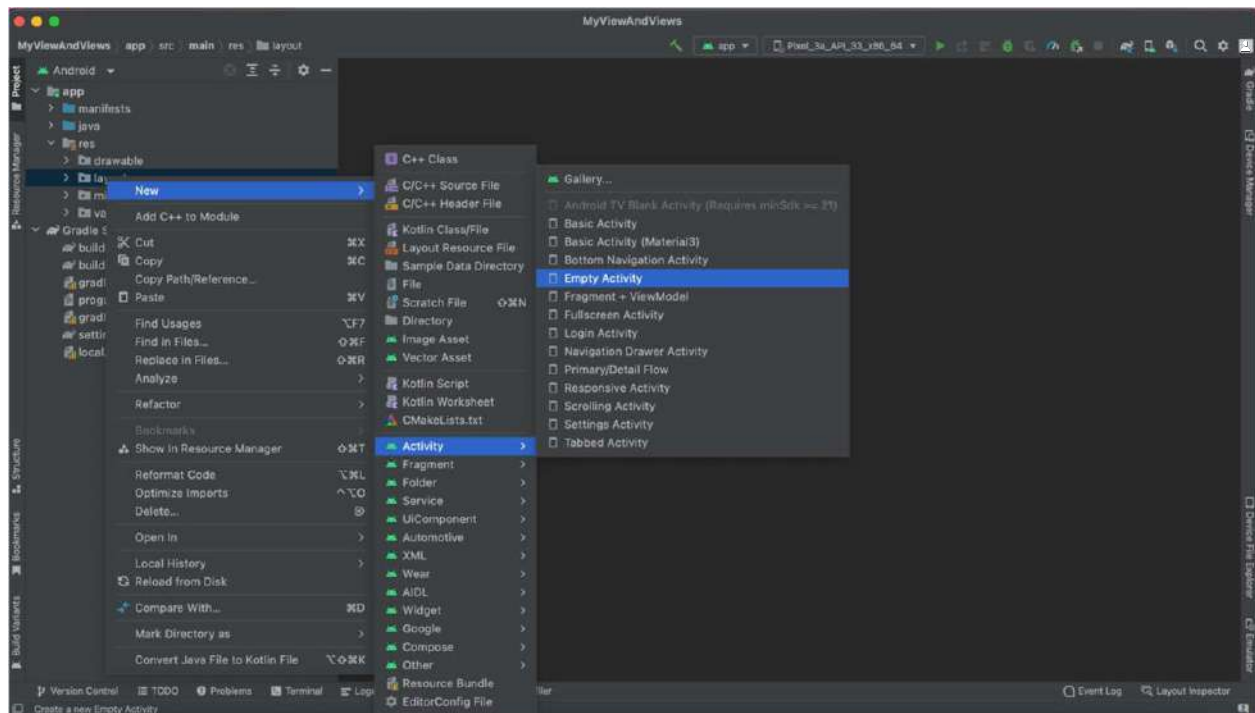
Ketika aplikasi Anda memuat informasi yang banyak dan melebihi ukuran layar, maka Anda membutuhkan tampilan yang memungkinkan pengguna untuk membaca informasi dengan lengkap. Kita membutuhkan komponen scrollview untuk mengatasi masalah ini.

Adalah layout yang memungkinkan komponen di dalamnya digeser (scroll) secara vertikal dan horizontal. Komponen di dalam scrollview hanya diperbolehkan memiliki 1 parent utama dari linearlayout, relativelayout, framelayout, atau tablelayout.

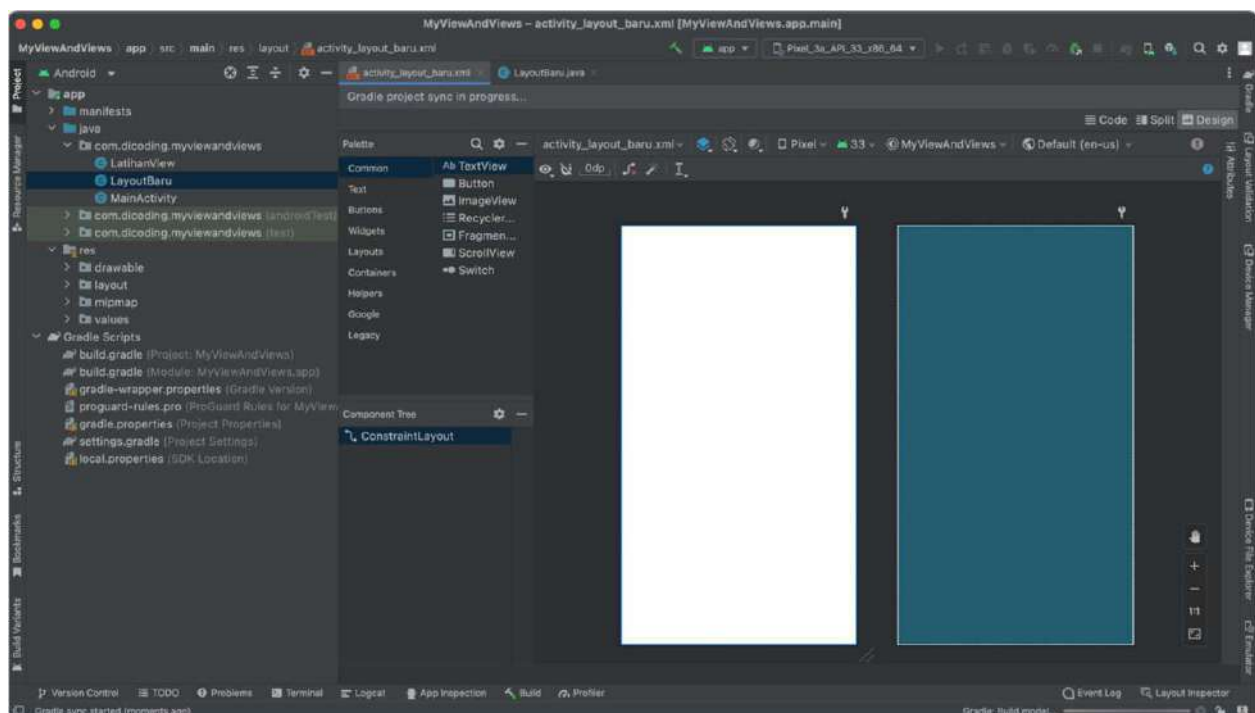
8.4. Studi Kasus

Berikut ini contoh cara membuat layout dan menggunakan komponen-komponen viewgroup di Android Studio.

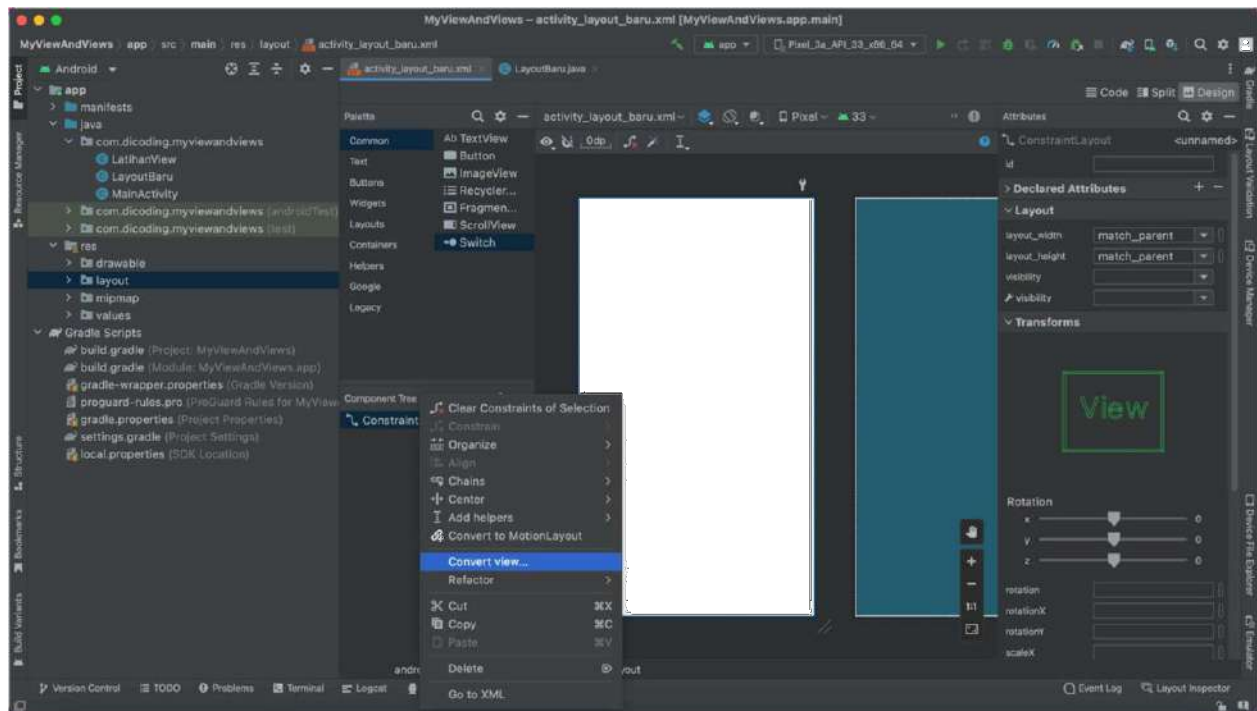
1. Membuat layout baru dengan klik kanan folder layout-> New -> Activity -> Empty Activity



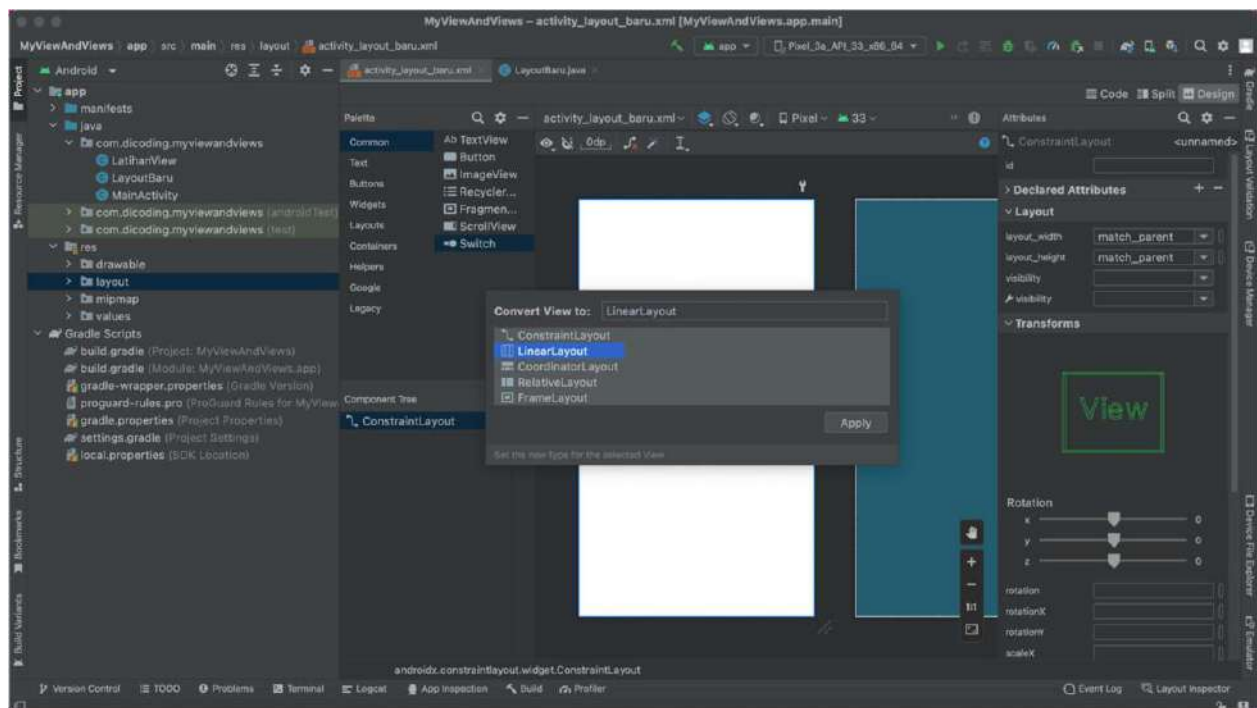
2. Setelah memberi nama layout kemudian klik -> Finish, akan masuk tampilan untuk mendesain layout seperti ini.



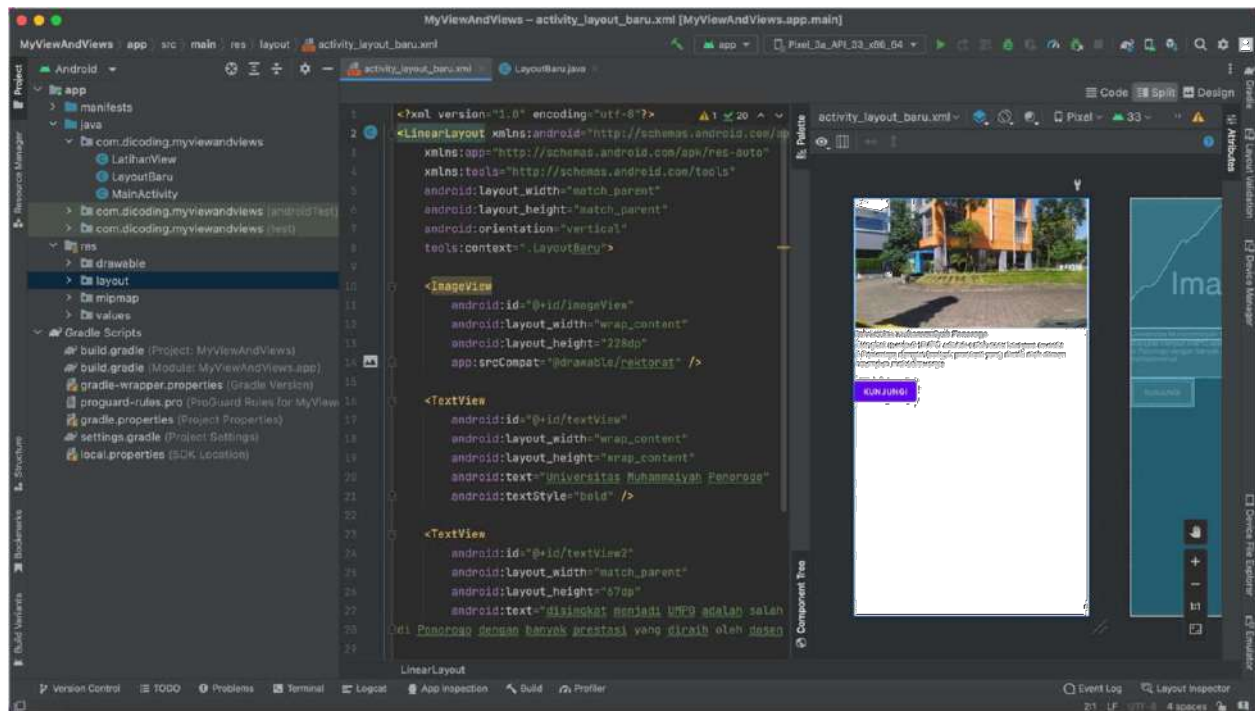
3. Layout atau komponen viewgroup yang telah dibuat menggunakan ConstraintLayout. Kita akan mengubah komponen viewgroup tersebut menjadi LinearLayout atau yang lain. Klik kanan ConstraintLayout -> klik Convert view



4. Setelah klik Convert view kita akan diberi pilihan Layout yang bisa kita pilih sebagai berikut.



5. Contohnya ini kita akan mengubah Layout menjadi LinearLayout seperti berikut hasilnya.



8.5. Test Akhir

Dari tutorial di atas, silahkan praktekkan membuat antar muka aplikasi Android dengan menggunakan komponen-komponen viewgroup minimal 3 komponen viewgroup.

8.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

Sumber :

<https://developer.android.com/studio/>

<https://developer.android.com/reference/android/view/ViewGroup>

BAB 9 – PEMBUATAN STYLE DAN THEMES

9.1. Identitas

Kompetensi

1. Mempelajari bagaimana menerapkan struktur tampilan view dalam sebuah berkas style.
2. Merancang antarmuka aplikasi Android dengan mematuhi kaidah yang ditetapkan oleh Design Guideline tim Android di Google.

Topik

1. Struktur dan tampilan view dalam sebuah berkas style
2. Antarmuka aplikasi Android

9.2. Test Awal

Kerjakan test awal dibawah ini!

- a. Apa yang Anda ketahui struktur dan tampilan view dalam Aplikasi Android?

- b. Apa yang Anda ketahui tentang berkas style aplikasi Android?

- c. Sebutkan prinsip kaidah yang ditetapkan oleh Design Guideline tim Android di Google!

9.3. Materi

Prinsip dasar antarmuka aplikasi Android

Prinsip dasar untuk mendesain antarmuka aplikasi Android harus mematuhi konvensi Pedoman Desain. Panduan ini dibuat oleh tim Android di Google. Beberapa prinsipnya adalah:

1. Tampilkan hanya informasi yang Anda butuhkan.
2. Jika aplikasi meminta izin pengguna untuk melakukan suatu tindakan, pengembang harus menyediakan mekanisme untuk mencabut izin tersebut.
3. Interupsi jika perlu
4. Gunakan teks secara ringkas. Gunakan gambar untuk menjelaskan informasi dengan lebih jelas.
5. Kami memelihara data pengguna.
6. Bantu pengguna menyelesaikan sesuatu dengan cepat.
7. Jika terlihat sama, itu harus bekerja sama.
8. Bantu pengguna membuat keputusan, tetapi serahkan keputusan kepada mereka.

Praktek terbaik

Ada beberapa praktik terbaik yang perlu dipertimbangkan saat mengembangkan aplikasi Android.

1. Rancang aplikasi Anda untuk kinerja

Aplikasi yang dirancang dengan baik harus berjalan cepat, dan jika ada proses yang memakan waktu lama, mereka berjalan secara tidak sinkron di latar belakang.

2. Desain yang bagus untuk membuat aplikasi Anda responsif

Memberikan umpan balik tentang tindakan yang dilakukan pengguna. Misalnya, ketika pengguna menekan tombol, aplikasi akan menampilkan efek tekan.

3. Desain yang memenuhi kebutuhan informasi pengguna

Aplikasi Anda harus menampilkan informasi yang dibutuhkan pengguna. Jika perlu, aplikasi harus menampilkan informasi yang terakhir diterima. Ini menyelamatkan pengguna dari keharusan menunggu aplikasi memuat data dari server.

4. Dirancang untuk mengoptimalkan penggunaan baterai

Coba jalankan aplikasi Anda dengan daya baterai rendah. Minimalkan penggunaan layanan latar belakang yang tidak perlu. Menghentikan semua pendengar saat aplikasi tidak berjalan. Jika Anda memiliki tugas yang perlu dijalankan secara teratur, gunakan Manajer Alarm dan Penjadwal Pekerjaan.

5. Desain untuk Penggunaan Koneksi Jaringan yang Efisien

Aplikasi yang baik adalah aplikasi yang menggunakan koneksi Anda ke Internet secara efisien. Urutkan tugas yang harus dilakukan saat perangkat pengguna terhubung ke Wi-Fi (jaringan flat rate) atau jaringan lain. Memiliki koneksi jaringan yang baik adalah wajib bagi pengguna untuk menggunakan aplikasi Anda.

Nilai titik tampilan yang baik mendukung kualitas fungsionalitas dari aplikasi yang dibuat. Selama aplikasi dibutuhkan dan memenuhi poin-poin di atas, pengguna akan tetap dan terus menggunakan aplikasi tersebut. Kembali ke Styles and Themes Jika Anda pernah mengembangkan aplikasi berbasis web, Anda pasti sudah tidak asing lagi dengan CSS (Cascading Style Sheets). Mengontrol tampilan halaman web. Pendekatan serupa berlaku untuk Android. Itu disebut Style/ gaya.

Gaya/Style adalah seperangkat properti yang diperlukan untuk menentukan bagaimana tampilan komponen dan jendela layar (aktivitas atau fragmen) ditampilkan. Contoh properti ini adalah tinggi, lebar, dan warna latar belakang. Gaya didefinisikan dalam file XML terpisah. Anda dapat menemukannya di `res -> Values -> Styles.xml`.

9.4. Studi Kasus

Misalnya, Anda memiliki tampilan teks dengan berbagai atribut seperti contoh kode berikut. Tampilan teks ini berguna untuk menampilkan isi informasi detail yang terdapat di seluruh aplikasi.

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/belajar"
    android:textColor="#00FF00"
    android:typeface="serif"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

Tidak terlalu efisien saat menyalin dan menempel dari satu tata letak XML ke tata letak lainnya. Kita dapat menyederhanakan ini menjadi:

```

<TextView
    style="@style/Styleku"
    android:text="@string/belajar">
</TextView>

```

Anda dapat memindahkan atribut `layout_width`, `layout_height`, `textcolor`, dan `fontface` ke dalam gaya/style tampilan teks Anda sendiri dan menggunakannya kembali untuk semua objek tampilan teks yang serupa.

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="Styleku" parent="@android:style/TextAppearance.Medium">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:textColor">#00FF00</item>
        <item name="android:typeface">serif</item>
    </style>
</resources>

```

Berikut adalah beberapa aturan yang perlu diingat saat menggunakan style/gaya:

1. Semua style/gaya yang dibuat harus berada di dalam tag sumber daya.
2. Semua style/gaya yang Anda tentukan harus dimuat dalam tag gaya.

```
<style name="Styleku" parent="@android:style/TextAppearance.Medium">
```

Nama : Nama gaya yang Anda buat. Anda dapat memodifikasi gaya yang diwariskan atau menambahkan atribut ke gaya yang baru dibuat. Android sudah dibangun dari SDK atau platform. Parent : Nilai gaya yang diwarisi gaya yang ada (termasuk atribut yang dikandungnya) biasanya merupakan nilai default yang memberikan gaya berbeda yang dapat digunakan untuk tampilan berbeda.

3. Semua atribut yang didefinisikan dalam style/gaya harus disertakan dalam tag item.

```
<item name="android:layout_width">match_parent</item>
```

Nama keluarga : Nama atribut yang akan ditentukan. match_parent: nilai atribut. Dalam satu kasus, kami ingin membuat instance dari gaya yang kami buat. Misalnya, jika Anda ingin gaya Styleku menjadi merah, Anda dapat melakukan:

```
<style name="Styleku.Red">
    <item name="android:textColor">#FF0000</item>
</style>
```

atau berwarna merah dan juga dengan ukuran yang besar menjadi seperti ini :

```
<style name="Styleku.Red">
    <item name="android:textColor">#FF0000</item>
    <item name="android:textSize">30sp</item>
</style>
```

Anda telah belajar cara membuat dan menerapkan style/gaya. Pelajari materi berikutnya. Tema/theme sendiri adalah style/gaya yang secara khusus diterapkan pada aktivitas dan aplikasi di file AndroidManifest.xml. Dalam proyek kita sebelumnya, kita mendefinisikan:

```
android:theme="@style/Theme"
```

Theme pada **styles.xml** berisi :

```

<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Theme" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
    <!-- Primary brand color. -->
    <item name="colorPrimary">@color/purple_500</item>
    <item name="colorPrimaryVariant">@color/purple_700</item>
    <item name="colorOnPrimary">@color/white</item>
    <!-- Secondary brand color. -->
    <item name="colorSecondary">@color/teal_200</item>
    <item name="colorSecondaryVariant">@color/teal_700</item>
    <item name="colorOnSecondary">@color/black</item>
    <!-- Status bar color. -->
    <item name="android:statusBarColor">?attr/colorPrimaryVariant</item>
    <!-- Customize your theme here. -->
  </style>
</resources>

```

Gaya yang mewarisi tema MaterialComponents varian Light dan memiliki DarkActionBar.

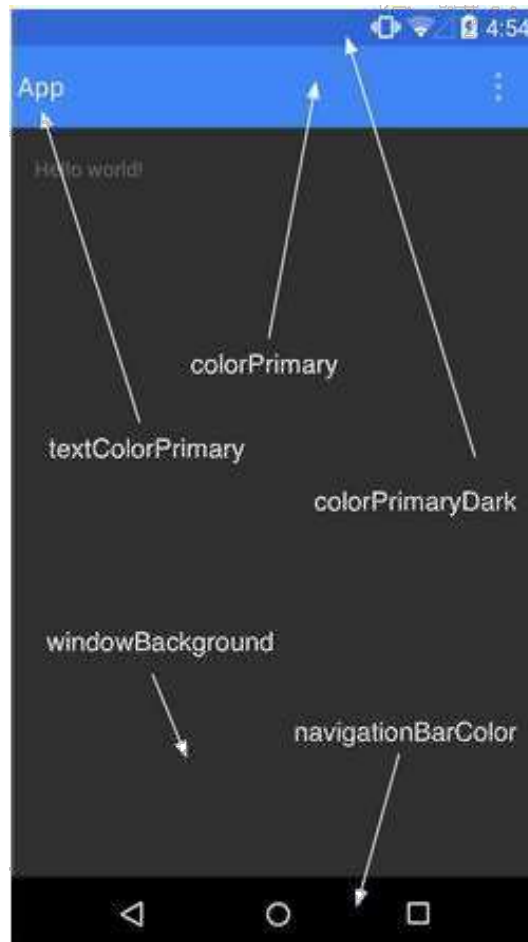
Semua nilai untuk atribut terkandung dalam file colors.xml.

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="purple_200">#FFBB86FC</color>
  <color name="purple_500">#FF6200EE</color>
  <color name="purple_700">#FF3700B3</color>
  <color name="teal_200">#FF03DAC5</color>
  <color name="teal_700">#FF018786</color>
  <color name="black">#FF000000</color>
  <color name="white">#FFFFFFFF</color>
</resources>

```

Pengembangan aplikasi Android sekarang mengacu pada penerapan prinsip-prinsip Desain Material untuk merancang antarmuka pengguna dan pengalaman pengguna. Untuk informasi lebih lanjut tentang Desain Material, silakan lihat tautan berikut:



9.5. Test Akhir

Dari tutorial di atas, silahkan praktekkan dengan membuat style baru dengan nama StyleNamakalian.

9.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

Sumber :

<https://developer.android.com/studio/>

<https://developer.android.com/develop/ui/views/theming/themes?hl=en>

BAB 10 – MEMBUAT BUTTON SOUND EFFECT DI ANDROID

10.1 IDENTITAS

Kompetensi

1. Memberikan pemahaman tentang membuat efek suara di Android
2. Dapat membuat efek suara di Android

Topik

1. Media Player di Android Studio
2. Studi Kasus Membuat button sound effect

10.2 Test Awal

Kerjakan test awal dibawah ini!

- a. Menurut Anda, apakah aplikasi yang dibangun menggunakan Android Studio dapat diberikan efek suara?

- b. Jika jawaban pada soal a adalah bisa, komponen apa yang dibutuhkan untuk dapat memutar audio/video di Android studio?

- c. Audio dalam format apa saja yang dapat digunakan?

10.3 Materi

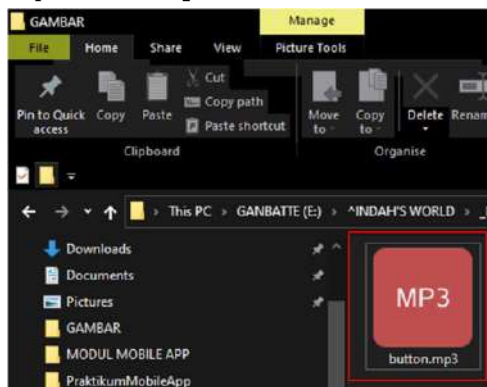
Dalam pemrograman Android terdapat Framework multimedia yang mendukung pemutaran berbagai macam media sehingga audio, video, dan gambar dapat dengan mudah terintegrasi ke dalam aplikasi yang kita bangun. Audio atau video dari file media disimpan dalam resource aplikasi dapat dengan mudah kita putar menggunakan MediaPlayer API. Salah satu komponen yang digunakan adalah class MediaPlayer. Objek class ini dapat mengambil, mendekode, serta memutar audio dan video dengan sedikit penyiapan. Class ini mendukung beberapa sumber media yang berbeda, seperti resource local, URL internal, dan URL eksternal (streaming).

Dalam praktikum kali ini kita akan mencoba memutar audio yang resource mentah lokal (dengan cara menyimpan resource ke direktori res/raw/). Resource "mentah" adalah file yang tidak akan diuraikan oleh sistem dengan cara khusus apa pun. Tetapi, konten resource ini tidak boleh berupa audio mentah, namun harus berupa file media yang termasuk ke dalam salah satu jenis format yang didukung dimana file tersebut telah diencode dan diformat dengan benar. Salah satu jenis format yang didukung adalah .mp3.

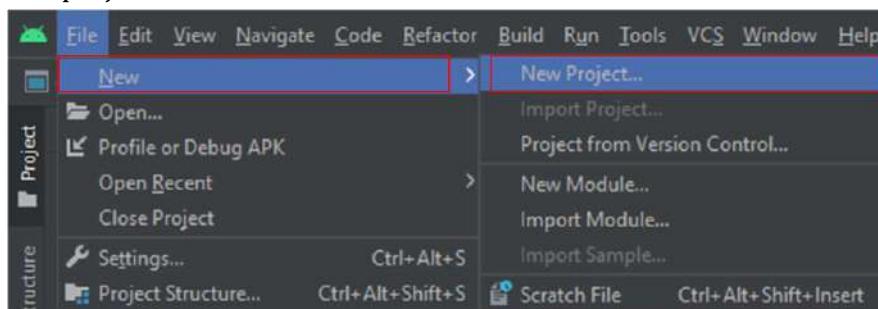
10.4 Studi Kasus

Pada studi kasus kali ini kita akan coba membuat button sound effect pada aplikasi. Untuk langkah-langkahnya silahkan diikuti tahapan-tahapan di bawah ini.

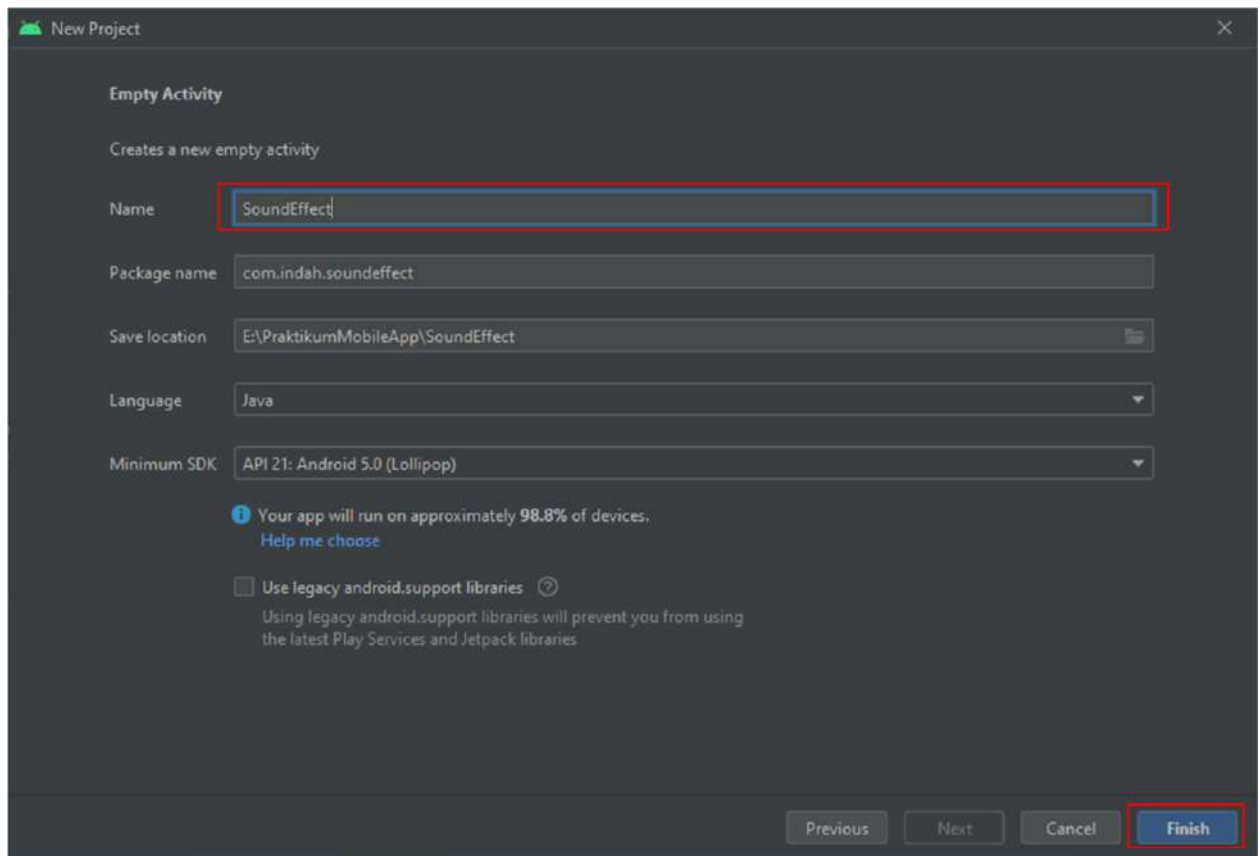
1. Siapkan file **.mp3**.



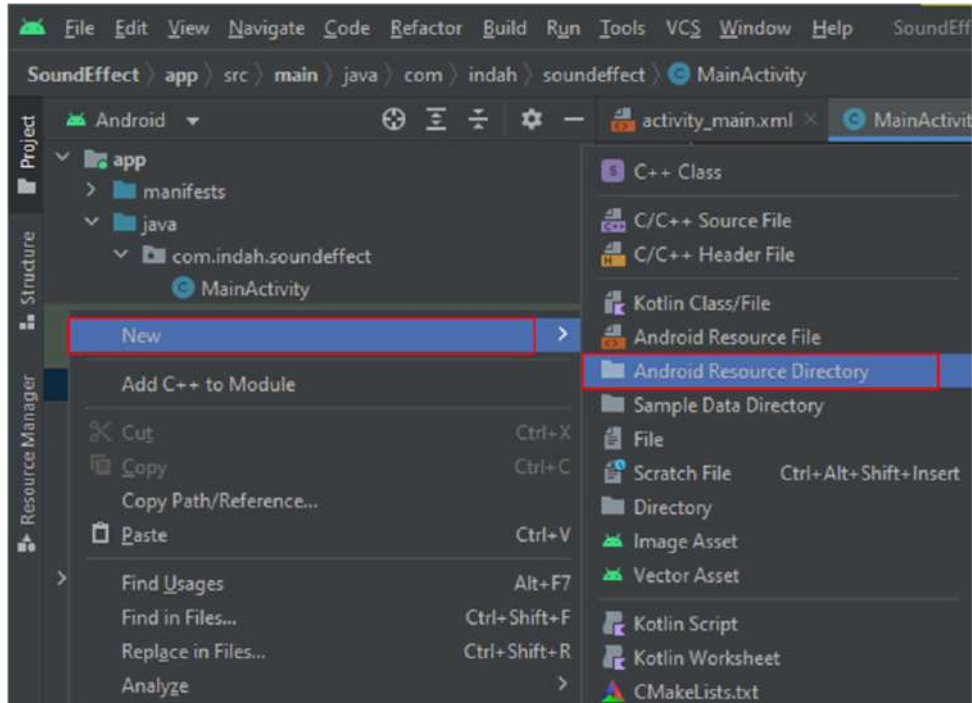
2. Buat project baru di Android Studio

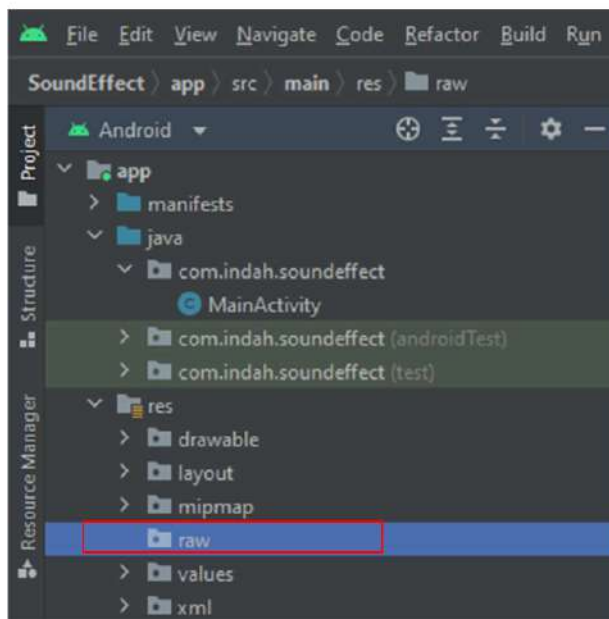
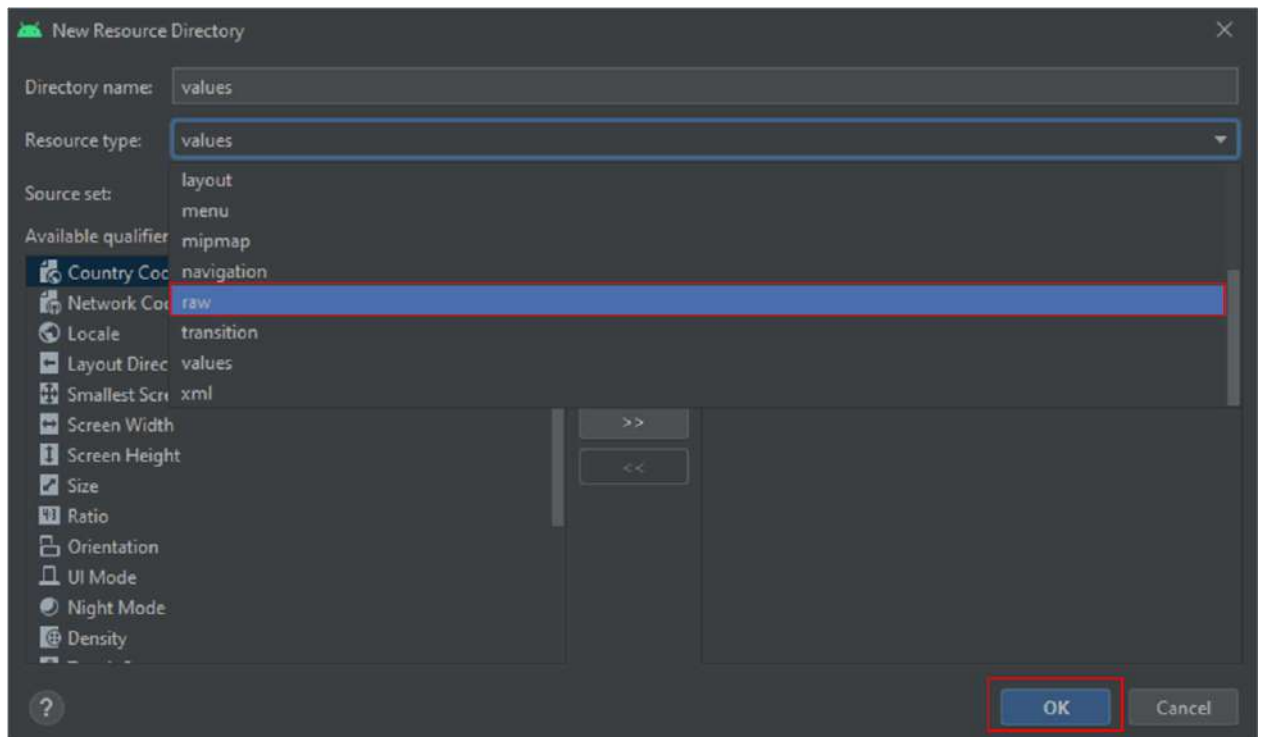


3. Beri nama project

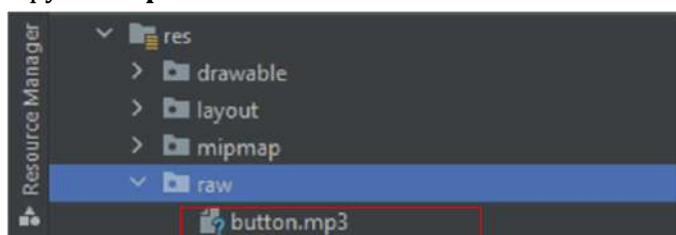


4. Buat folder **raw** di direktori **res**





5. Copy file **.mp3** ke folder **raw**



6. Buka berkas **activity_main.mp3** dan buat sintaks seperti pada gambar di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/sound"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="@string/cek_sound"
        android:textColor="@color/white"
        android:textSize="25dp" />

</RelativeLayout>
```

7. Maka tampilan layout akan berubah menjadi seperti ini



8. Selanjutnya, buka berkas **MainActivity.java**, dan modifikasi sintaks menjadi seperti pada gambar di bawah ini

```

package com.indah.soundeffect;

import androidx.appcompat.app.AppCompatActivity;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    private Button sound;

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

        sound = findViewById(R.id.sound);
        final MediaPlayer = MediaPlayer.create(this, R.raw.button);

        sound.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mediaPlayer.start();
            }
        });
    }
}

```

9. Jalankan aplikasi dengan klik **Run'app'**. Maka hasil output akan menjadi seperti pada gambar di bawah ini. Dan ketika di klik button **CEK SOUND** akan muncul suara.



10.5 Test Akhir

Dari contoh studi kasus di 10.4, silahkan membuat aplikasi yang menampilkan nama-nama binatang beserta gambar dan suara. Gunakan button untuk dapat memberikan efek suara.

10.6. Tugas

1. Kerjakan percobaan pada praktikum
2. Kerjakan soal-soal latihan
3. Berikan kesimpulan dari praktikum ini.
4. Buatlah laporan resmi dari hasil praktikum ini.
5. Laporan dikumpulkan ke instruktur masing-masing sebelum melaksanakan praktikum selanjutnya.

Sumber :

<https://developer.android.com/reference/android/media/MediaPlayer>

BAB 11 – STUDI KASUS - 1

12.1. IDENTITAS

Kompetensi

1. Memantapkan pemahaman praktikan tentang Activity
2. Memantapkan pemahaman praktikan tentang Intent
3. Memantapkan pemahaman praktikan tentang Button Sound Effect

Topik

1. Studi Kasus pada Bab 11

12.2. Studi Kasus

Membuat aplikasi sederhana berbasis mobile yang berisi media pembelajaran untuk anak-anak mengenal kosa kata dalam bahasa inggris, dimana aplikasi menggabungkan teks, gambar, audio ataupun video (opsional).

12.3. Test Akhir

1. Berdasarkan petunjuk pada poin 12.2. rancanglah aplikasi sederhana berbasis mobile untuk media pembelajaran kosa kata dalam bahasa inggris dengan menggunakan **Activity, Intent, dan Button Sound Effect**
2. Menampilkan kosa kata bahasa inggris dengan jumlah **minimal 10 kosa kata yang berbeda**
3. Ketika salah satu **kosa kata diklik** maka akan menampilkan **halaman baru** yang berisi terjemahan dari kosa kata, gambar serta cara pelafalan dan arti dari kosa kata dalam bentuk audio.
4. Tambahkan **1 fitur** dalam aplikasi yang dibangun **berisi identitas diri mahasiswa (nama, NIM, foto)**
5. Dikumpulkan dalam bentuk **apk** yang siap untuk diinstall di perangkat mobile dan **Folder Project aplikasi**

BAB 12 – STUDI KASUS - 2

12. 1. IDENTITAS

Kompetensi

1. Memantapkan pemahaman praktikan tentang View
2. Memantapkan pemahaman praktikan tentang ViewGroup
3. Memantapkan pemahaman praktikan tentang Style dan Theme

Topik

1. Studi Kasus pada Bab 11

12.2. Studi Kasus

Membuat tampilan halaman media pembelajaran pada aplikasi android dari studi kasus bab 11 dengan menerapkan **view, viewgroup, style** dan **theme**.

12.3. Test Akhir

1. Berdasarkan petunjuk pada poin 12.2. rancanglah tampilan halaman media pembelajaran pada aplikasi android dari studi kasus Bab 11 dengan menerapkan **view, viewgroup, style** dan **theme**.
2. Membuat halaman yang **berisi identitas diri mahasiswa (nama, NIM, foto)**
3. Dikumpulkan dalam bentuk **apk** yang siap untuk diinstall di perangkat mobile dan **Folder Project aplikasi**

BIBLIOGRAPHY

<https://developer.android.com/>

<https://developer.android.com/studio/>

<https://www.oracle.com/java/technologies/javase-downloads.html>

<https://developer.android.com/about/versions/>

[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

<https://developer.android.com/studio/run/device>

<https://developer.android.com/studio/run/emulator>

<https://developer.android.com/reference/android/content/Intent?hl=en>

<https://developer.android.com/reference/android/view/View>

<https://developer.android.com/reference/android/view/ViewGroup>

<https://developer.android.com/develop/ui/views/theming/themes?hl=en>

<https://developer.android.com/reference/android/media/MediaPlayer>

Murya, Y. 2014. Pemrograman Android Black Box. Jasakom Publisher.



LABORATORIUM TEKNIK INFORMATIKA
FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH PONOROGO