

MODUL PRAKTIKUM PEMROGRAMAN UNTUK PERANGKAT BERGERAK 1

Indra Azimi, S.T., M.T.

Reza Budiawan, S.T., M.T., OCA

Cahyana S.T., M.Kom.



D3 Rekayasa Perangkat Lunak Aplikasi
Telkom University

Daftar Isi

Modul 01: Pengenalan Android Studio	2
1. Overview	2
2. Getting Started	2
3. Task	2
3.1. Membuat Project Android “Hello World”	2
3.2. Mengetahui Struktur Folder dan File di Android Studio	6
3.3. Mengetahui Tampilan Aplikasi Android	8
3.4. Mengetahui Kode Activity Aplikasi Android	8
3.5. Menjalankan Aplikasi dengan Smartphone	9
3.6. Menjalankan Aplikasi dengan Emulator	11
3.7. Mengekspor Project untuk Dikirim	13
4. Summary	14
5. Challenge	14

Modul 01: Pengenalan Android Studio

1. Overview

Di sini kita akan memulai perjalanan untuk menjadi seorang Android Programmer. Langkah awal untuk mencapai ini adalah dengan mengenal Integrated Development Environment (IDE) yang umum digunakan untuk membangun aplikasi Android, yaitu Android Studio. Kita akan membangun aplikasi sederhana yang menampilkan teks “Hello World” di layar smartphone masing-masing. Aplikasi ini kemudian akan kita jalankan menggunakan smartphone/emulator.

2. Getting Started

Pertama, download terlebih dahulu Android Studio di link: <https://developer.android.com/studio>. Selanjutnya install di komputer/laptop masing-masing dengan mengikuti petunjuk yang tampil saat proses instalasi. Proses download dan install ini mungkin membutuhkan waktu yang agak lama jika koneksi internetmu terbatas dan spesifikasi komputer/laptop kurang memadai. Jadi, sediakan waktu yang cukup dan cobalah untuk bersabar jika itu terjadi 😊

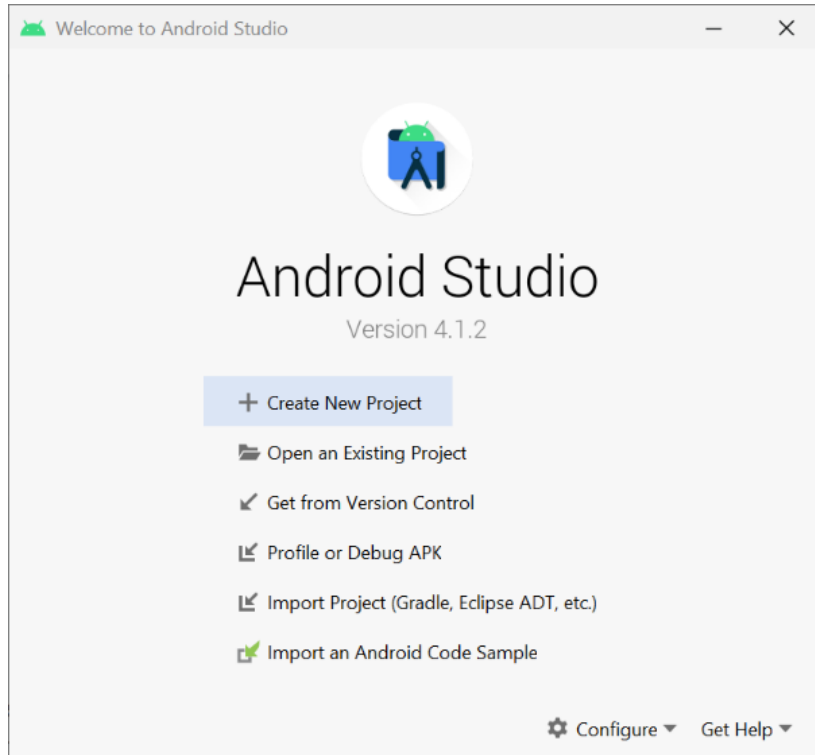
Catatan: Jika komputer/laptopmu tidak memungkinkan untuk menggunakan Android Studio karena spesifikasi yang terlalu rendah, kamu bisa membuat aplikasi Android menggunakan command line tools. Kode program dapat diketik menggunakan teks editor seperti Notepad++, compile dengan menggunakan Command Prompt, lalu jalankan di smartphone/emulator. Silahkan dipelajari di <https://developer.android.com/studio/build/building-cmdline>

3. Task

3.1. Membuat Project Android “Hello World”

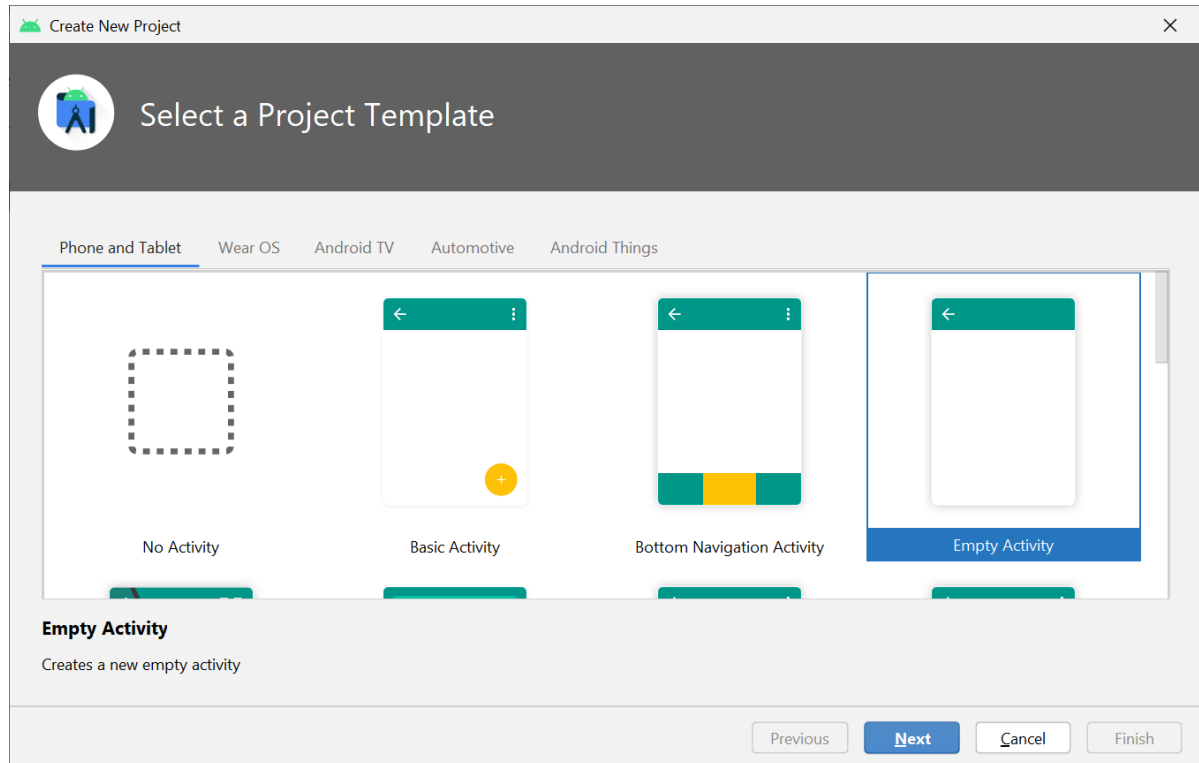
Pada task ini kita akan memulai membuat aplikasi Android pertama kita di mata kuliah ini. Langkah awal adalah, jalankan Android Studio. Versi yang digunakan di mata kuliah kita minimal versi 4.1.

Tampilan awal seperti di bawah ini. Terdapat beberapa pilihan dengan penjelasan sebagai berikut:



1. Create New Project: Membuat project baru. Umumnya, sebuah aplikasi Android dibuat dari 1 project. Jadi jika ingin membuat aplikasi baru, pilih pilihan ini.
2. Open an Existing Project: Membuka project yang telah dibuat sebelumnya.
3. Get from Version Control: Jika project sebelumnya telah dibuat dan disimpan di version control system (VCS) repository online, project tersebut dapat dibuka melalui menu ini. Akan dibahas lebih lanjut pada modul berikutnya.
4. Profile or Debug APK: memilih APK yang dihasilkan sebelumnya untuk kemudian dilakukan aksi profiling/debugging.
5. Import Project (Gradle, Eclipse ADT, etc): membuka project yang telah dibangun sebelumnya pada environment yang berbeda selain Android Studio.
6. Import an Android Code Sample: membuka project berupa contoh kode yang telah disediakan Android Studio.
7. Configure: melakukan konfigurasi seperti SDK, AVD, setting, update Android Studio, dan lainnya.
8. Get Help: mendapatkan bantuan terkait Android Studio, melihat versi, dan lainnya.

Kali ini, pilih “Create New Project”. Pemilihan menu ini akan memunculkan pilihan aplikasi yang dapat kita buat beserta template-nya. Pilih “Empty Activity” untuk menggunakan template sederhana yang telah disediakan Android Studio.

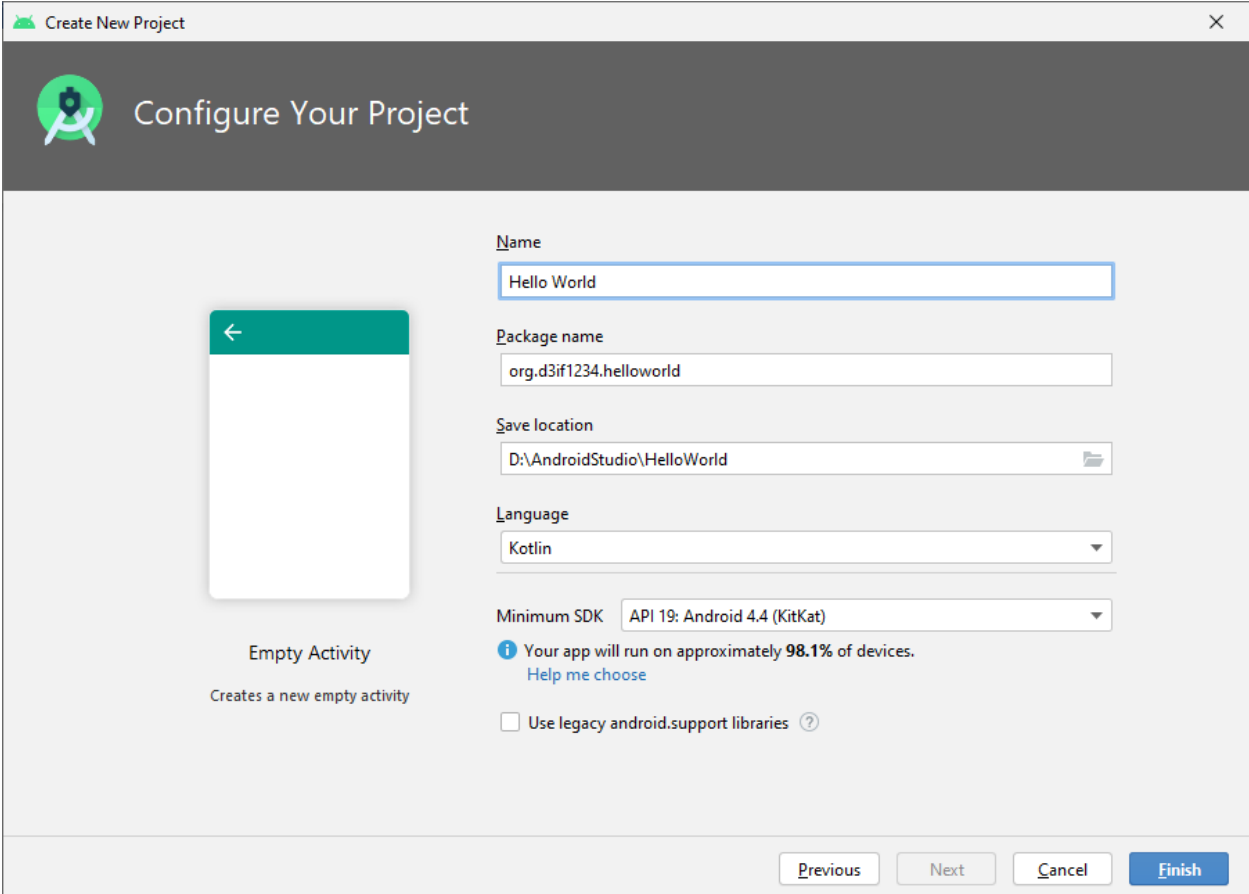


Berikutnya, tekan tombol Next. Akan muncul windows “Configure Your Project”. Lakukan konfigurasi terhadap project yang dibuat.

1. Nama project: merupakan nama dari project Android Studio. Nama ini merupakan nama dari aplikasi yang kita bangun. Isi dengan “Hello World”.
2. Nama package: merupakan package dari aplikasi yang dibuat. Nama package ini, biasanya akan digunakan sebagai ID aplikasi. Oleh karena itu, nama package harus unik, sebab digunakan untuk mengidentifikasi aplikasi pada device dan juga di Google Play Store. Untuk menjamin keunikan tadi, di mata kuliah ini, **WAJIB** menggunakan format package “org.d3ifxxx.helloworld” dimana xxxx adalah 4 digit NIM terakhir kalian.
3. Save location: adalah tempat penyimpanan project dari Android Studio.

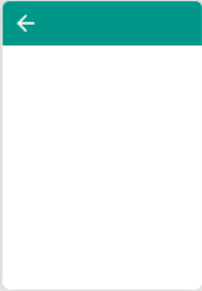
4. Language: merupakan bahasa yang digunakan untuk membangun aplikasi Android. Ada 2 pilihan bahasa yang tersedia, yaitu Kotlin dan Java. Kotlin merupakan first language untuk membangun aplikasi Android, dan bahasa ini yang akan kita gunakan di mata kuliah Pemrograman untuk Perangkat Bergerak 1 dan 2.
5. Minimum SDK: adalah versi SDK yang dapat meng-install aplikasi yang kita bangun. Mengubah hal ini akan berdampak pada jumlah persentase device yang dapat meng-install aplikasi. Pilihlah API 19 (Kitkat). Akan terlihat aplikasi kita dapat dijalankan di 98,1% device, dan terdapat 1,9% smartphone dengan sistem operasi JellyBean ke bawah tidak dapat meng-install aplikasi kita.

Terdapat satu pilihan lagi, yaitu “Use legacy android.support.libraries”. Sesuai namanya, jika kita mencentang pilihan ini artinya kita akan menggunakan library yang bersifat legacy (pusaka). Sebisanya mungkin, gunakan library terbaru saat membangun aplikasi. Jadi pilihan ini bisa diabaikan. Pastikan tampilanmu telah sama dengan gambar berikut, selanjutnya klik “Finish”.



Create New Project

Configure Your Project

 Empty Activity
Creates a new empty activity


Name
Hello World


Package name
org.d3if1234.helloworld

Save location
D:\AndroidStudio\HelloWorld

Language
Kotlin

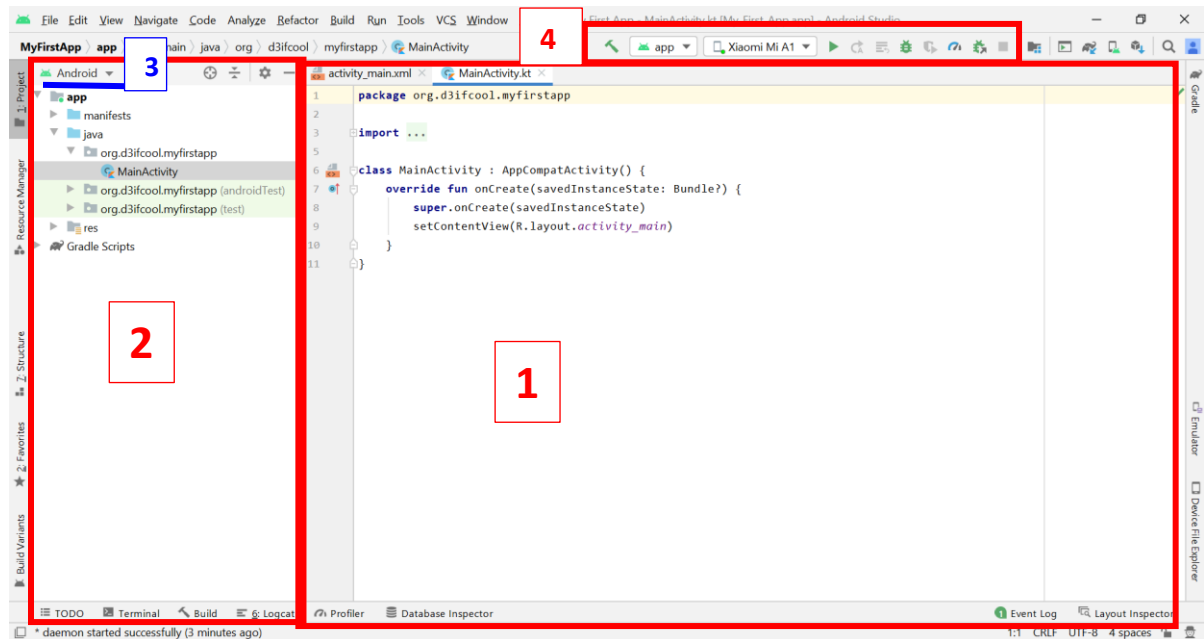
Minimum SDK
API 19: Android 4.4 (KitKat)

 Your app will run on approximately **98.1%** of devices.
[Help me choose](#)

☐ Use legacy android.support.libraries 

Previous Next Cancel Finish

Jika ini merupakan pertama kalinya kita membuat project baru di Android Studio, maka akan ada download yang dilakukan sebelum IDE terbuka sempurna. Pastikan internet dalam keadaan aktif. Indikator loading, dapat dilihat di bagian bawah dari Android Studio. Jika sudah selesai, IDE menunjukkan tampilan berikut:

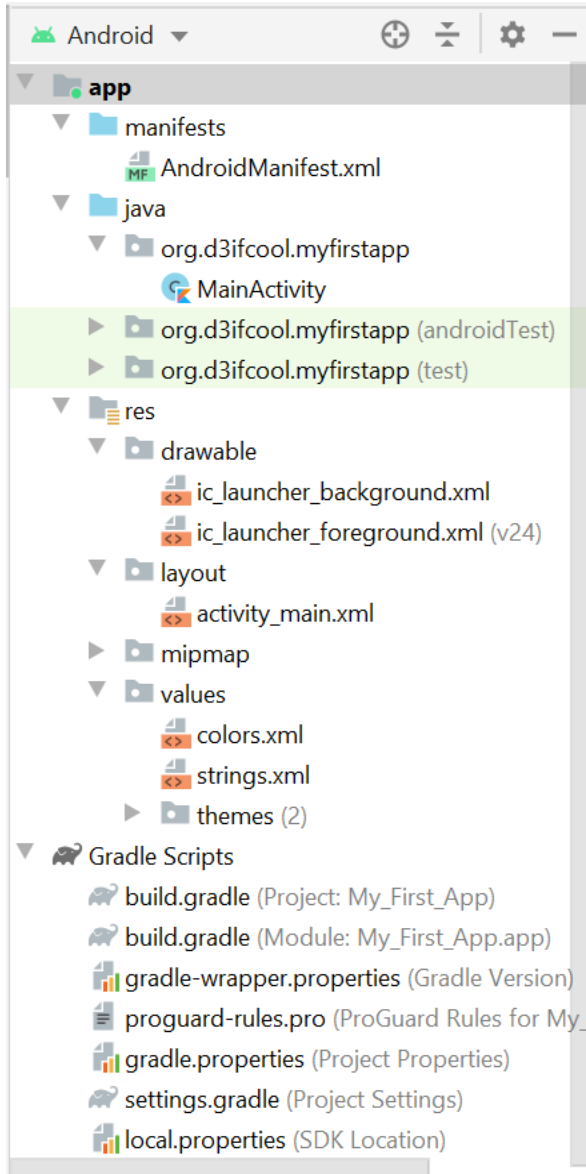


Mari membiasakan diri untuk mengenal GUI dari Android Studio. Paling tidak ketahuilah 4 area ini:

1. Area 1: merupakan area editor. Di sini kita akan mengetikkan kode untuk membangun aplikasi.
2. Area 2: merupakan area file. Pada area ini, kita dapat memilih file apa yang akan di-edit.
3. Area 3: area yang digarisbawah ini merupakan pilihan bagaimana struktur folder diperlihatkan. Defaultnya, pilihan "Android" yang dimunculkan. Selain pilihan ini, ada juga pilihan "Project" yang memperlihatkan bagaimana struktur file jika dilihat melalui File Explorer.
4. Area 4: merupakan area yang harus diperhatikan ketika ingin menjalankan aplikasi. Di sini juga terlihat perangkat yang sudah terhubung ke komputer/laptop dan dapat di-install aplikasi.

3.2. Mengetahui Struktur Folder dan File di Android Studio

Pada langkah sebelumnya, kita telah membuat aplikasi pertama kita dan siap untuk di-run di smartphone masing-masing atau emulator. Akan tetapi sebelum masuk ke tahap menjalankan, ada baiknya kita mengenal struktur folder di Android Studio.

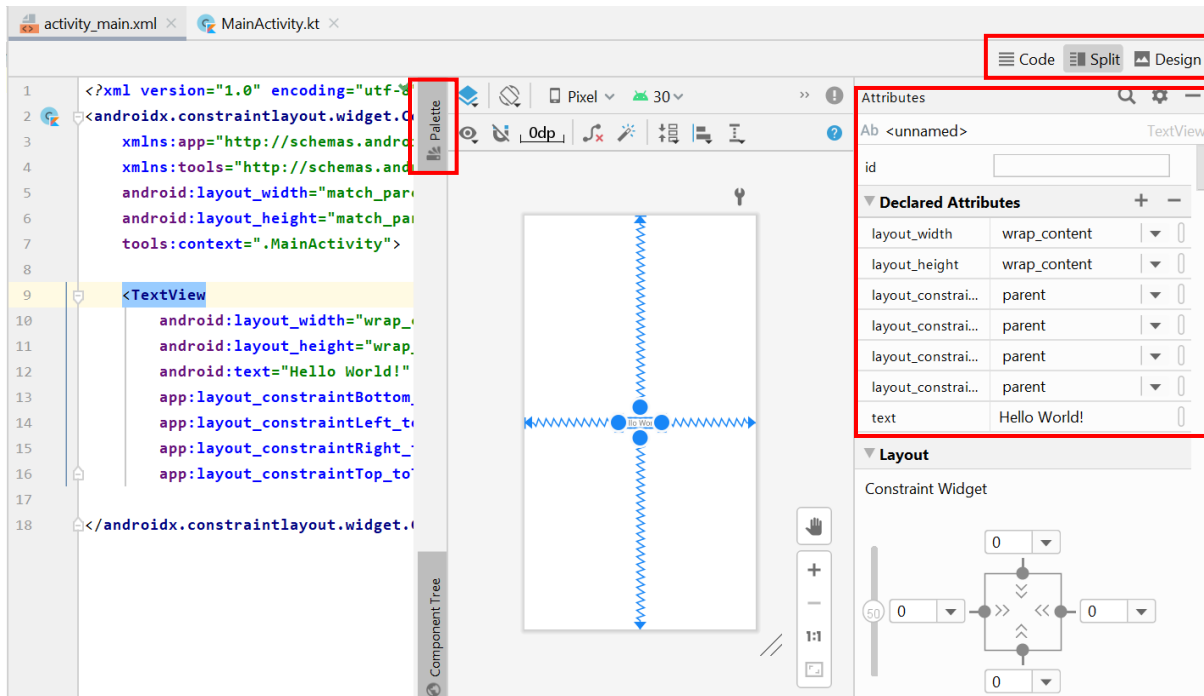


Gambar di samping merupakan struktur file yang akan sering kita temui saat membangun aplikasi Android. Beberapa yang penting diketahui sebagai dasar untuk modal dalam membangun aplikasi ke depannya yaitu:

1. app: merupakan folder module dari sebuah project. Satu project dapat memiliki beberapa module. Module app merupakan defaultnya. Di mata kuliah semester ini, 1 project Android Studio menggunakan 1 module app saja.
2. manifest: Pada folder ini terdapat AndroidManifest.xml. Setting dari aplikasi Android terdapat di sini, seperti icon yang digunakan, screen yang ditampilkan (activity), theme yang diterapkan, dan lainnya.
3. java: folder ini mengandung kode yang ditulis menggunakan bahasa Java atau Kotlin. Kode ini merupakan bentuk buah pikir programmer dalam membuat aplikasi.
4. res: Folder res (R) memiliki beberapa subfolder yang menyimpan resource dari aplikasi Android yang kita bangun.
5. drawable: merupakan subfolder dari /res untuk menyimpan gambar yang digunakan.
6. layout: subfolder ini mengandung tampilan yang diperlukan untuk aplikasi
7. mipmap: digunakan untuk menampung resource gambar sama seperti drawable, akan tetapi dikhususkan untuk launcher icon.
8. values: subfolder dari res ini mengandung beberapa file seperti colors.xml (resource warna yang digunakan) atau strings.xml (teks yang digunakan untuk aplikasi).
9. build.gradle: merupakan file untuk setting dari aplikasi. Ada 2 level pada build.gradle, project dan module. Pada level project, terdapat setting seperti versi Kotlin yang digunakan, classpath, repo, dan lainnya. Sedangkan pada level module, terdapat setting applicationId, minimum SDK, dependency, dan lainnya.

3.3. Mengetahui Tampilan Aplikasi Android

Tampilan aplikasi Android dapat dibuat melalui kode Kotlin atau melalui layout yang berada pada folder /res/layout. Layout ini dibuat menggunakan markup language XML, yang dapat dilihat dari 3 cara: design, code, dan gabungan keduanya. Pada bagian design, kita dapat menambahkan komponen yang ada di sebuah aplikasi seperti button, text field dan lainnya dengan cara drag-and-drop. Setiap komponen dapat diatur juga tampilannya melalui pengaturan di bagian Attributes.



Dalam mengatur tampilan, perlu diperhatikan juga cara menempatkan komponen tersebut dengan konsep layouting. Ada beberapa layout yang umum digunakan seperti: Linear Layout, Constraint Layout, atau Frame Layout. Hal ini akan kita kenal seiring pembuatan aplikasi di modul-modul berikutnya. Cara memposisikan komponen yang tidak tepat menghasilkan aplikasi yang tidak baik bagi user untuk digunakan.

3.4. Mengetahui Kode Activity Aplikasi Android

Activity merepresentasikan single screen (tampilan layar) kepada pengguna. Activity ini mirip seperti sebuah page saat membangun aplikasi web, atau JFrame saat membangun aplikasi menggunakan Swing Java. Tampilan aplikasi Android dapat dibentuk dengan membuat activity atau fragment. Akan tetapi, kali ini kita hanya membahas activity saja.

Saat membuat project baru menggunakan template “Empty Activity” sebelumnya, akan terbentuk sebuah activity (MainActivity.kt) dan tampilannya (activity_main.xml). Bahasa yang digunakan untuk activity di mata kuliah ini adalah bahasa pemrograman Kotlin (*.kt). Pada file ini, kita dapat memberikan behaviour untuk klik tombol, mengambil nilai pengguna dari text field, dan lainnya.

Kode default yang dibentuk sebelumnya adalah sebagai berikut:

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

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Pada kode di atas, kita membuat sebuah class bernama MainActivity yang meng-inherit sifat dari AppCompatActivity. Simbol double colon/titik dua (:) pada bahasa Kotlin, memiliki sifat yang sama dengan “extends” atau “implements” pada bahasa pemrograman Java. Pada class ini, terdapat sebuah method onCreate. Method ini dijalankan ketika aplikasi Android dijalankan.

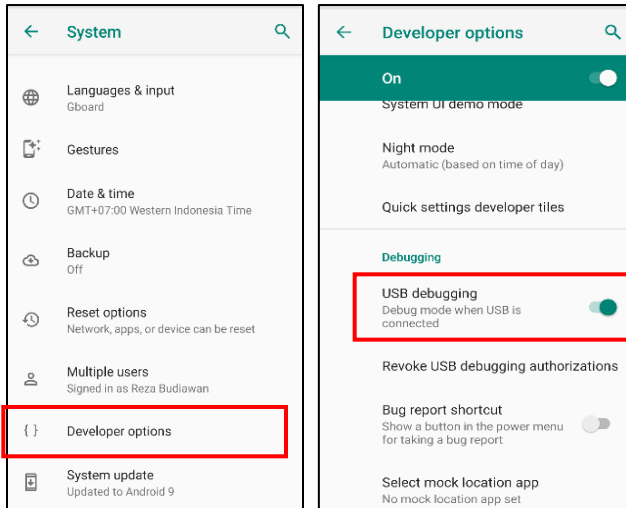
Pada method onCreate terlihat bahwa ada “setContentView” yang berarti, ketika MainActivity menggunakan tampilan dari layout “R.layout.activity_main”. Dengan kata lain, layout yang digunakan berada pada folder /res/layout/activity_main.xml. Ketika dijalankan, aplikasi menampilkan tulisan “Hello World” seperti yang ditampilkan pada activity_main.xml. Menghapus kode “setContentView” akan menghasilkan tampilan kosong pada smartphone ketika aplikasi dijalankan.

3.5. Menjalankan Aplikasi dengan Smartphone

Aplikasi Android merupakan aplikasi yang berjalan di smartphone dengan sistem operasi Android. Untuk itu, kita memerlukan smartphone Android untuk menjalankan aplikasi yang sebelumnya telah dibuat. Pastikan 2 hal ini sebelum menjalankan aplikasi di smartphone:

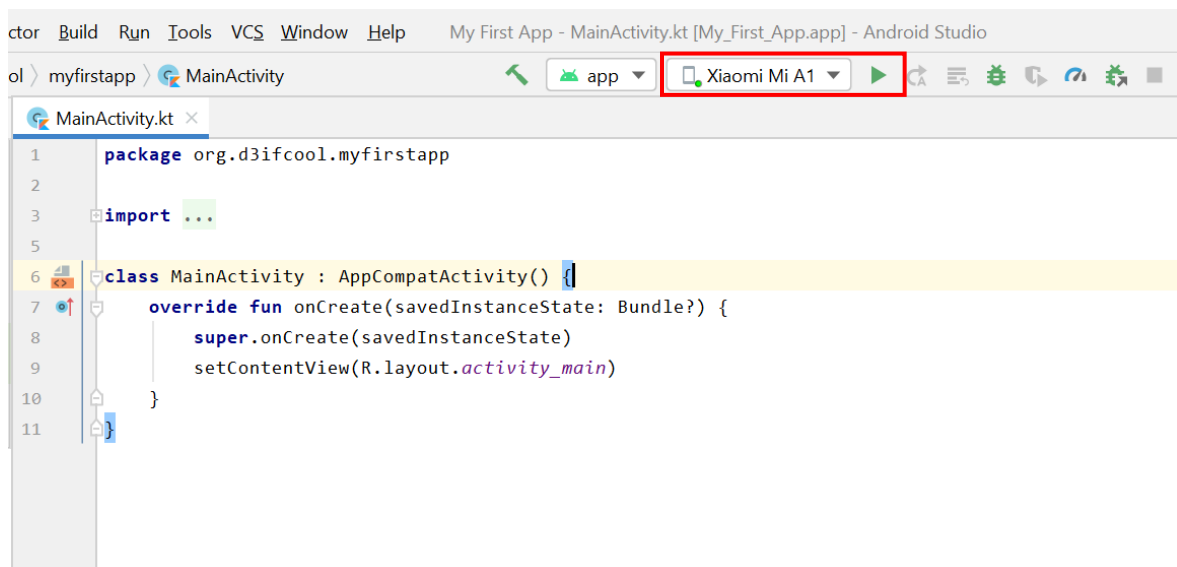
1. Smartphone terhubung ke PC dan dapat melakukan transfer data
2. Mode USB debugging dalam keadaan aktif

Jika mode USB debugging belum aktif, langkah pertama yang harus dilakukan adalah **mengaktifkan developer options**. Menu ini biasanya dapat ditemui di Settings → System → Advanced → Developer Options. Atau pada Settings → Additional Settings → Developer Options. Setiap brand memiliki struktur menu yang berbeda, jadi hal ini disesuaikan dengan smartphone masing-masing.

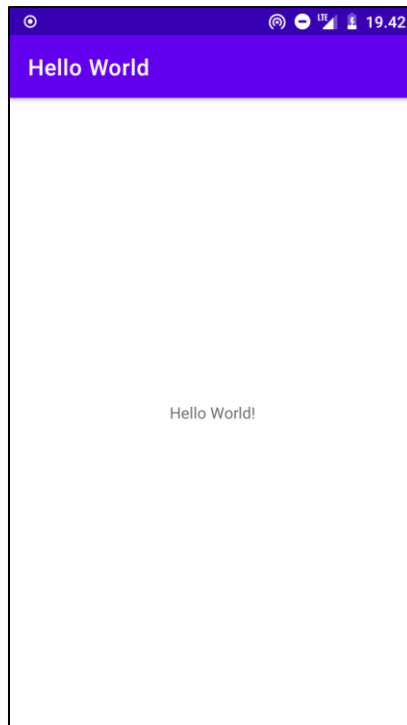


Jika menu ini tidak dapat ditemui, artinya mode ini sedang dalam keadaan tidak aktif. Mengaktifkan mode developer options dilakukan dengan menekan Build number sebanyak 7x. Caranya, masuk ke Settings → System → About phone → cari Build number → tekan 7x hingga muncul tulisan yang memberitahu akan mengaktifkan developer options.

Jika menu ini sudah aktif, ketika build number ditekan, akan muncul tulisan “No need, you are already a developer”. Melalui menu developer options, aktifkan USB debugging mode. Mode ini akan memungkinkan kita sebagai developer untuk melakukan instalasi di smartphone. Hubungkan smartphone ke PC, dan akan muncul permission di smartphone (jika hal ini pertama kali dilakukan), klik OK. Setelahnya, pilihan smartphone akan muncul di layar atas. Jalankan aplikasi dengan meng-klik tombol segitiga hijau di sebelah brand smartphone yang muncul.



Aplikasi yang dijalankan pada smartphone akan tampil sebagai berikut:

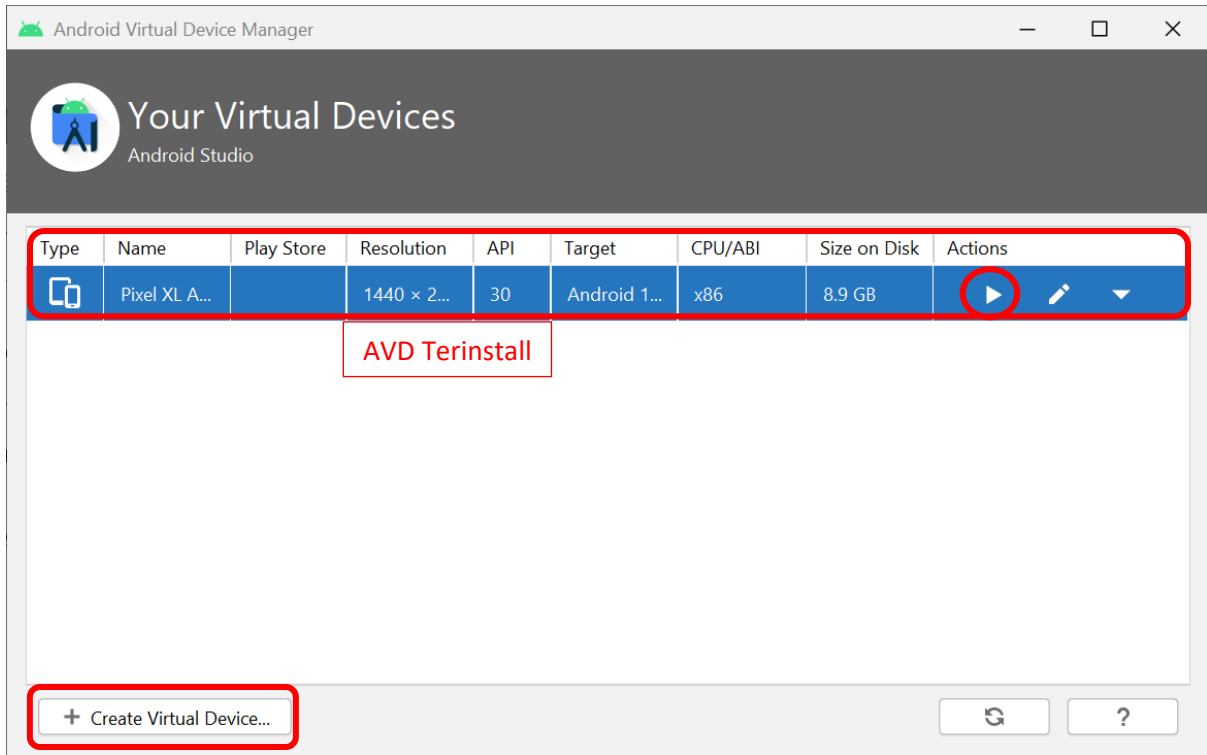


Selamat, kita sudah membangun aplikasi Android pertama di mata kuliah ini, dan juga menginstallnya di smartphone masing-masing.

3.6. Menjalankan Aplikasi dengan Emulator

Aplikasi Android merupakan aplikasi yang dijalankan di smartphone, akan tetapi tidak semua orang memiliki smartphone dengan sistem operasi Android. Untuk itu, kita dapat menjalankan aplikasi ini di emulator. Emulator merupakan aplikasi untuk membuat smartphone virtual di komputer/laptop kita. Banyak nama-nama emulator seperti Genymotion, NoxPlayer, BlueStacks, dan lainnya. Akan tetapi, kali ini kita akan menggunakan emulator dari Android Studio.

Jika penginstalan Android Studio dilakukan secara default, emulator ini juga diinstall dengan nama AVD (Android Virtual Device). Jika pada saat install awal sebelumnya, hal ini tidak dipilih, AVD dapat diakses dari Tools → AVD Manager. Jika belum ada virtual device terinstall, pilih "Create virtual device", lalu ikuti langkah-langkahnya.

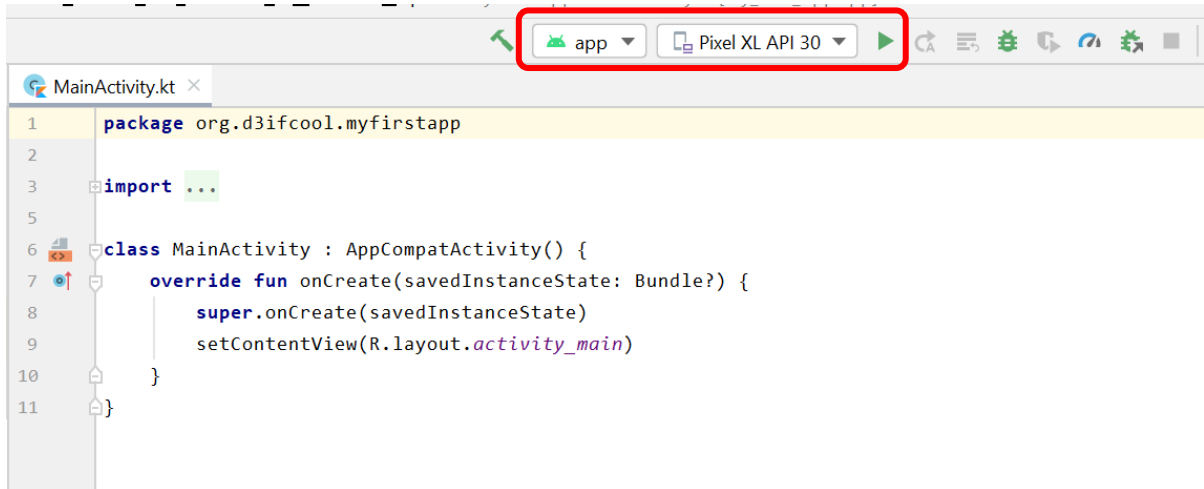


Jika sudah ada device yang diinstall, tekan tombol segitiga untuk menjalankan emulator. Menjalankan emulator ini akan membutuhkan waktu yang lama jika komputer/laptop yang digunakan kurang memadai. Di kondisi ini, sebaiknya gunakan smartphone saja, agar beban komputer/laptopmu tidak begitu berat.

Jika sudah berjalan, emulator dapat dijalankan seperti menghidupkan smartphone (dengan menekan tombol power), menunggu loading, dan pastikan berjalan seperti sebuah smartphone.



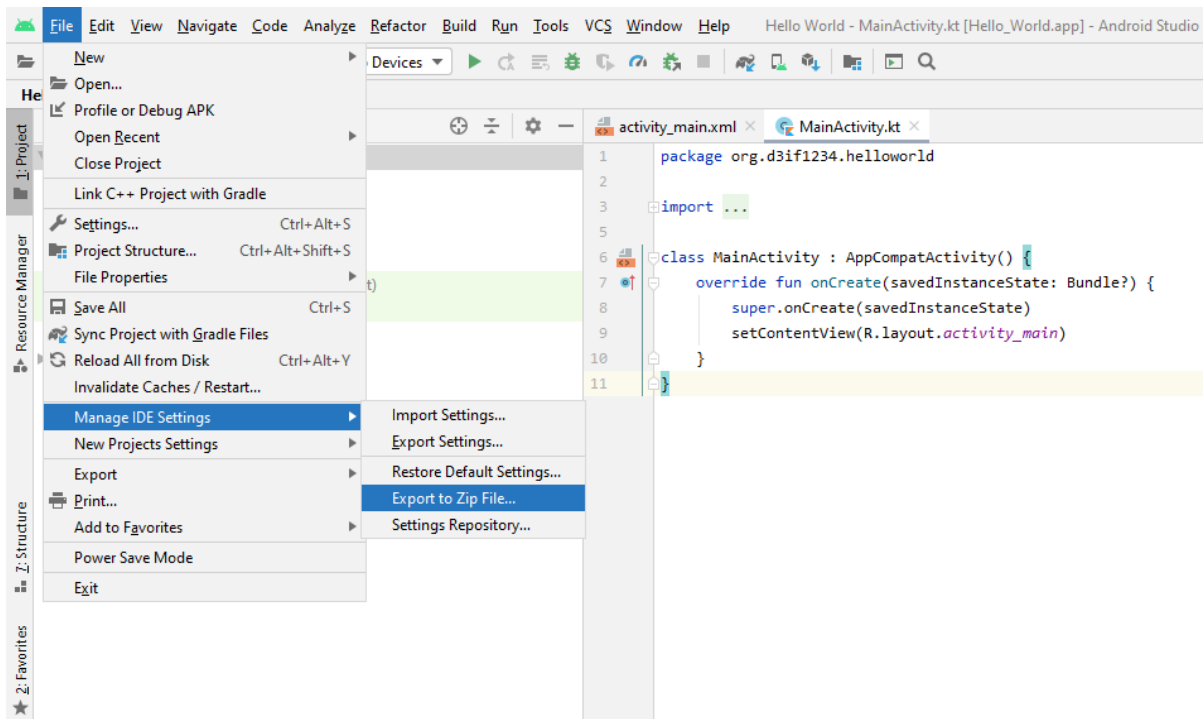
Setelah selesai loading, emulator akan muncul di pilihan Android Studio. Klik ikon Run.



Di tahap ini, kita sudah menjalankan aplikasi Android di emulator. Selamat!

3.7. Mengekspor Project untuk Dikirim

Untuk mengirimkan project Android Studio, klik menu File > Manage IDE Settings > Export to Zip File... Dengan cara ini, ukuran file yang dihasilkan akan lebih kecil (kurang dari 1MB) daripada men-zip dari Windows Explorer.



4. Summary

Pada modul kali ini kita sudah praktik bagaimana membuat aplikasi Android sederhana pertama di mata kuliah ini, dan juga menjalankannya di smartphone/emulator. Selain itu, kita juga sudah mengenal user interface dari Android Studio, struktur folder, mengenal apa itu activity dan juga layout untuk tampilan di aplikasi Android.

5. Challenge

Ubah kalimat Hello World yang ditampilkan menggunakan identitas masing-masing seperti nama, nim, dan kelas. Gunakan “\n” jika kamu membutuhkan Enter.