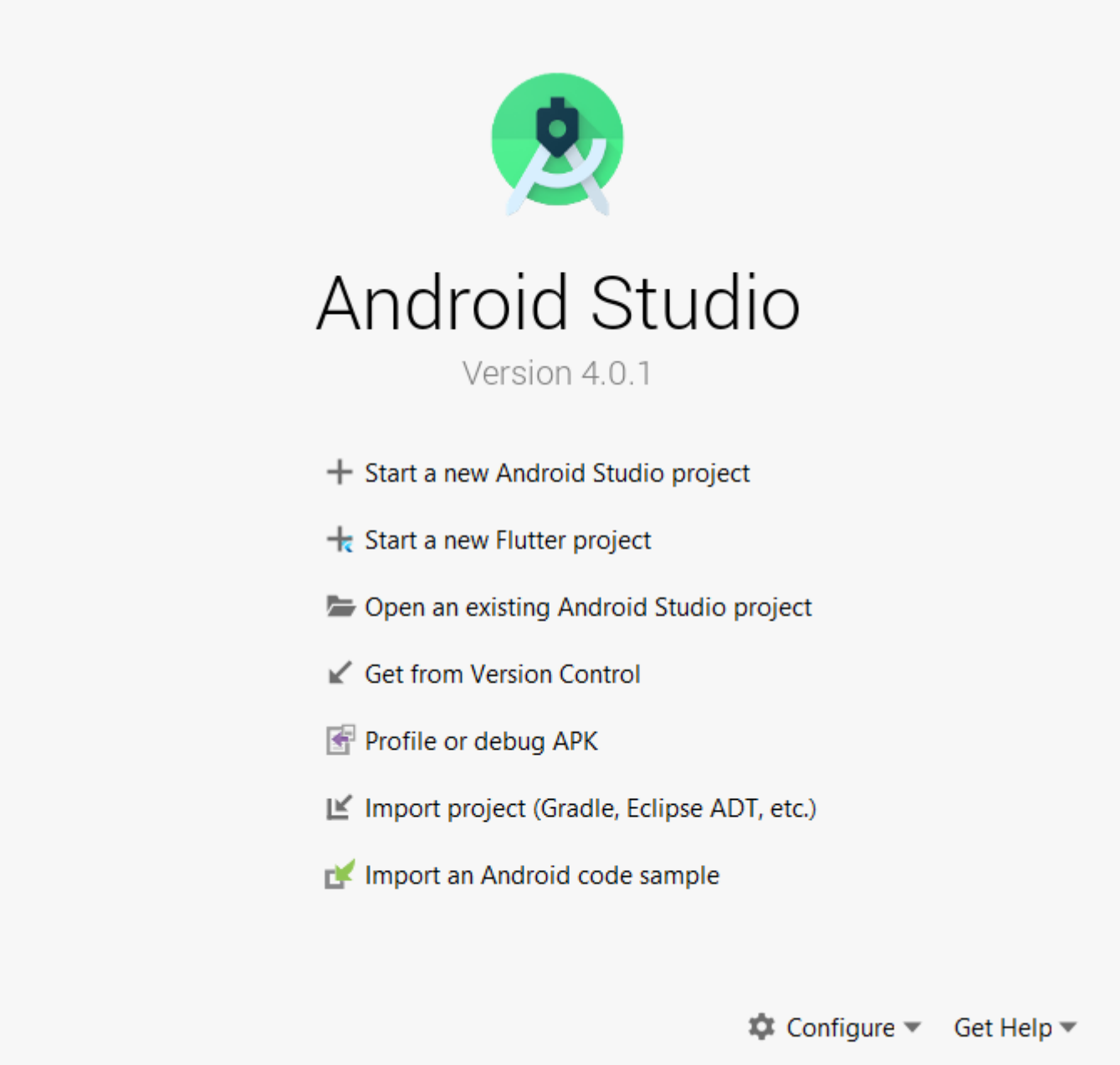
Introduction to Android Studio

Agar mulus menggunakan Android Studio, baiknya Anda tahu bagaimana strukturnya terlebih dahulu. Bila terbiasa menggunakan produk IntelliJ IDEA lainnya, maka akan mudah bagi Anda untuk menavigasi tata letak dan struktur Android Studio ini. Yang berbeda adalah komponen tambahan pendukung yang membantu pengembangan dan pembuatan aplikasi Android. Yuk, kita kenali Android Studio, *right from the git-go!!*

Project Android Pertama

Kali pertama menjalankan Android Studio, Anda akan melihat tampilan seperti berikut
ini.



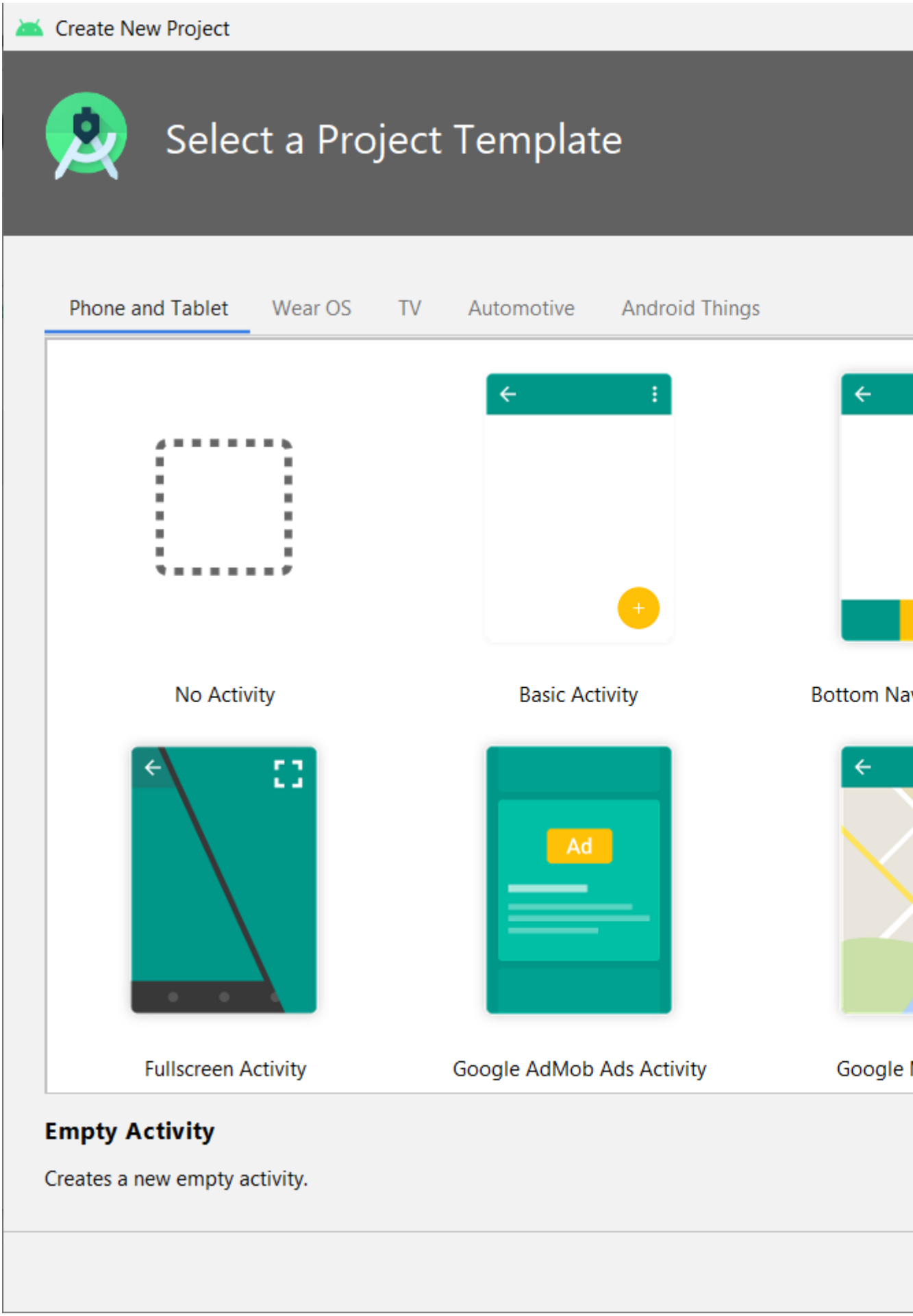
Untuk memulai proyek baru pilihlah **"Start a new Android Studio project"**.

Project Wizard

Setelah proses instalasi Android Studio, mari kita mulai membuat sebuah aplikasi **Hallo World,** aplikasi pertama Anda.

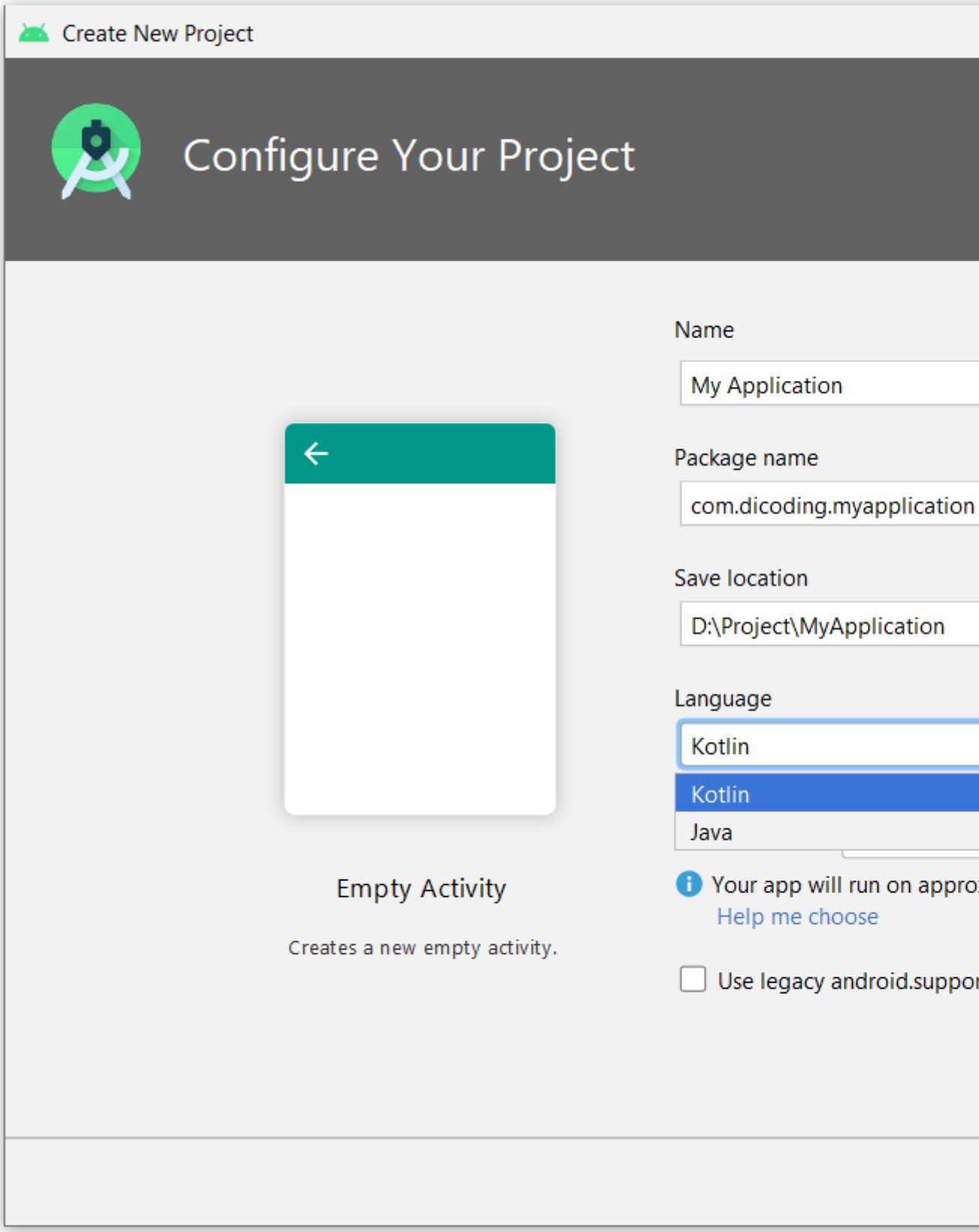
1. Setelah melakukan **"Start a new Android Studio project"**, Anda diminta untuk melakukan konfigurasi dalam pembuatan proyek baru Anda. Dalam dialog ini Anda bisa memilih **template** dari Aplikasi yang akan Anda buat. Terdapat beberapa *template* yang bisa kita gunakan seperti **Empty Activity**, **Login Activity**, **Navigation Drawer Activity** dan lain-lain. Di dalam dialog ini kita bisa

memilih target *devices* dari aplikasi yang akan kita buat.


Create New Project
Select a Project Template
Phone and Tablet
Wear OS
TV
Automotive
Android Things
No Activity
Basic Activity
Bottom Nav
Fullscreen Activity
Ad
Google AdMob Ads Activity
Google M
Empty Activity
Creates a new empty activity.

2. Dalam dialog ini Anda bisa memberi nama dari aplikasi, lokasi proyek Anda dan nama *package*. Nama *package* akan digunakan dalam identifikasi unik dari aplikasi kita ketika sudah di-*publish*. Kita juga dapat mengganti dari direktori di mana proyek kita akan disimpan. Kita juga bisa mengganti nilai minimum SDK, yang berfungsi untuk membatasi penggunaan API pada sebuah aplikasi. Di dalam dialog ini juga Anda bisa memberi nama dari Activity yang pertama kali yang Anda buat. Selain itu Anda juga bisa mengganti bahasa *default* pada aplikasi tersebut menjadi bahasa Java atau

Kotlin.



Catatan: Untuk Android Studio versi **3.4 ke bawah,** penggunaan **AndroidX** belum menjadi *default*. silakan beri tanda centang pada **"Use AndroidX.* artifacts"** di setiap kali membuat *project*. Informa

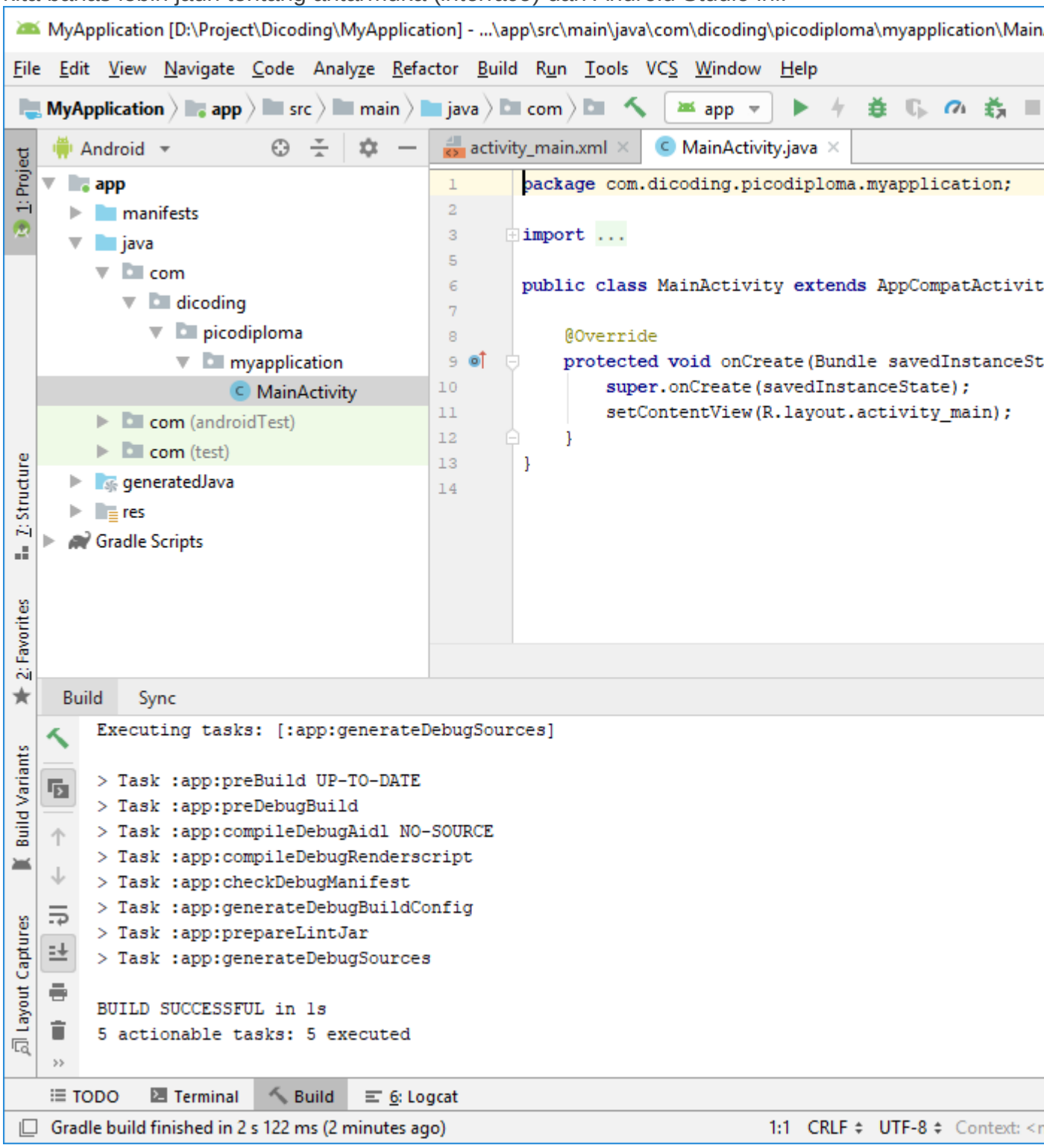
> detail mengenai **AndroidX** bisa Anda lihat di sini, dan untuk melihat perubahan antara sebelum dan sesudah **AndroidX**, atau cara migrasi menggunakan **AndroidX** bisa lihat di sini. Perlu Anda ketahui ketika Anda sudah menggunakan **AndroidX** maka Anda tidak bisa menambahkan library lama atau sebelum **AndroidX**.

3.

Antarmuka Android Studio

OK, membuat proyek pertama kali di Project Wizard, *done*! Kali ini kita akan menemui tampilan penuh Android Studio. Untuk meningkatkan produktivitas, mari

kita bahas lebih jauh tentang antarmuka (*interface*) dari Android Studio ini.



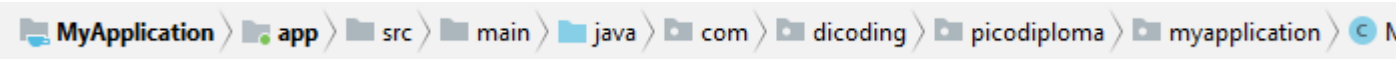
Di atas adalah *screenshot* tampilan penuh IDE Android Studio berbasis IntelliJ IDEA. Mungkin tampilan tersebut akan berbeda dengan tampilan di layar Anda karena perbedaan konfigurasi dan versi Android Studio.

Tools



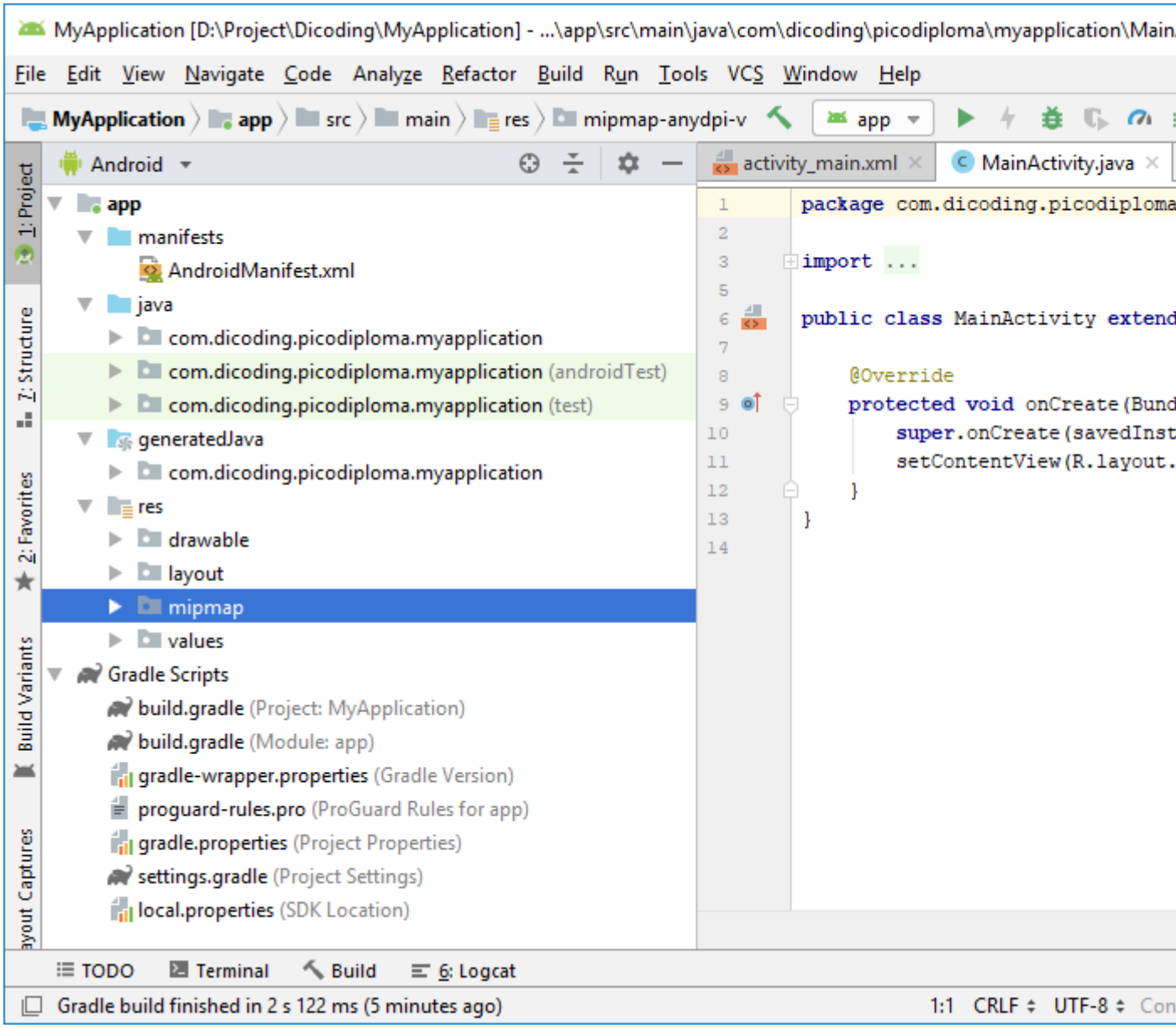
Merupakan *Tools* yang sering digunakan dalam *development* seperti *copy/paste, build,* menjalankan aplikasi, hingga menjalankan *emulator*.

Navigasi



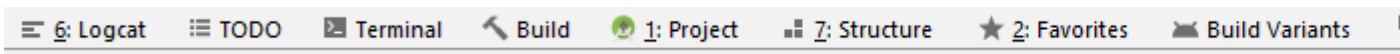
Membantu melihat struktur dari kedalaman (*depth*) dan posisi proyek yang saat ini sedang dibuka.

Project Explorer dan Editor



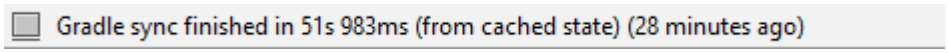
Merupakan bagian utama dari IDE Android Studio di mana kita menuliskan kode. Pada tampilan di atas, sebelah kiri adalah struktur proyek kita dan sebelah kanan adalah editor. Bagian ini akan dibahas lebih detail di poin selanjutnya.

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



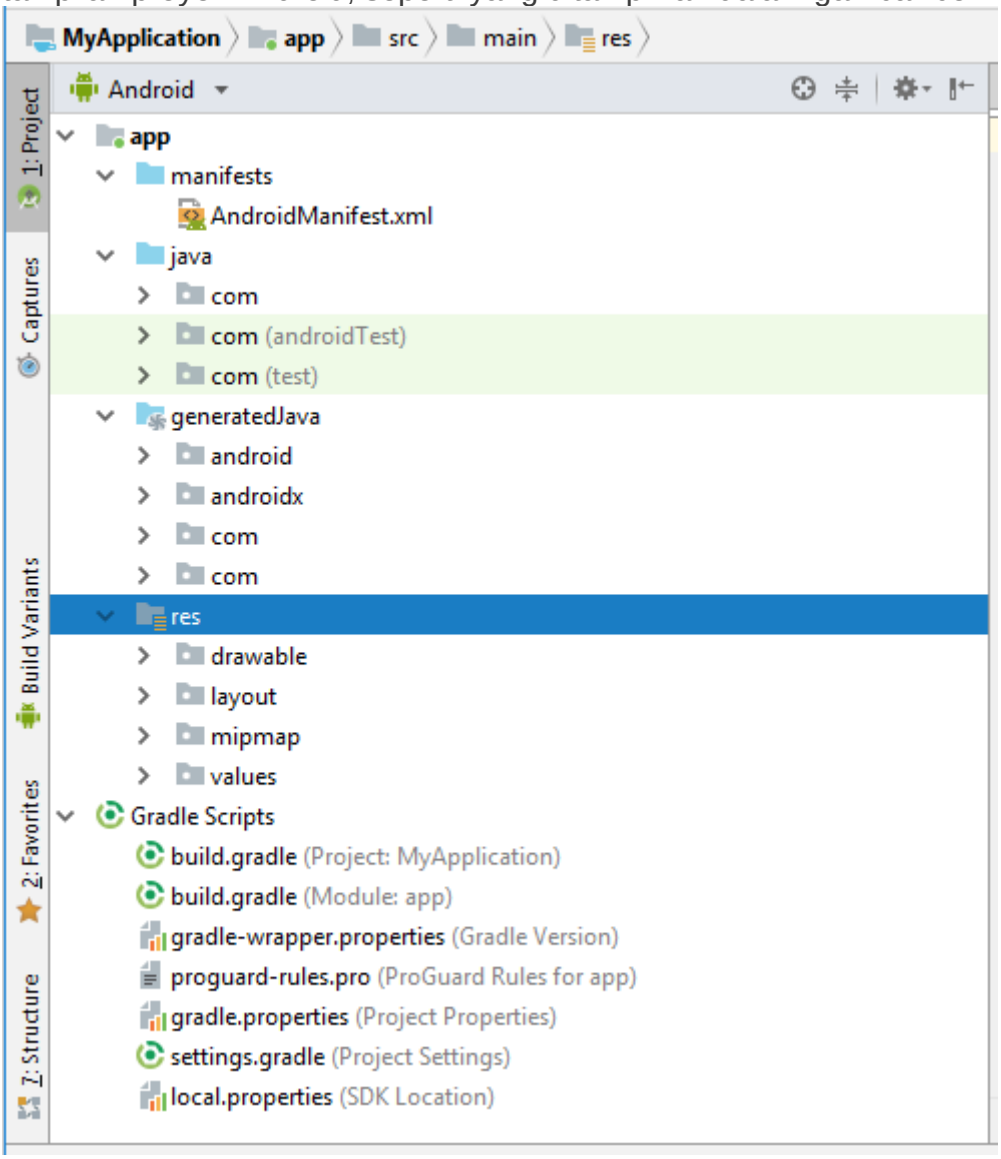
Terletak di bagian terbawah di Android Studio, ia berfungsi untuk menampilkan status proyek kita dan pesan peringatan (*warning message*), bila ada.

Struktur Proyek

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

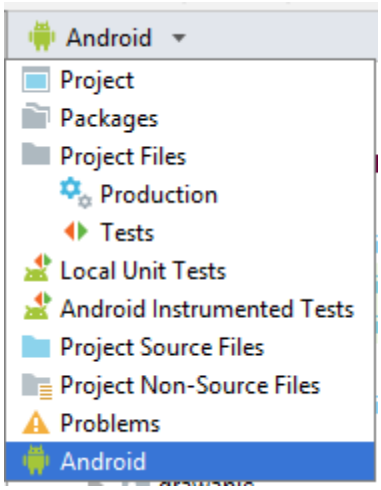
- Modul Aplikasi Android
- Modul Pustaka
- Modul Google App Engine

Secara *default*, Android Studio akan menampilkan berkas proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan dalam gambar berikut :



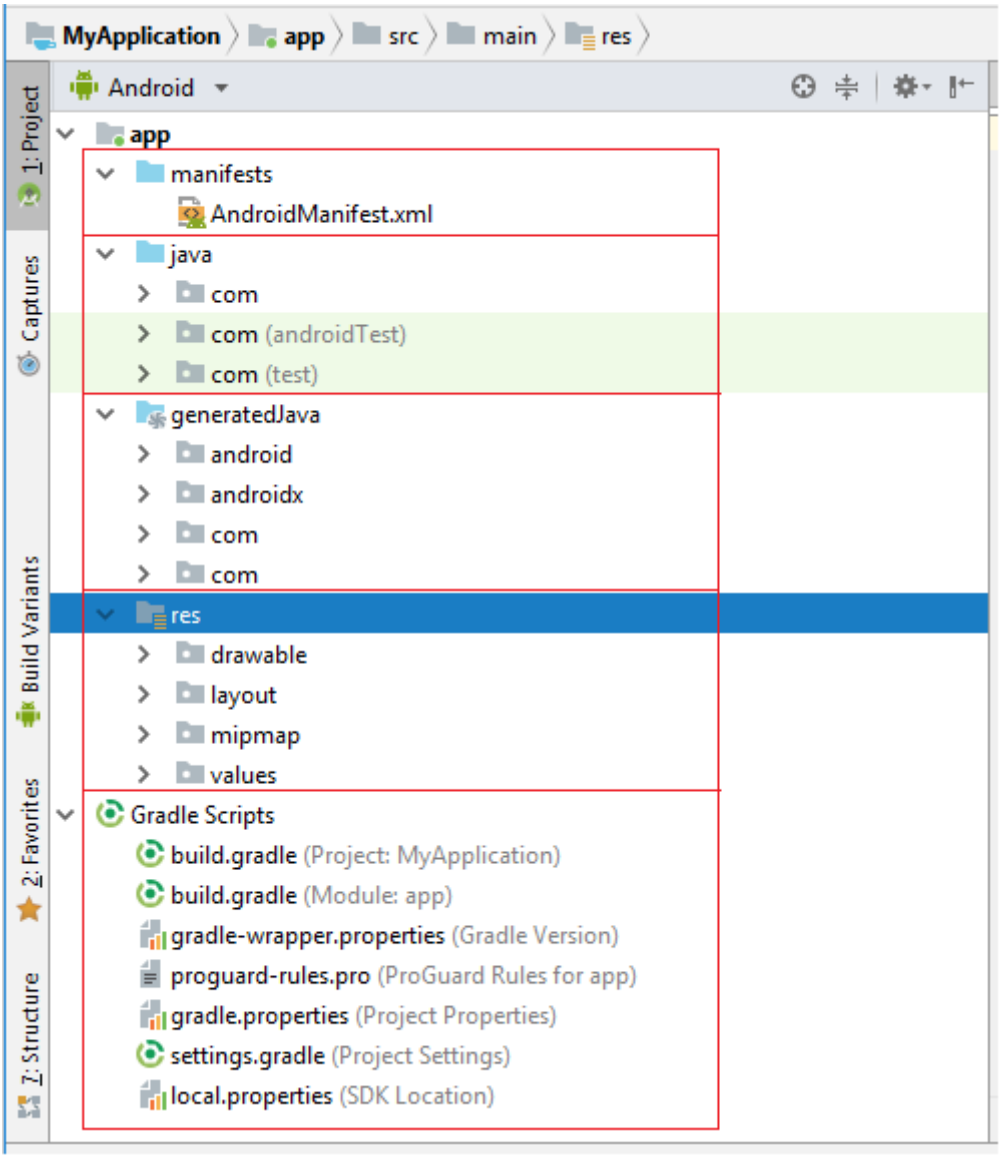
Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke berkas sumber utama proyek Anda.

Secara *default* ketika kita membuat proyek baru, Android Studio akan menampilkan struktur yang lebih ringkas dan cepat sesuai dengan kebutuhan pengembangan Android. Bila ingin melihat struktur proyek dalam bentuk selain standar Android, kita dapat mengubahnya dengan tombol *dropdown* yang terdapat di atas *project structure*.



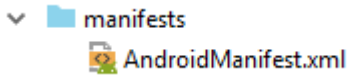
Pada bagian ini kita dapat mengganti tampilan *project structure* sesuai kebutuhan.

Mari kita bahas lebih detail tentang proyek yang baru saja kita buat.



Masing-masing modul aplikasi berisi folder berikut:

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.

Beberapa fungsi yang ada di dalam manifest adalah sebagai berikut.

- **Nama Package**

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

Package name merupakan sebagai identitas unik dari sebuah aplikasi. Identitas ini juga yang akan digunakan di dalam store untuk membedakan suatu 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**

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

```
1.      <application
2.          android:allowBackup="true"
3.          android:icon="@mipmap/ic_launcher"
4.          android:label="@string/app_name"
5.          android:roundIcon="@mipmap/ic_launcher_round"
6.          android:supportsRtl="true"
7.          android:theme="@style/AppTheme">
8.          <activity android:name=".MainActivity">
9.              <intent-filter>
10.                 <action android:name="android.intent.action.MAIN" />
11.
12.                 <category
    android:name="android.intent.category.LAUNCHER" />
13.             </intent-filter>
14.         </activity>
15.
16.         <service
17.             android:name=".MyIntentService"
18.             android:exported="false" />
19.
20.         <receiver
21.             android:name=".MyReceiver"
22.             android:enabled="true"
23.             android:exported="true"></receiver>
24.     </application>
```

  Komponen aplikasi semuanya berada di antara tag <application>. Ia juga berfungsi sebagai penamaan kelas yang mengimplementasi komponen dan mendeskripsikan kemampuannya seperti intent-filter, di mana fungsinya mendeskripsikan bahwa komponen itu adalah yang pertama kali dijalankan.

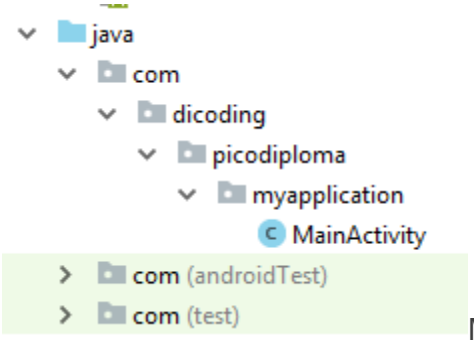
- **Permission**

  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

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

  Kode ini biasanya diletakkan di atas tag <application> dan masih di dalam tag <manifest>

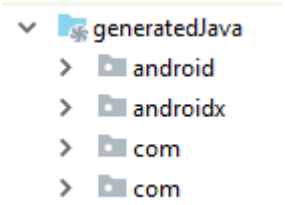
Java



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).

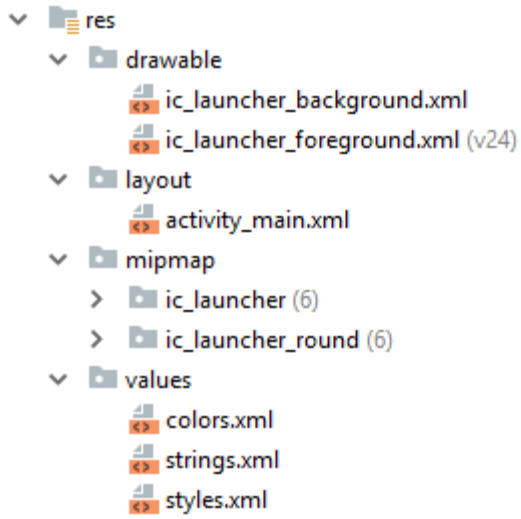
Catatan: Penamaan folder "java" bisa juga diganti sesuai dengan bahasa yang dipakai, misal "kotlin."

GeneratedJava



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

Res / Resource

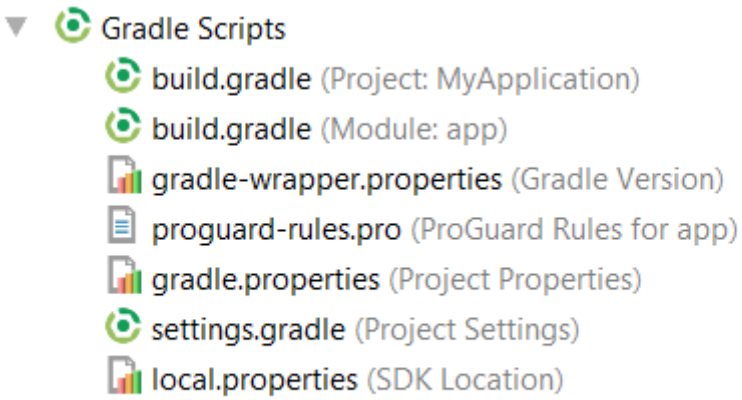


Mengatur *resource* di dalamnya, yang mana bukan berupa kode, melainkan *layout* aplikasi, sumber gambar, ikon, hingga *style*. Di dalam folder **res** ini juga terdapat sejumlah folder yang sudah diatur dan dikategorikan sesuai kebutuhan, seperti :

- **Drawable**
  Untuk menyimpan file gambar maupun ikon.
- **Layout**
  Salah satu folder yang sering dipakai untuk file 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 themes.xml/ 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.

```
1.      apply plugin: 'com.android.application'
2.
3.      android {
4.          compileSdkVersion 29
5.          defaultConfig {
```

```
6.              applicationId "com.dicoding.picodiploma.myapplication"
7.              minSdkVersion 15
8.              targetSdkVersion 29
9.              versionCode 1 //(Incremental)
10.             versionName "1.0"
11.             testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
12.         }
13.         buildTypes {
14.             release {
15.                 minifyEnabled false
16.                 proguardFiles getDefaultProguardFile('proguard-android.txt'),
    'proguard-rules.pro'
17.             }
18.         }
19.     }
20.
21.     dependencies {
22.         implementation fileTree(dir: 'libs', include: ['*.jar'])
23.         implementation 'androidx.appcompat:appcompat:1.0.2'
24.         implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
25.         testImplementation 'junit:junit:4.12'
26.         androidTestImplementation 'androidx.test:runner:1.2.0'
27.         androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
28.     }
```

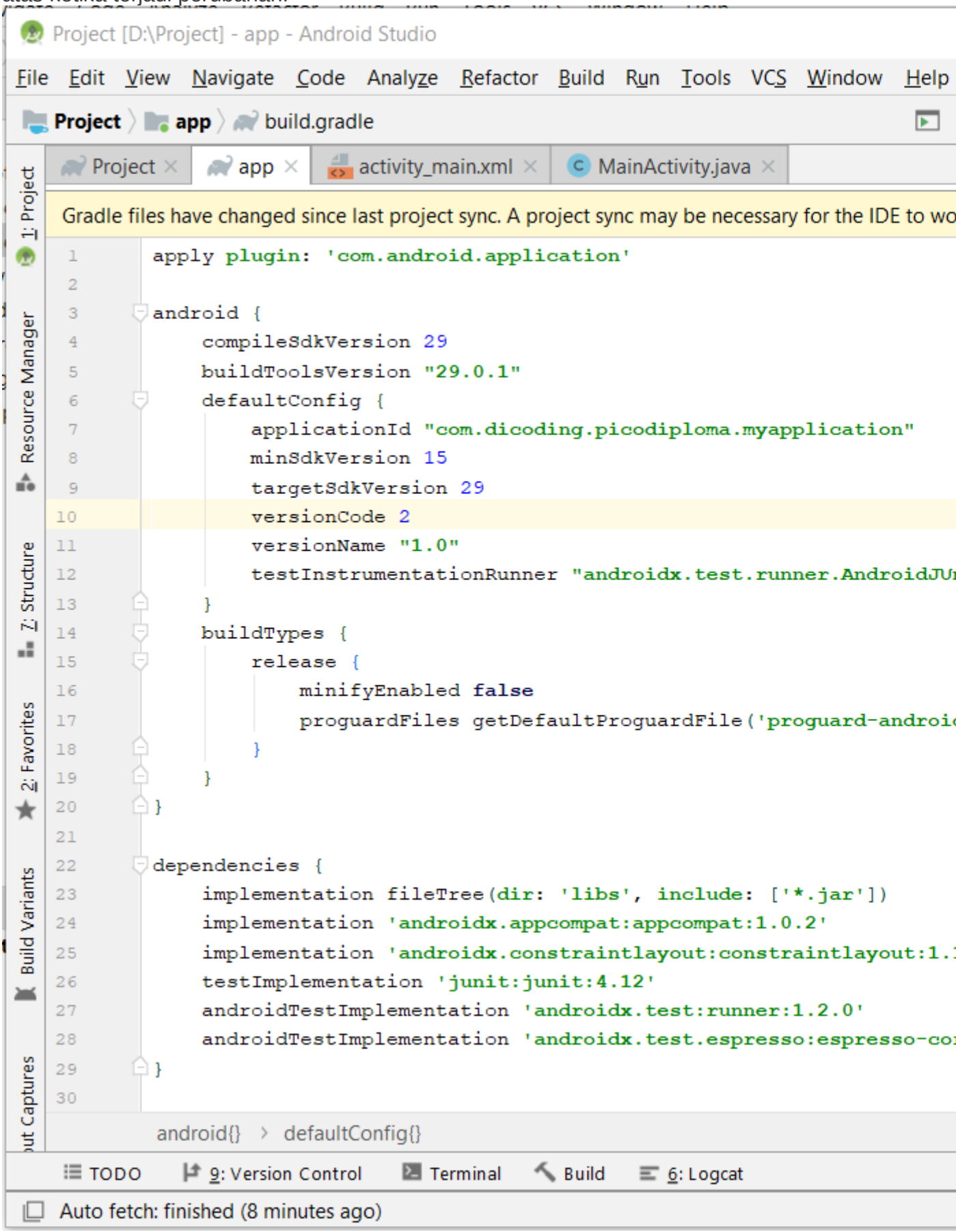
- **Android Settings**
  Di dalam *block* android kita bisa menetapkan compileSDKVersion dan buildToolsVersion.
- **Default Config**
  Di dalamnya terdapat *properties* seperti applicationID, minSdkVersion, targetSdkVersion dan test information.
- **Build Types**
  Di dalamnya terdapat *properties* dari debuggable*, ProGuard enabling, debug signing, version name suffix* dan *test information*.
- **Dependencies**
  Di dalamnya terdapat informasi tentang *library* yang digunakan oleh aplikasi.

Yang perlu diperhatikan di sini yaitu di bagian versionCode, misal kita sudah *publish* aplikasi ke PlayStore dengan *version code* 1, maka ketika kita ingin meng-*update* aplikasinya lagi ke Playstore, kita perlu merubah *version code*-nya menjadi 2 (*incremental*), kalau tidak kita ubah maka PlayStore akan menolak APK yang kita *upload*.

Sync Project

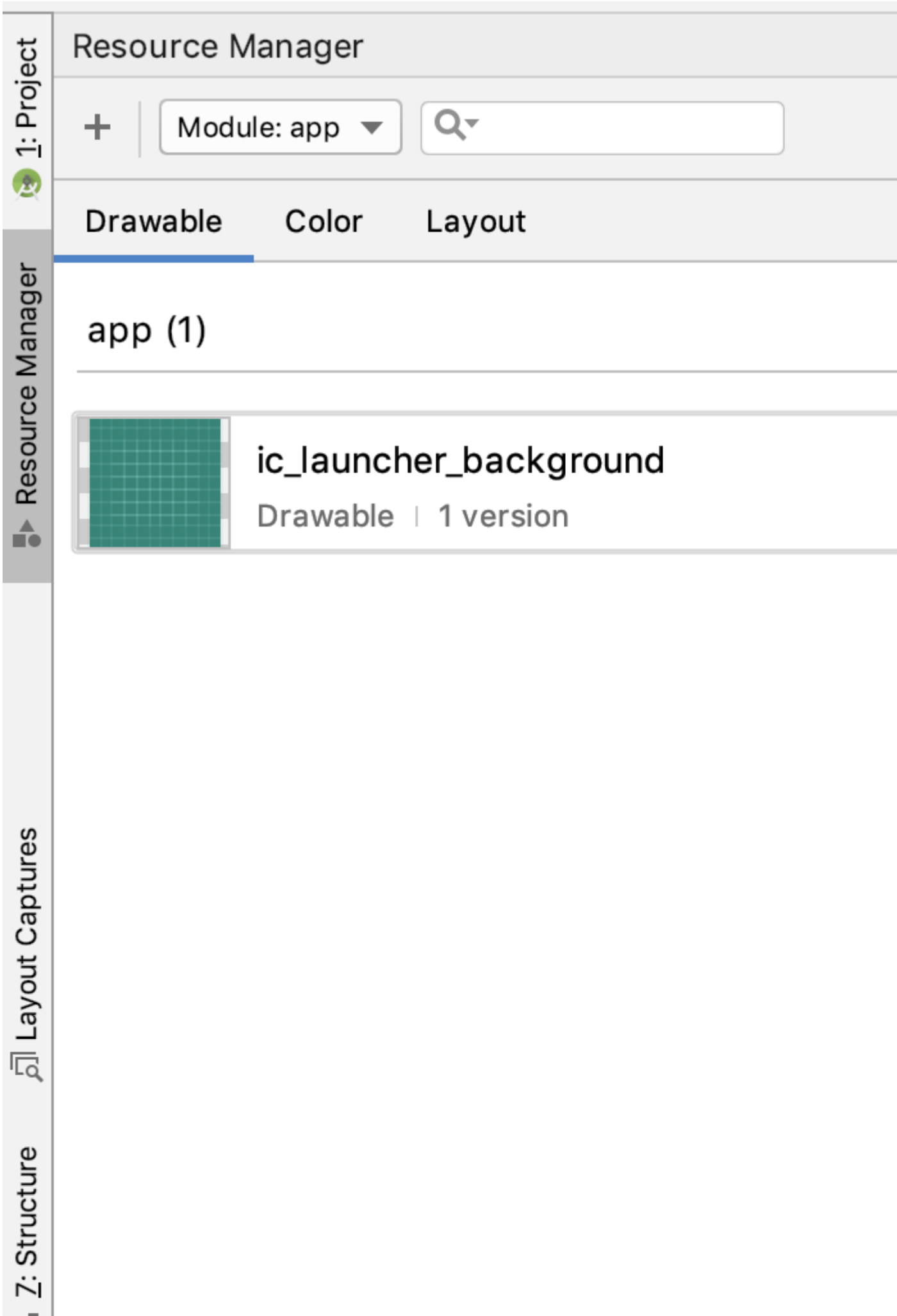
Setiap kali terjadi perubahan informasi di dalam build.gradle kita harus melakukan sinkronisasi terlebih dahulu. Tombol **sync now** akan muncul pada sebelah kanan

atas ketika terjadi perubahan.



Setelah proses sinkronisasi selesai maka akan muncul informasi pada log apakah proses sinkronisasi berhasil atau tidak.

Resource Manager


1: Project
Resource Manager
Layout Captures
Z: Structure
Resource Manager
Module: app
Drawable
Color
Layout
app (1)
ic_launcher_background
Drawable | 1 version

Resource Manager, berfungsi untuk memanajemen segala *resource* yang ada di proyek Anda seperti gambar, warna, layout dll. Jika Anda perhatikan struktur dari **res**, dengan menggunakan Resource Manager semua *resource* akan tampil di sini dan Anda bisa memanipulasinya sesuai kebutuhan Anda.

Useful Tools pada Android Studio

Android Studio menyediakan fasilitas yang *powerful* di bawah IntelliJ IDEA ini. Banyak *tools* milik Android yang membantu kita saat mengembangkan Aplikasi. Mari kita bahas *tools* yang sering digunakan dan manfaatnya.

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

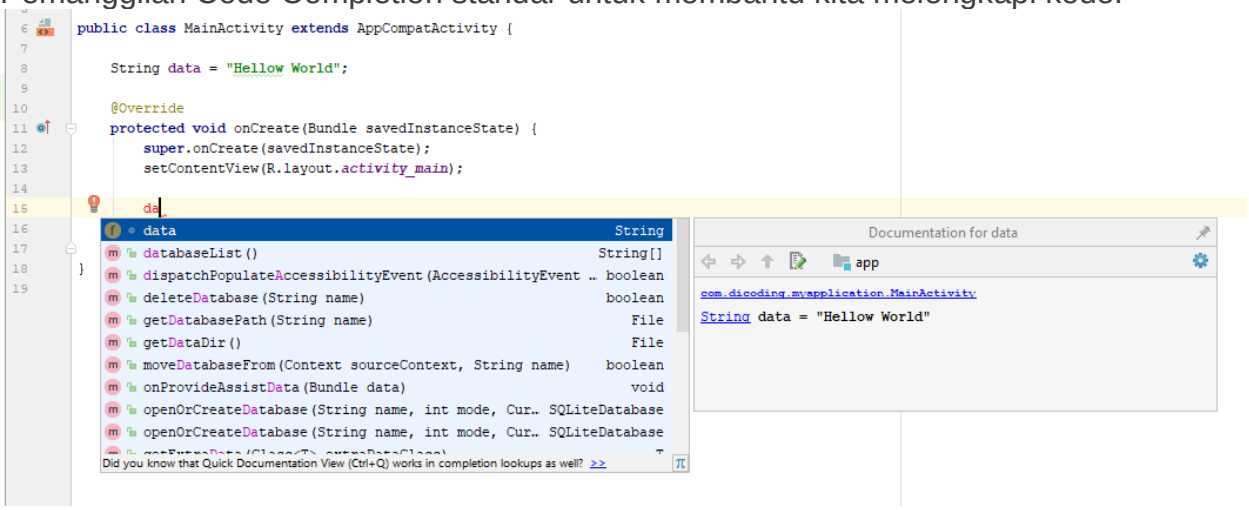
- **Ctrl+F9**
  Make project, build project.
- **Ctrl+Shift+F9**
  Melakukan kompilasi pada berkas, package atau modul.
- **Shift+F10**
  Run. Menjalankan aplikasi ke *emulator* atau *devices.*
- **Shift+F9**
  Debug. Menjalankan aplikasi ke *emulator* atau *devices* dalam mode **Debug**, biasanya untuk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

Code Completion

Untuk meminimalisir salah ketik (*typo*) dalam pemanggilan class, method hingga variabel sebaiknya kita memanfaatkan Code Completion di Android Studio. Terdapat dua (2) jenis Code Completion yang sering digunakan di Android Studio:

- **Basic Code Completion**

Ctrl+Space
Pemanggilan Code Completion standar untuk membantu kita melengkapi kode.



Ketika kita akan memanggil sebuah variabel, cukup ketikkan code completion di atas. Saran pun akan diberikan.

- **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.
Kode di bawah ini ditulis sebelum menggunakan shortcut:

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

    data = "mar

}
```

Kemudian kita menggunakan Statement Completion. Lihat apa yang terjadi!

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

    data = "mar";

}
```

Statement kita yang belum tuntas akan diselesaikan oleh Android Studio. Tentu hal ini akan mempercepat waktu kita dalam menggarap aplikasi.

Selengkapnya Anda dapat mempelajarinya di sini: https://www.jetbrains.com/help/idea/2016.1/code-completion-2.html

Style dan Formatting

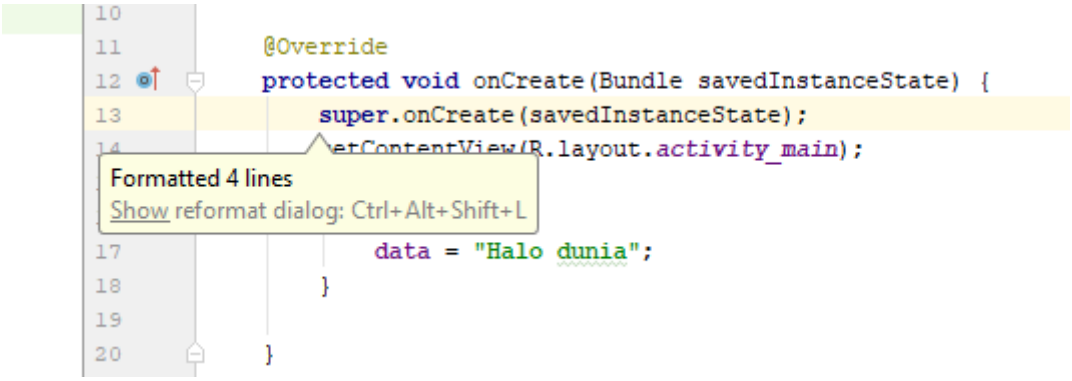
Gaya penulisan kode adalah seni dalam pemrograman. Kita memiliki *signature style* masing-masing, Semua tergantung pilihan kita sendiri. Tetapi kita tetap perlu memperhatikan bagaimana tata letak kode, apalagi bila suatu saat nanti kita membuat aplikasi bersama orang lain. Kode yang rapi itu enak dilihat dan memudahkan, baik kita maupun orang lain untuk membacanya.
Secara *default* Android Studio memberikan code style formatting untuk tata letak kode yang kita miliki. Untuk menyesuaikan setelan Code Style, klik **File > Settings > Editor > Code Style** (**Android Studio > Preferences > Editor > Code Style** pada Mac.)

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

   if(isTrue){
   data="Halo dunia";
   }

}
```

Bagaimana menurut Anda tentang kode di atas? Ya tidak ada yang salah. Namun Code Style berantakan dan tidak indah untuk dilihat.
Nah, kini kita akan melakukan kode *formatting* dengan menggunakan shortcut **Ctrl+Alt+L.**



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

    if (isTrue) {
        data = "Halo dunia";
    }

}
```

Hasilnya lebih baik, bukan?

Mungkin bila kode yang kita miliki sedikit, tidak terlalu berpengaruh. Tapi bila baris kode sudah mulai kompleks, *formatting code* seperti ini akan sangat membantu.

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.

Selengkapnya dapat kita jumpai di sini: https://developer.android.com/studio/write/sample-code.html

Keren kan? Jadi biasakan diri Anda menggunakan alat bantu dari Android Studio ini. Tak lain agar membantu dan mempercepat pembuatan aplikasi kita!