

# Modul 0 (Introduction)

## Pendahuluan

### Apa itu Android?

Android adalah sistem operasi yang dikeluarkan oleh Google. Android dibuat khusus untuk menjadi sistem operasi smartphone dan tablet. Android juga mempunyai store dengan lebih dari 2 miliar pengguna aktif per bulannya, per Januari 2018.



### Mengapa menggunakan Android?

Android memiliki beberapa kondisi seperti :

#### 1. Sistem Operasi Smartphone Terpopuler

- Pada tahun 2013, Android menjadi *operation system* (OS) terlaris pada *tablet* dan *smartphone*. Kini *market share* Android sedikitnya 80 % dari total penjualan *smartphone* di tingkat global (statista.com). Tercatat pada tahun 2016 Android store memiliki lebih dari 2.8 juta aplikasi.
- Android menarik bagi perusahaan teknologi yang membutuhkan barang siap jadi, biaya rendah dan kustomisasi OS untuk perangkat teknologi tinggi mereka. Hal ini menjadi daya tarik bagi banyak perusahaan, sehingga mereka memilih Android.

- *Source code* dari Android bersifat *open source*. Ini adalah hal menarik bagi komunitas developer, karena lisensi *open source* sangat mendukung untuk mengembangkan produknya dengan aman.

## 2. Store

- Aplikasi Android bisa didistribusikan menggunakan web, copy APK, dan store. Android store, yaitu Google Play, merupakan cara termudah bagi para developer untuk mendistribusikan aplikasinya ke pasar dengan miliaran pengguna.



- Strategi monetisasi aplikasi yang ditawarkan Google Play ada bermacam-macam. Dimulai dari app berbayar (*paid distribution*), pembelian dalam aplikasi (*in-app purchase*), langganan (*subscriptions*), dan iklan (*ads*). Tentunya developer harus mengikuti aturan yang ada untuk memastikan bahwa pengguna mendapatkan pengalaman (*user experience*) terbaik.

## 3. Development Kit untuk Developer

- Android Software Development Kit (SDK) merupakan *kit* yang bisa digunakan oleh para *developer* untuk mengembangkan aplikasi berbasis Android. Di dalamnya, terdapat beberapa *tools* seperti *debugger*, *software libraries*, *emulator*, dokumentasi, *sample code* dan tutorial.
- Bahasa pemrograman yang sering digunakan untuk mengembangkan aplikasi Android adalah Java. Namun ada beberapa bahasa lainnya yang dapat digunakan, seperti C++ dan Go. Pada IO 2017 Google juga menetapkan Kotlin sebagai tambahan bahasa resmi.

## IDE android

- Pada 2014 Google mengeluarkan IDE yang bernama Android Studio yang berbasis IntelliJ IDEA.






- Dengan menggunakan Android Studio, para developer dapat membuat aplikasi dari nol hingga dipublikasikan ke dalam store. Android Studio juga mempunyai beberapa fitur built-in yang sangat membantu para developer untuk memaksimalkan proses pembuatan aplikasi. Fitur-fitur ini misalnya Gradle, Code Completion, dan beragam integrasi dengan layanan dari Google, seperti Firebase.
- Android Studio sebagai IDE (Integrated Development Environment). Android Studio dirilis 16 Mei 2013 saat Google IO berlangsung. Android Studio berbasis JetBrains IntelliJ IDEA, dan dikhususkan untuk pengembangan software berbasis Android.
- Android Studio menawarkan fitur lebih demi meningkatkan produktifitas Anda saat membuat aplikasi Android, misalnya:
  - Sistem versi berbasis gradle yang fleksibel.
  - Emulator yang cepat dan kaya fitur.
  - Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
  - Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
  - Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
  - Alat pengujian dan kerangka kerja yang ekstensif.

- Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
- Dukungan C++ dan NDK.
- Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine.
- Persyaratan Sistem



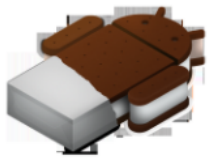



Windows	Mac
<ul style="list-style-type: none"> <li>• Microsoft® Windows® 7/8/10 (64-bit)</li> <li>• RAM minimum 4 GB, RAM yang disarankan 8 GB; tambah 1 GB untuk Android Emulator</li> <li>• Ruang disk minimum yang tersedia 2 GB, Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)</li> <li>• Resolusi layar minimum 1280 x 800</li> </ul>	<ul style="list-style-type: none"> <li>• Mac® OS X® 10.10 (Yosemite) atau lebih baru, hingga 10.14 (macOS Mojave)</li> <li>• RAM minimum 4 GB, RAM yang disarankan 8 GB; tambah 1 GB untuk Android Emulator</li> <li>• Ruang disk minimum yang tersedia 2 GB, Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)</li> <li>• Resolusi layar minimum 1280 x 800</li> </ul>

Linux	Chrome OS
<ul style="list-style-type: none"> <li>• Desktop GNOME atau KDE <i>Telah diuji pada gLinux yang berbasis Debian</i></li> <li>• Distribusi 64-bit yang mampu menjalankan aplikasi 32-bit</li> <li>• GNU C Library (glibc) 2.19 atau lebih baru</li> <li>• RAM minimum 4 GB, RAM yang disarankan 8 GB; tambah 1 GB untuk Android Emulator</li> <li>• Ruang disk minimum yang tersedia 2 GB, Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)</li> <li>• Resolusi layar minimum 1280 x 800</li> </ul>	<ul style="list-style-type: none"> <li>• 8 GB RAM atau lebih yang disarankan</li> <li>• Ruang disk minimum yang tersedia 4 GB</li> <li>• Resolusi layar minimum 1280 x 800</li> <li>• Disarankan Intel i5 atau lebih (U series atau lebih)</li> </ul> <p>Perangkat yang direkomendasikan:</p> <ul style="list-style-type: none"> <li>• <b>Acer:</b> Chromebook 13/Spin 13, Chromebox CX13</li> <li>• <b>Lenovo:</b> Yoga C630 Chromebook</li> <li>• <b>HP:</b> Chromebook x360 14, Chromebox G2</li> <li>• <b>Dell:</b> Inspiron Chromebook 14</li> <li>• <b>ASUS:</b> Chromebox 3</li> <li>• <b>ViewSonic:</b> NMP660 Chromebox</li> <li>• <b>CTL:</b> Chromebox CBx1</li> </ul>

# Sejarah Perkembangan OS Android

Version	Code name	Release date	API level	DVM/ART	New features	Icon
11	11	September 8, 2020	30	ART	<ul style="list-style-type: none"> <li>• Chat Bubbles</li> <li>• Screen Recorder</li> <li>• Device Control</li> <li>• Predictive Tool</li> <li>• One-time permission</li> </ul>	
10	10	September 3, 2019	29	ART	<ul style="list-style-type: none"> <li>• Live Caption</li> <li>• Smart Reply</li> <li>• Sound Amplifier</li> <li>• Dark Theme</li> <li>• Privacy &amp; Security</li> <li>• Digital Wellbeing</li> </ul>	
9	Pie	August 6, 2018	28	ART	<ul style="list-style-type: none"> <li>• Adaptive Battery</li> <li>• Adaptive Brightness</li> </ul>	

<b>8.0-8.1</b>	Oreo	October 25, 2017	26 - 27	ART	<ul style="list-style-type: none"> <li>• Picture-in-Picture</li> </ul>	
<b>7.1 - 7.1.2</b>	Nougat	August 22, 2016	24 - 25	ART	<ul style="list-style-type: none"> <li>• Multi window</li> <li>• GIF Keyboard</li> </ul>	
<b>6.0 - 6.0.1</b>	Marshmallow	October 5, 2015	23	ART	<ul style="list-style-type: none"> <li>• Now On Tap</li> <li>• Permissions</li> <li>• Battery (Doze &amp; App Standby)</li> </ul>	
<b>5.1 - 5.1.1</b>	Lollipop	November 12, 2014	21 - 22	ART	<ul style="list-style-type: none"> <li>• Material Design</li> <li>• Multiscreen</li> <li>• Notifications</li> </ul>	

4.4 - 4.4.4	KitKat	October 31, 2013	19 - 20	DVM (and ART 1.6.0)	<ul style="list-style-type: none"> <li>• Voice : Ok Google</li> <li>• Immersive Design</li> <li>• Smart Dialer</li> </ul>	
4.1 - 4.3.1	Jelly Bean	July 9, 2012	18	DVM	<ul style="list-style-type: none"> <li>• Google Now</li> <li>• Actionable Notifications</li> <li>• Account Switching</li> </ul>	
4.0 - 4.6	Ice Cream Sandwich	October 19, 2011	15	DVM	<ul style="list-style-type: none"> <li>• Custom Home Screen</li> <li>• Data Usage Control</li> <li>• Android Beam</li> </ul>	
3.0 - 3.2.6	HoneyComb	February 22, 2011	11 - 13	DVM	<ul style="list-style-type: none"> <li>• Tablet-Friendly Design</li> <li>• System Bar</li> <li>• Quick Settings</li> </ul>	
2.3 - 2.3.7	Gingerbread	February 9, 2011	9 - 10	DVM	<ul style="list-style-type: none"> <li>• Gaming APIs</li> <li>• NFC</li> <li>• Battery Management</li> </ul>	
2.2 - 2.2.3	Froyo	May 20, 2010	8	DVM	<ul style="list-style-type: none"> <li>• Voice Action</li> <li>• Portable Hotspot</li> <li>• Dalvik JIT</li> </ul>	 <div> <b>Froyo</b>  <small>Android 2.2 / 2.2.3</small> </div>

2.0 - 2.1	Eclair	October 26, 2009	5	-	<ul style="list-style-type: none"> <li>• Google Maps Navigation</li> <li>• Home Screen Customization</li> <li>• Speech-to-Text</li> </ul>	
1.6	Donut	September 15, 2009	4	-	<ul style="list-style-type: none"> <li>• Quick Search Box</li> <li>• Screen Size Diversity</li> <li>• Android Market</li> </ul>	
1.5	Cupcake	April 27, 2009	3	-	-	

## ART dan DVM

### DVM (Dalvik Virtual Machine)

DVM menerapkan pendekatan JIT (Just-In-Time), di mana kompilasi dijalankan ketika ada permintaan untuk menjalankan aplikasi.

### ART (Android Runtime)

ART menerapkan pendekatan berbeda yaitu AOT (Ahead-Of-Time). AOT melakukan kompilasi pada saat proses instalasi aplikasi.

## Perbedaan DVM dan ART

### DVM



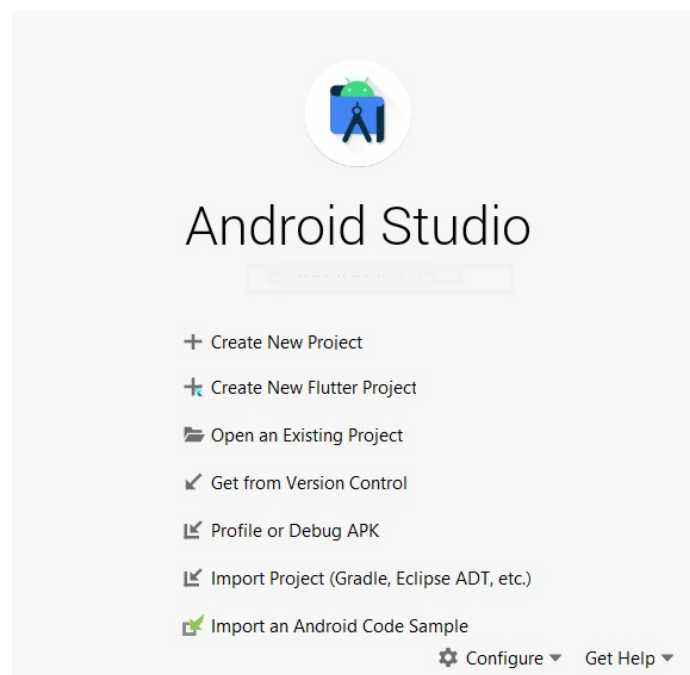
- DVM menggunakan JIT yang berarti kompilasi dilakukan setiap kali aplikasi dijalankan.
- Hal ini sangat mempengaruhi kecepatan respon aplikasi.
- Setiap kali kita menyentuh ikon aplikasi maka kompilasi akan dilakukan.
- Tentu proses ini menghabiskan CPU dan berimbas pada relatif lebih borosnya penggunaan baterai.

## ART

- ART melakukan proses kompilasi pada saat proses instalasi.
- Hal ini meningkatkan performa dalam menjalankan aplikasi.
- Selain itu karena penggunaan sumber daya CPU bisa dikurangi, pemakaian baterai jadi lebih hemat.
- Akan tetapi ART membutuhkan space (ukuran berkas) yang lebih besar jika dibandingkan dengan DVM.

# Project Pertama pada android

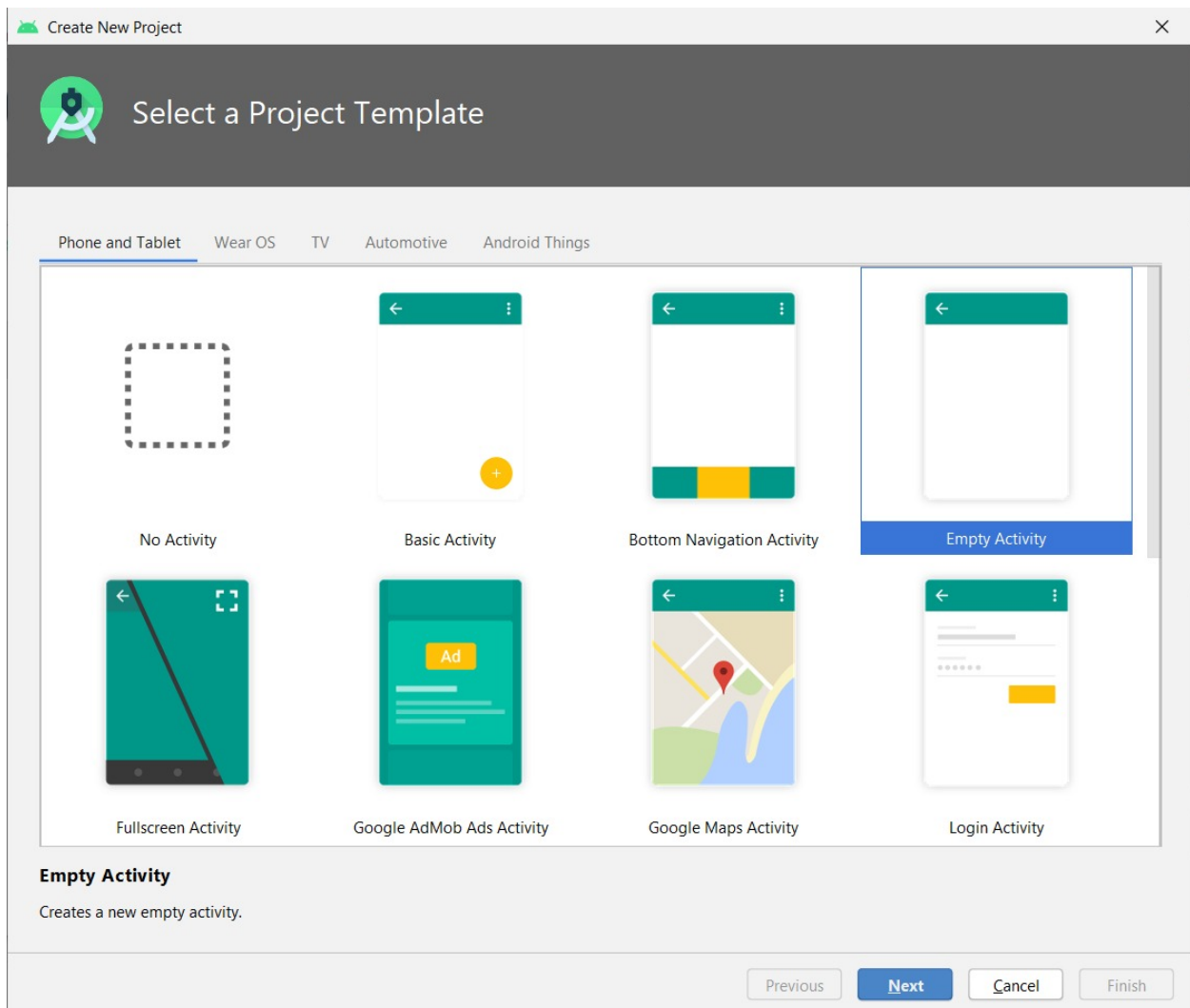
## Tampilan Awal



Untuk memulai proyek baru pilihlah “Start a new Android Studio project”.

## Project Wizard

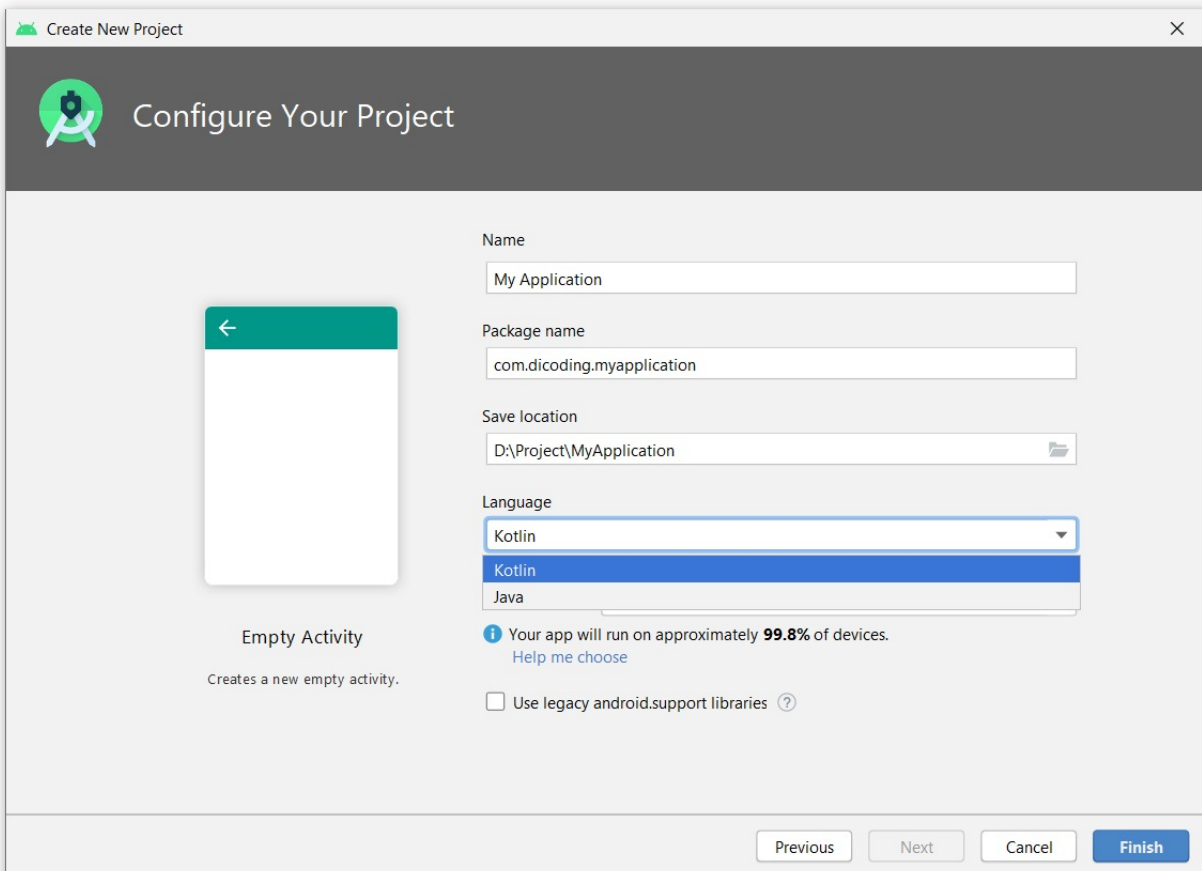
Pada Project wizard kita sudah disuguhkan dengan beberapa template seperti Empty Activity, Login Activity, Navigation Drawer Activity dan lain-lain. Di dalam project wizard juga dapat memilih target devices dari aplikasi yang akan kita buat.



## Configure Project

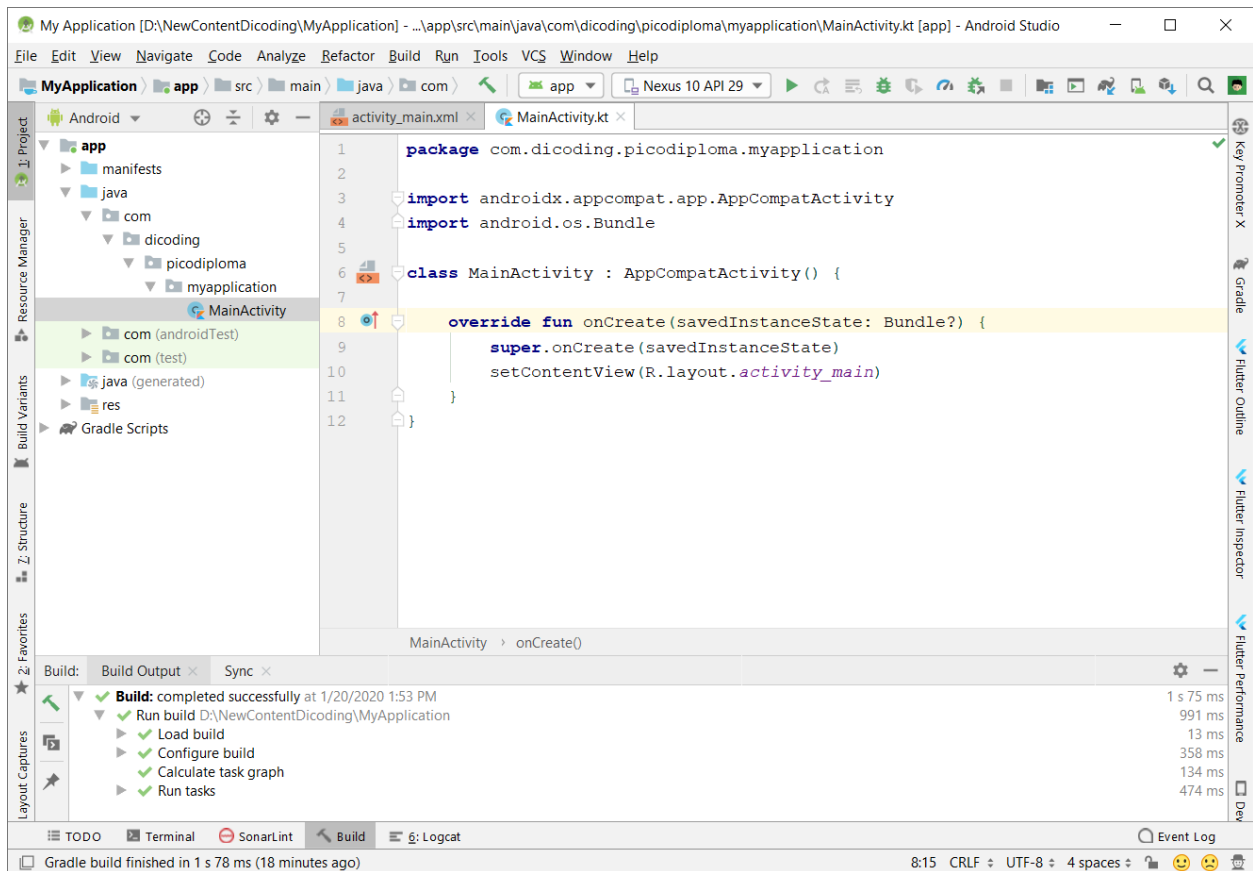
- Pada saat konfigurasi project setelah memilih template kita bisa memberi nama dari aplikasi, lokasi proyek dan nama package.
- Nama package akan digunakan dalam identifikasi unik dari aplikasi kita ketika sudah di-publish.

- Kita juga bisa mengganti nilai minimum SDK, yang berfungsi untuk membatasi penggunaan API pada sebuah aplikasi.
- Kita juga bisa mengganti bahasa default pada aplikasi tersebut menjadi bahasa Java atau Kotlin.



## Antarmuka Android Studio

Tampilan di bawah merupakan tampilan Full dari android studio



## Tools



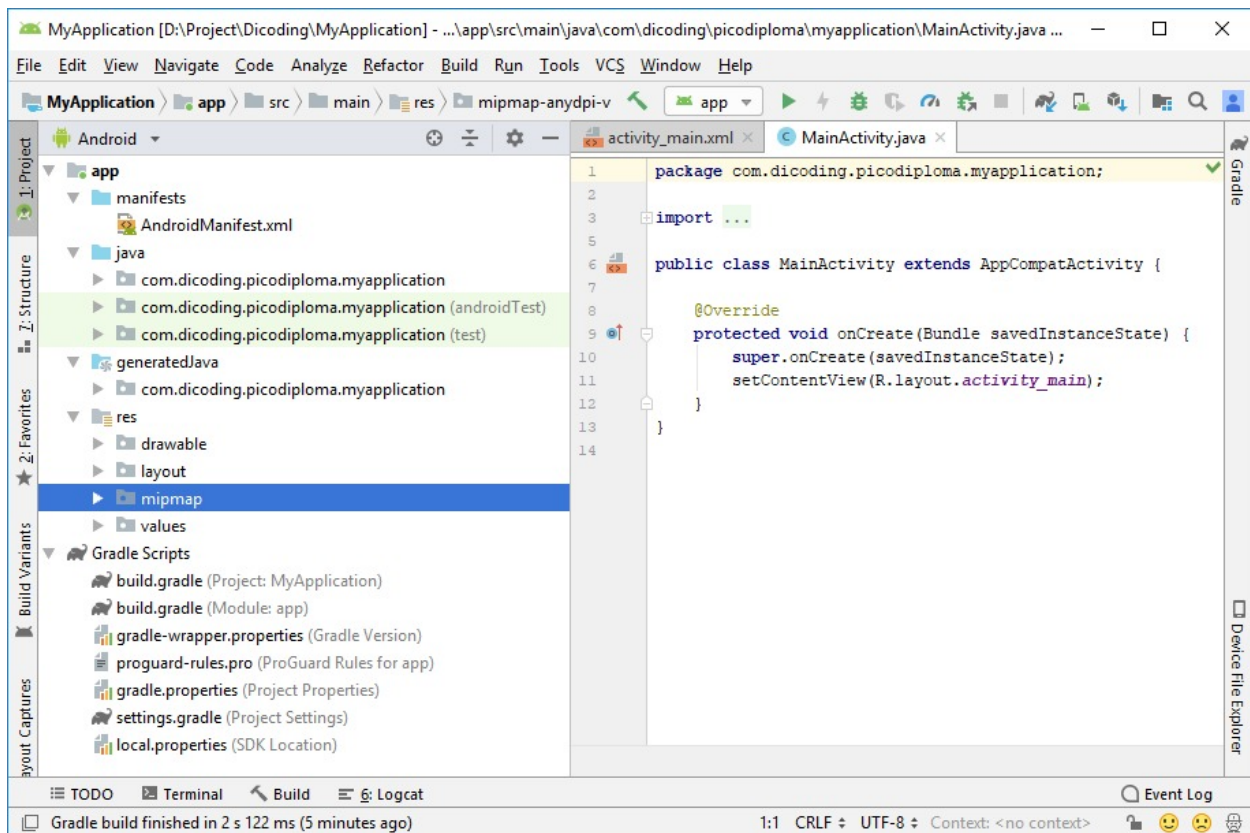
Merupakan tools yang sering digunakan dalam development seperti copy/paste, build, menjalankan aplikasi, hingga menjalankan emulator.

## Navigasi



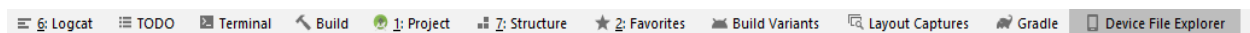
Membantu melihat struktur dari kedalaman (depth) dan posisi proyek yang saat ini sedang dibuka.

## Project Explorer dan Editor



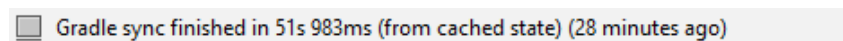
Merupakan bagian utama dari IDE Android Studio di mana kita menuliskan kode. Pada tampilan di atas, **sebelah kiri adalah struktur proyek kita** dan **sebelah kanan adalah editor**.

## Tool Window Bar



Tools menu yang mengelilingi editor ini merupakan button yang dapat di-expand ataupun untuk menampilkan tools secara detail dan individual.

## Status Bar



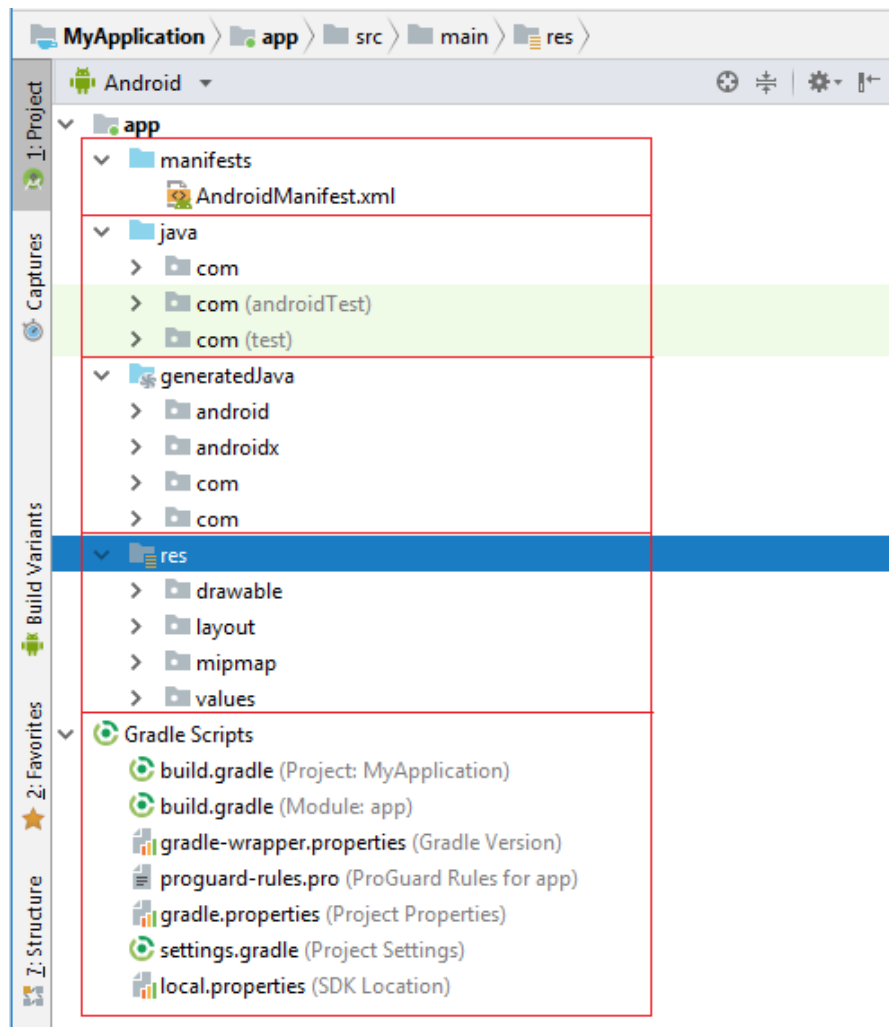
Terletak di bagian paling bawah dalam Android Studio, ia berfungsi untuk menampilkan status proyek kita dan pesan peringatan (warning message), bila ada.

## Struktur Project

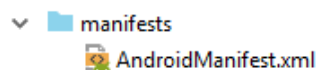
Ada 3 modul pada android studio

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

Berikut adalah Pembagian Projectnya



## Android Manifest



Manifest adalah salah satu berkas yang harus ada di dalam sebuah proyek Android. Manifest akan memberikan beragam informasi penting kepada sistem Android. Sistem perlu

mengetahui apa yang akan digunakan oleh aplikasi sebelum dijalankan.

Fungsi pada Manifest

- **Nama Package**

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.dicoding.picodiploma.myapplication">
```

Nama dari sebuah package bertindak sebagai identitas unik dari sebuah aplikasi. Identitas ini juga yang akan digunakan di dalam store untuk membedakan satu aplikasi dengan aplikasi lainnya. Jangan pernah mengganti value di dalam package karena nantinya akan dikenali sebagai aplikasi yang lain jika sudah masuk ke dalam store.

- **Komponen Aplikasi**

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="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>

    <service
        android:name=".MyIntentService"
        android:exported="false" />

    <receiver
        android:name=".MyReceiver"
        android:enabled="true"
        android:exported="true"></receiver>
</application>
```

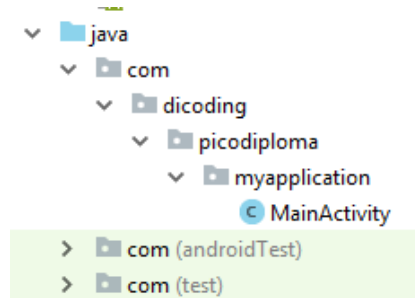
Berfungsi untuk mendeskripsikan komponen dari aplikasi mulai dari activity, services, broadcast receiver, dan content provider.

- **Permission**

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

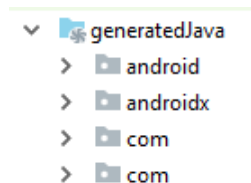
Mendeklarasikan permission apa saja yang harus dimiliki oleh aplikasi untuk akses ke dalam komponen API seperti internet, external storage, contact, dan juga untuk berinteraksi dengan aplikasi lainnya. Sebagai contoh ini adalah kode untuk permission Internet

## Java



Merupakan salah satu folder yang sering dipakai, berisi berkas source code kita yang ditulis dalam bahasa Java/Kotlin, termasuk juga kode Unit Test dan androidTest (Instrumentation Test).

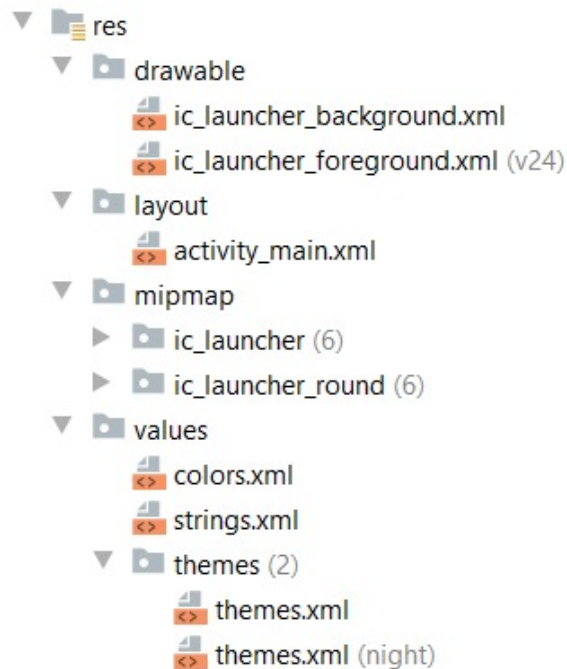
## GeneratedJava



Berisi file hasil dari generate library atau sebuah kelas dari proyek Android.

## Res / Resource

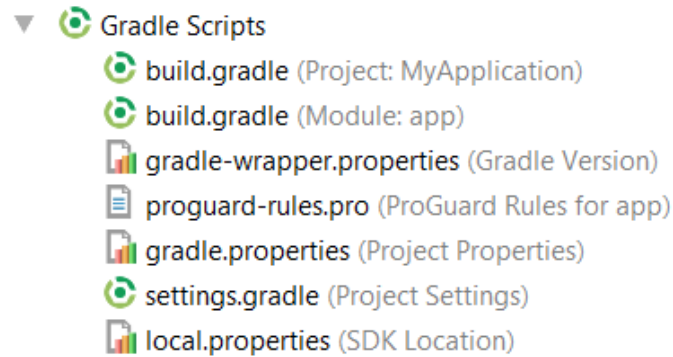




Mengatur resource di dalamnya, yang mana bukan berupa kode, melainkan layout aplikasi, sumber gambar, ikon, hingga style. Di dalam folder res ini juga terdapat sejumlah folder yang sudah diatur dan dikategorikan sesuai kebutuhan, seperti :

- **Drawable**  
Untuk menyimpan berkas gambar maupun ikon.
- **Layout**  
Salah satu folder yang sering dipakai untuk berkas desain aplikasi.
- **Mipmap**  
Untuk menyimpan logo dalam berbagai dimensi.
- **Values**  
Berisi berbagai macam sumber data, seperti `colors.xml` untuk warna, `strings.xml` untuk teks, `dimens.xml` untuk ukuran, dan `styles.xml` untuk membuat *style* atau *template*.

## Gradle



Gradle merupakan open source build automation system. Automation system berguna untuk mengotomatisasi proses pembuatan dari software build dan proses-proses terkait lainnya termasuk compile source code menjadi binary code, packaging binary code, dan menjalankan automated test.

- **build.gradle (Project: MyApplication)**

Merupakan software build tingkat teratas yang meliputi keseluruhan dari proyek dari sebuah aplikasi. Di dalamnya berisi konfigurasi semua modul yang ada di dalam proyek.

- **build.gradle (Module: app)**

Merupakan software build yang ada pada setiap modul di dalam proyek sebuah aplikasi. Beberapa konfigurasi yang diedit di antaranya adalah android settings, defaultConfig dan productFlavors, buildTypes, dan dependencies.

```
plugins {
    id 'com.android.application'
}

android {
    compileSdkVersion ...
    buildToolsVersion "..."

    defaultConfig {
        applicationId "com.dicoding.picodiploma.myapplication"
        minSdkVersion ...
        targetSdkVersion ...
        versionCode 1 //incremental
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}
```

```

        compileOptions {
            sourceCompatibility JavaVersion.VERSION_1_8
            targetCompatibility JavaVersion.VERSION_1_8
        }
    }

    dependencies {
        implementation ...
        testImplementation ...
        androidTestImplementation ...
    }

```

- **Android Settings**

Di dalam *block* android kita bisa menetapkan `compileSdkVersion` dan `buildToolsVersion`.

- **Default Config**

Di dalamnya terdapat *properties* seperti `applicationID`, `minSdkVersion`, `targetSdkVersion` dan *test information*.

- **Build Types**

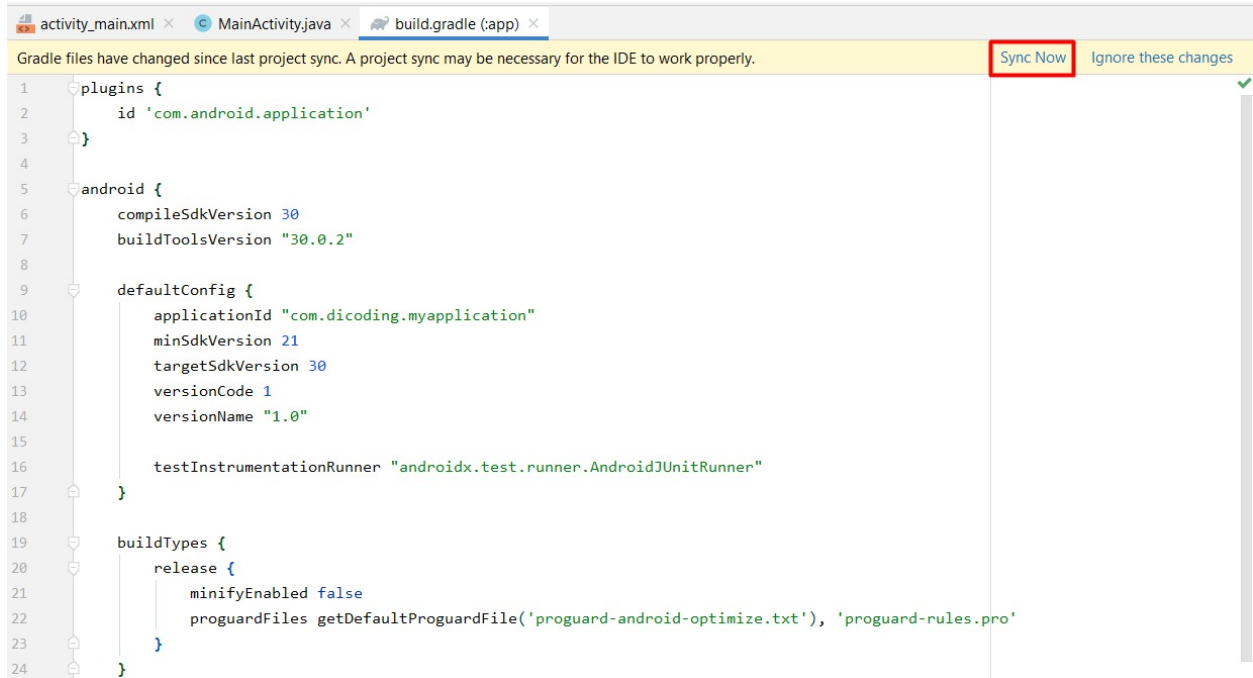
Di dalamnya terdapat *properties* dari `debuggable`, *ProGuard enabling*, *debug signing*, *version name suffix* dan *test information*.

- **Dependencies**

Di dalamnya terdapat informasi tentang *library* yang digunakan oleh aplikasi.

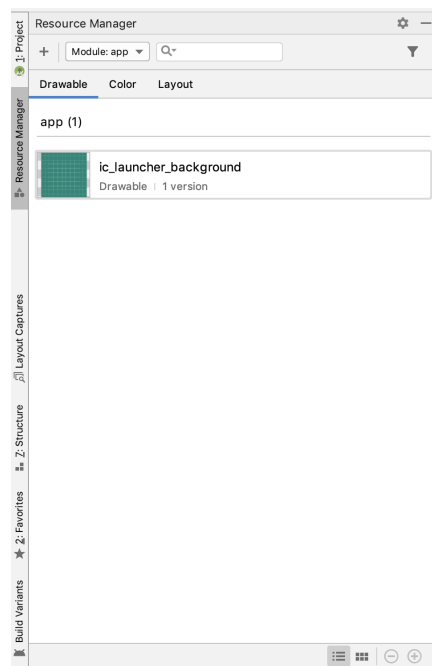
## Sync Project

Setiap kali terjadi perubahan informasi di dalam `build.gradle` kita harus melakukan sinkronisasi terlebih dahulu.



## Resource Manager

Resource Manager, berfungsi untuk manajemen segala resource yang ada di proyek Anda seperti gambar, warna, layout dll.



# Useful Tools pada Android Studio

## Shortcut

Pencarian:

- **Shift+Shift**

*Search Everywhere*, atau dapat dikatakan pencarian semua jenis berkas yang masih dalam 1 proyek.

- **Ctrl+F**

*Find*, pencarian teks dalam salah satu berkas.

- **Ctrl+Shift+F**

*Find in path*, pencarian teks di seluruh berkas proyek.

- **Ctrl+Shift+A**

*Find action*, pencarian aksi atau perintah-perintah yang ada di Android Studio.

- **Ctrl+R**

*Replace*, mengganti teks di dalam berkas.

## Navigasi:

- **Ctrl+N**  
*Find Class*, navigasi ke kelas tertentu.
- **Ctrl+Shift+N**  
*Find file*, navigasi ke berkas.
- **Ctrl+B**  
*Go to declaration*, lompat ke deklarasi yang dipilih.
- **Alt+↑**  
Lompat ke method sebelumnya.
- **Alt+↓**  
Lompat ke method sesudahnya.
- **Ctrl+G**  
*Go to line*, lompat ke baris tertentu.
- **Ctrl+E**  
Membuka berkas teranyar (*recent file*).
- **Ctrl+Left Mouse (or) Ctrl+Alt+F7**  
Melihat penggunaan pada variabel/objek yang diklik.
- **Alt+F7 / Ctrl+F7**  
Melihat penggunaan variabel/objek yang dipilih di seluruh berkas proyek.
- **Ctrl+Shift+B**  
Mencari tahu implementasi dari variabel/objek yang dipilih.

Redaksi:

- **Ctrl+D**  
Menggandakan bagian yang dipilih.
- **Ctrl+Q**  
Melihat dokumentasi dengan tampilan minimal.
- **Ctrl+P**  
Melihat isi dari parameter, penting ketika melihat method dari Android atau library lain.
- **Ctrl+Space**  
*Basic code completion*, menampilkan saran untuk melengkapi kode Anda.
- **Ctrl+Shift+Space**  
*Smart code completion*, menampilkan saran kode untuk melengkapi kode Anda dengan lebih pintar (menampilkan apa yang benar-benar terkait dengan kode Anda).
- **Alt+Insert**  
*Generate code*, menghasilkan (*generate*) kode. Perintah ini sangat memudahkan ketika membuat constructor dan setter/getter.
- **Ctrl+Alt+L**  
Memformat ulang kode, merapikan kode.
- **Ctrl+Y**  
*Delete One Line*, Menghapus satu baris kode.
- **Ctrl+Alt+V**  
*Create variable*, Membuat teks yang diblok menjadi sebuah variabel.
- **Ctrl+Alt+M**  
*Create method*, Membuat teks yang diblok menjadi sebuah fungsi.
- **Shift+F6**  
Rename, untuk mengganti nama suatu file atau variabel maupun fungsi.

## Run

- **Ctrl+F9**  
*Make project, build project.*
- **Ctrl+Shift+F9**  
Melakukan kompilasi pada berkas, package atau modul.
- **Shift+F10**  
*Run.* Menjalankan aplikasi ke *emulator* atau *devices*.
- **Shift+F9**  
*Debug.* Menjalankan aplikasi ke *emulator* atau *devices* dalam mode **Debug**, biasanya untuk keperluan *testing*.

## Code Completion

Untuk meminimalisir salah ketik (typo) dalam pemanggilan class, method hingga variabel sebaiknya kita memanfaatkan Code Completion di Android Studio.

- Basic Code Completion  
**Ctrl+Space** ⇒ Pemanggilan code completion standar untuk membantu kita melengkapi kode.
- Statement Completion  
**Ctrl+Shift+Enter** ⇒ Perintah ini sangat membantu karena kita bisa menyelesaikan kode tanpa harus mengetik lengkap dan tanpa tanda kurung, kurung siku, kurung kurawal, dan banyak macam pemformatan lainnya.

## Style and Formating

- Gaya penulisan kode adalah seni dalam pemrograman. Kita memiliki signature style masing-masing, Semua tergantung pilihan kita sendiri.
- Tetapi kita tetap perlu memperhatikan bagaimana tata letak kode, apalagi bila suatu saat nanti kita membuat aplikasi bersama orang lain.
- Kode yang rapi itu enak dilihat dan memudahkan, baik kita maupun orang lain untuk membacanya.
- Secara default Android Studio memberikan code style formatting untuk tata letak kode yang kita miliki.
- Nah, kini kita akan melakukan kode formatting dengan menggunakan shortcut  
**Ctrl+Alt+L**.

## Sample Code

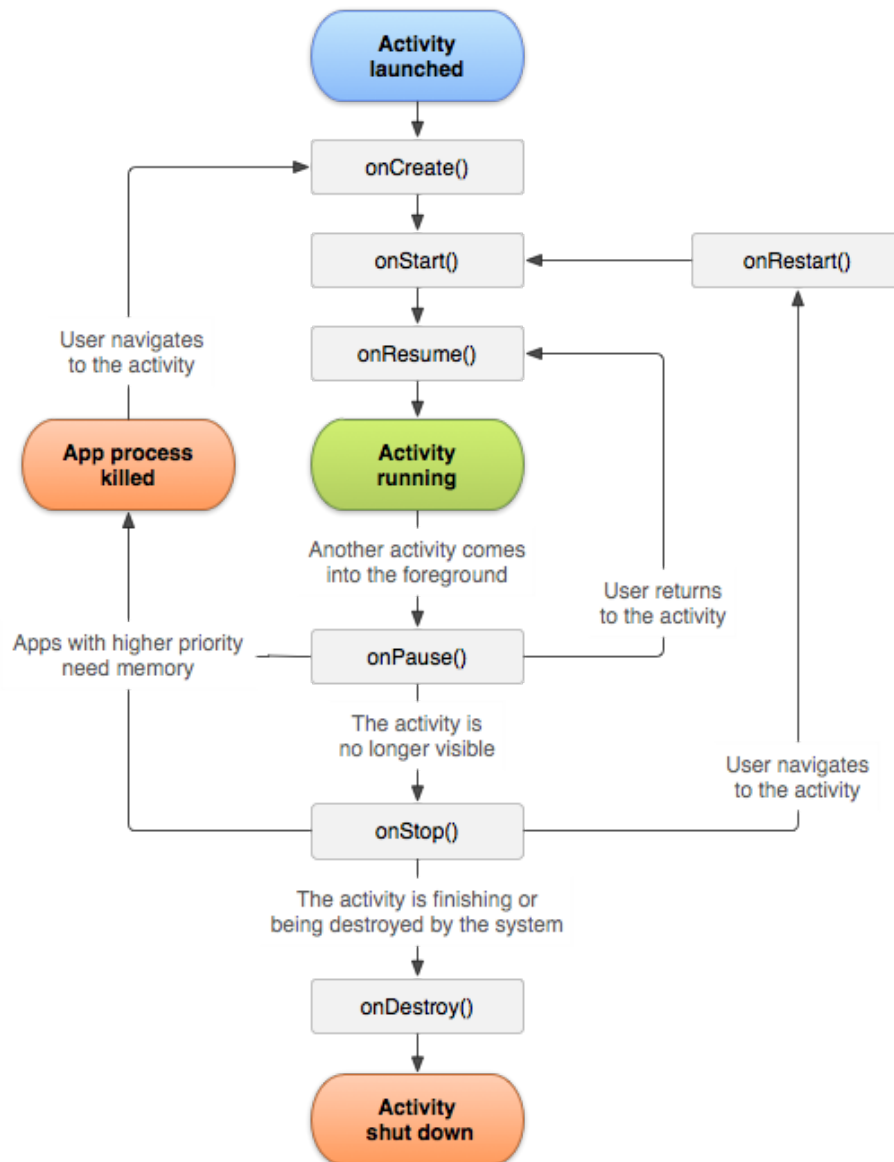


- Android Studio juga membantu kita menemukan kode yang berkualitas dan best practice-nya.
- Melalui Google, Android Studio memiliki sample code yang bebas kita gunakan dan manfaatkan untuk kebutuhan kita belajar atau membuat aplikasi Android.
- Dengan mengakses File > New > Import Sample, kita punya banyak pilihan contoh kode yang bisa dipakai.
- referensi Link <https://developer.android.com/studio/write/sample-code>

## Komponen dasar pada Android

### 1. Activity

Activity menangani window (tampilan) mana yang akan di tampilkan ke dalam interface (antarmuka). Activity memiliki daur hidup (life cycle) tersendiri yang dimulai dari onCreate hingga onDestroy.



## 2. Intent

Beberapa fungsi dari intent adalah dapat digunakan untuk menjalankan sebuah activity, mengirimkan pesan ke broadcast receiver, dan dapat juga digunakan untuk berkomunikasi dengan service yang sedang berjalan.

## 3. Views and ViewGroup

Sebuah android dibangun oleh 2 komponen inti yaitu Views dan ViewGroup. 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.

#### 4. Style and Theme

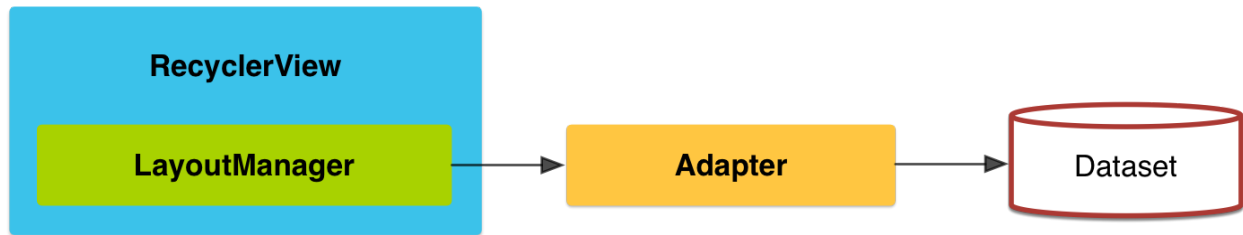
Prinsip dasar untuk merancang Antarmuka pada android harus berdasar pada Design Guideline.

Beberapa prinsip Guideline yang dibuat oleh Google sebagai berikut :

- **Menampilkan informasi yang hanya dibutuhkan.**
- **Jika aplikasi meminta izin pengguna untuk melakukan sebuah aksi, maka pengembang harus menyediakan mekanisme untuk membatalkan izin tersebut.**
- **Lakukan interupsi jika diperlukan.**
- **Menggunakan teks secara singkat.**
- **Gunakan gambar untuk menjelaskan informasi secara lebih deskriptif.**
- **Jaga data pengguna.**
- **Permudah pengguna untuk melakukan sesuatu yang penting secara cepat.**
- **Jika terlihat sama, maka perilaku haruslah sama.**
- **Bantu pengguna untuk membuat keputusan tapi tetap biarkan pengguna menentukan keputusannya.**

#### 5. Recycler View

RecyclerView adalah sebuah komponen tampilan (widget) yang lebih canggih ketimbang pendahulunya listview. RecyclerView memiliki kemampuan untuk menampilkan data secara efisien dalam jumlah yang besar. Terlebih jika Anda memiliki koleksi data dengan elemen yang mampu berubah-ubah sewaktu dijalankan (runtime).



- **RecyclerView dan LayoutManager:** Komponen antarmuka yang bertugas untuk menampilkan data set yang dimiliki di dalamnya. Layoutmanager akan mengatur posisi tampilan data baik itu secara list (vertikal), grid (baris dan kolom) atau staggered grid (grid yang memiliki susunan tak seragam / tak beraturan)
- **Adapter:** Komponen yang akan mengatur bagaimana menampilkan data set ke dalam **RecyclerView**. Di sinilah terjadi proses pengisian tampilan (**ViewInflate**) dari file layout xml untuk tiap elemen dari data yang sebelumnya terpasang (bind) ke dalam **RecyclerView**.
- **Dataset:** Kumpulan data yang dimiliki dan ingin ditampilkan. Bisa berupa array, list maupun obyek map
- **Item Animator:** Ini yang spesial. Kita bisa pasang animasi untuk tiap item di dalamnya. Contoh animasi yang umum seperti penambahan (add) dan penghapusan (removal) item. Kita akan mempelajari hal ini pada materi terpisah.