

# Bab 1. Pengenalan SDK Android Studio

## 1.1. Dasar Teori

**Android** adalah sistem operasi open source dan platform pemrograman yang dikembangkan oleh Google untuk telepon seluler dan perangkat seluler lainnya, seperti tablet. Anda dapat mendownload beragam aplikasi Android melalui Play Store.

Untuk mengembangkan suatu aplikasi Android, Anda membutuhkan **Android SDK**. Android SDK membantu Anda menulis kode asli dan merakit modul perangkat lunak untuk membuat aplikasi bagi pengguna Android. Anda dapat menggunakan Android Studio sebagai official Android IDE.

**Android studio** adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Android Studio menyediakan alat tercepat untuk membangun aplikasi di setiap jenis perangkat Android.

Sebelum memulai instalasi perlu untuk memperhatikan spesifikasi yang direkomendasikan Android. Spesifikasi akan berpengaruh pada cepat atau lambatnya performa Android Studio. Berikut syarat untuk instalasi android studio di Windows:

1. Microsoft® Windows® 64-bit 8/10/11
2. Arsitektur CPU x86\_64; Intel Core generasi ke-2 atau yang lebih baru, atau CPU AMD dengan dukungan untuk Windows Hypervisor
3. RAM 8 GB atau lebih
4. Ruang disk minimum yang tersedia 8 GB (IDE + Android SDK + Android Emulator)
5. Resolusi layar minimum 1280 x 800

Berikut ini adalah persyaratan sistem untuk Android Studio di macOS.

1. MacOS® 10.14 (Mojave) atau yang lebih tinggi
2. Chip berbasis ARM, atau Intel Core generasi ke-2 atau yang lebih baru dengan dukungan untuk Hypervisor Framework
3. RAM 8 GB atau lebih
4. Ruang disk minimum yang tersedia 8 GB (IDE + Android SDK + Android Emulator)
5. Resolusi layar minimum 1280 x 800

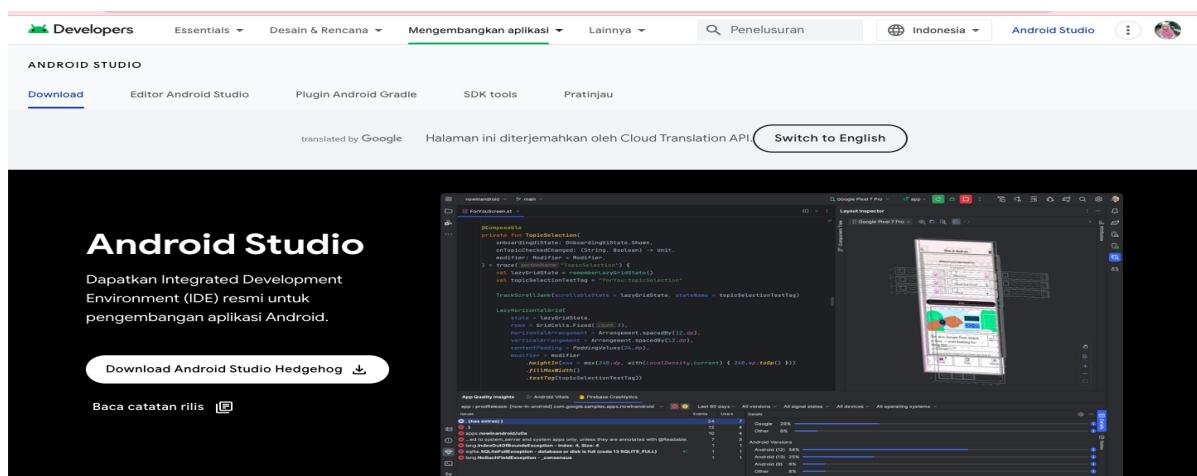
Berikut adalah persyaratan sistem Android Studio untuk Linux.

1. Distribusi Linux 64-bit apa pun yang mendukung Gnome, KDE, atau Unity DE; GNU C Library (glibc) 2.31 atau lebih baru.

2. Arsitektur CPU x86\_64; Prosesor Intel Core atau generasi ke-2, atau prosesor AMD dengan dukungan untuk AMD Virtualization (AMD-V) dan SSE3
3. RAM 8 GB atau lebih
4. Ruang disk minimum yang tersedia 8 GB (IDE + Android SDK + Android Emulator)
5. Resolusi layar minimum 1280 x 800

## 1.2. Mengunduh Android Studio

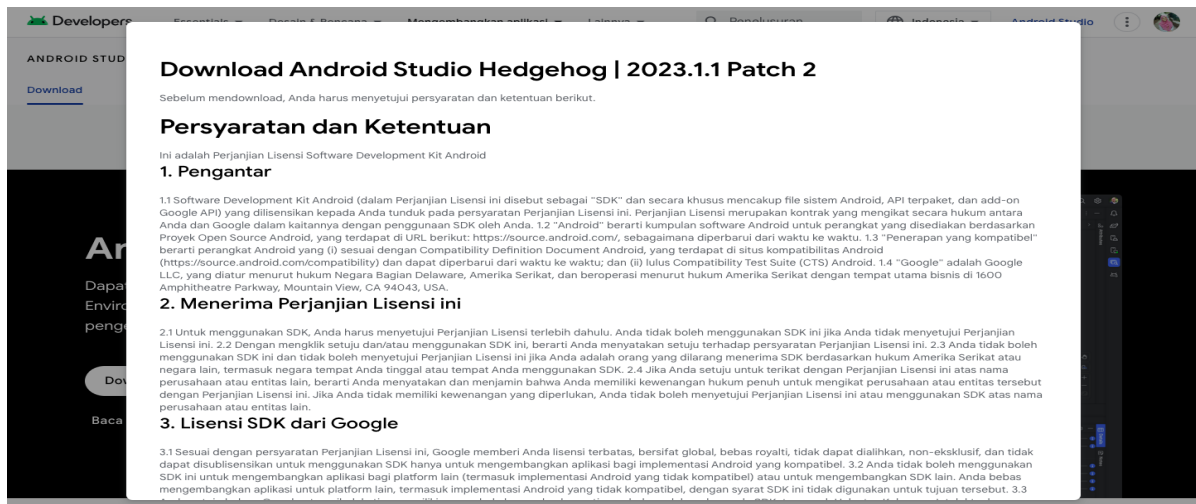
1. Download Android Studio di <https://developer.android.com/studio>.



Gambar 1.1 Website Developers Android

2. Menyetujui *Terms and Conditions*.

Setelah itu akan muncul *Terms and Conditions*. Scroll kebawah hingga bagian akhir, kemudian tandai *check box* “*I have read and agree with the above terms and conditions*”.

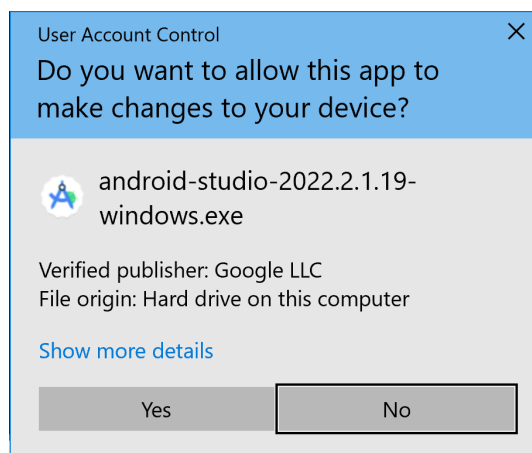


Gambar 1.2 Terms and Conditions

Klik pada tombol unduh “Download Android Studio....” untuk memulai. Tunggu download selesai. Proses ini mungkin memerlukan waktu lama, bahkan Anda bisa melakukan aktivitas lain.

### 1.3. Instalasi Android Studio

1. Buka folder tempat Anda menyimpan file penginstalan Android Studio yang di-download.
2. Klik dua kali file tersebut.
3. Jika Anda melihat dialog User Account Control tentang mengizinkan penginstalan melakukan perubahan pada komputer, klik **Yes** untuk mengonfirmasi penginstalan.



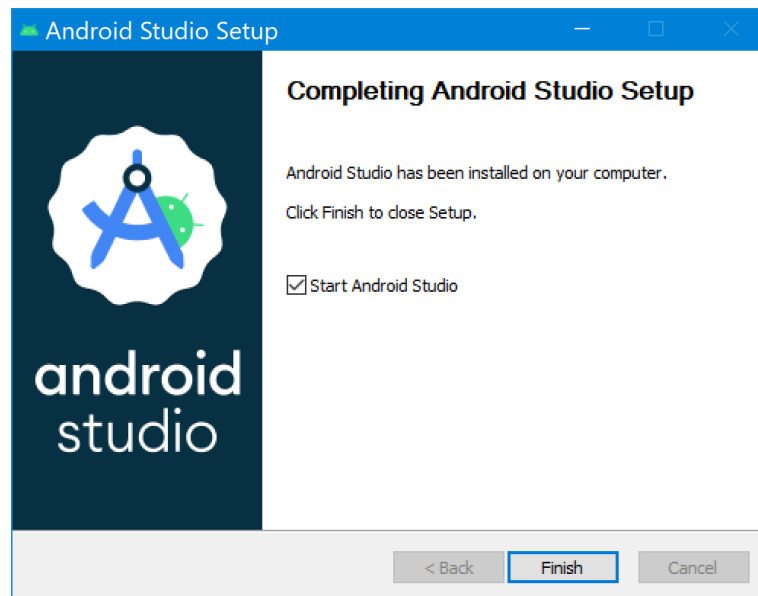
Gambar 1.3 User Account Control

4. Dialog Welcome to Android Studio Setup akan ditampilkan.



Gambar 1.4 Dialog Welcome to Android Studio Setup

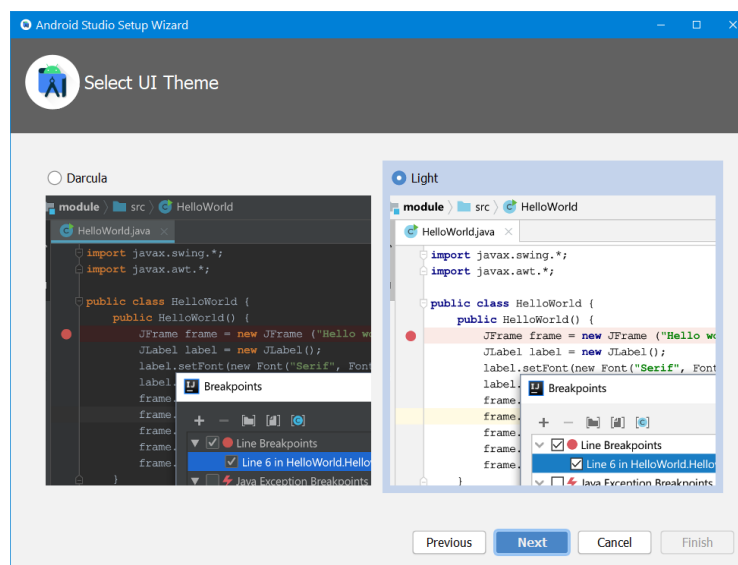
5. Klik **Next** untuk memulai penginstalan.
6. Setujui setelan penginstalan default untuk semua langkah.
7. Tunggu instalasi hingga selesai. Klik **Finish** saat penginstalan selesai untuk meluncurkan Android Studio.



Gambar 1.5 Instalasi Android Studio Selesai

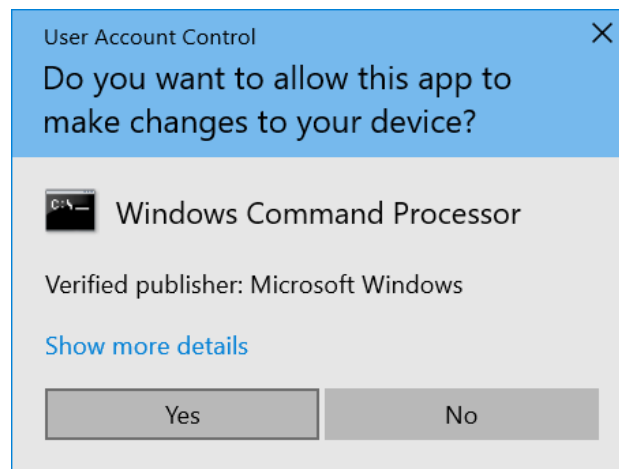
#### 1.4. Setup Software Development Kit (SDK)

1. Pilih preferensi tema terang atau gelap saat Android Studio pertama kali diluncurkan. Screenshot dalam modul ini menggunakan tema terang, tetapi Anda dapat memilih tema mana pun sesuai keinginan.



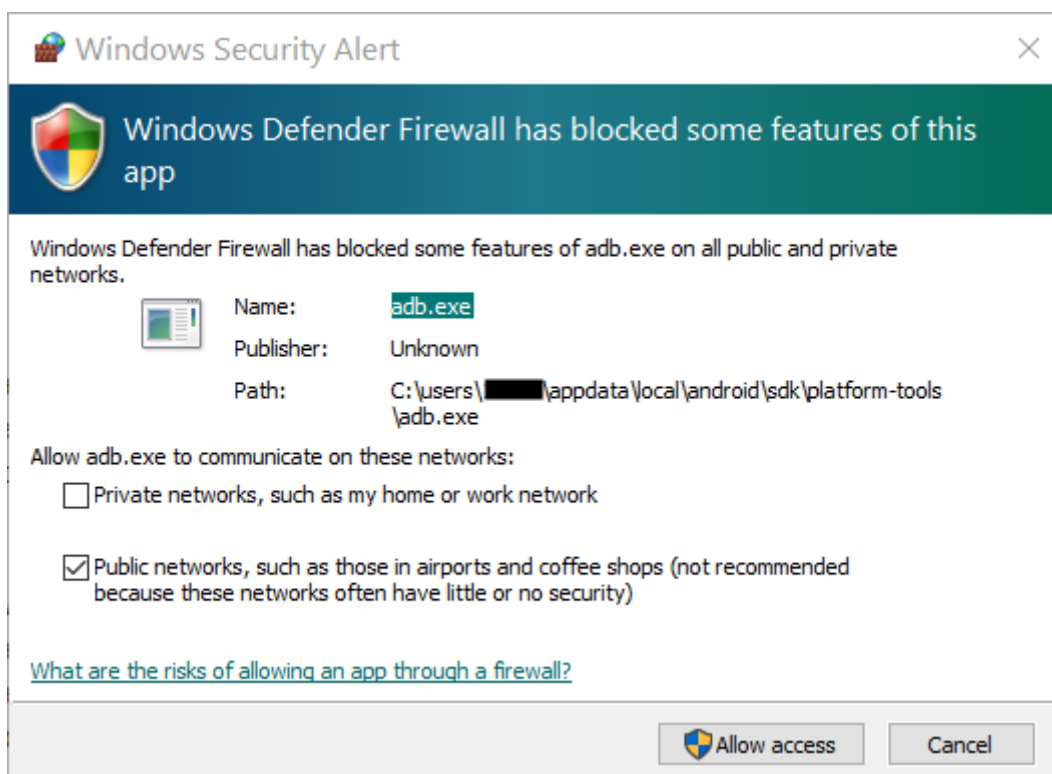
Gambar 1.6 Memilih Preferensi Tema Android Studio

2. Selama penginstalan, wizard penyiapan akan mendownload dan menginstal komponen serta alat tambahan yang diperlukan untuk pengembangan aplikasi Android. Proses ini mungkin berlangsung beberapa saat, bergantung pada kecepatan internet. Selama proses ini, Anda mungkin melihat **dialog User Account Control untuk Windows Command Processor**. Klik **Yes** untuk menerima dialog.



Gambar 1.7 Dialog User Account Control untuk Windows Command Processor

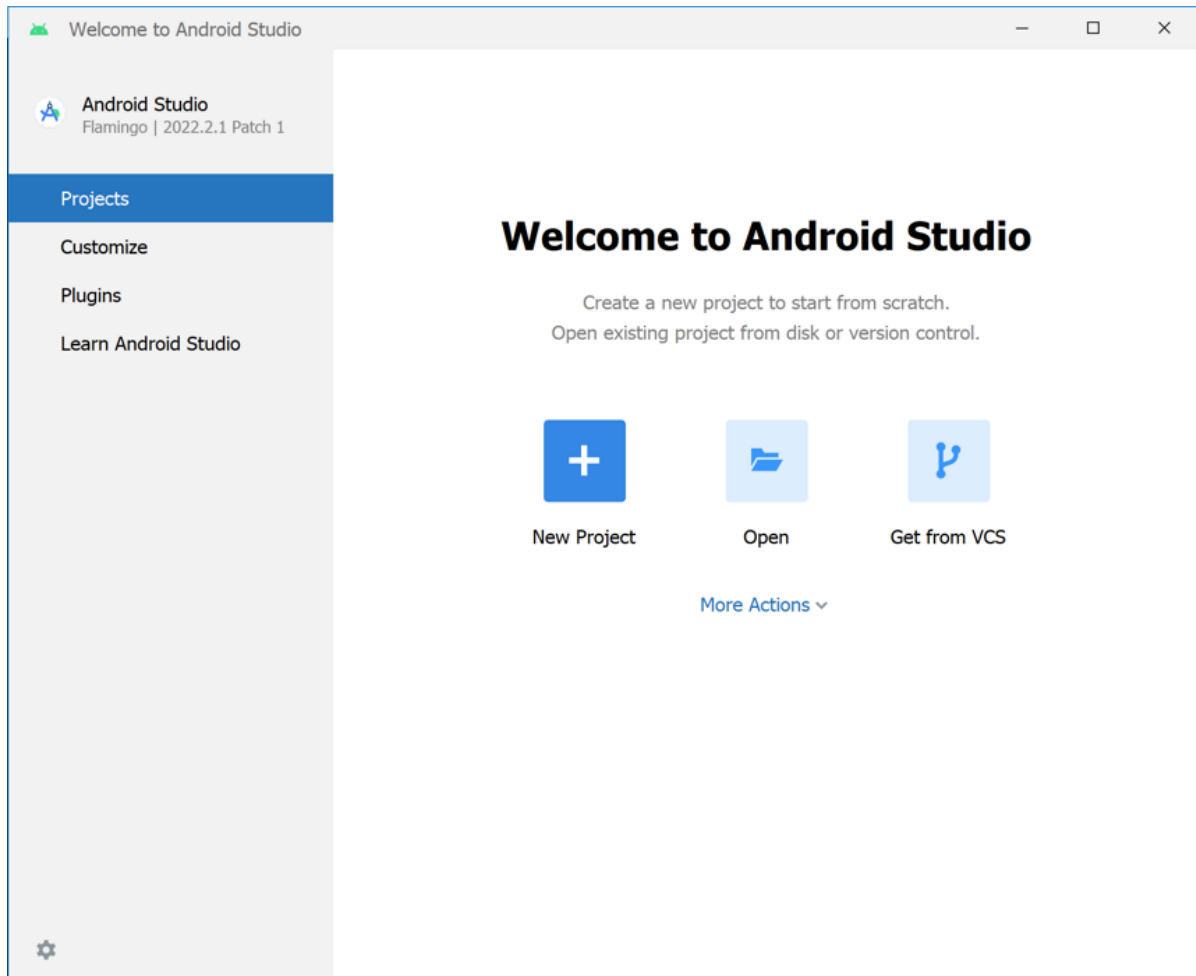
3. Anda mungkin juga menerima **Windows Security Alert** tentang adb.exe. Klik **Allow Access** jika diperlukan, untuk melanjutkan penginstalan.



Gambar 1.8 Verifikasi Komponen SDK yang akan Diinstall

4. Proses download sekaligus instalasi komponen SDK sedang dilakukan oleh Android Studio. Ketika prosesnya telah selesai, klik **Finish**.

5. Jika instalasi setup Android SDK telah berhasil, Anda dapat mulai **membuat project Android** dengan klik **New Project**.

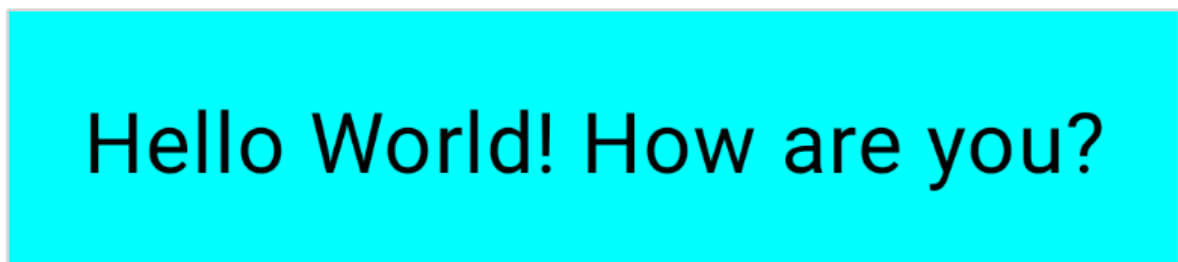


Gambar 1.9 Tampilan UI Welcome to Android Studio

### 1.5. Membuat Aplikasi Android Pertama

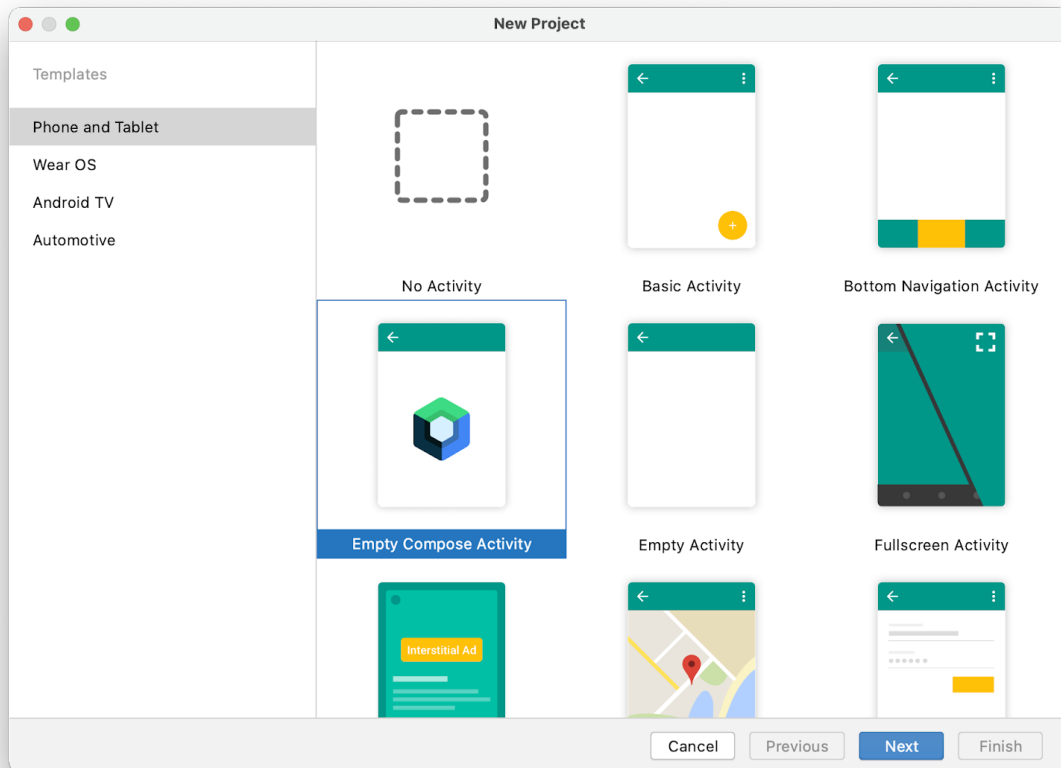
Berikut adalah tampilan aplikasi saat Anda menyelesaikan praktikum ini.

GreetingPreview



Gambar 1.10 Tampilan Aplikasi Android Pertama Anda

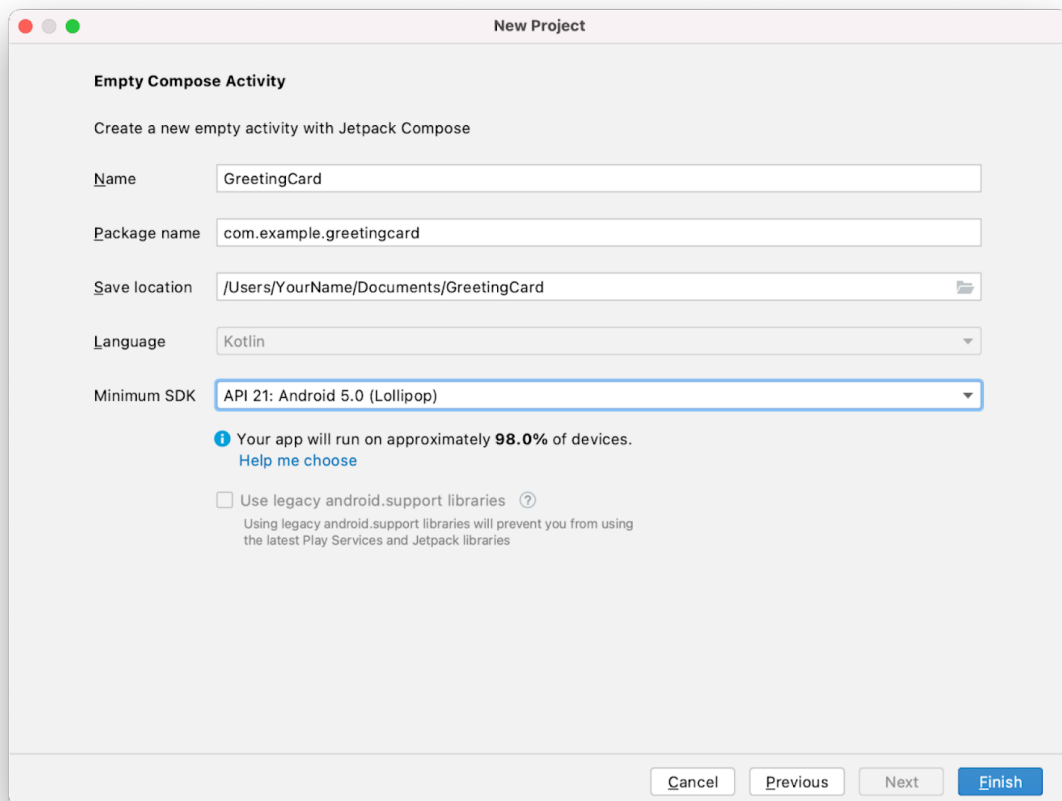
1. Setelah klik New Project pada proses sebelumnya, kemudian pilih **template** sesuai kebutuhan Anda. Pada praktikum ini, gunakan template yang mendasar yaitu pilih **Empty Compose Activity** untuk Phone and Tablet. Empty Compose Activity adalah template untuk membuat project sederhana yang dapat Anda gunakan untuk membuat aplikasi Compose. Kemudian klik **Next**. Template ini telah disediakan langsung oleh Android Studio.



Gambar 1.11 Memilih Template Empty Compose Activity

2. Isi **configuration** sesuai kebutuhan Anda.
  - a. Berikan nama aplikasi yang akan Anda buat pada field **Name**.
  - b. Biarkan kolom **Package Name** apa adanya. Ini adalah cara file Anda akan disusun dalam struktur file. Saat upload aplikasi Anda ke Play Store, Anda membutuhkan package name. Secara umum, penamaan package aplikasi Android berformat “namadomain.namadeveloper.namaaplikasi”.
  - c. Pada **Save Location**, Anda dapat menentukan lokasi untuk menyimpan project Anda.
  - d. **Kotlin** sudah dipilih di kolom **Language**. Bahasa menentukan bahasa pemrograman yang ingin Anda gunakan untuk project. Karena Compose hanya kompatibel dengan Kotlin, Anda tidak dapat mengubah kolom ini.

- e. Pilih API 21: Android 5.0 (Lollipop) dari menu di kolom Minimum SDK. Minimum SDK menunjukkan versi minimum Android yang dapat dijalankan oleh aplikasi Anda.
- f. Jika aplikasi Anda memerlukan dukungan *legacy library*, tandai checkbox Use legacy android.support libraries. Pada praktikum ini, **kosongkan checkbox Use legacy android.support libraries**.



Gambar 1.12 Pengaturan Empty Compose Activity

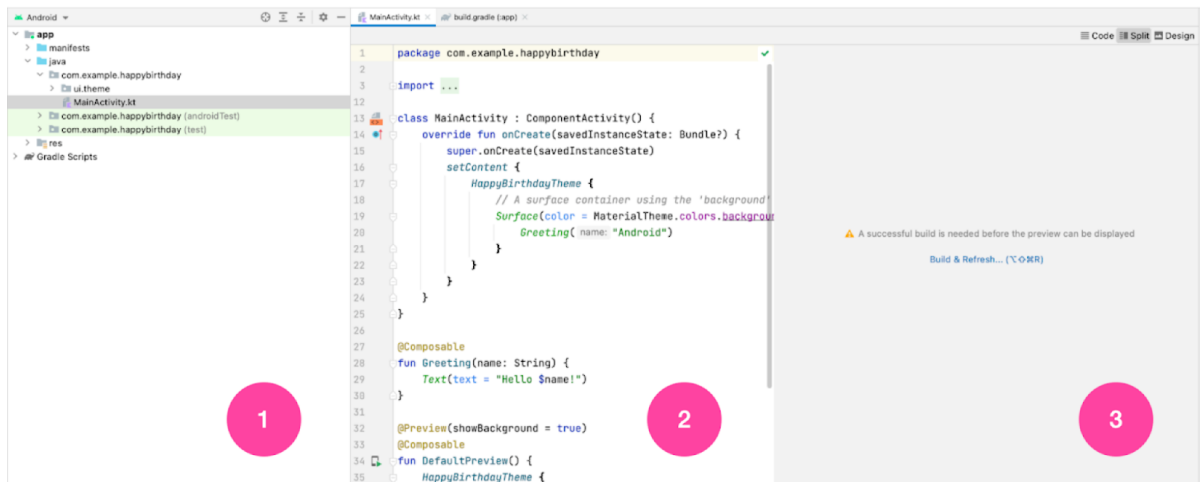
3. Klik **Finish**. Setelah beberapa waktu pemrosesan, jendela utama Android Studio muncul.
4. Setiap membuat suatu project baru, Android Studio akan otomatis melakukan import dan download Gradle. Maka dari itu, **pastikan personal komputer Anda terkoneksi dengan Internet**.



Gambar 1.13 Import dan Download Gradle



5. Setelah proses Gradle telah selesai, project yang anda buat siap untuk pengembangan lebih lanjut. Klik **Split** di kanan atas Android Studio, ini memungkinkan Anda untuk melihat kode dan desain. Anda juga dapat mengklik Code untuk melihat kode saja atau klik Design untuk melihat desain saja.

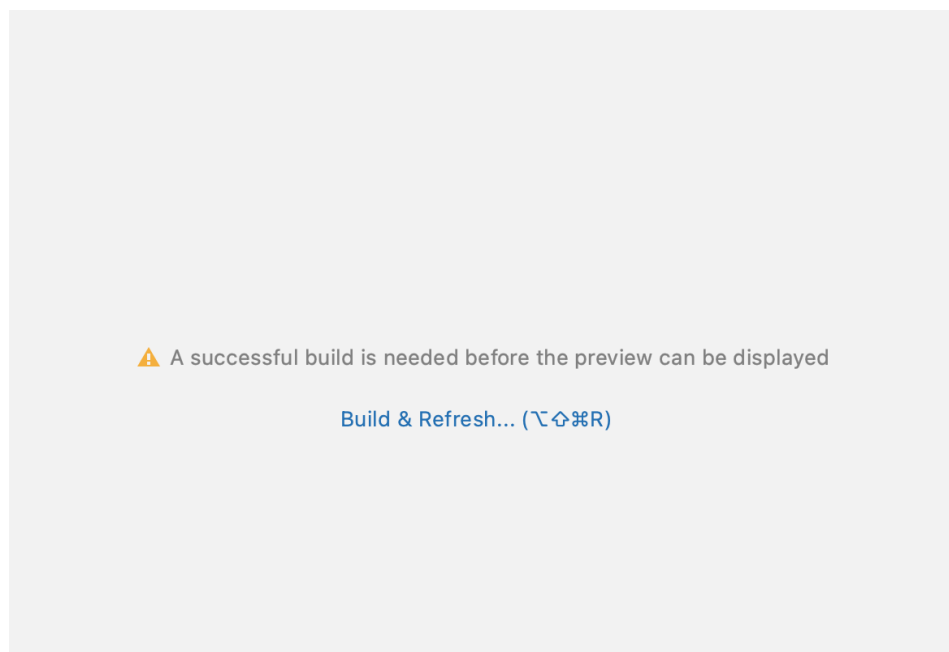


Gambar 1.14 Jendela Utama Android Studio

Setelah menekan Split, Anda akan melihat 3 area:

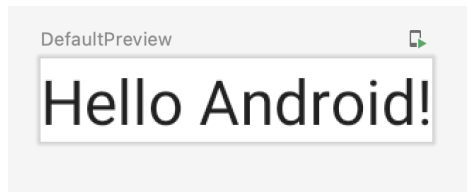
1. Tampilan **Project** (1) menampilkan file dan folder project Anda
2. Tampilan **Code** (2) adalah tempat Anda mengedit kode
3. Tampilan **Design** (3) adalah tempat Anda melihat pratinjau tampilan aplikasi

Di tampilan Design, Anda akan melihat panel kosong dengan teks ini:



Gambar 1.15 Tampilan Area Design Android Studio

6. Klik **Build & Refresh**. Proses build mungkin memerlukan waktu beberapa saat, tetapi setelah selesai, pratinjau akan menampilkan kotak teks yang bertuliskan "Hello Android!". Aktivitas Compose kosong berisi semua kode yang diperlukan untuk membuat aplikasi ini.



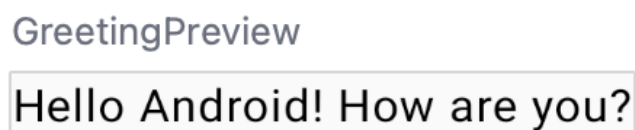
Gambar 1.16 Default Preview

7. Memperbarui **teks** menjadi **Hello World! How are you?** Cara dengan modifikasi fungsi Greeting

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

Gambar 1.17 Modifikasi Text pada Fungsi Greeting

8. Build ulang DefaultPreview dengan menekan tombol  di kiri atas panel desain:



Gambar 1.18 Preview Setelah Build Ulang

9. Ubah fungsi DefaultPreview() dengan World. Kemudian, build ulang dan lihat preview yang dipersonalisasi.

```
@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    FirstAppKotlinTheme {
        Greeting(name: "World")
    }
}
```

Gambar 1.19 Ubah Default Preview dengan World

## GreetingPreview

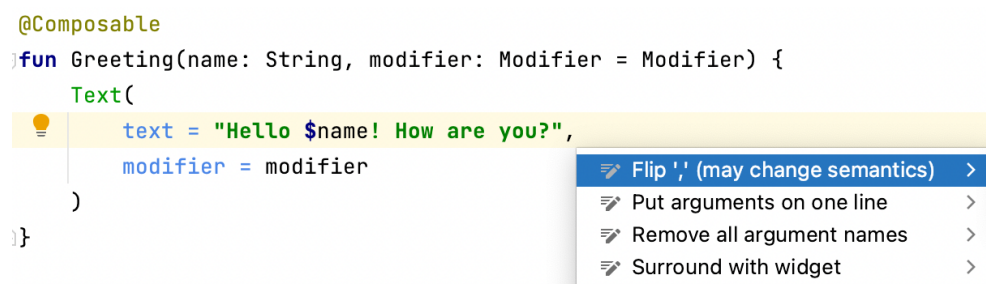
Hello World! How are you?

Gambar 1.20 Preview Setelah Diubah dengan World

10. Untuk menetapkan **warna background**, Anda harus mengapit teks dengan Surface.

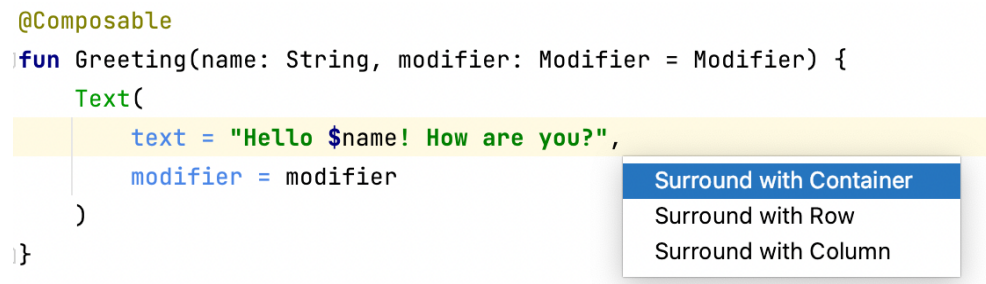
**Surface** adalah penampung yang menampilkan bagian UI tempat Anda dapat mengubah tampilan, seperti warna background atau padding.

11. Untuk mengapit teks dengan Surface, tandai baris teks, tekan (Alt+Enter untuk Windows atau Option+Enter di Mac), lalu pilih **Surround with widget**.



Gambar 1.21 Surround with widget

12. Pilih **Surround with Container**.



Gambar 1.22 Surround with Container

13. Penampung default yang akan diberikan kepada Anda adalah Box, tetapi Anda dapat mengubahnya ke jenis penampung lainnya.

14. Hapus **Box** dan ketik **Surface()** sebagai gantinya.

15. Penampung **Surface** memiliki parameter **color**, tetapkan ke **Color**. Pilih warna untuk permukaan Anda.

```

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Surface(color = Color.Cyan) {
        Text(
            text = "Hello $name! How are you?",
            modifier = modifier
        )
    }
}

```

Gambar 1.23 Menambahkan Surface dan Parameter Color

GreetingPreview

Hello World! How are you?

Gambar 1.24 Preview Setelah Ditambah Warna Background

16. Selanjutnya Anda akan menambahkan padding di sekitar teks.
17. **Modifier** digunakan untuk menambah atau mendekorasi composable. Satu Pengubah yang dapat Anda gunakan adalah pengubah padding, yang menerapkan ruang di sekitar elemen (dalam hal ini, menambahkan ruang di sekitar teks). Hal ini diperoleh dengan menggunakan fungsi **Modifier.padding()**.

```

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Surface(color = Color.Cyan) {
        Text(
            text = "Hello $name! How are you?",
            modifier = Modifier.padding(24.dp)
        )
    }
}

```

Gambar 1.25 Menambahkan Modifier.padding()

GreetingPreview

Hello World! How are you?

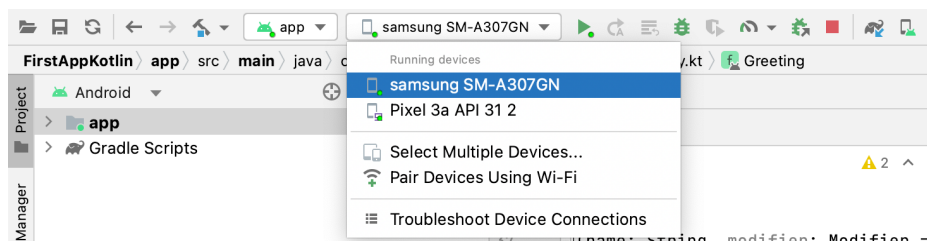
Gambar 1.26 Preview Setelah Ditambah Padding

18. Selamat - Anda telah mem-build aplikasi Android pertama Anda di Compose!

## 1.6. Menjalankan Project di Real Device dan Emulator

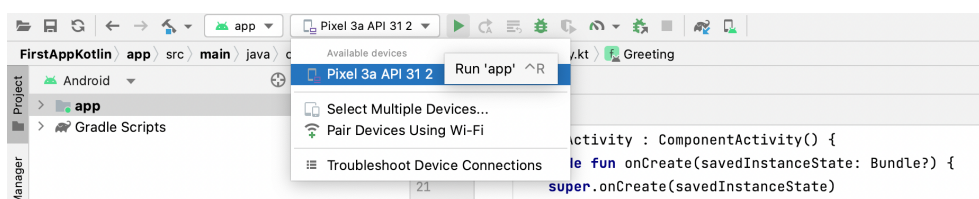
Tahap berikutnya yaitu menjalankan aplikasi. Terdapat 2 cara yaitu menjalankan di **real device dan emulator**. Jika laptop/PC Anda mempunyai spesifikasi yang kurang, sebaiknya Anda memilih menjalankan di real device.

1. Untuk **menjalankan di real device**, Anda harus mengatur device Anda sebagai berikut:
  - a. Hubungkan device Android Anda ke laptop/PC Anda dengan **kabel USB**.
  - b. Aktifkan USB debugging di jendela Developer options device Android Anda dengan cara pilih **Settings > Cari Build Number > Tap 7 kali pada Build Number > Kembali ke Settings** maka **Developer options** telah aktif. Pada jendela Developer options, aktifkan **USB debugging**.
  - c. Untuk Android 11+, Anda juga dapat menjalankan melalui **Pair Devices Using Wi-Fi**. Caranya dengan mengaktifkan Wireless Debugging di jendela Developer options device Android Anda kemudian pilih Pair using QR Code.
  - d. Pada Android Studio, pastikan device Android Anda sudah terpilih sebagai running device. Kemudian klik **Run**  untuk menjalankan aplikasi.



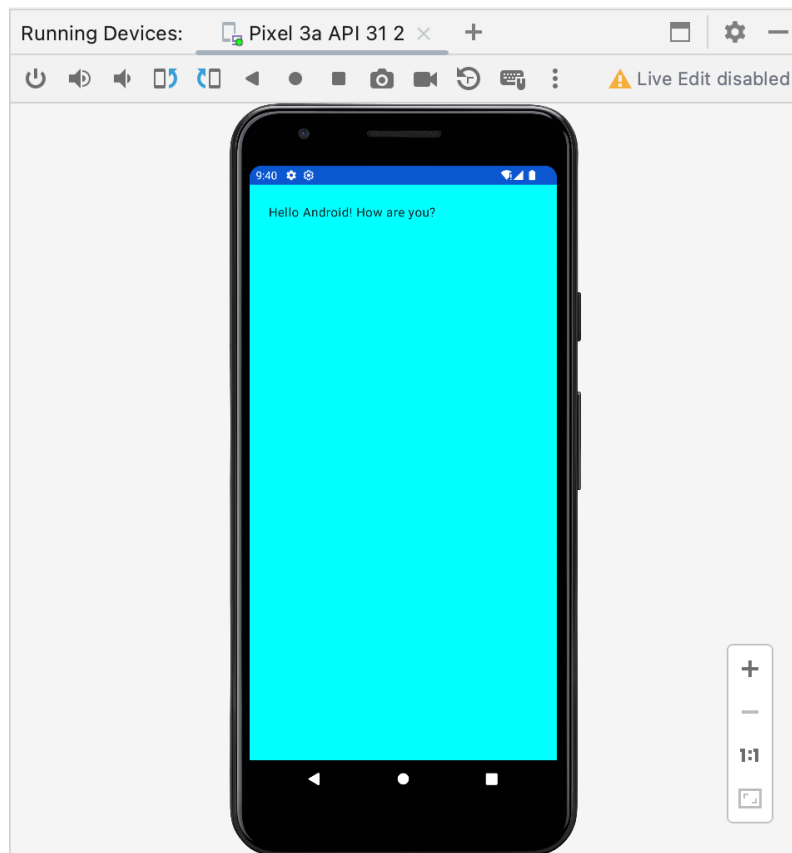
Gambar 1.27 Menjalankan Aplikasi di Real Device

2. Untuk **menjalankan di emulator**, Anda dapat mengikuti langkah-langkah sebagai berikut:
  - a. Di Android Studio, buat **Android Virtual Device (AVD)** yang dapat digunakan emulator untuk menginstal dan menjalankan aplikasi Anda. Jika AVD sudah ada, pilih AVD tempat Anda ingin menjalankan aplikasi.



Gambar 1.28 Menjalankan Aplikasi di Emulator

- b. Klik Run  untuk menjalankan aplikasi.



Gambar 1.29 Aplikasi Berjalan di Emulator

Sekian, proses membuat aplikasi pertama dengan compose telah selesai. Selamat mencoba dan silahkan kerjakan tugasnya.

## 1.7. Tugas Bab 1

Kerjakan tugas berikut secara individu.

**Modifikasi** aplikasi Android pertama Anda dengan ketentuan berikut:

1. Ubah **nama** menjadi nama lengkap Anda.
2. Ubah **tata letak** tulisan menjadi di tengah.
3. Perbesar ukuran **font**.