Modul Perkuliahan

Pemrogaman Mobile

Dr. Habibullah Akbar

Session 2

2019

SESSION OUTCOMES

1. Mahasiswa mengetahui dan melakukan *refreshment* / mempelajari secara mandiri mengenai konsep pemrograman berorientasi objek, Java dan XML sebagai persyaratan kuliah ini.
2. Mahasiswa dapat memahami prinsip kerja sistem operasi *mobile* Android
3. Mahasiswa dapat melakukan instalasi perangkat pengembangan aplikasi Android Studio
4. Mahasiswa dapat membuat Aplikasi sederhana

OUTLINE MATERI:

1. Persyaratan Mata Kuliah
2. Pendahuluan Sistem Operasi *Mobile* Android
3. Perangkat Pengembangan Aplikasi: Android Studio
4. Aplikasi Sederhana *Hello World*

ISI MATERI

1. Persyaratan Mata Kuliah

Sebelum membuat aplikasi Android, ada beberapa persyaratan yang perlu anda *refresh* atau pelajari yaitu

• Java Programming Language
• Object-oriented programming
• XML - properties / attributes

Pada kuliah lainnya, anda diharapkan telah memiliki pengetahuan mengenai ketiga persyaratan diatas. Prasyarat mata kuliah sebelumnya adalah Pemrograman Berorientasi Objek dan juga anda diharapkan memiliki dasar kemampuan pemrograman Java. Bahasa Java merupakan bahasa pemrograman yang digunakan untuk mendefinisikan aktifitas-aktifitas aplikasi Android yang akan dikembangkan. Jika diperlukan, silahkan pelajari lagi di https://www.w3schools.com/java/. Adapun pengetahuan mengenai XML akan diperlukan untuk mengatur tata letak layout dari tampilan aplikasi. Silahkan pelajari XML yang merupakan pengembangan Bahasa HTML di https://www.w3schools.com/xml/.

2. Pendahuluan Sistem Operasi *Mobile* Android

Android adalah sistem operasi seluler yang dikembangkan oleh Google. Android didasarkan pada modifikasi kernel Linux dan perangkat lunak *open source* lainnya, dan dirancang terutama untuk perangkat seluler layar sentuh seperti *smartphone* dan tablet. Selain itu, Google telah mengembangkan *platform* Android untuk perangkat lainnya seperti Android *TV* untuk televisi, Android *Auto* untuk mobil, dan *Wear OS* untuk jam tangan. Android menyediakan antarmuka pengguna khusus untuk masing-masing perangkat. Bahkan, varian Android juga digunakan pada konsol game, kamera digital, PC, dan alat elektronik lainnya.

Saat ini terdapat dua perusahaan utama yang mengisi ekosistem perangkat *smartphone* (perangkat yang dilengkapi fitur standar komputer, foto, video kamera), yaitu Android dan iOS. Namun Android digunakan pada lebih dari 80% dari semua pengguna smartphone sehingga menjadikan kemampuan membuat aplikasi Android menjadi satu *skill* yang layak untuk diasah dari sekian banyak *skill* lainnya yang diperlukan baik bagi industri maupun untuk usaha mandiri.

Selain kamera dan *microphone*, Android juga memiliki beberapa sensor untuk mendeteksi user dan aksinya. Konten dari perangkat *smartphone* dapat berputar sesuai dengan posisi perangkat. Android juga sudah dilengkapi dengan layanan berbasiskan lokasi untuk menunjukkan posisi perangkat didalam map.

Sebelum adanya Android, masyarakat menggunakan SonyEricsson, Nokia dan BlackBerry yang masih berbasiskan *keypad*. Hal ini menyebabkan pengetikan sedikit lebih lambat. Adapun

Android *smartphone*, menyediakan layer sentuh yang dapat digesek, diketuk dan dicubit. Selain itu terdapat juga fitur lainnya seperti virtual keyboard untuk emoji dan kapabilitas untuk sambungan ke perangkat lain melalui Bluetooth dan USB.

Kelebihan utama Android sebenarnya terletak pada sistem operasi mobile yang dibangun berbasiskan kernel Linux sehingga bersifat *open source*. Sehingga vendor-vendor *smartphone* dapat mengembangkan perangkat Android mereka sendiri. Saat ini juga terdapat sekitar 2.6 Juta aplikasi tersedia di Google *Play Store*.

2.1 Varian Android

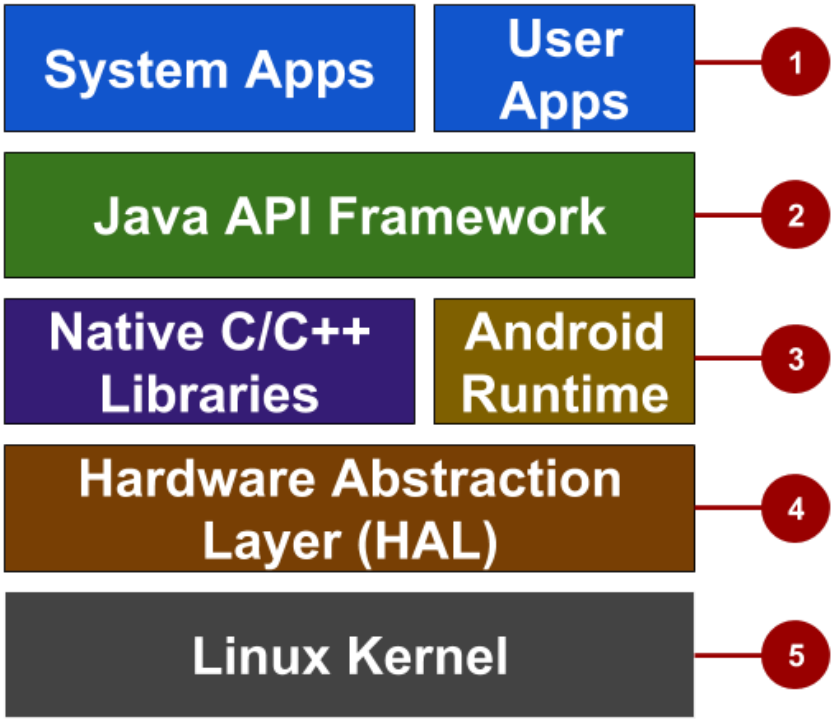
Perangkat Android komersil yang pertama dikeluarkan pada tahun 2008. Adapun versi terbaru saat ini adalah Android *Pie* yang dikeluarkan pada 2018 yang lalu. Table 2.1 menampilkan versi-versi Android sejak 2008 hingga 2018.

Table 2.1 Daftar versi Android sejak berdasarkan kronologis level API.

Codename	***Version***	***Released date***	***API level***
(No codename)	1.0	September 23, 2008	1
Petit Four	1.1	February 9, 2009	2
Cupcake	1.5	April 27, 2009	3
Donut	1.6	September 15, 2009	4
Eclair	2.0 – 2.1	October 26, 2009	5 – 7
Froyo	2.2 – 2.2.3	May 20, 2010	8
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18
KitKat	4.4 – 4.4.4	October 31, 2013	19 – 20
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22
Marshmallow	6.0 – 6.0.1	October 5, 2015	23
Nougat	7.0 – 7.1.2	August 22, 2016	24 – 25
Oreo	8.0 – 8.1	August 21, 2017	26 – 27
Pie	**9.0**	August 6, 2018	28

2.2 Arsitektur *Platform* Android

Android dapat dipandang sebagai tumpukan perangkat lunak yang berbasiskan *kernel Linux* yang dibuat untuk beragam jenis perangkat *mobile* dengan berbagai tipe *form-factor* (ukuran layar). Gambar 2.1 menunjukkan komponen-komponen utama dari *platform* Android: kernel Linux, *hardware abstraction layer*, *Android Runtime* (ART), C/C++ *libraries*, dan Java API.



Gambar 2.1 Lapisan dari tumpukan perangkat lunak Android.

Kernel Linux

Kernel linux (kernel inti) merupakan kumpulan program inti yang mengontrol segala aktifitas dari sistem operasi Android seperti *memory management*, *device management*, dan *process management*. Kernel inti juga mencakup *driver* yang digunakan untuk mengontrol *hardware* Android seperti *wifi*, *keypad*, *bluetooth*, *audio*, *memory*, USB dan sebagainya.

Hardware Abstraction Layer (HAL)

Lapisan ini (HAL) menyediakan jembatan bagi developer untuk mengakses *driver* daripada *hardware* diatas. Hal ini memudahkan vendor/produsen untuk melakukan kustomisasi pada perangkat Android yang mereka kembangkan.

Android Runtime (ART)

Perangkat Android versi 5.0 (Android *Lollipop*) keatas sudah dapat menggunakan kompilator ART (kompilasi yang hanya dilakukan satu kali yaitu saat aplikasi diinstal pada perangkat). Sehingga tidak memerlukan kompilasi yang berulang-ulang setiap kali aplikasi dijalankan.

Native C/C++ libraries

Walaupun kebanyakan aplikasi Android dikembangkan dengan bahasa Java, namun ada kalanya pengembang ingin menggunakan *libraries* asli yang ditulis dalam Bahasa C dan C++. Hal ini cocok digunakan saat membangun aplikasi yang memerlukan performa yang lebih baik (misal *game*) ataupun untuk mengakses sensor dengan lebih efisien.

Java API Framework

Seluruh fitur/perangkat dari sistem Android diatas hanya dapat diakses menggunakan API yang ditulis dengan bahasa Java. Biasanya, aplikasi yang dikembangkan berinteraksi secara langsung dengan komponen-komponen yang terdapat pada framework ini. Komponen-komponen ini mencakup:

- *View System*: sistem tampilan yang kaya dan dapat digunakan untuk membangun UI aplikasi, termasuk *lists*, *grids*, *text boxes*, *buttons*, dan *embeddable web browser*.
- *Resource Manager*: manajer yang menyediakan akses ke file-file yang digunakan oleh kode yang dibuat seperti *strings*, *graphics*, and *layout files*.
- *Notification Manager* yang memungkinkan aplikasi menampilkan peringatan khusus di status bar.
- *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan tumpukan untuk navigasi kembali ke aktifitas sebelumnya.
- Content provider yang memungkinkan aplikasi untuk berbagi data dengan aplikasi lainnya.

3. Perangkat Pengembangan Aplikasi: Android Studio

Pada kuliah ini, anda akan menggunakan IDE Android Studio sebagai aplikasi lingkungan pengembangan aplikasi terpadu yang juga akan digunakan untuk *debugging*. Android Studio dibangun di atas perangkat lunak *IntelliJ IDEA JetBrains*. Android Studio tersedia untuk diunduh pada sistem operasi berbasis Windows, macOS dan Linux. Android Studio dapat diunduh dari situs https://developer.android.com/studio. Note: perhatikan juga kebutuhan minimal sistem (*system requirement*) agar komputer anda dapat menginstal Android Studio, lihat bagian paling bawah pada situs tersebut.

Setelah mengunduh Android Studio, anda hanya membutuhkan beberapa langkah berikut ini

- Klik file .exe yang Anda unduh.
- Ikuti kotak dialog instalasi Android Studio dengan menekan Next dan install paket SDK jika diperlukan.

Untuk yang menggunakan Mac atau Linux, petunjuk instalasi Android Studio dapat dilihat di https://developer.android.com/studio/install?hl=id.

Pada sebagian sistem Windows (versi 32-bit), instalasi IDE tidak menemukan tempat JDK (Java Development Kit). Jika anda menemui masalah ini, anda harus mengatur variabel lingkungan yang menunjukkan lokasi yang benar. Caranya anda harus yaitu

Pilih menu **Start > Computer > System Properties > Advanced System Properties**. Lalu buka tab **Advanced > Environment Variables** dan tambahkan variabel sistem yang baru JAVA_HOME yang menunjuk ke folder JDK Anda, misalnya C:\Program Files\Java\jdk1.8.0_77.

Jika masih belum berhasil, coba anda ikuti petunjuk sesuai dengan situs http://www.kodingindonesia.com/cara-install-jdk/.

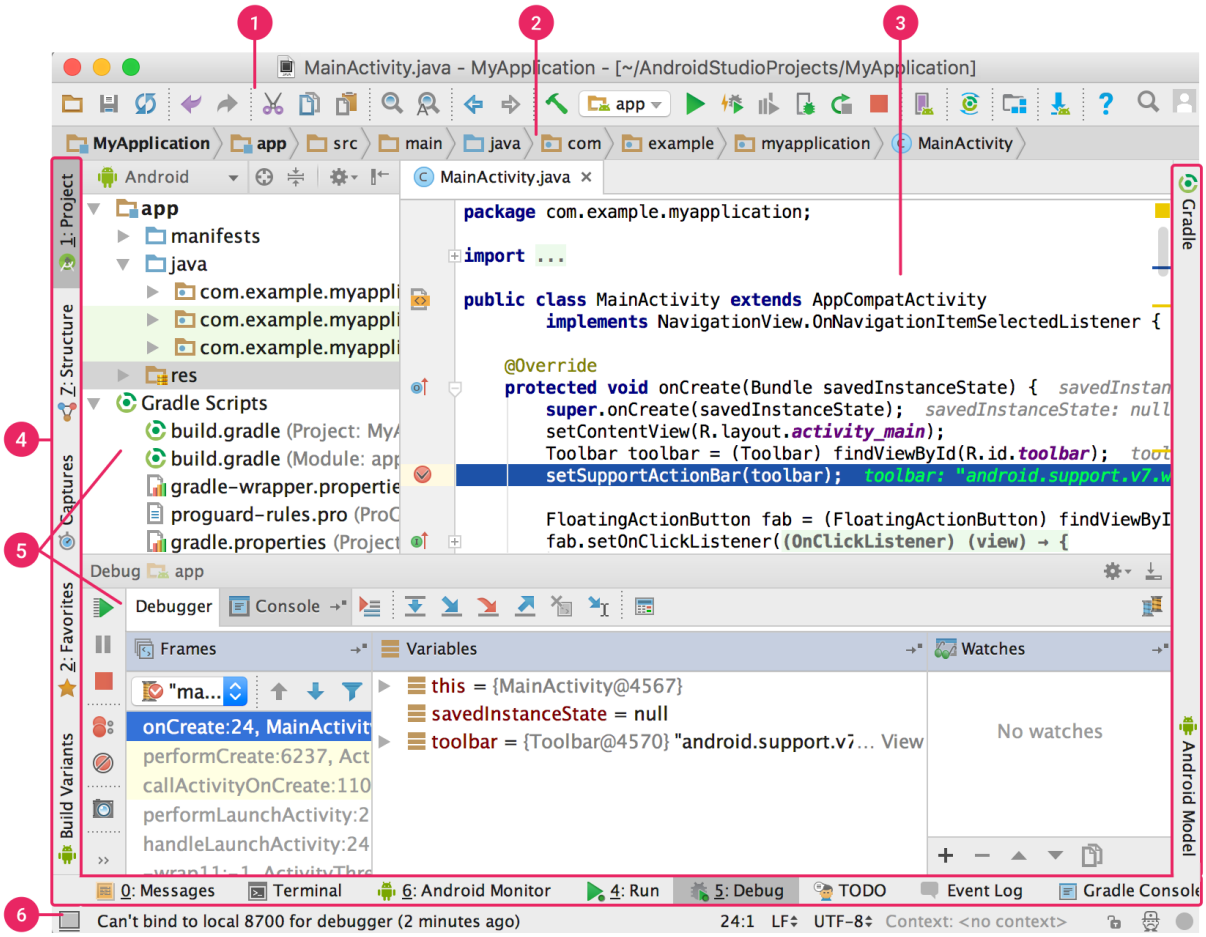
Setelah berhasil melakukan instalasi IDE Android Studio, anda perlu mengenal lebih dalam mengenai editor ini. Secara umum, IDE ini digunakan bukan hanya membuat kode, namun juga untuk menjalankan aplikasi, mencari *bug*, pengujian, monitoring performa dan pengaturan proyek aplikasi. Lebih spesifik lagi, editor Android Studio menawarkan fitur yang dapat meningkatkan produktivitas anda saat membangun aplikasi Android, seperti:

- Emulator yang efisien dan kaya fitur
- Lingkungan editor terpadu tempat untuk semua perangkat Android
- Fitur *Instant Run* untuk mendorong perubahan pada aplikasi Anda yang sedang berjalan tanpa harus membuat APK baru
- Integrasi dengan GitHub untuk membantu pembangunan fitur aplikasi umum dan *import* kode sampel
- Alat bantu *Lint* untuk menampilkan performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- Dukungan C++ dan NDK
- Dukungan bawaan untuk *Google Cloud Platform*, untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine*
- Sistem pembangunan berbasis *Gradle* yang fleksibel

3.1 Tampilan *User Interface* Android Studio

Jendela utama Android Studio terdiri dari beberapa area yang dijelaskan pada gambar dibawah.

1. Bagian **toolbar** memberikan berbagai pilihan, termasuk menjalankan aplikasi.
2. Bagian **navigation bar** membantu untuk menelusuri proyek dan membuka file yang akan diedit. Tampilan yang lebih kompak dari anatomi struktur aplikasi dapat dilihat di jendela **Project**.
3. Jendela **editor window** untuk membuat dan memodifikasi kode. Tergantung pada jenis file yang sedang dibuka, tampilan editor akan menyesuaikan.
4. Bagian **tool window bar** yang dapat memperluas atau menutup jendela-jendela yang tersedia.
5. Jendela **tool windows** memberikan akses tertentu seperti manajemen proyek, pencarian, dan kontrol versi.
6. Bagian **status bar** menampilkan status proyek dan IDE itu sendiri seperti peringatan atau pesan lainnya.



Untuk mempelajari IDE Android studio secara lebih mendetil, silahkan lihat di situs https://developer.android.com/studio/intro/index.html.

4. Aplikasi Sederhana Hello World

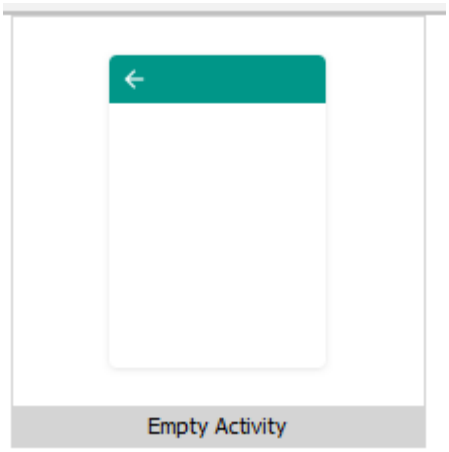
Setelah lebih memahami lingkungan Android, bagian berikut akan mengajarkan anda cara membangun aplikasi Android sederhana yang dapat menampilkan "Hello World" pada display *smartphone*. Aplikasi ini akan merupakan tradisi lama dalam belajar suatu bahasa pemrograman agar programer dapat menangkap sintak-sintak dasar.

4.1 Membuat Proyek Aplikasi Baru

Ikuti langkah-langkah berikut untuk membuat sebuah aplikasi Android

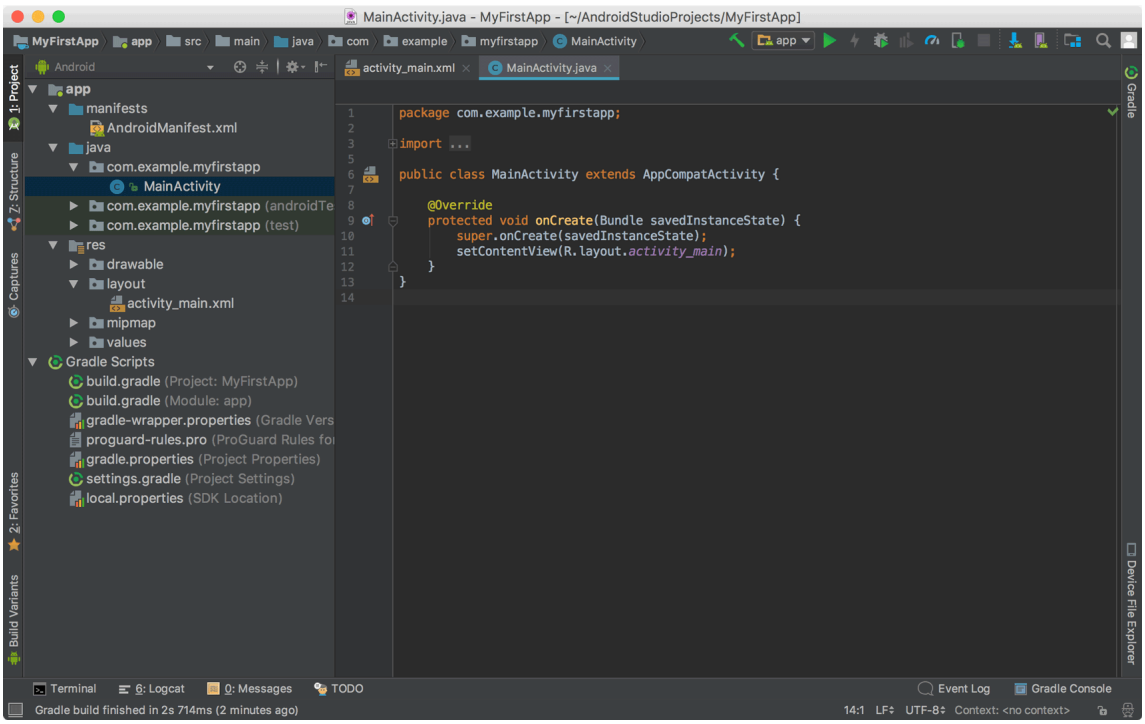
1. Pertama-tama, jalankan Android Studio dan klik **Start a new Android Studio project**.
2. Pada jendela **Create New Project**, masukkan nilai berikut:
   - **Application Name**: "Aplikasi Pertama Saya"
   - **Company Domain**: "contoh.com"

3. Klik **Next.**
4. Pada layar **Target Android Devices**, tidak perlu ubah nilai *default* dan klik **Next**.
5. Pada layar **Add an Activity to Mobile**, pilih **Empty Activity** dan klik **Next**.



6. Pada layar **Configure Activity**, biarkan nilai *default* dan klik **Finish**.

Setelah Android Studio akan memproses konfigurasi aplikasi anda (lama waktu bergantung komputer yang digunakan), maka akan muncul tampilan berikut



4.2 Anatomi Struktur Proyek Aplikasi Android

Sebelum menjalankan aplikasi ini, , anda wajib mempelajari anatomi dari struktur proyek aplikasi "Hello World" ini. Hal ini menjadi penting karena sebuah proyek di Android Studio berisi gambaran umum tentang komponen-komponen utama di dalam proyek anda, seperti kode, aset, informasi pengujian dan konfigurasi kompilasi.

Saat Anda memulai proyek aplikasi baru, Android Studio secara otomatis membuat semua struktur yang diperlukan untuk semua file yang akan dibutuhkan dan membuatnya terlihat di jendela *Project* di sisi kiri IDE. Note: jika jendela ini tidak nampak, maka klik **View > Tool Windows > Project**).

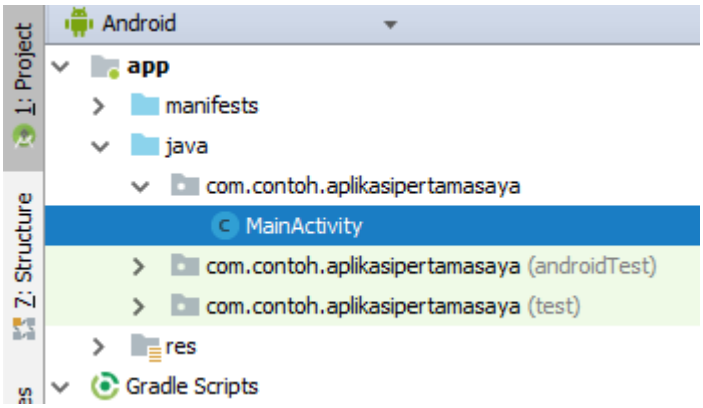
File-file yang penting untuk diperhatikan mencakup

- **app > manifests > AndroidManifest.xml**

File *AndroidManifest* berisi informasi dasar mengenai aplikasi bagi sistem operasi, Google Play and kebutuhan kompilasi dari kode menjadi APKs.

- **app > java > com.example.myfirstapp > MainActivity**

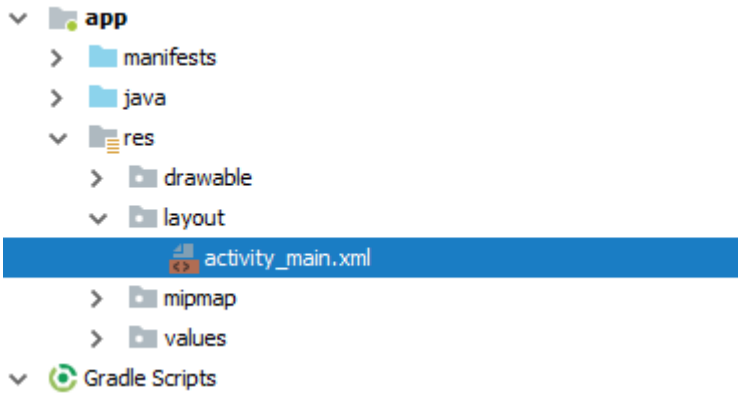
Folder **java** berisi file-file dari kode Java yang digunakan untuk membuat aktifitas pada aplikasi. Secara *default*, IDE akan membuatkan file *MainActivity* secara otomatis.



Saat Anda menjalankan aplikasi, sistem akan memulai aktivitas berdasarkan file *MainActivity* dan menampilkan *layout* (tata letak tampilan) yang didefinisikan pada file XML berikut.

- **app > res > layout > activity_main.xml**

Folder **res** berisi semua sumber daya yang bukan kode utama, utamanya adalah tata letak layout yang didefinisikan pada file *activity-main.xml*. Selain itu, folder ini juga berisi file string.xml yang menyimpan variabel string global, dan juga file grafis (folder *drawable*).



File *activity-main.xml* mendefinisikan tata letak *layout* agar pengguna aplikasi dapat melakukan aktivitas sesuai dengan tampilan *User Interface* (UI).

Pada sesi ini, file inilah yang paling penting untuk anda pahami. Secara *default*, **Empty Activity** sudah mengatur elemen *TextView* dengan teks "Hello world!". Artinya ketika dijalankan, aplikasi akan menampilkan teks tersebut. Perhatikan isi dari file tersebut.

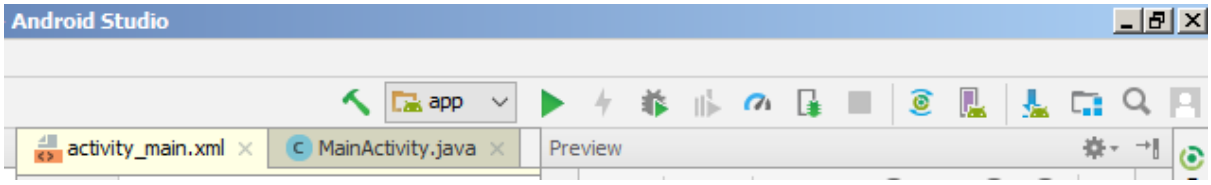
activity_main.xml | MainActivity.java

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/ap
    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=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

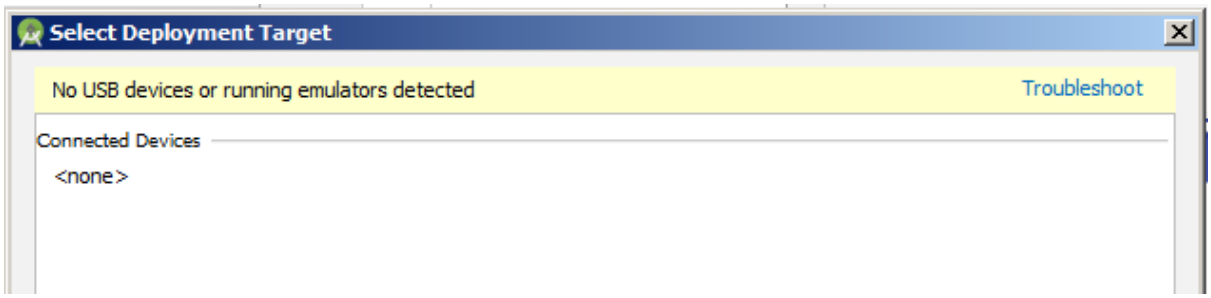
- **Gradle Scripts > build.gradle**

Terdapat dua file dengan nama yang sama yaitu *build.gradle*: satu untuk proyek dan satu untuk modul aplikasi. Sedangkan, file *build.gradle* untuk aplikasi berada pada folder *app*. Setingannya hanya berlaku pada modul *app* saja. File *build.gradle* yang untuk proyek berada pada folder *root* sehingga setingan konfigurasi akan berlaku untuk setiap modul pada proyek aplikasi

Setelah memahami anatomi, anda dapat menjalankan aplikasi dengan menekan tombol *run app* ▶ berbentuk panah hijau pada bagian kanan atas IDE.



IDE akan menampilkan jendela *Select Deployment Target*. Anda akan mendapatkan bahwa belum ada target *device* dari aplikasi.



Untuk mendapatkan target *device*, anda dapat menggunakan *smartphone* Android ataupun *emulator*.

4.2 Menjalankan aplikasi pada *smartphone* Android

Siapkan *smartphone* Android anda sebagai berikut

1. Hubungkan *smartphone* dengan computer menggunakan kabel USB. Jika Anda mengembangkan pada versi Windows 32-bit, anda mungkin perlu menginstal *driver* USB yang sesuai untuk perangkat anda.
   (cari *driver* yang sesuai disini https://developer.android.com/studio/run/oem-usb.html).

2. Berikutnya, aktifkan **USB debugging** di **Developer options** dengan cara sebagai berikut.

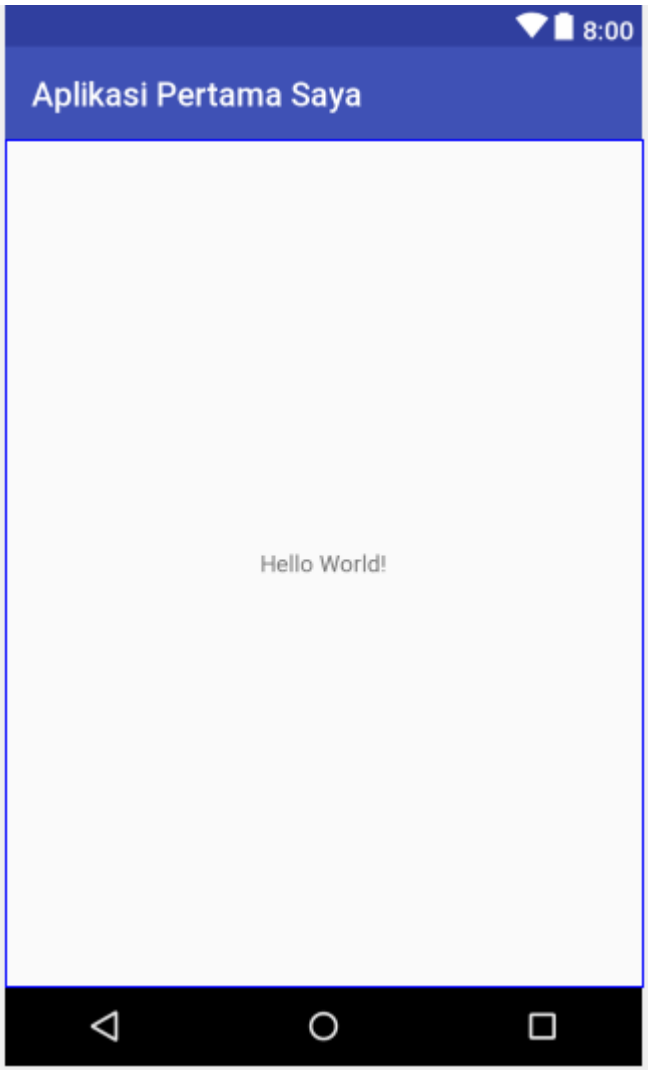
   - Buka Pengaturan **Settings**
   - Pilih **System** (untuk Android 8.0 keatas)
   - Geser ke bawah dan pilih **About phone**
   - Geser ke bawah dan ketuk **Build number** 7 kali.
   - Kembali ke layar sebelumnya untuk menemukan **Developer options** di dekat bagian bawah.
   - Buka kembali **opsi Pengembang**, lalu geser ke bawah untuk menemukan dan mengaktifkan **USB debugging**.

Setelah selesai mengaktifkan **USB debugging**, tekan tombol *run app* lagi.

Jalankan aplikasi di *smartphone* anda sebagai berikut:

Pada jendela *Select Deployment Target*, pilih *smartphone* anda lalu klik OK.

Android Studio akan menginstal aplikasi pada *smartphone* anda yang sudah terhubung dan memulainya. Anda seharusnya sekarang sudah berhasil menampikan aplikasi Android pertama anda yaitu "Hello World!" yang berjalan pada *smartphone* Anda.



Selain menggunakan device *smartphone*, anda juga dapat menggunakan *emulator* yang sudah tersedia pada Android Studio. Untuk lebih detil, silahkan ikuti panduan di https://developer.android.com/training/basics/firstapp/running-app#RealDevice.

Simpulan

Sesi ini telah menjelaskan beberapa dasar-dasar dalam pemrograman Android. Pertama, anda diingatkan untuk melakukan *refreshment* / mempelajari secara mandiri mengenai konsep pemrograman berorientasi objek, Java dan XML sebagai persyaratan kuliah ini. Kemudian, sesi ini juga membahas fitur dan versi dari Android. Setelah itu, anda diminta untuk melakukan instalasi aplikasi lingkungan pengembangan aplikasi Android yang terpadu (IDE). Terakhir, anda telah belajar membuat Aplikasi sederhana "Hello World" dan melakukan setingan agar aplikasi tersebut dapat ditampilkan pada *smartphone* anda. Pada sesi berikutnya, anda akan mempelajari bagaimana membuat aplikasi dengan tampilan layout yang lebih kompleks.