

RUNNING MODUL PEMROGRAMAN UNTUK PERANGKAT BERGERAK 1

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

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

Dwiko Indrawansyah, A.Md.Kom.

D3 Rekayasa Perangkat Lunak Aplikasi
Telkom University

Daftar Isi

Modul 01: Running Modul	1
1. Overview	1
2. Getting Started.....	1
3. Task	1
3.1. Membuat Project Pertama “Hello Android!”	1
3.2. Mengenal Struktur Folder dan File Project	6
3.3. Mengenal Kode Activity Aplikasi Android	7
3.4. Mengenal Tampilan 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. Self-Reflection.....	14
6. Challenge.....	14

Modul 01: Running Modul

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 Android!” di layar smartphone masing-masing. Aplikasi ini kemudian akan kita jalankan menggunakan smartphone/emulator.

2. Getting Started

Pertama, unduh terlebih dahulu Android Studio di tautan: <https://developer.android.com/studio>. Disarankan untuk mengunduh **Android Studio Hedgehog 2023.1.1 Patch 2** atau versi yang lebih tinggi. Selanjutnya install di komputer/laptop masing-masing dengan mengikuti petunjuk yang tampil pada saat proses instalasi. Proses unduh 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 😊

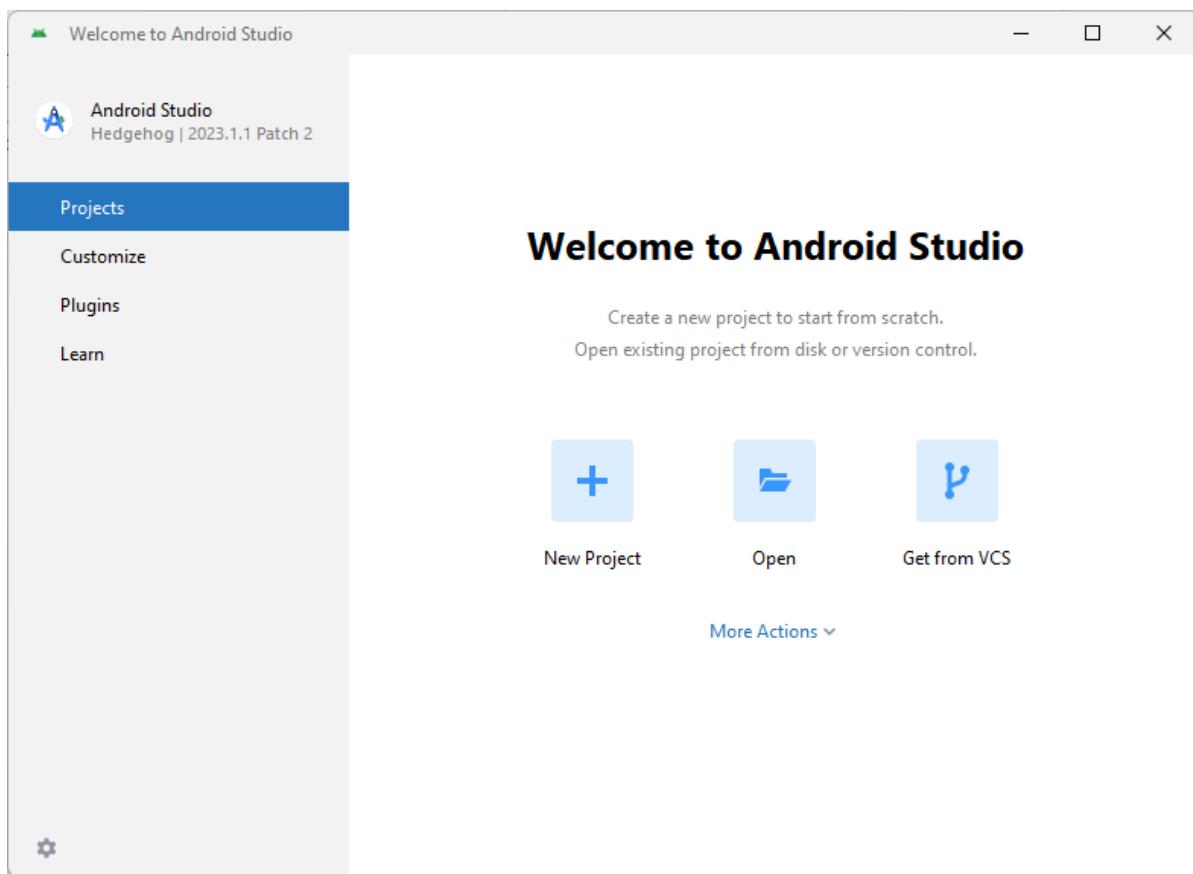
Catatan: Jika komputer/laptopmu tidak memungkinkan untuk menggunakan Android Studio karena spesifikasi yang terlalu rendah, kamu bisa membuat aplikasi Android menggunakan command line tools. Kode program dapat diketik menggunakan teks editor seperti Visual Studio Code, compile dengan menggunakan Terminal,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

3. Task

3.1. Membuat Project Pertama “Hello Android!”

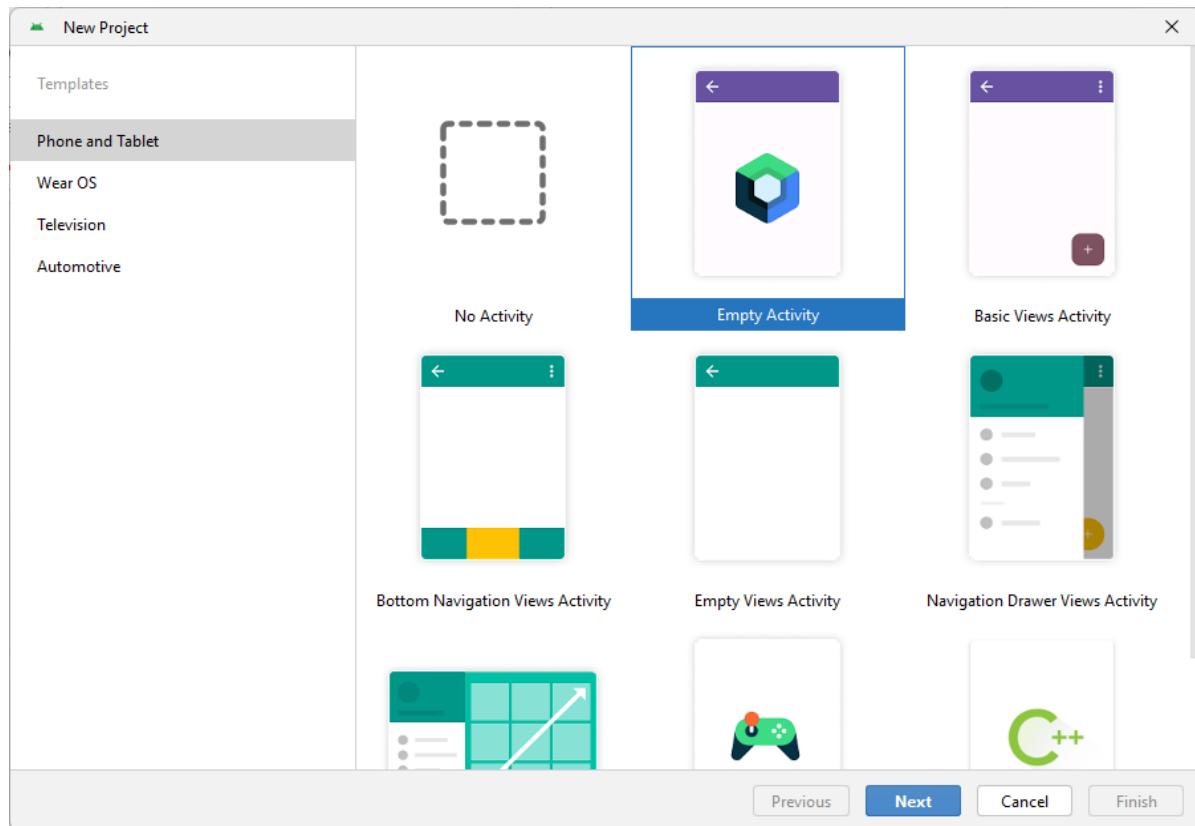
Pada task ini kita akan mulai membuat aplikasi Android pertama kita. Langkah awal adalah menjalankan Android Studio. Ingat, versi yang digunakan di mata kuliah kita **minimal** versi Hedgehog. Jika versimu yang lebih rendah, update terlebih dahulu.

Tampilan awal setelah proses loading selesai seperti di bawah ini. Terdapat info versi di kiri atas dan beberapa pilihan dengan penjelasan sebagai berikut:



1. **New Project:** Membuat project baru. Umumnya, sebuah aplikasi Android dibuat di satu project. Jadi jika ingin membuat aplikasi baru, pilih pilihan ini.
2. **Open:** Membuka project yang telah dibuat sebelumnya.
3. **Get from VCS:** Jika project yang akan dibuka berasal dari repository online yang tersimpan di version control system seperti Github, project tersebut dapat dibuka melalui menu ini. Akan dibahas lebih lanjut pada modul berikutnya.
4. **More Actions:** Untuk melakukan aksi lainnya seperti mengimpor contoh kode Android project, melakukan profiling/debugging APK atau mengatur emulator (virtual device) dan SDK yang akan digunakan dalam proses pengembangan aplikasi.
5. **Customize:** Untuk melakukan kustomisasi tampilan Android Studio, seperti tema terang atau gelap, font yang digunakan, dan lain sebagainya.
6. **Plugins:** Untuk mengatur plugin yang digunakan seperti menambahkan Flutter.
7. **Learn:** Untuk mendapatkan bantuan dan resource lainnya terkait Android Studio.

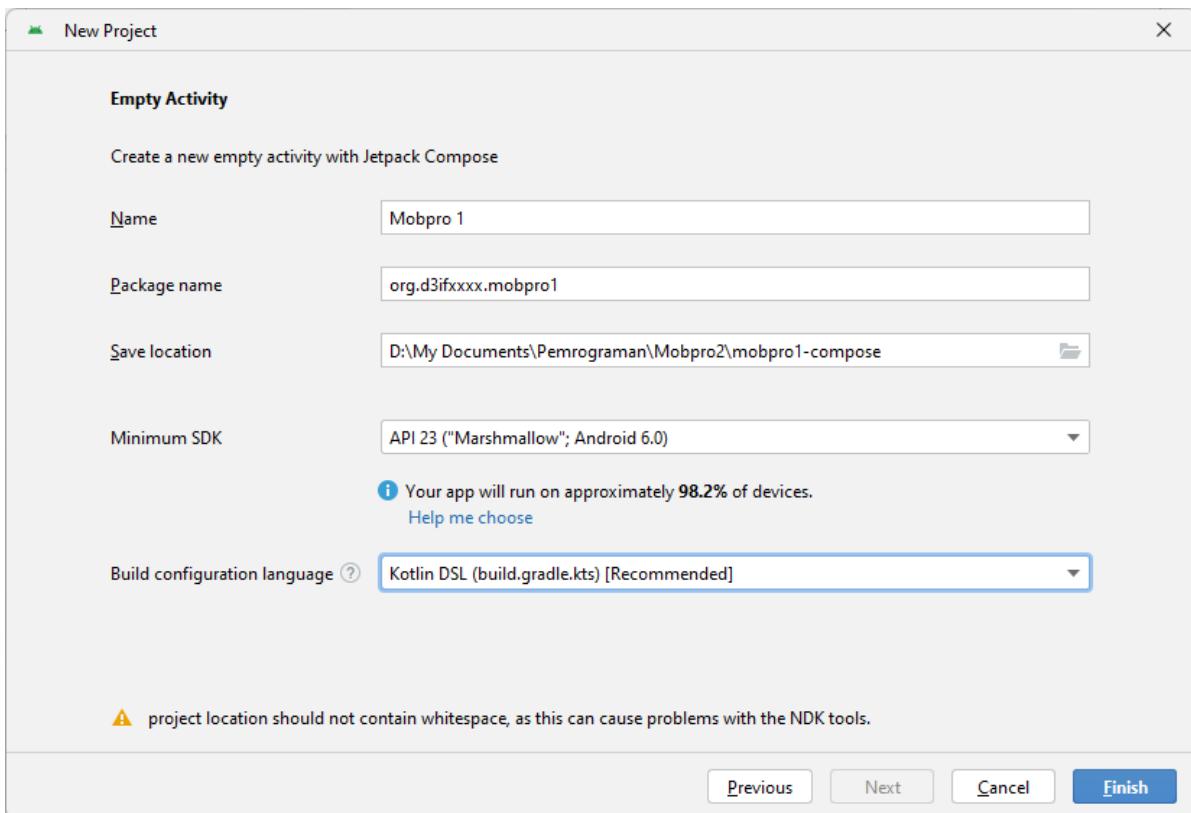
Kali ini, pilih “New Project”. Pemilihan menu ini akan memunculkan pilihan aplikasi yang dapat kita buat beserta template-nya. Pilih “Empty Activity” untuk menggunakan template sederhana yang telah disediakan Android Studio.



Setelah menekan Next, akan muncul window baru untuk mengkonfigurasi project yang akan dibuat.

1. **Name:** merupakan nama dari aplikasi yang dibuat. Isi dengan “Mobpro 1”.
2. **Package name:** merupakan ID dari aplikasi yang dibuat. Oleh karena itu, nama package harus unik, sebab digunakan untuk mengidentifikasi aplikasi pada device dan juga di Google Play Store. Maka untuk menjamin keunikan tadi, di mata kuliah ini, **WAJIB** menggunakan format package “org.d3ifxxxx.mobpro1” dimana xxxx adalah 4 digit NIM terakhir kalian.
3. **Save location:** adalah tempat penyimpanan project dari Android Studio.
4. **Minimum SDK:** adalah versi Android paling rendah yang dapat meng-install aplikasi yang kita bangun. Pilihlah API 23 (Marshmallow). Akan terlihat aplikasi kita dapat dijalankan di 98,2% device. Itu berarti terdapat 1,8% smartphone dengan sistem operasi Lollipop ke bawah tidak dapat meng-install aplikasi kita. Karena cukup kecil, jadi tidak apa-apa.
5. **Build configuration language:** biarkan default (Kotlin DSL – Recommended).

Pastikan tampilanmu telah sama dengan gambar berikut, selanjutnya klik Finish.



Jika ini merupakan pertama kalinya kamu membuat project baru di Android Studio, maka akan ada download yang dilakukan sebelum IDE terbuka sempurna. **Pastikan internetmu dalam keadaan aktif.**

Indikator loading dapat dilihat di bagian bawah dari Android Studio. Lama tidaknya proses loading ini bergantung kepada kecepatan koneksi internet dan juga spesifikasi komputer/laptop yang kamu gunakan. Ketika proses loading sedang berlangsung, harap tidak melakukan apa-apa yang dapat menambah beban kerja laptop karena akan memperlama proses.

Ketika loading selesai, di kiri bawah Android Studio muncul teks ini:

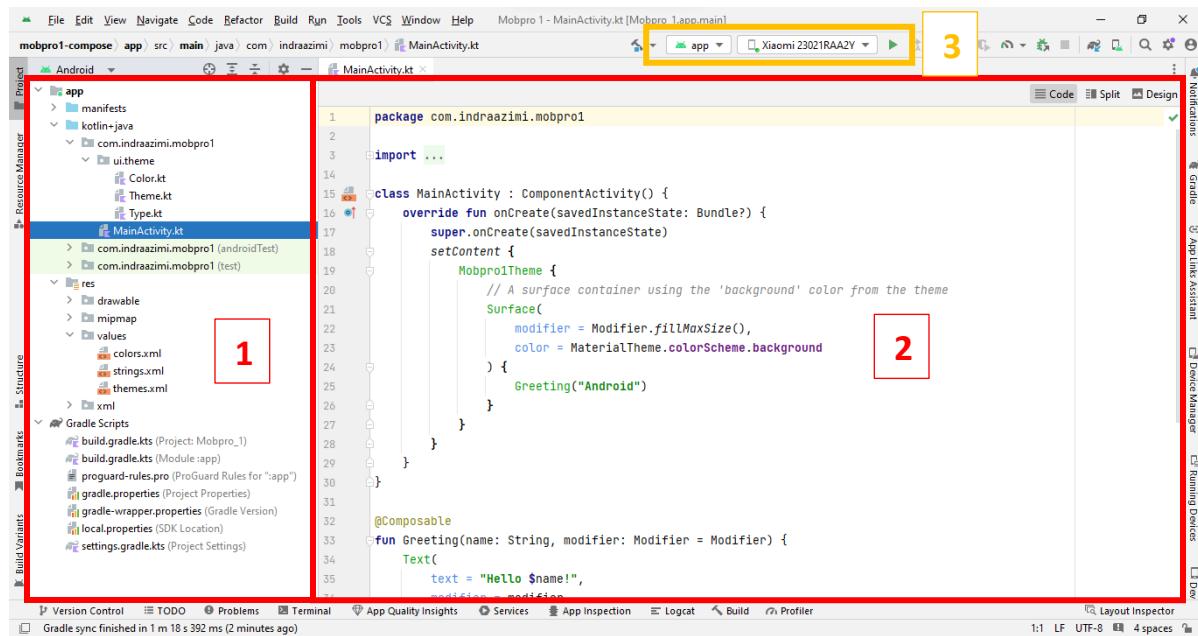
```
gradle-wrapper.properties (Gradle Version)
local.properties (SDK Location)
settings.gradle.kts (Project Settings)

32 @Composable
33 fun Greeting(name: String, mo
34     Text(
35         text = "Hello $name!"
36         modifier = modifier
```

Version Control TODO Problems Terminal App Quality Insights Services App

Gradle sync finished in 1 m 18 s 392 ms (15 minutes ago)

Tampilan utama Android Studio terlihat pada gambar. Terdapat 3 area utama, yaitu:

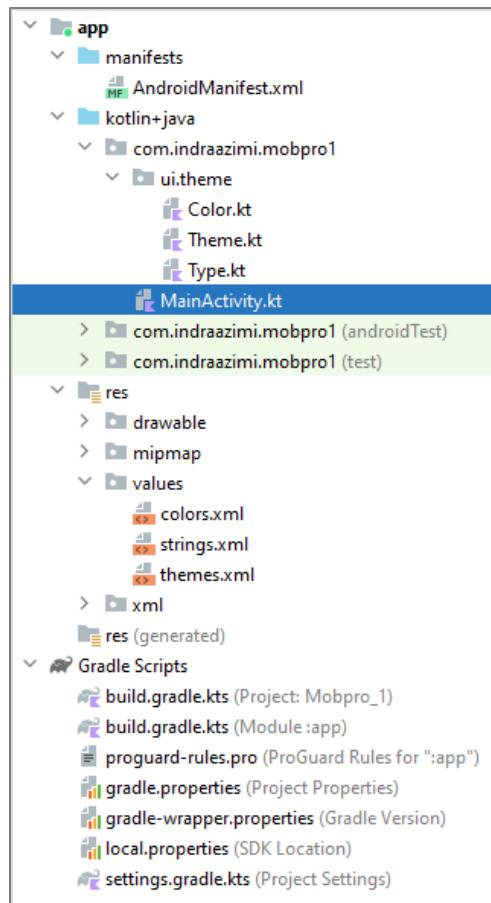


- Project Explorer.** Pada area ini, kita dapat melihat daftar file pada project yang muncul sesuai mode yang dipilih. Terdapat dua mode utama yang akan sering kita pakai. Mode default adalah “Android” yang hanya menampilkan daftar file penting di Android. Ketika ada file yang tidak muncul di mode ini, kita harus mengubah mode menjadi “Project” yang akan menampilkan daftar file apa adanya tanpa filter. Untuk berpindah dari mode Android ke mode Project dan sebaliknya, gunakan dropdown yang terletak di bagian atas Project Explorer.
- Code Editor.** Di sini tempat kita menuliskan kode yang diperlukan untuk membangun aplikasi, baik itu kode layout menggunakan Jetpack Compose, maupun kode Kotlin. Di bagian kanan atas kita disediakan pilihan tampilan Code, Split dan Design. Ketika sedang membuat layout, biasanya kita menggunakan tampilan Split agar ketika menuliskan kode layout, kita dapat langsung melihat preview-nya. Sedangkan ketika kita menuliskan kode Kotlin untuk logic aplikasi, kita akan menggunakan tampilan Code saja agar editornya terasa lebih lega.
- Toolbar.** Bagian ini merupakan area yang harus diperhatikan ketika ingin menjalankan project. Kita dapat memilih menjalankan aplikasi dari modul mana, memilih perangkat yang terhubung ke Android Studio untuk menjalankan aplikasi, dan lain sebagainya.

Selain itu, jika kamu perhatikan, di sekeliling pinggir kiri, bawah dan kanan Android Studio terdapat banyak tool window yang dapat digunakan agar proses pengembangan aplikasi lebih optimal. Kita akan membahasnya nanti ketika studi kasusnya tiba.

3.2. Mengenal Struktur Folder dan File Project

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 bawah merupakan struktur file yang akan sering kita temui saat membangun aplikasi Android. Beberapa yang penting diketahui sebagai dasar dalam membangun aplikasi ke depannya yaitu:



1. **app**: merupakan module default dari sebuah project. Dalam project yang besar, satu project dapat memiliki beberapa module. Saat ini, kita hanya akan menggunakan satu module di satu project Android Studio.
2. **manifest**: pada folder ini terdapat sebuah file AndroidManifest.xml. Setting dari aplikasi Android terdapat di sini, seperti icon yang digunakan, screen yang ditampilkan (activity), theme yang diterapkan, dan lainnya.
3. **kotlin+java**: folder ini mengandung kode project dalam bahasa Kotlin/Java.
4. **res**: folder untuk menyimpan resource dari aplikasi Android yang kita bangun. Dibagi lagi ke dalam beberapa subfolder sesuai jenisnya.
5. **drawable**: subfolder ini untuk menyimpan gambar yang digunakan di aplikasi.
6. **mipmap**: digunakan untuk menampung resource gambar sama seperti drawable, akan tetapi dikhususkan untuk launcher icon yang tampil di home screen.
7. **values**: subfolder dari res ini mengandung beberapa file seperti colors.xml (resource warna yang digunakan) atau strings.xml (teks yang digunakan untuk aplikasi).
8. **build.gradle**: merupakan file untuk setting dari aplikasi. Ada 2 level build.gradle: project dan module. Pada level project, terdapat setting seperti versi Gradle plugin dan Kotlin yang digunakan. Sedangkan pada level module, terdapat setting applicationId, minimum SDK, versi aplikasi, dependency atau library yang digunakan, dan lainnya.

3.3. Mengenal Kode Activity Aplikasi Android

Di dalam project Android yang telah dibuat, kita akan menemukan file bernama MainActivity. Apakah Activity itu? Di 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. Oleh karena satu aplikasi dapat memiliki banyak tampilan, maka satu project dapat memiliki lebih dari satu activity.

Secara umum, struktur kode pada Activity kurang lebih sama seperti struktur kode Java yang telah kalian pelajari di mata kuliah terdahulu. Di bagian paling atas terdapat package, kemudian import, baru masuk ke definisi kelas. Kelas selanjutnya dapat dibagi lagi menjadi method-method sesuai kebutuhan. Berikut adalah contoh kode MainActivity. File extension *.kt berarti bahasa pemrograman yang digunakan untuk activity ini adalah bahasa Kotlin.

```
package com.indraazimi.mobpro1

import ...

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Mobpro1Theme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}
```

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

Pada method onCreate terlihat bahwa ada perintah “setContent” yang digunakan untuk menentukan tampilan yang akan muncul. Di sini kita membuat tampilan menggunakan **Jetpack Compose**, jadi tidak lagi menggunakan XML karena itu teknologi lama.

3.4. Mengenal Tampilan Aplikasi Android

Di Jetpack Compose, setContent digunakan untuk menentukan tampilan UI dari suatu Activity. Di dalam setContent, terdapat tema yang akan dipakai oleh Activity. Tema ini berasal dari file Theme.kt pada package ui.theme. Oleh karena project kita bernama Mobpro1, maka tema default-nya bernama Mobpro1Theme. Di baris selanjutnya, terdapat Surface sebagai lapisan utama yang akan memuat konten aplikasi. Perhatikan kembali kode MainActivity berikut ini.

```
setContent {  
    Mobpro1Theme {  
        Surface(  
            modifier = Modifier.fillMaxSize(),  
            color = MaterialTheme.colorScheme.background  
        ) {  
            Greeting("Android")  
        }  
    }  
}
```

Di bawah class MainActivity terdapat Composable Function dengan nama Greeting. Perhatikan bahwa function ini menggunakan anotasi @Composable untuk menandai bahwa fungsi tersebut merupakan suatu komponen Jetpack Compose. Fungsi Greeting menerima dua parameter, yaitu:

1. name: sebuah string yang digunakan untuk membuat teks sapaan.
2. modifier: sebuah parameter opsional yang menyediakan modifikasi tambahan pada tata letak komponen. Secara default, ini diatur sebagai Modifier kosong.

```
@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
    Text(  
        text = "Hello $name!",  
        modifier = modifier  
    )  
}
```

Oleh karena Activity hanya akan menampilkan teks, maka di dalam fungsi Greeting hanya terdapat satu komponen, yaitu Text. Text merupakan komponen dasar untuk menampilkan teks di Jetpack Compose. Parameter text digunakan untuk menentukan teks yang akan ditampilkan, yang dalam hal ini berupa pesan sapaan "Hello \$name!". Sedangkan parameter modifier digunakan untuk mengatur modifikasi tambahan pada tata letak Text (jika ada).

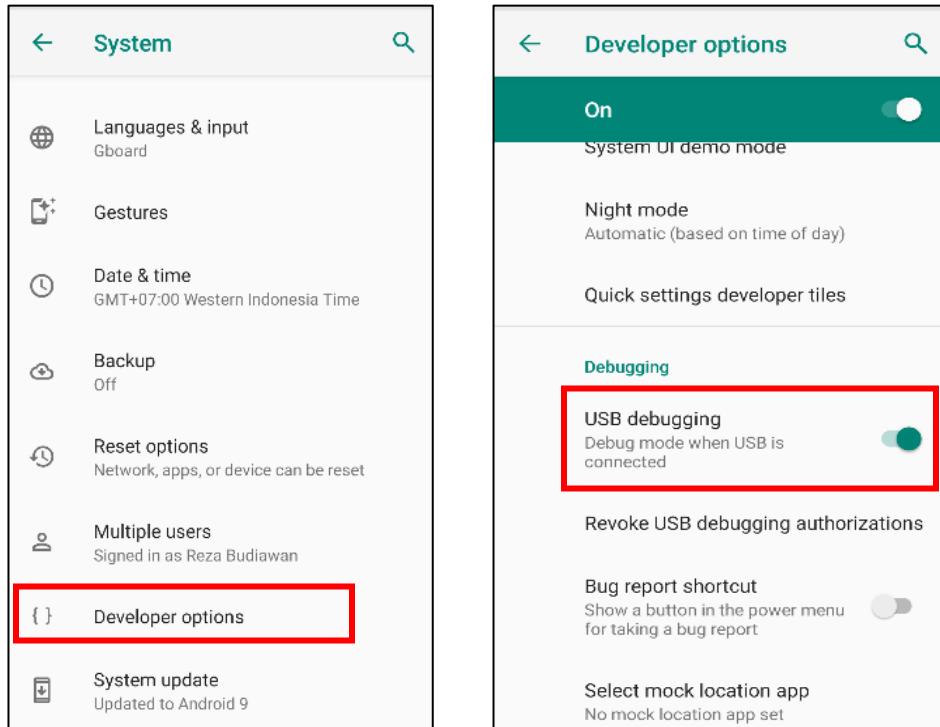
Method selanjutnya, yaitu GreetingPreview digunakan untuk membuat preview dari UI Greeting. Perhatikan bahwa method ini menggunakan anotasi @Preview dan parameter showBackground diatur true agar latar belakang preview ikut ditampilkan. Dengan demikian, kita dapat melihat tampilan visual dari komponen Greeting langsung dari mode Split di Code Editor, tanpa perlu repot menjalankan aplikasi di perangkat/emulator terlebih dahulu.

```
@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Mobpro1Theme {
        Greeting("Android")
    }
}
```

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:

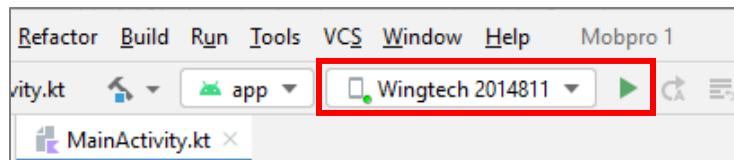
1. Smartphone terhubung ke PC dan dapat melakukan transfer data
2. Mode USB debugging di smartphone dalam keadaan aktif



Untuk itu, langkah pertama yang harus dilakukan adalah mencari menu Developer Options. Menu ini biasanya dapat ditemui di Settings → System → Advanced → Developer Options. Atau pada Settings → Additional Settings → Developer Options. Struktur menu mungkin berbeda, sesuai brand dan versi Android di perangkat yang kamu miliki. Untuk info lebih lengkap, silahkan melihat dokumentasi <https://developer.android.com/studio/debug/dev-options>

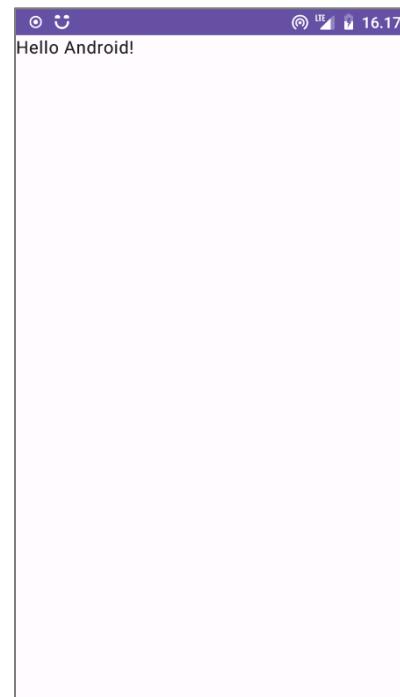
Jika menu Developer Options tidak dapat ditemui, artinya mode ini sedang dalam keadaan tidak aktif. Mengaktifkan mode developer dilakukan dengan menekan Build number sebanyak 7x. Caranya, masuk ke Settings → System → About phone → Build number → tekan 7x hingga muncul tulisan yang memberitahu akan mengaktifkan developer options. Jika menu ini sudah aktif, ketika build number ditekan, 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 aplikasi di smartphone. Hubungkan smartphone ke PC, dan akan muncul request permission di smartphone ketika pertama dilakukan. Klik OK atau Accept. Selanjutnya, nama smartphone kita akan muncul di toolbar.



Jalankan aplikasi dengan meng-klik tombol segitiga hijau di sebelah brand smartphone yang muncul. Android Studio akan melakukan proses build project, dan setelah selesai, aplikasi akan muncul di smartphone kamu dengan tampilan seperti pada gambar di samping.

Sampai di tahap ini, kita sudah berhasil menjalankan aplikasi Android pertama kita.. Selamat! 😊



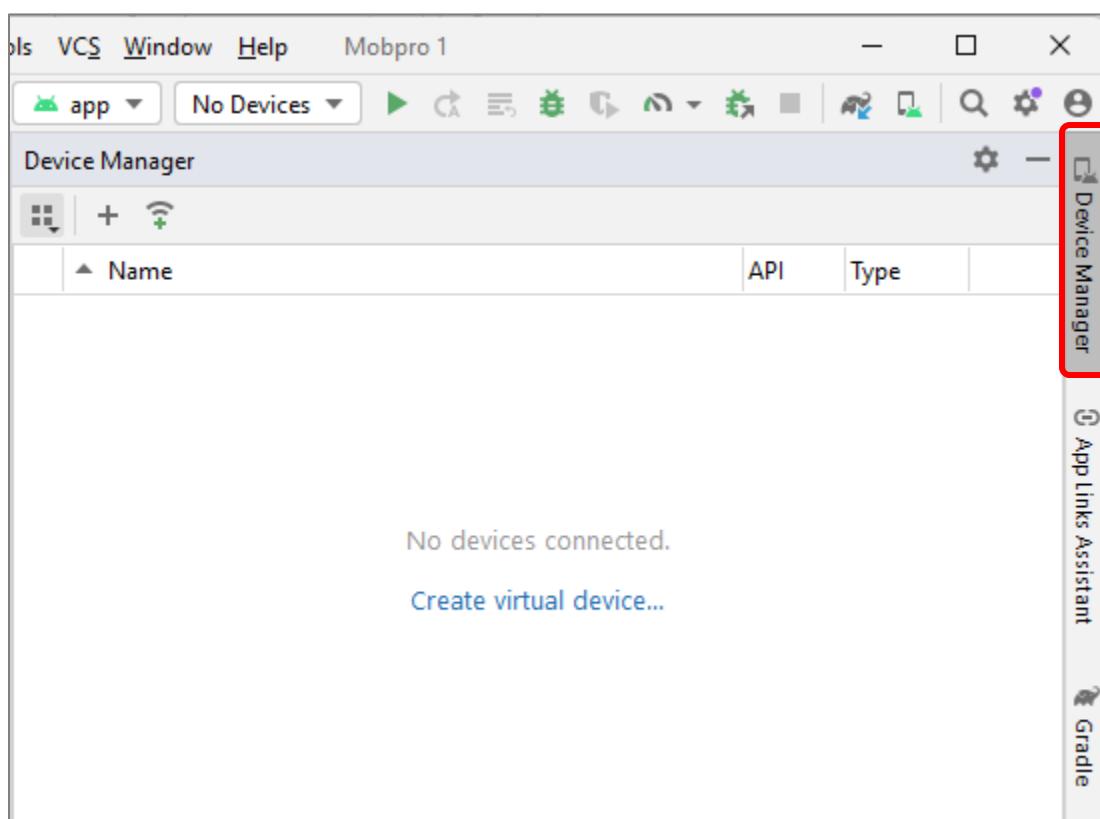
*Programming is not about typing,
it's about thinking.*

Rich Hickey

3.6. Menjalankan Aplikasi dengan Emulator

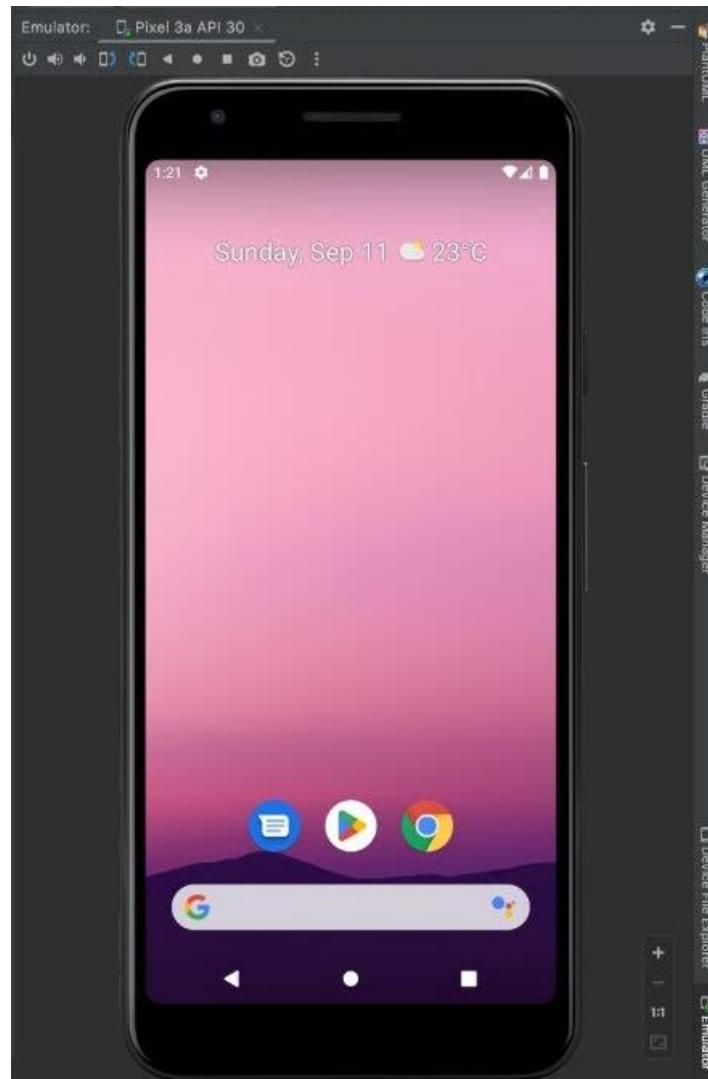
Aplikasi Android merupakan aplikasi yang dijalankan di smartphone, akan tetapi tidak semua orang memiliki smartphone dengan sistem operasi Android. Untuk itu, kita dapat menjalankan aplikasi ini di emulator. Emulator merupakan aplikasi untuk membuat smartphone virtual di komputer/laptop kita. Banyak 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 komponen ini tidak dipilih, AVD dapat diakses dari menu Tools → Device Manager. Atau bisa juga dengan klik langsung pada tab Device Manager di pinggir kanan Android Studio seperti ini.

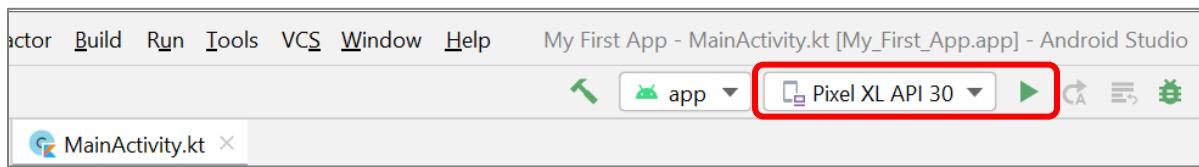


Jika belum ada virtual device terinstall, pilih "Create virtual device", lalu ikuti langkah-langkah yang muncul di layar. Jika sudah ada device yang diinstall, tekan tombol segitiga untuk menjalankan emulator. Menjalankan emulator ini akan membutuhkan waktu yang lama jika komputer/laptop yang digunakan memiliki spesifikasi yang kurang memadai. Di kondisi ini, sebaiknya gunakan smartphone saja, agar beban komputer/laptopmu tidak begitu berat.

Jika sudah berjalan, emulator dapat dipakai seperti smartphone pada umumnya. Hidupkan dengan menekan tombol power, tunggu loading selesai, dan home screen muncul sempurna.



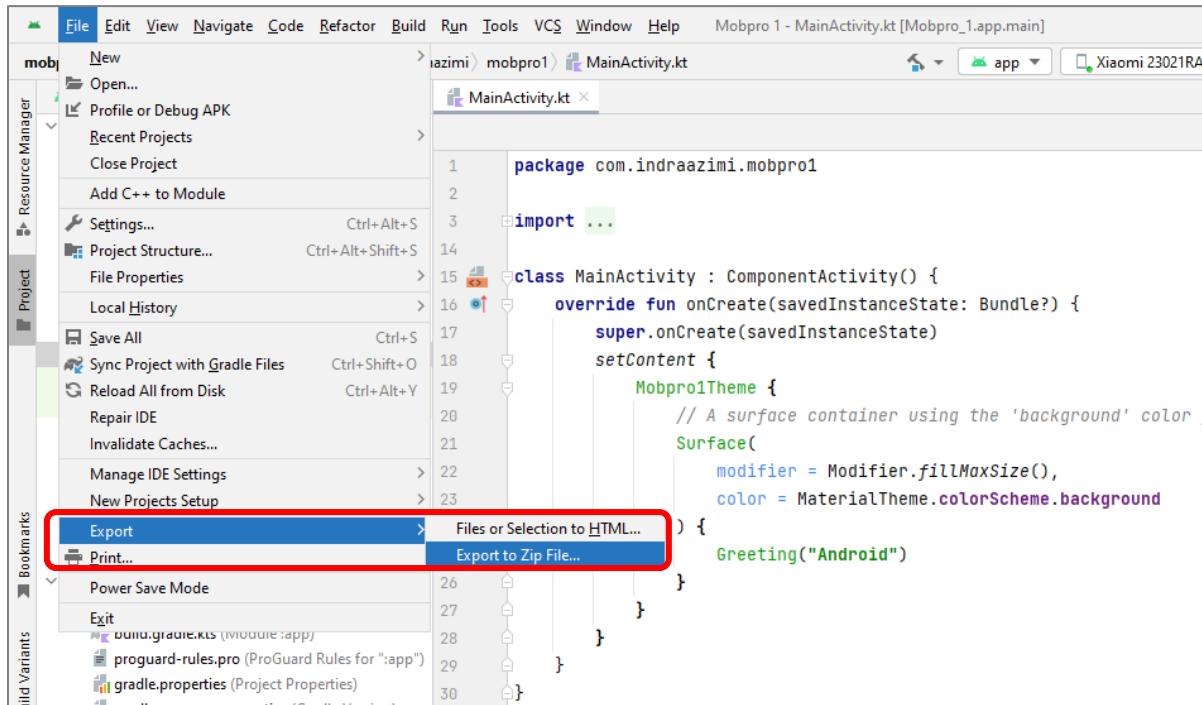
Setelah selesai loading, emulator akan muncul di pilihan Android Studio. Klik ikon Run.



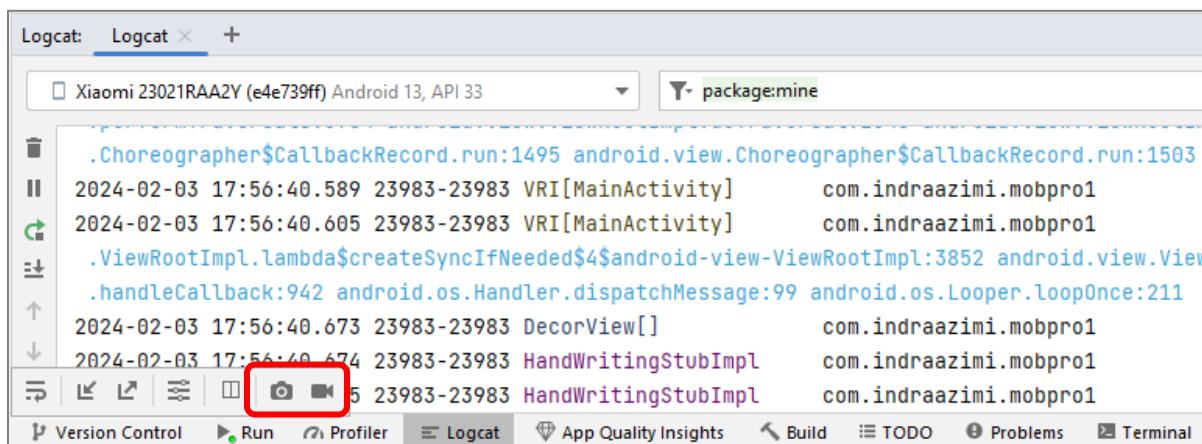
Tampilan aplikasi di emulator akan sama seperti tampilan aplikasi di smartphone.

3.7. Mengekspor Project untuk Dikirim

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



Kadang, kita juga butuh untuk mengambil screenshot dari aplikasi kita. Caranya, buka tab Logcat di pinggir bawah Android Studio, kemudian klik ikon kamera untuk mengambil screenshot smartphone.



Selain mengambil screenshot, kita juga bisa melakukan screen recording dengan menekan ikon video.

4. Summary

Pada modul kali ini kita sudah praktik bagaimana membuat aplikasi Android pertama kita, walaupun masih sederhana. Kita juga sudah mengenal user interface dari Android Studio, struktur file project, mengenal apa itu activity dan juga Jetpack Compose untuk tampilan di aplikasi Android. Selain itu, kita juga sudah belajar bagaimana menjalankan aplikasi kita di smartphone atau emulator, serta cara mengekspor project agar dapat dikirim ke pihak lain.

5. Self-Reflection

Apakah praktik kita di modul ini telah berhasil..? Silahkan isi tabel berikut.

No.	Pertanyaan	Ya / Tidak
1	Saya dapat membuat project baru Android	
2	Saya telah mengenal bagian-bagian Android Studio	
3	Saya dapat menjalankan aplikasi di smartphone/emulator	
4	Saya dapat mengekspor project dan mengambil screenshot aplikasi	

6. Challenge

Kerjakan tugas-tugas berikut:

- a. Buatlah project baru dengan nama “About Me”.
- b. Ubah kalimat Hello Android! dari project default menjadi identitas masing-masing (nama, NIM dan kelas). Untuk memisahkan baris, gunakan “\n”.
- c. Jalankan aplikasi. Jika sudah benar, ambil screenshotnya.
- d. Ekspor project dan screenshot ke dalam satu file ZIP.