

LAPORAN PRAKTIKUM

PEMROGRAMAN BERBASIS MOBILE

PERTEMUAN KE-3



Disusun Oleh :

NAMA : Raden Isnawan Argi Aryasatya
NIM : 195410257
JURUSAN : Informatika
JENJANG : S1
KELAS : 5

Laboratorium Terpadu
Sekolah Tinggi Manajemen Informatika Komputer
AKAKOM
YOGYAKARTA

2021

PERTEMUAN KE-3 **(PENGENALAN APLIKASI ANDROID)**

TUJUAN

1. Mahasiswa mampu memahami dan mengetahui tentang aplikasi Android, cara penginstallan Android Studio, debug dengan virtual device dan debug dengan physical device.
2. Mahasiswa mampu memahami tentang Pengembangan Aplikasi Android

DASAR TEORI

Pada modul ini Anda membuat dan menjalankan aplikasi Android pertama Anda, Hello World, pada sebuah emulator dan pada perangkat fisik. Anda juga mengeksplorasi seperti apa proyek Android itu.

Apa yang harus anda ketahui:

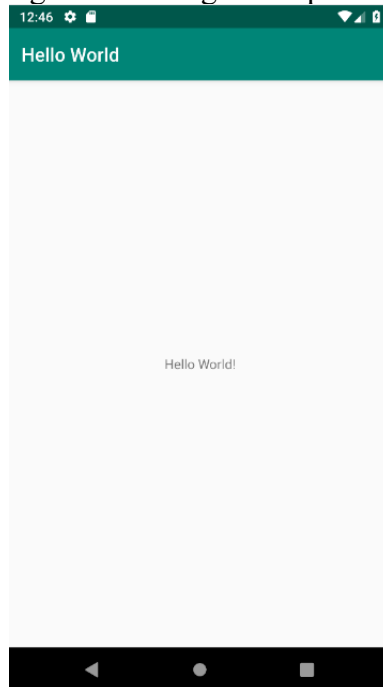
- Anda harus memahami proses pengembangan perangkat lunak umum untuk aplikasi berorientasi objek menggunakan IDE (lingkungan pengembangan terintegrasi) seperti Android Studio.
- Anda harus memiliki setidaknya satu tahun pengalaman dalam pemrograman berorientasi objek, dengan setidaknya pemahaman tentang Java dan Kotlin.

Anda akan mempelajari membangun aplikasi Android dasar di Android Studio, membuat proyek Android dari template, menemukan komponen utama dari proyek Android, dan menjalankan aplikasi Android pada emulator atau perangkat fisik.

PRAKTIK

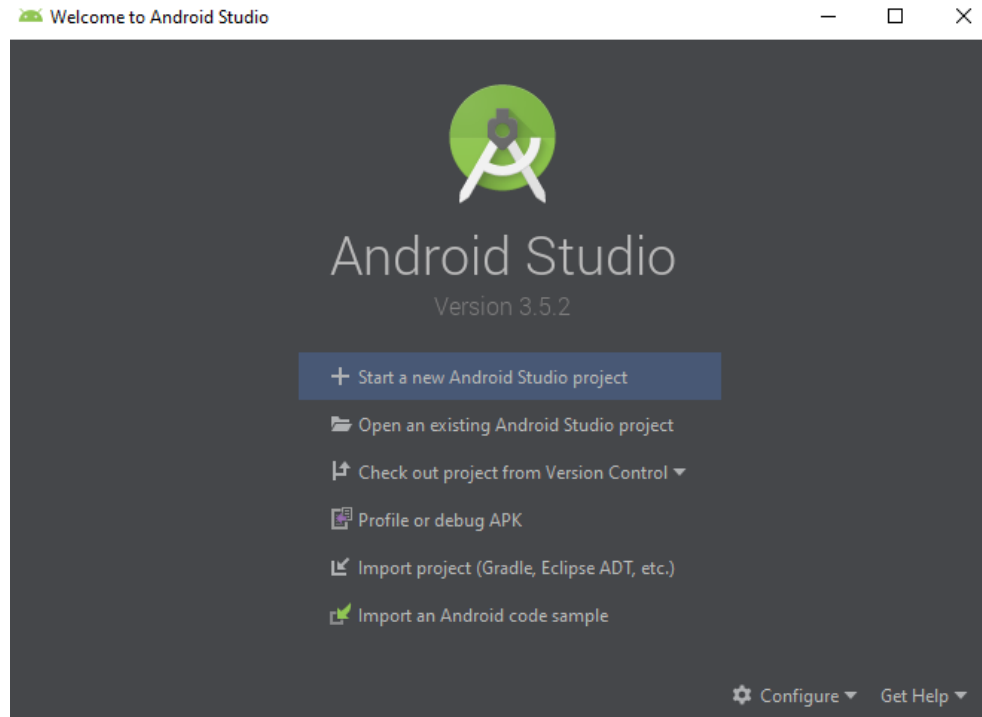
Praktik 1. Menciptakan Proyek HelloWorld

Anda akan membuat aplikasi HelloWorld yang menampilkan string "Hello World" pada layar perangkat virtual Android atau perangkat fisik. Begini tampilannya:

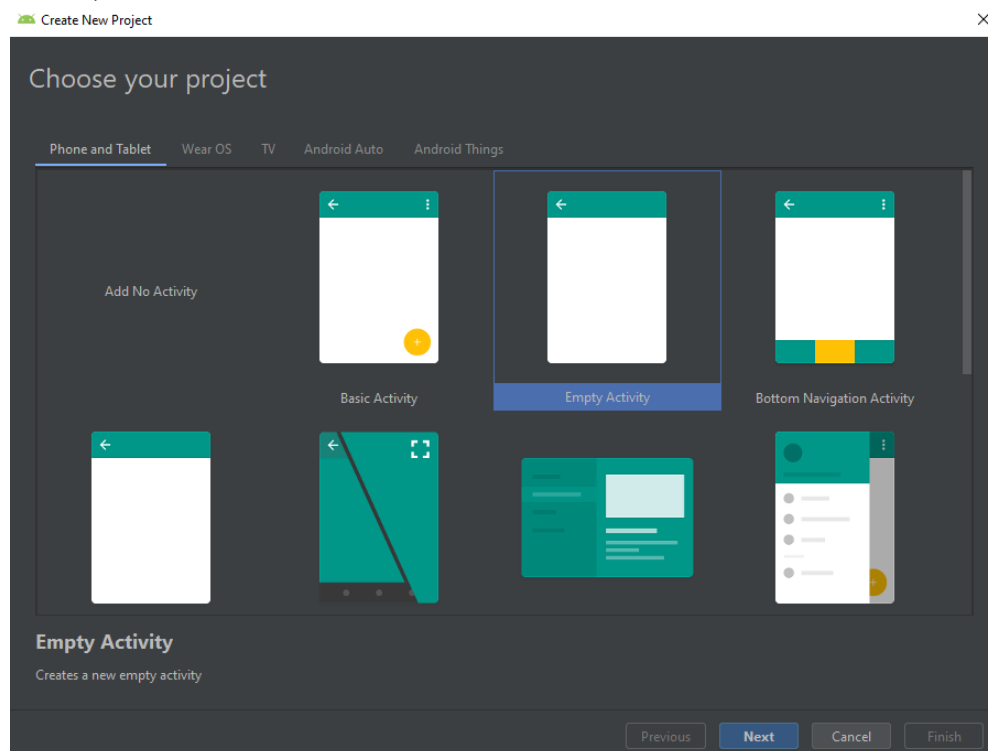


1. Buat proyek hello world
 - a. Buka Android Studio.

- b. Dalam dialog **Welcome to Android Studio**, klik **Start a new Android Studio project**.

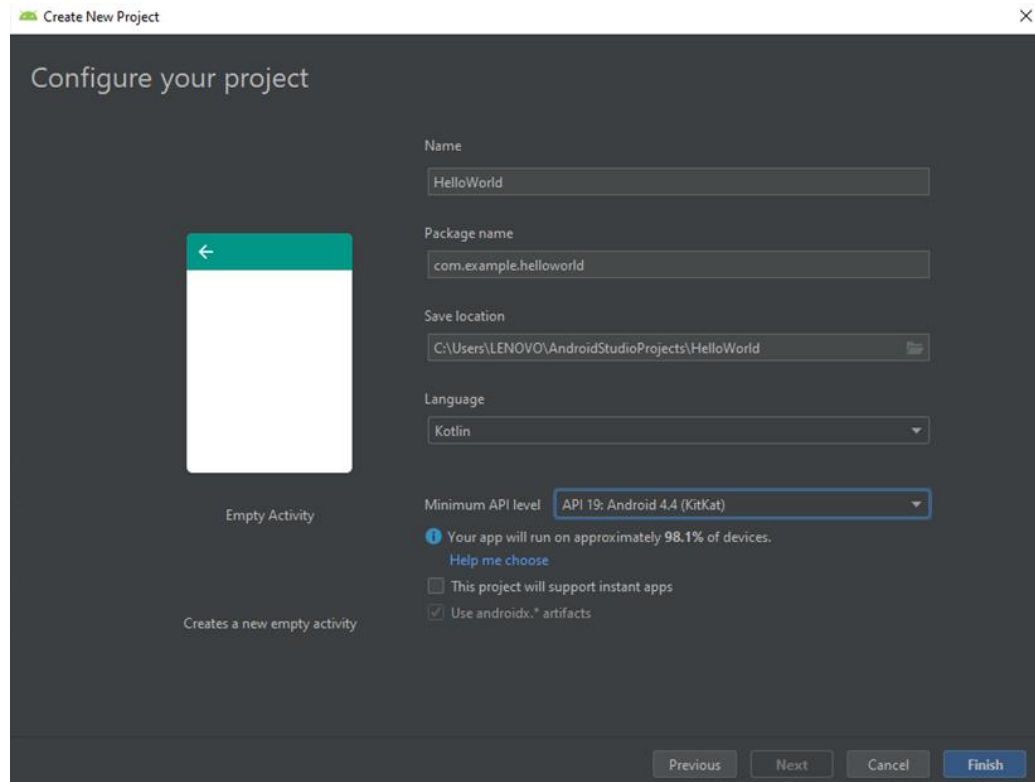


- c. Dialog **Choose your project** muncul. Pilih **Empty Activity** seperti ditunjukkan di bawah, dan klik **Next**.

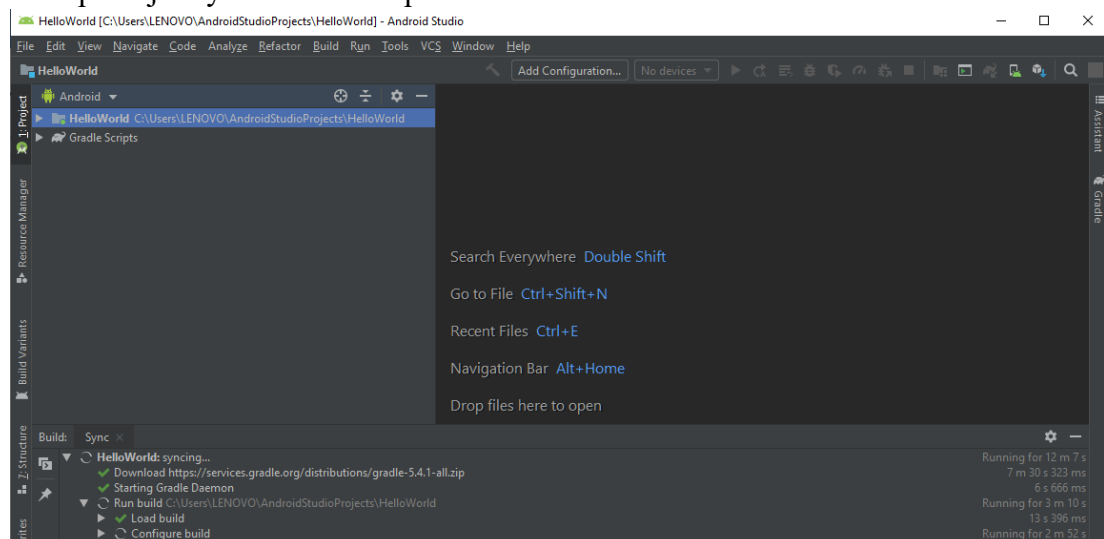


Activity adalah suatu yang terfokus yang dapat dilakukan pengguna. Setiap aplikasi harus memiliki setidaknya satu activity sebagai titik masuknya. Anggap activity entry-point ini sebagai fungsi `main()` di program lain. Suatu activity biasanya memiliki tata letak yang terkait dengannya yang menentukan bagaimana elemen antarmuka pengguna (UI) muncul di layar. Android Studio menyediakan beberapa template Activity untuk membantu Anda memulai.

- d. Dalam dialog **Configure your project**, masukkan "**HelloWorld**" sebagai **Name**.

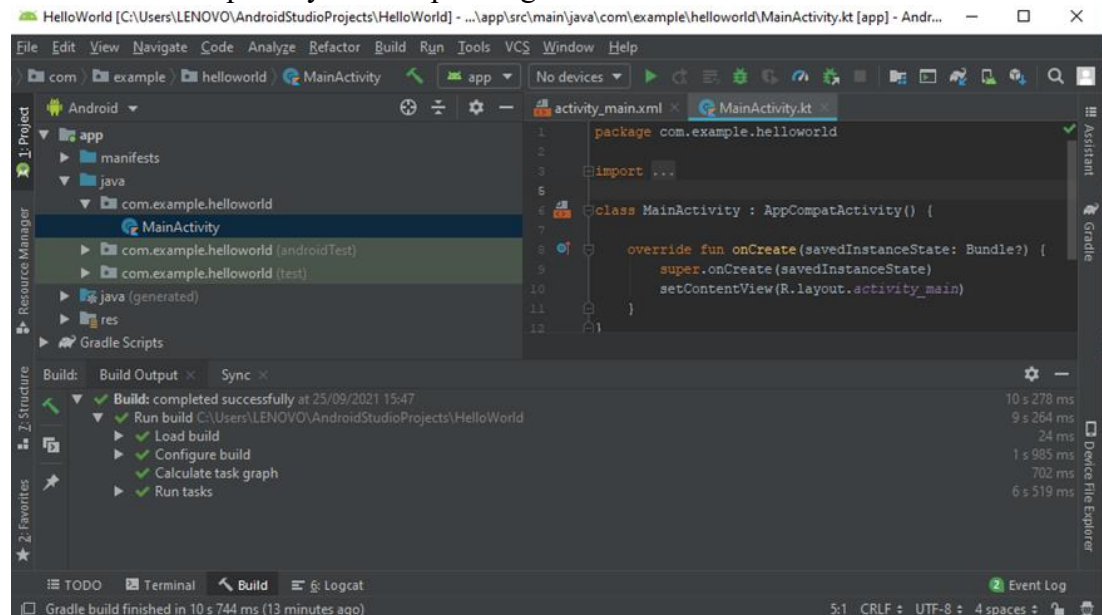


- e. Gunakan **android.example.com** default untuk Package name, atau buat domain unik lainnya. Nilai ini plus nama aplikasi adalah nama paket untuk aplikasi Anda. Jika Anda tidak berencana untuk menerbitkan aplikasi Anda, terima defaultnya. Anda dapat mengubah nama paket aplikasi Anda nanti, tetapi ini adalah pekerjaan tambahan.
- f. Verifikasi bahwa **Save location** default adalah tempat Anda akan menyimpan aplikasi Anda. Jika tidak, ubah lokasi ke direktori pilihan Anda.
- g. Pastikan **Language** adalah **Kotlin**.
- h. Pastikan level API Minimum adalah **API 19: Android 4.4 (KitKat)**.
- i. Pilih checkbox **Use AndroidX artifacts**.
- j. Biarkan checkbox lainnya dikosongkan, dan klik Finish. Jika proyek Anda memerlukan lebih banyak komponen untuk SDK target yang Anda pilih, Android Studio menginstalnya secara otomatis, yang mungkin memakan waktu cukup lama. Ikuti petunjuknya dan terima opsi default.



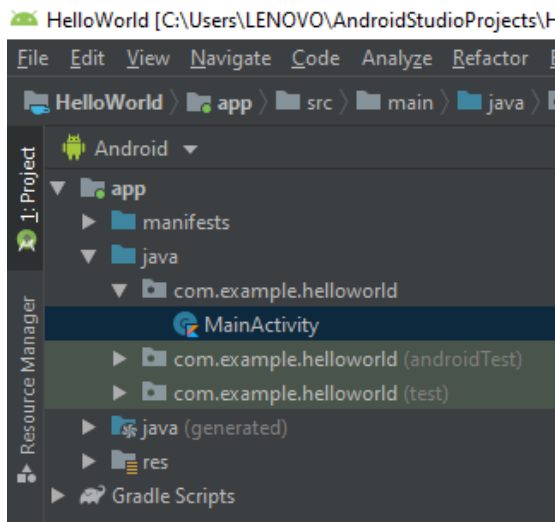
Penjelasan:

Proses tersebut dinamakan proses gradle. Kita harus menunggu proses gradle sampai selesai. Proses ini biasanya memakan waktu selama beberapa menit. Lama proses gradle ini tergantung spesifikasi ram dan processor yang digunakan pada computer kita. Berikut tampilannya ketika proses gradle sudah selesai:



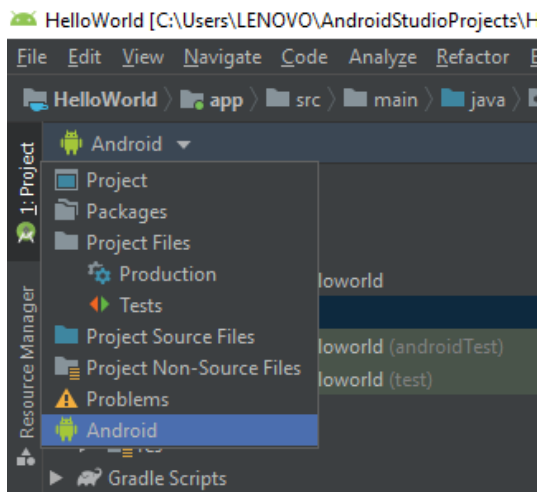
Praktik 2. Explore Android Studio

1. Pilih tab Project. Tab Project berada di kolom tab vertikal di sisi kiri jendela Android Studio

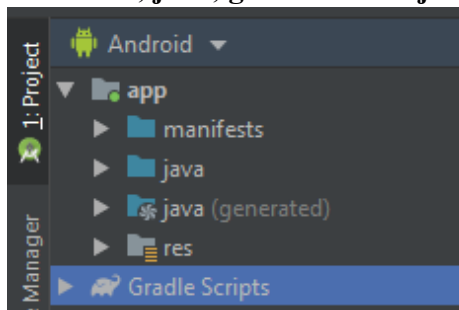


2. Untuk melihat proyek sebagai hierarki proyek Android standar, pilih Android dari menu drop-down di bagian atas panel Project. (Android adalah defaultnya). Anda dapat melihat file proyek dengan berbagai cara, termasuk melihat file tersebut seperti apa yang terlihat dalam hierarki sistem file. Namun, proyek ini lebih mudah digunakan dengan menggunakan tampilan Android.

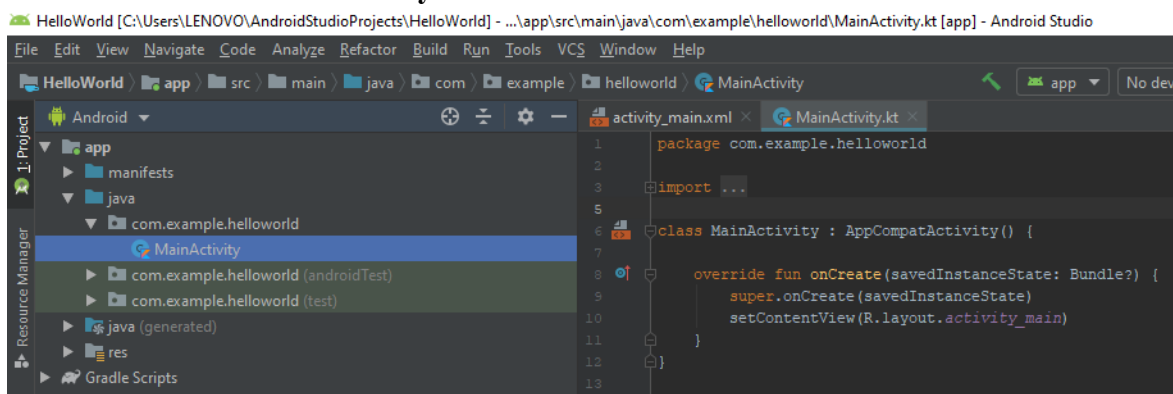
(screenshot di halaman selanjutnya)



3. Di panel **Proyek > Android**, perluas folder app. Di dalam folder app ada empat subfolder: **manifests**, **java**, **generateJava/java(generated)**, dan **res**.



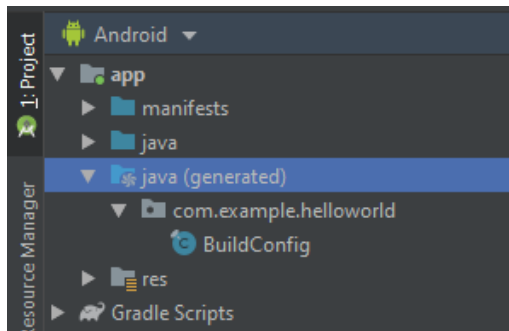
4. Perluas folder **java**, dan kemudian perluas folder **com.example.android.HelloWorld** untuk melihat file Kotlin **MainActivity**.



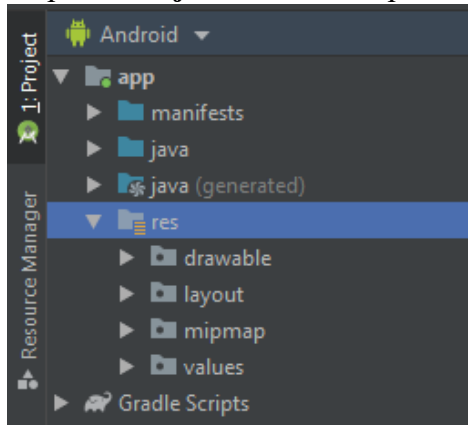
Folder **java** berisi semua kode Kotlin utama untuk aplikasi Android. Ada alasan historis mengapa kode Kotlin Anda muncul di folder java. Konvensi itu memungkinkan Kotlin untuk beroperasi tanpa hambatan dengan kode yang ditulis dalam bahasa pemrograman Java, bahkan dalam proyek dan aplikasi yang sama.

5. Perhatikan folder **generatedJava/java(generated)**. Folder ini berisi file yang dihasilkan Android Studio saat membangun aplikasi. Jangan edit apa pun di folder ini, karena perubahan Anda mungkin ditimpa ketika Anda membangun kembali aplikasi. Tetapi penting untuk mengetahui tentang folder ini ketika Anda perlu melihat file-file ini selama debugging

(screenshot di halaman selanjutnya)

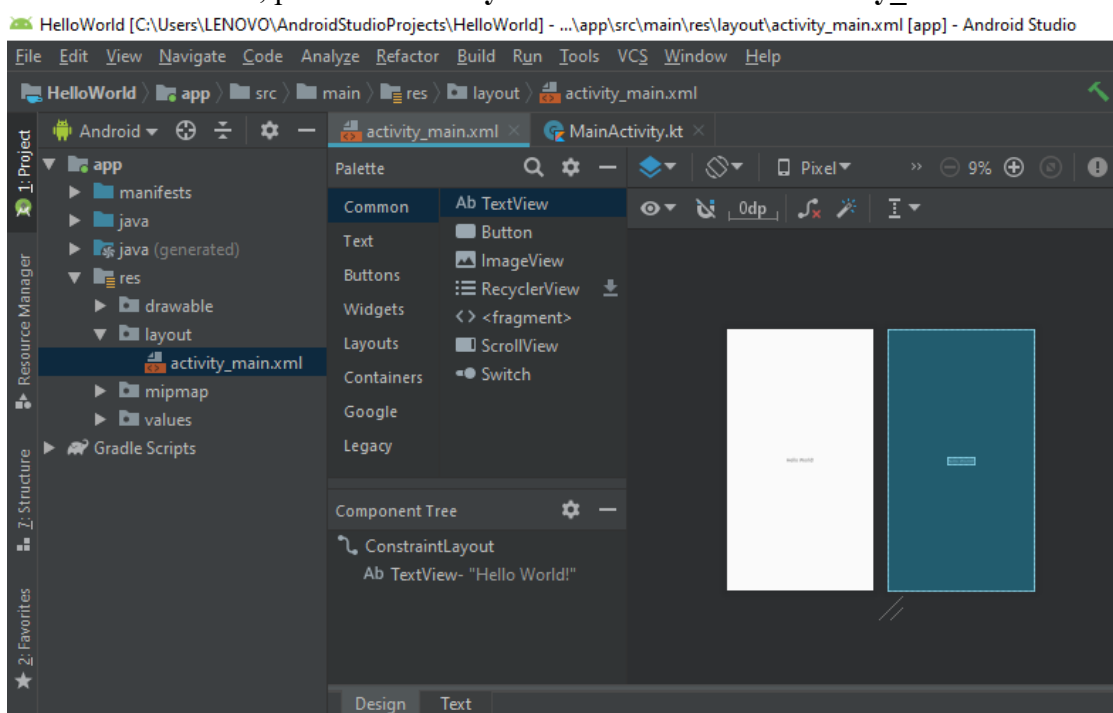


6. Di panel **Project** > **Android**, perluas folder **res**



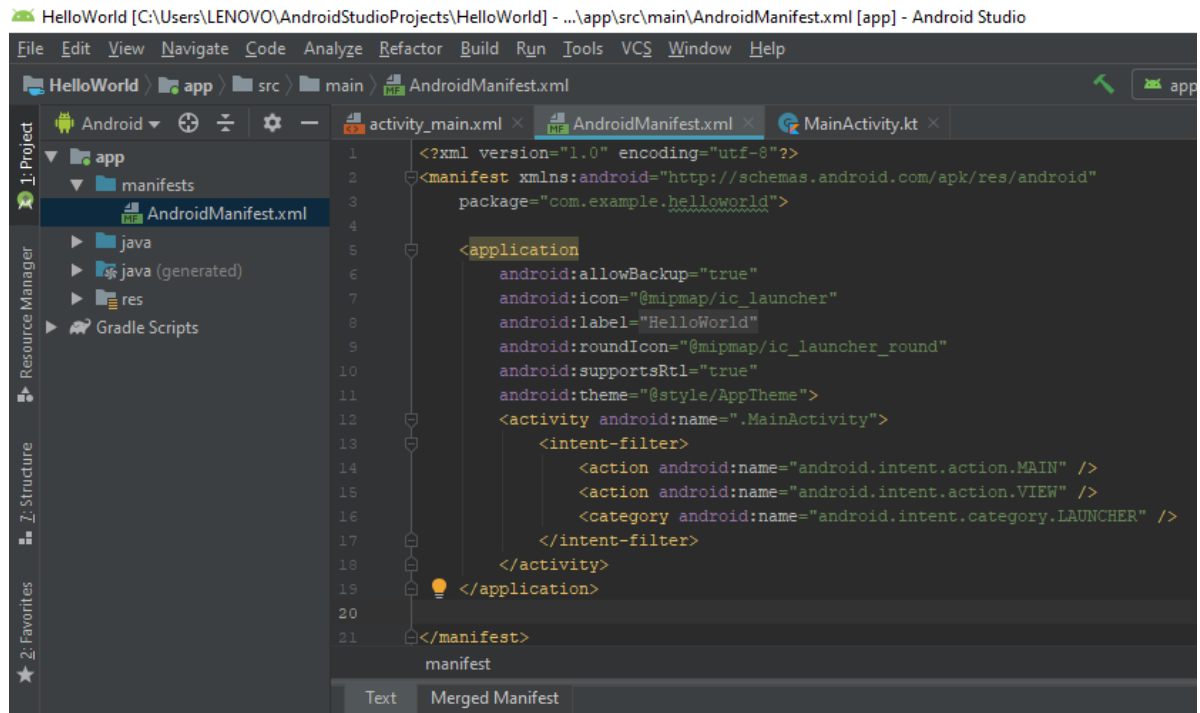
Folder **res** menyimpan sumber daya. Sumber daya di Android adalah konten statis yang digunakan dalam aplikasi Anda. Sumber daya termasuk gambar, string teks, tata letak layar, style, dan nilai-nilai seperti warna heksadesimal atau dimensi standar. Aplikasi Android memisahkan kode dan sumber daya Kotlin sebanyak mungkin. Itu membuatnya lebih mudah untuk menemukan semua string atau ikon yang digunakan di UI aplikasi. Juga, ketika Anda mengubah salah satu file sumber daya ini, perubahan itu berlaku di mana-mana bahwa file tersebut digunakan dalam aplikasi.

7. Di dalam folder **res**, perluas folder **layout** untuk melihat file **activity_main.xml**



Activity Anda biasanya dikaitkan dengan file layout UI, didefinisikan sebagai file XML di direktori **res/layout**. File layout itu biasanya dinamai berdasarkan activity. Dalam hal ini nama activity adalah **MainActivity**, jadi layout yang terkait adalah **activity_main**.

8. Buka folder **manifest** dan klik dua kali **AndroidManifest.xml** untuk membukanya. File **AndroidManifest.xml** mencakup detail yang dibutuhkan sistem Android untuk menjalankan aplikasi Anda, termasuk kegiatan apa yang merupakan bagian dari aplikasi.

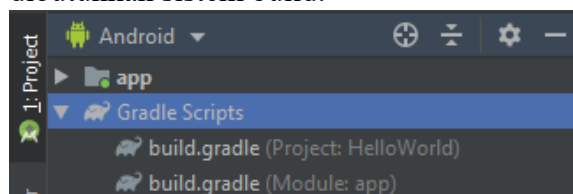


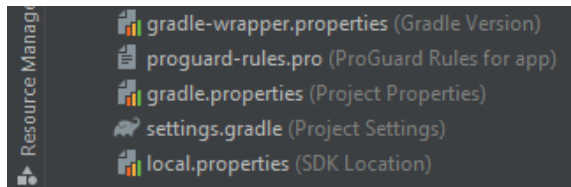
9. Perhatikan bahwa **MainActivity** dirujuk dalam elemen **<activity>**. **Activity** apa pun di aplikasi Anda harus dinyatakan dalam manifest. Berikut ini contoh untuk **MainActivity**:

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

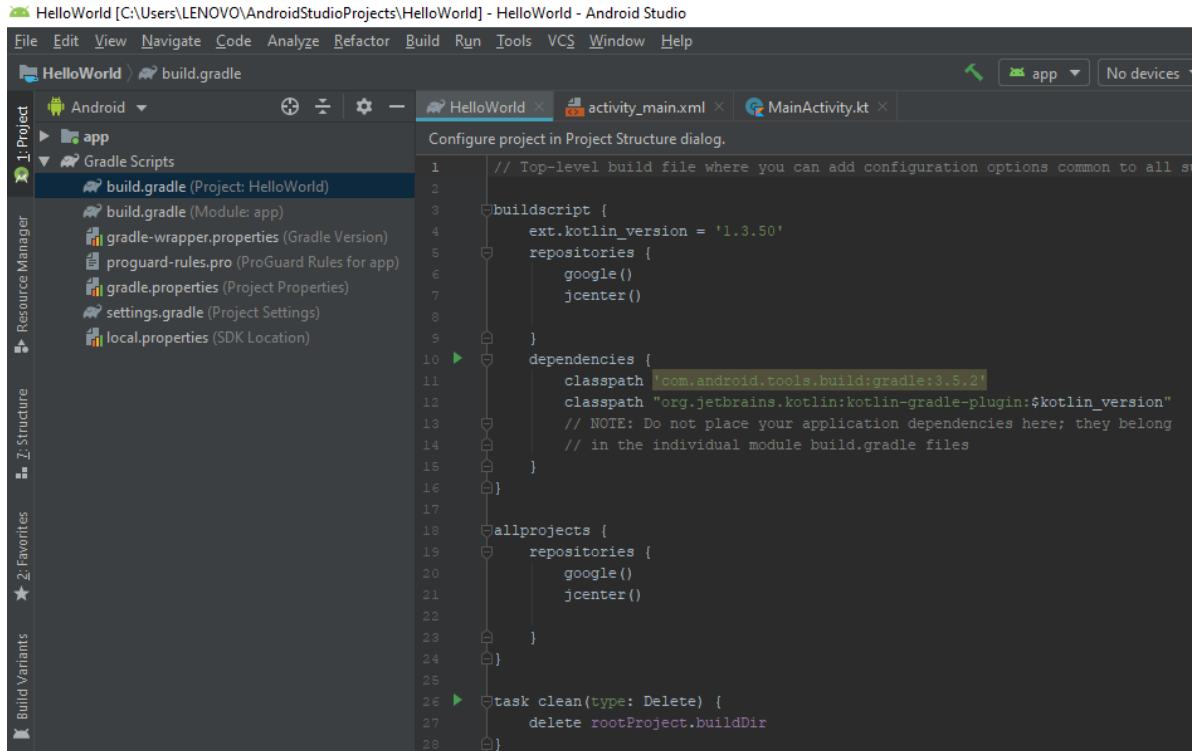
10. Perhatikan elemen **<intent-filter>** di dalam **<activity>**. Elemen **<action>** dan **<category>** dalam filter intent ini memberi tahu Android tempat memulai aplikasi ketika pengguna mengklik ikon peluncur (launcher). File **AndroidManifest.xml** juga merupakan tempat Anda menentukan izin (permission) apa pun yang dibutuhkan aplikasi Anda. Izin mencakup kemampuan aplikasi Anda untuk membaca kontak telepon, mengirim data melalui internet, atau mengakses perangkat keras seperti kamera perangkat.

11. Buka folder **Gradle Scripts**. Di panel **Project > Android**, folder ini berisi semua file yang dibutuhkan sistem build.



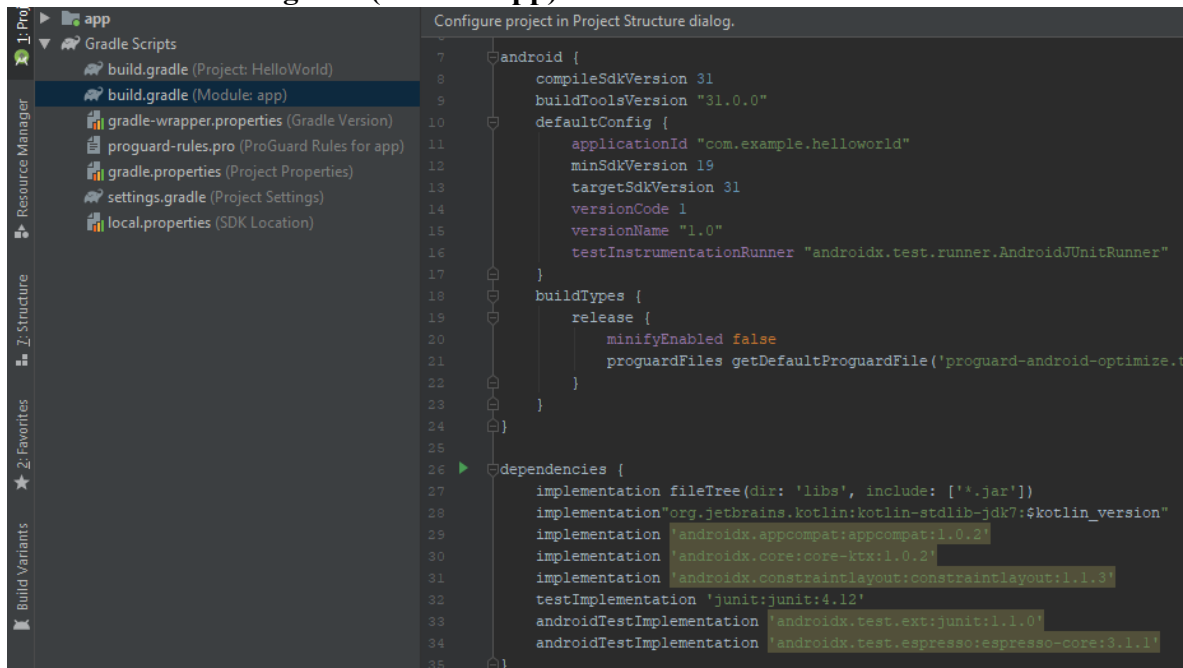


12. Perhatikan file **build.gradle(Project: HelloWorld)**.



File ini berisi opsi konfigurasi yang umum untuk semua modul yang membentuk proyek Anda. Setiap proyek Android Studio berisi satu, file build Gradle tingkat atas. File ini mendefinisikan repositori dan dependensi Gradle yang umum untuk semua modul dalam proyek.

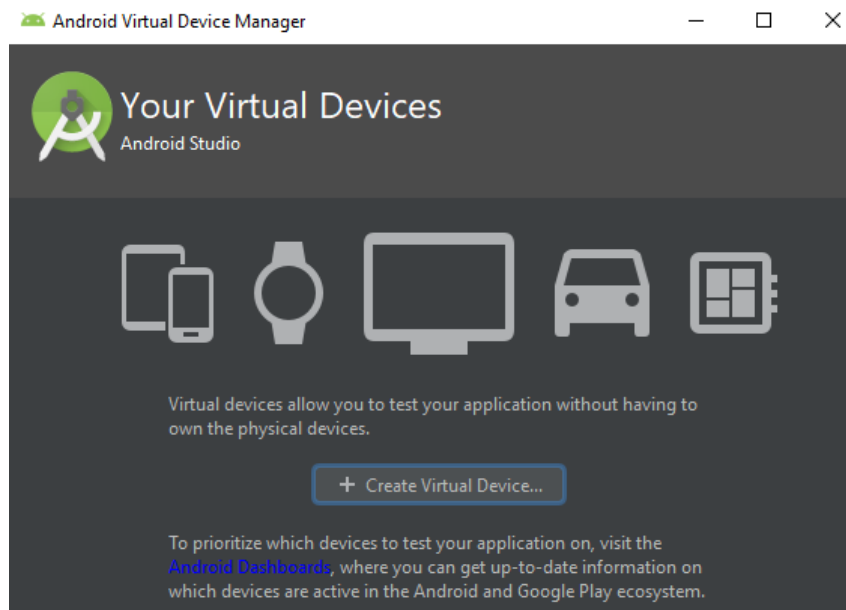
13. Perhatikan file **build.gradle(Module:app)**



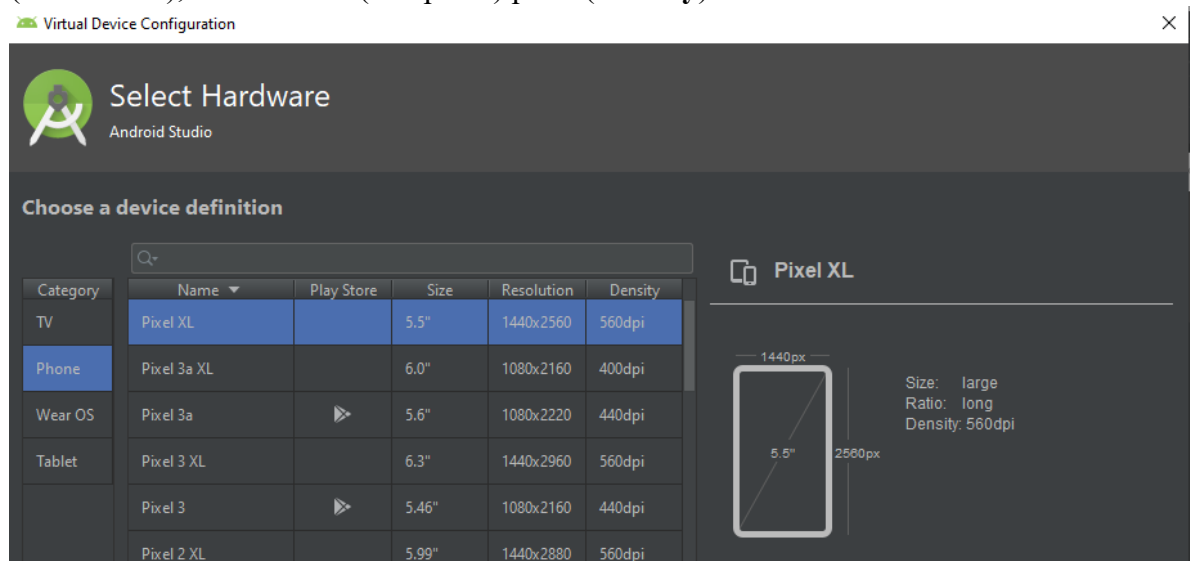
Selain file build.gradle tingkat proyek, setiap modul memiliki file build.gradle sendiri. File build.gradle tingkat modul memungkinkan Anda mengkonfigurasi pengaturan build untuk setiap modul. (Aplikasi HelloWorld hanya memiliki satu modul, modul untuk aplikasi itu sendiri.) File build.gradle ini adalah yang paling sering Anda edit ketika mengubah konfigurasi pembangunan tingkat aplikasi. Misalnya, Anda mengedit file build.gradle ini ketika Anda mengubah level SDK yang didukung aplikasi Anda, atau ketika Anda mendeklarasikan dependensi baru di bagian dependensi.

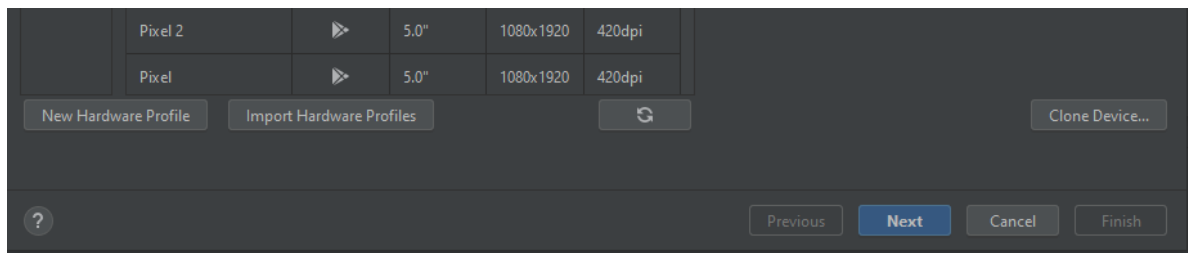
Praktik 3. Menjalankan Aplikasi pada Virtual Device (Emulator)

1. Di Android Studio, pilih **Tools > AVD Manager**, atau klik ikon **AVD Manager** di toolbar. Dialog **Your Virtual Devices** muncul. Jika Anda sudah membuat perangkat virtual, dialog menunjukkannya. Jika tidak, Anda melihat daftar kosong.

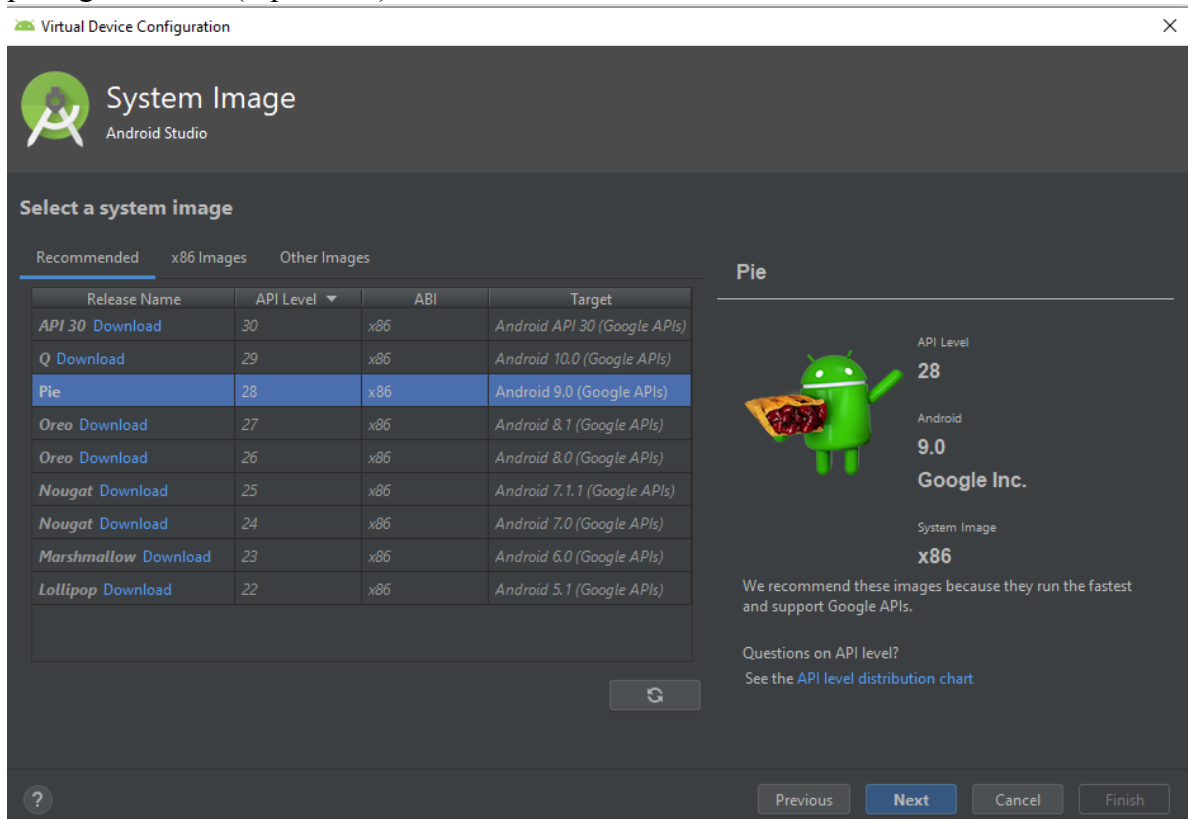


2. Klik **+Create Virtual Device**. Dialog **Select Hardware** muncul, menunjukkan daftar perangkat hardware yang sudah dikonfigurasi sebelumnya. Untuk setiap perangkat, tabel menyediakan kolom untuk ukuran layar diagonal (**Size**), resolusi layar dalam pixels (**Resolution**), dan densitas (kerapatan) pixel (**Density**).

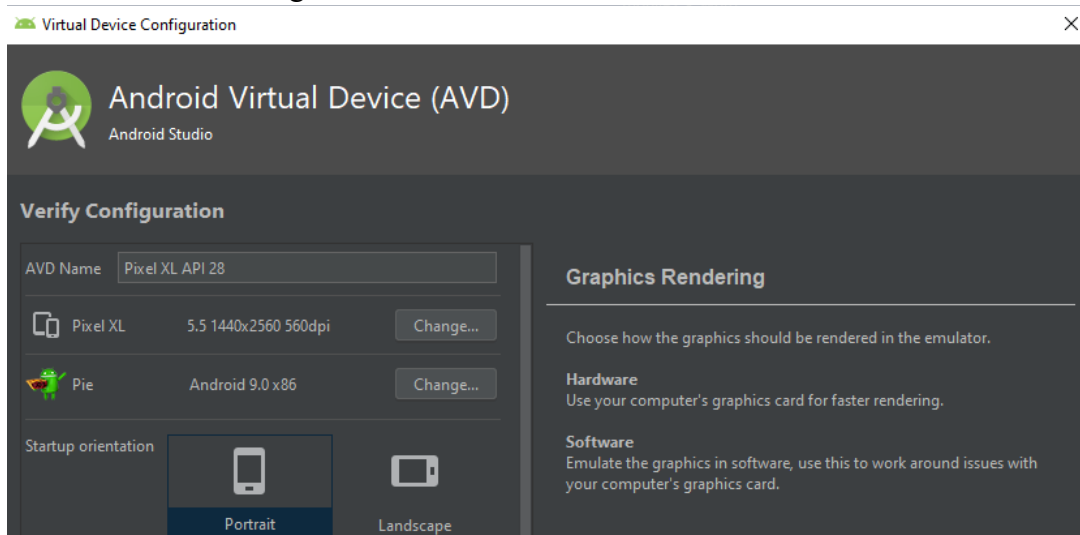


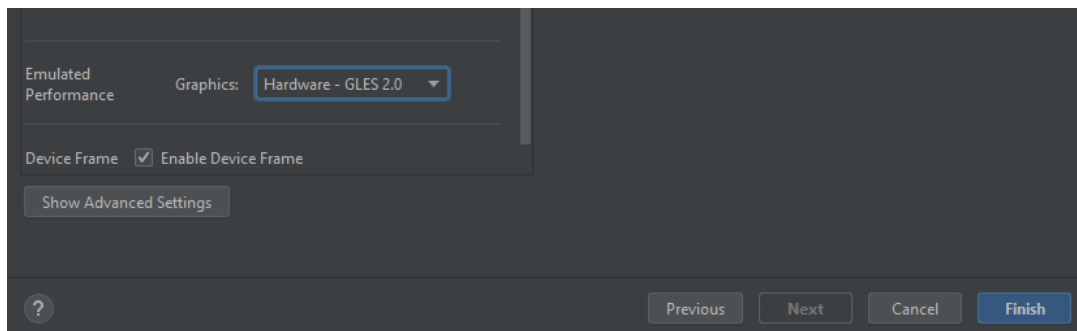


- Pilih perangkat seperti **Nexus 5x** atau **Pixel XL**, dan klik **Next**. Dialog System Image muncul.
- Klik tab **Recommended**, dan pilih versi sistem Android mana yang akan dijalankan pada perangkat virtual (seperti Pie).

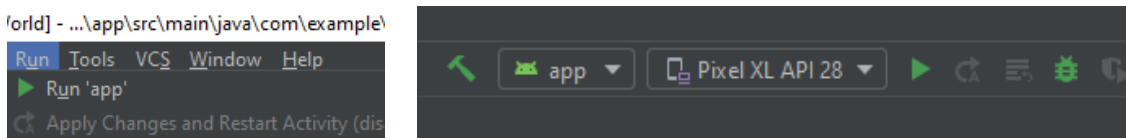


- Setelah Anda memilih system image, klik **Next**. Dialog **Android Virtual Device (AVD)** terbuka. Periksa konfigurasi Anda dan klik **Finish**.

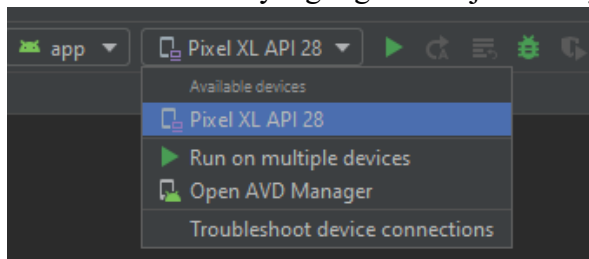




6. Di Android Studio, pilih **Run > Run app** atau klik ikon **Run** di toolbar untuk menjalankan emulator

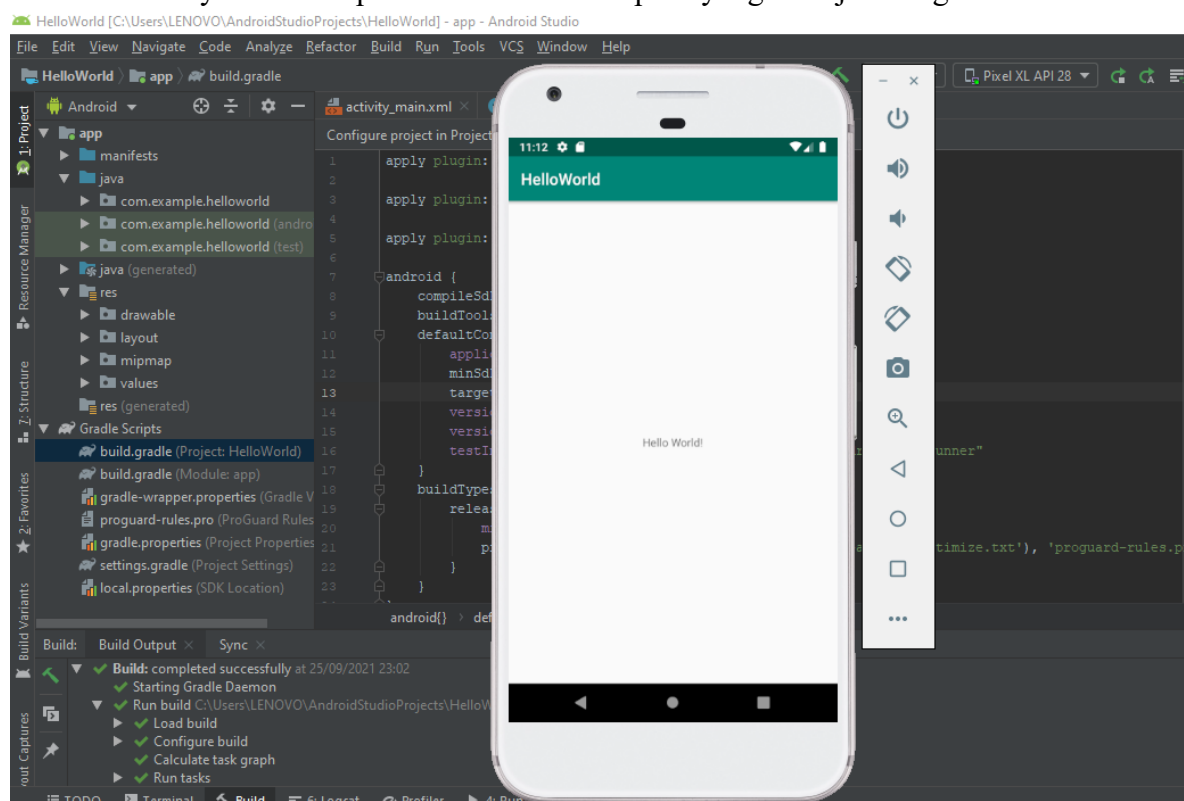


7. Pilih virtual device yang ingin anda jalankan yaitu **Pixel XL API 28**



Emulator memulai dan menjalankan boot seperti perangkat fisik. Tergantung pada kecepatan komputer Anda, proses ini mungkin memakan waktu. Aplikasi Anda dibuat, dan ketika emulator siap, Android Studio mengunggah aplikasi ke emulator dan menjalankannya.

Anda seharusnya melihat aplikasi HelloWorld seperti yang ditunjukkan gambar berikut:

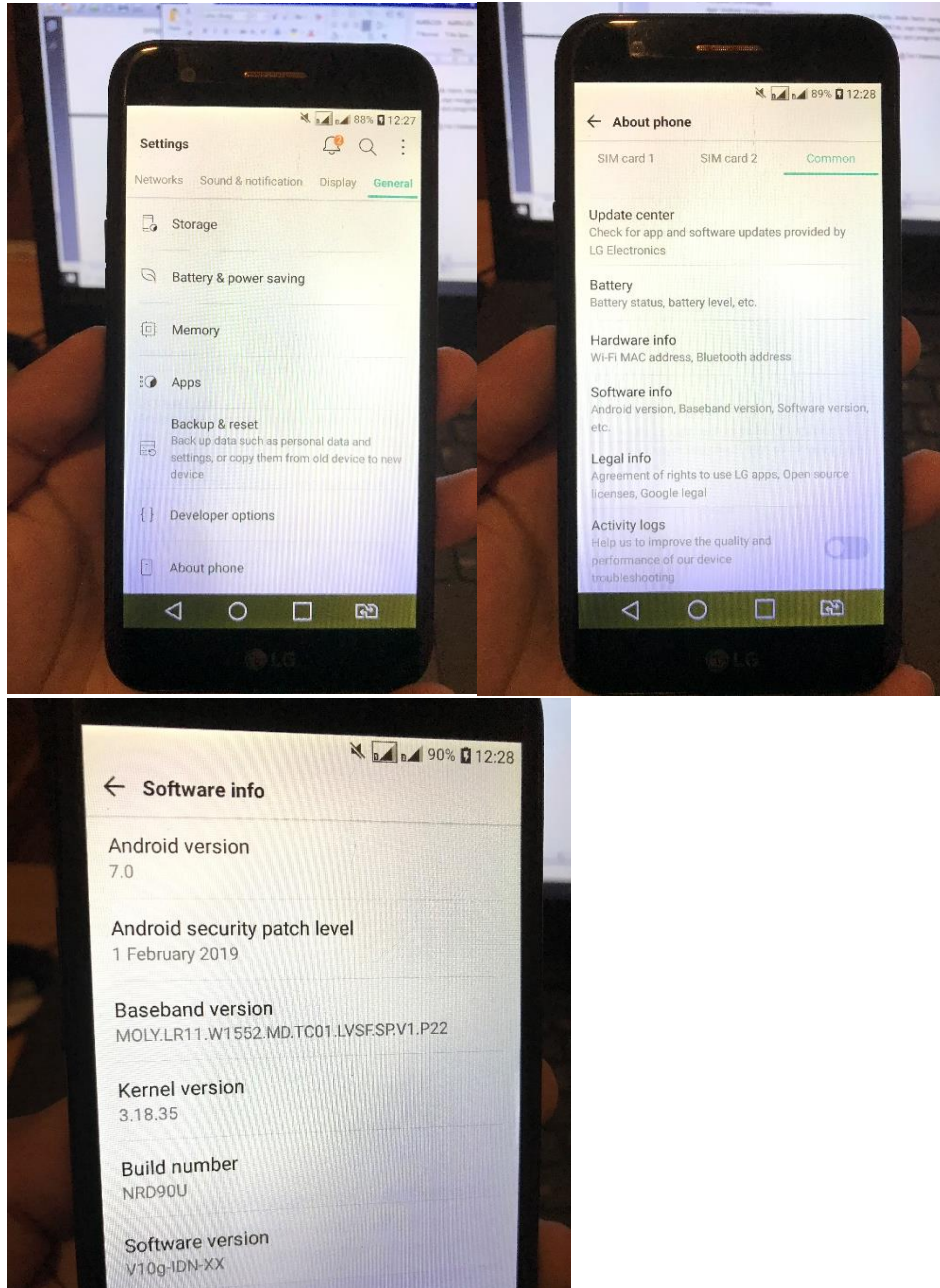


Praktik 4. Menjalankan Aplikasi pada Perangkat Fisik

1. Hidupkan USB debugging

Agar Android Studio berkomunikasi dengan perangkat Android Anda, Anda harus mengaktifkan debugging USB di pengaturan opsi Pengembang perangkat. Pada praktik ini, saya menggunakan LG K10 2017 dengan versi Android 7.0 atau Nougat. Berikut cara menampilkan opsi pengembang dan mengaktifkan debugging USB pada smartphone saya:

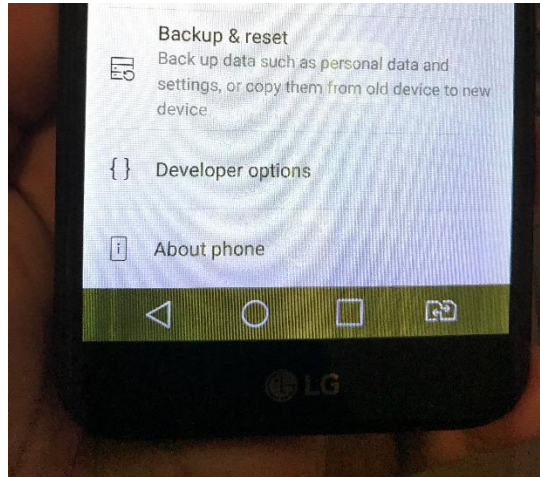
- 1) Buka **settings**, klik **General**, scroll ke bawah lalu klik **About phone**, klik bar Common, scroll ke bawah lalu klik **Software info**, kemudian klik **Build number** 7x



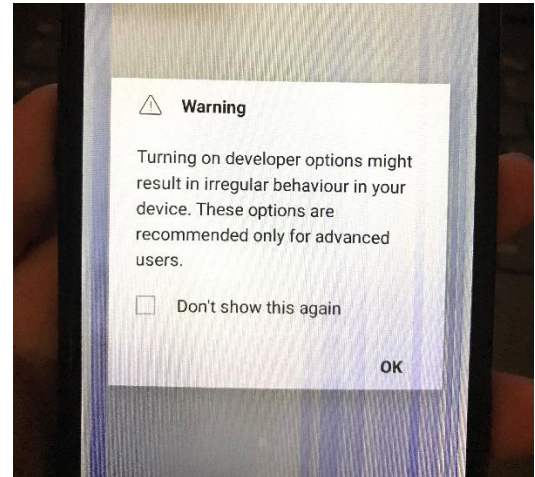
- 2) Kembali ke **General**, lalu klik **Developer options**, jika ada **Warning** klik OK saja, kemudian kita klik tombol switch disamping kanan **Developer options**, jika ada **Allow Development Settings?**, klik OK. Developer options berhasil dihidupkan.

Langkah-langkah dengan foto:
(di halaman selanjutnya)

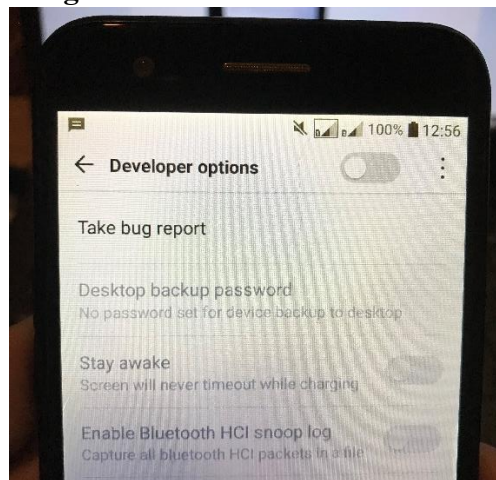
Langkah 1. General -> Developer options



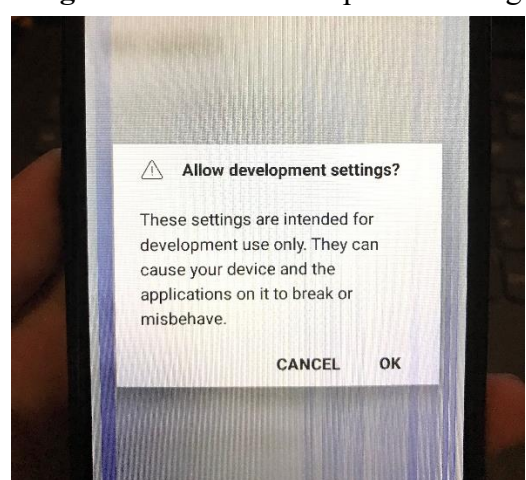
Langkah 2. Warning -> klik OK



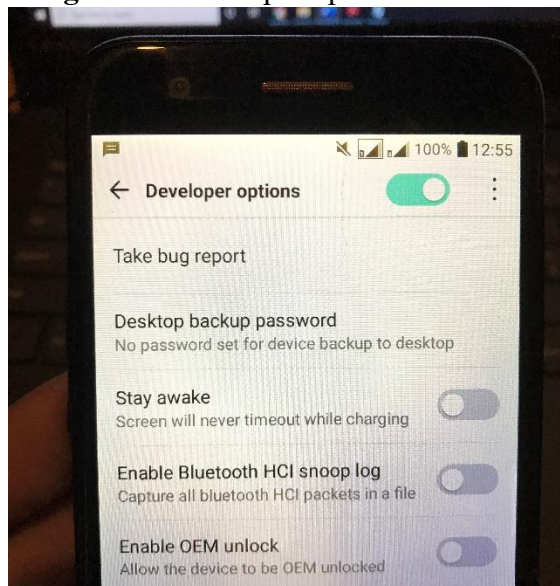
Langkah 3. Klik tombol switch ON



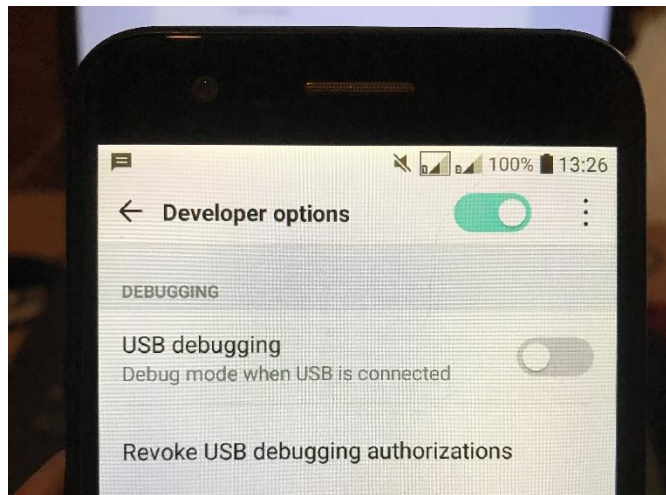
Langkah 4. Allow Development Settings



Langkah 5. Developer options berhasil dihidupkan

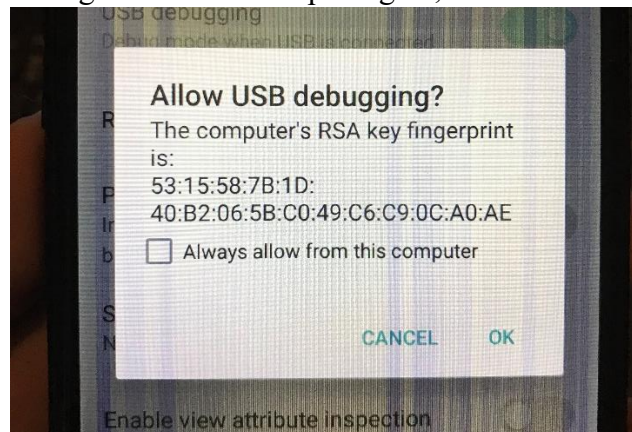


- 3) Pilih USB debugging dengan klik tombol switch supaya menjadi hijau (hidup) (foto di halaman selanjutnya)

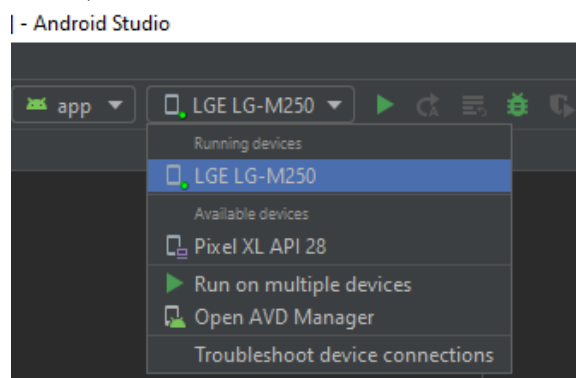


2. Jalankan aplikasi Anda di perangkat Android

- a. Hubungkan perangkat Android ke mesin pengembangan Anda dengan kabel USB. Dialog akan muncul di perangkat, meminta untuk mengizinkan USB debugging.

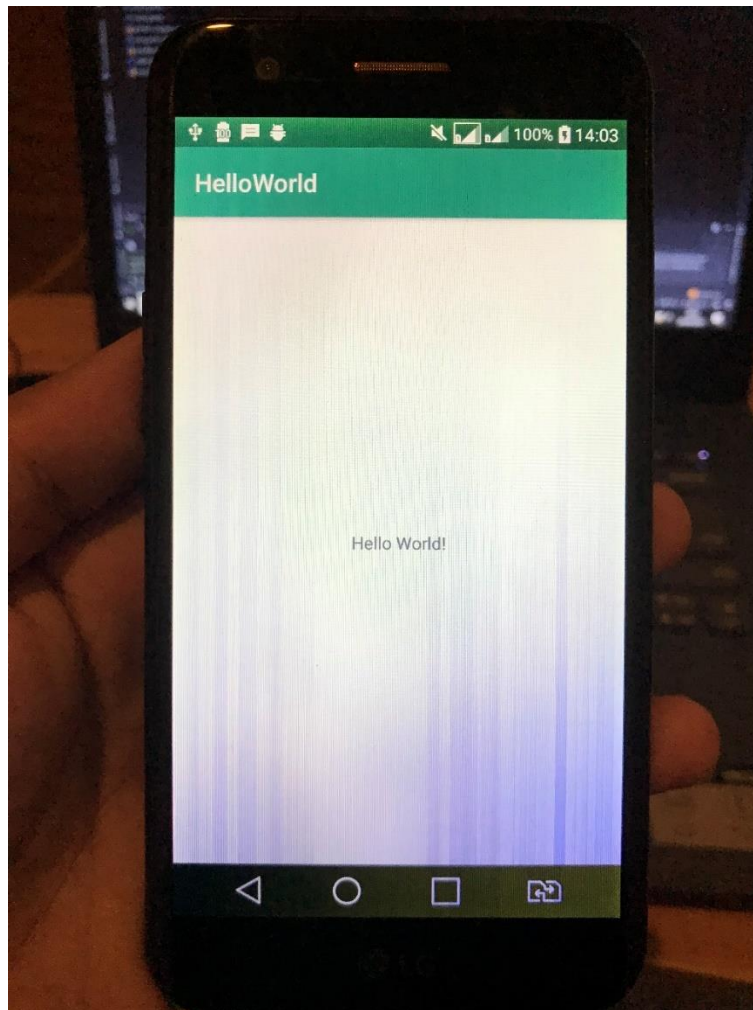


- b. Boleh pilih opsi **Always allow** untuk mengingat komputer ini. Ketuk OK.
- c. Di android studio, menuju ke toolbar untuk memilih device LG K10 2017 (LGE LG-M250)



- d. Kemudian ketuk tombol Run di sebelah kanan device. Android Studio menginstal aplikasi pada perangkat Anda dan menjalankannya.

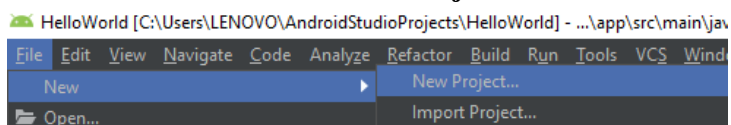
(foto di halaman selanjutnya)



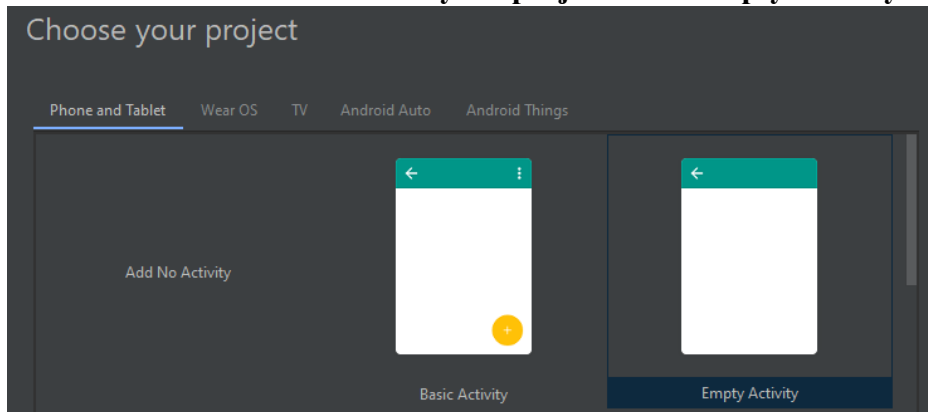
LATIHAN

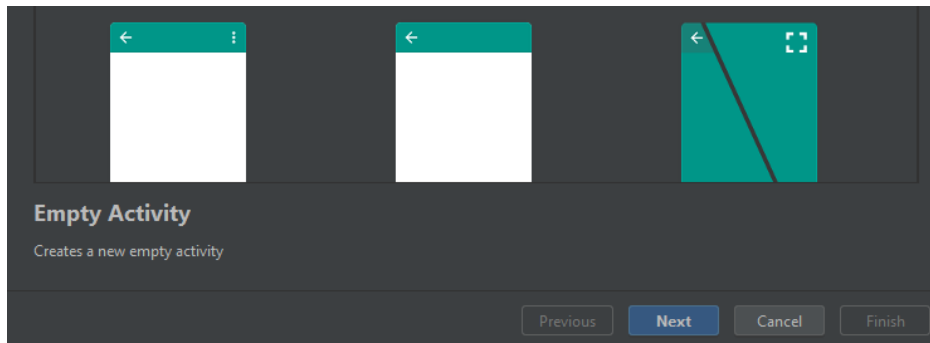
Latihan 1. Buat Proyek Baru di Android Studio

1. Pilih menu **File > New > New Project**

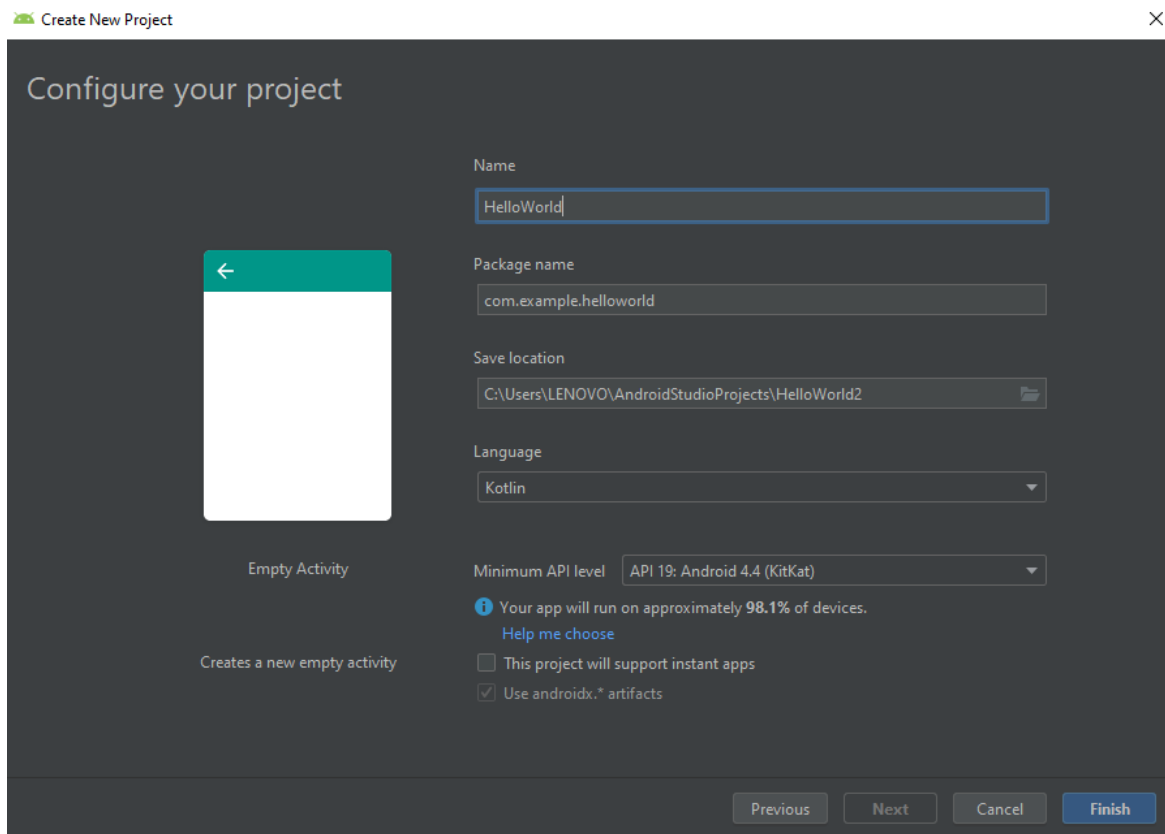


2. Kemudian muncul menu **Choose your project**. Pilih **Empty Activity**. Lalu klik **Next**

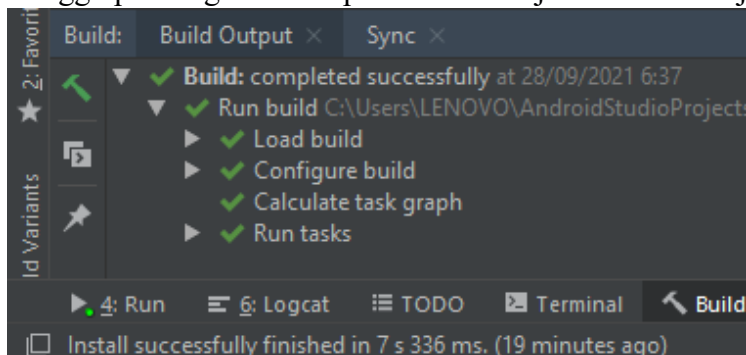




3. Di dialog **Configure your project**, masukkan nama “HelloWorld”. Tidak masalah nama nya sama dengan project tadi, karena project ini akan otomatis tersimpan di folder yang berbeda. Setting lain dibiarkan default saja. Lalu klik **Finish**



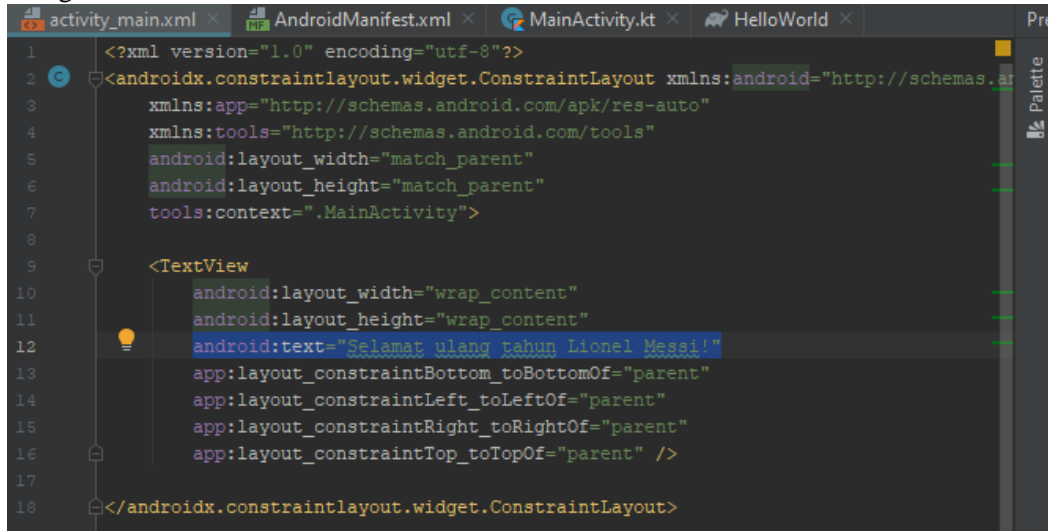
4. Tunggu proses gradle sampai selesai. Project baru sudah jadi.



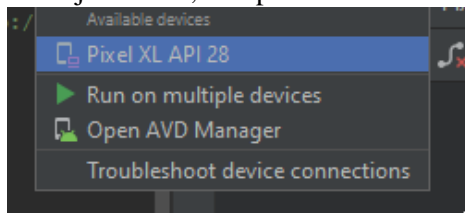
(latihan 2 ada di halaman selanjutnya)

Latihan 2. Ubah salam "Hello World" menjadi "Selamat Ulang Tahun" dan nama seseorang yang baru saja berulang tahun.

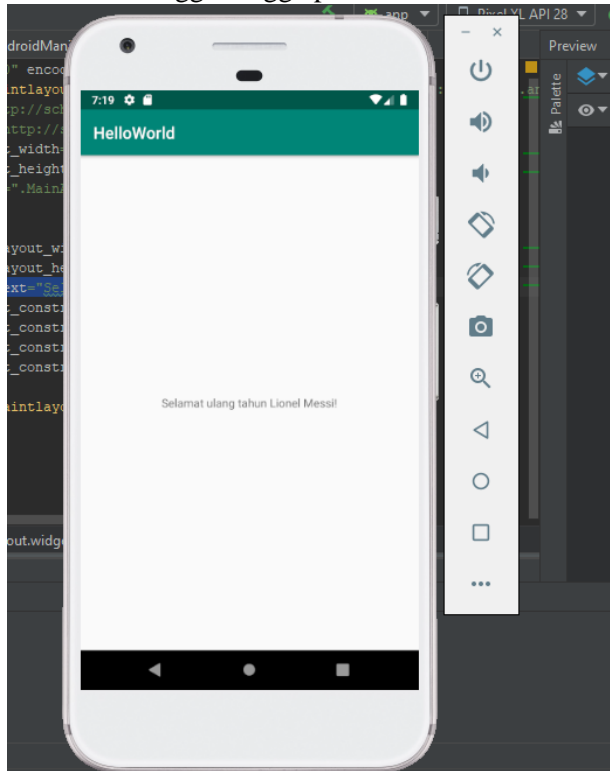
1. Di project HelloWorld kedua yang sudah kita buat tadi, langsung kita masuk ke **res > layout > activity_main.xml**. Di dalam tag **<TextView>**, kita ganti tulisan "Hello World!" menjadi "Selamat ulang tahun Lionel Messi!"



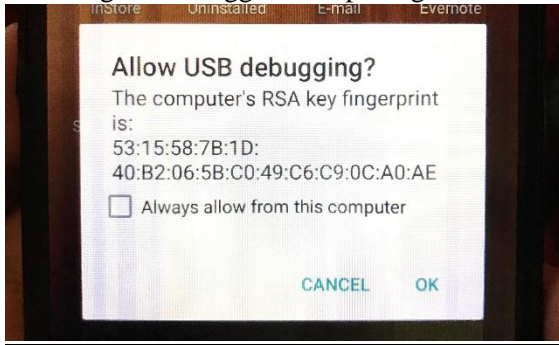
2. Menuju toolbar, lalu pilih virtual device emulator yang sudah kita buat di praktik tadi lalu klik run



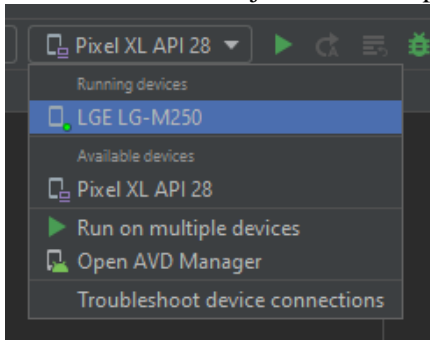
3. Kemudian tunggu hingga proses build selesai. Jika berhasil, maka emulator akan menampilkan:



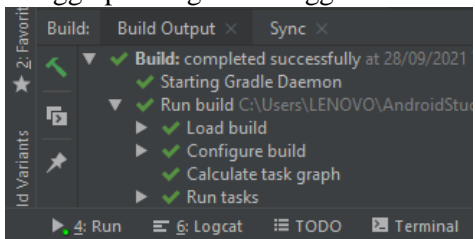
4. Sekarang kita menggunakan perangkat fisik. Kita lakukan USB Debugging terlebih dahulu.



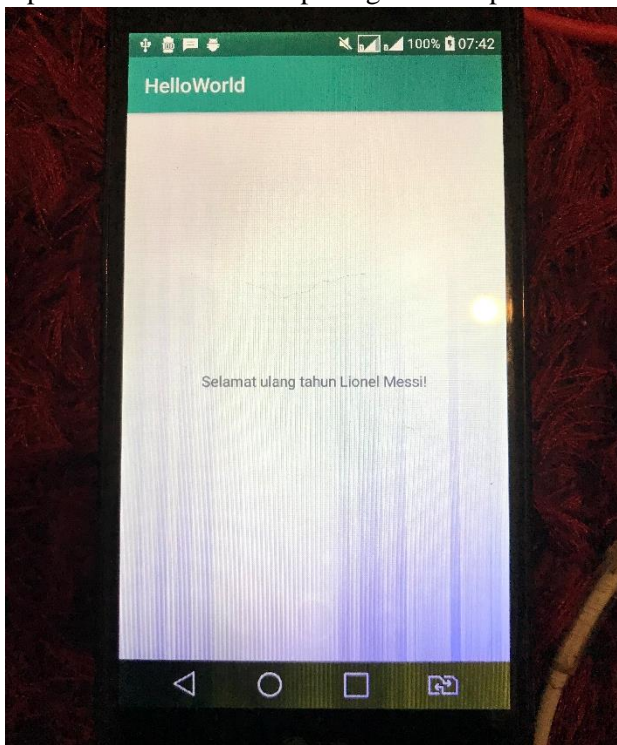
5. Kemudian kita menuju toolbar dan pilih LGE LG-M250 yaitu perangkat smartphone milik saya.



6. Tunggu proses gradle hingga selesai



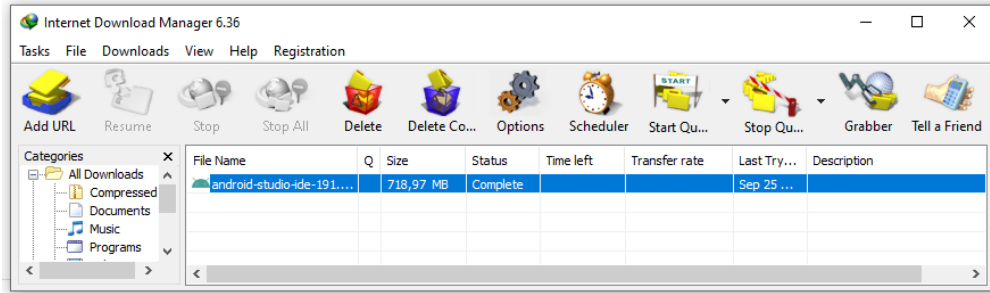
7. Aplikasi akan terbuka di perangkat smartphone



TUGAS

Tugas 1. Install Android Studio pada perangkat komputer anda masing-masing di rumah.

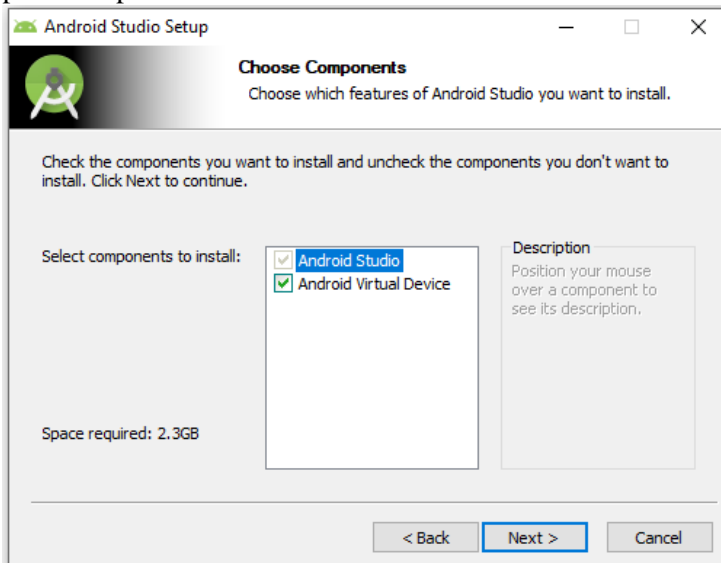
1. Sebelum melakukan instalasi, hal pertama yang harus dilakukan adalah mengunduh Android Studio



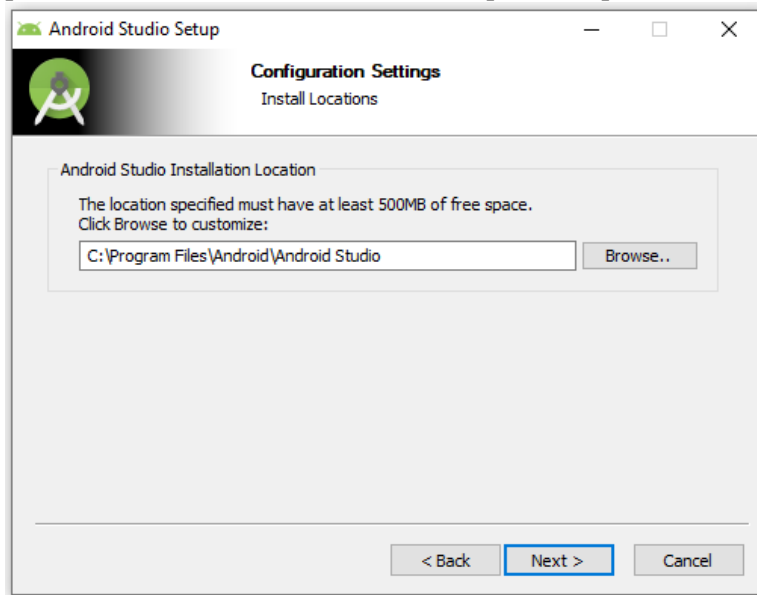
2. Setelah itu buka file instalasi Android Studio. Akan muncul halaman Welcome to Android Studio Setup. Klik Next.



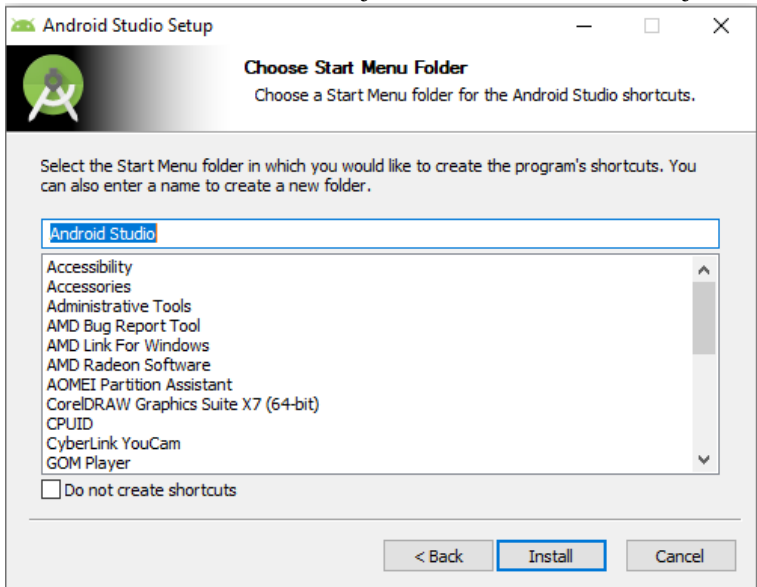
3. pilih komponen tambahan untuk install Android Studio



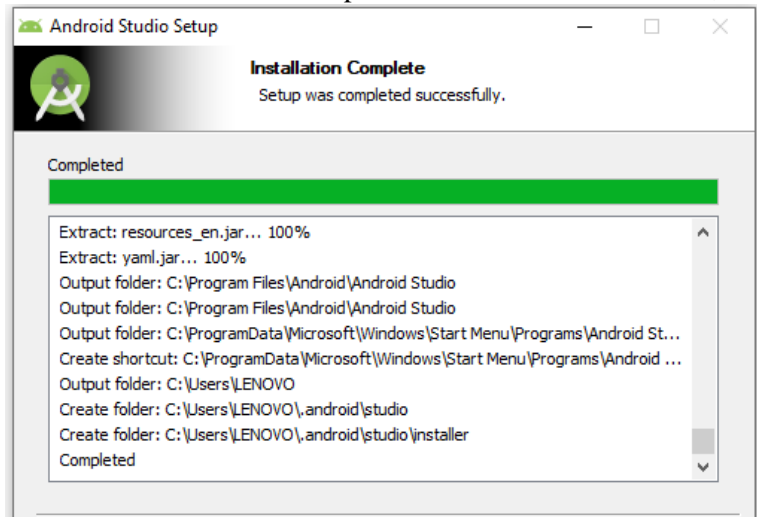
4. pilih lokasi untuk install Android Studio pada komputer



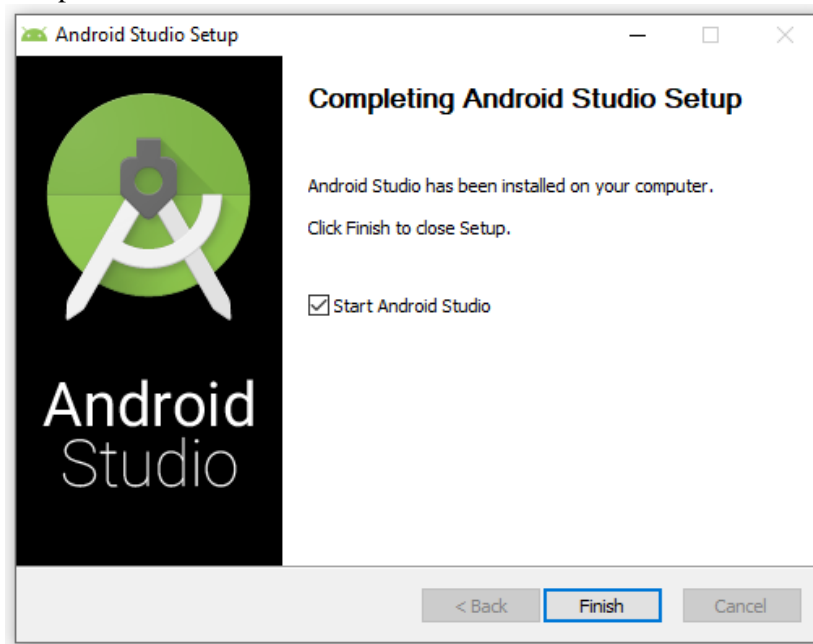
5. Ganti nama Android Studio saja. Klik Install untuk melanjutkan proses.



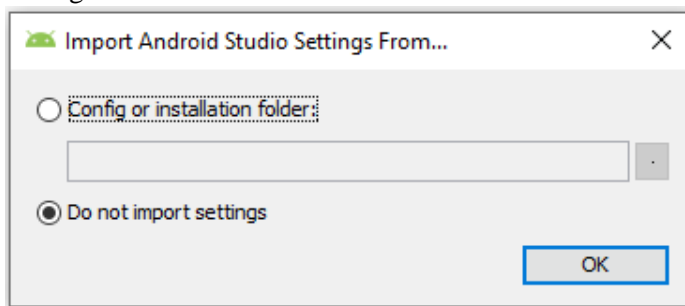
6. Gambar di bawah ini adalah proses instalasi Android Studio, tunggu hingga proses selesai.



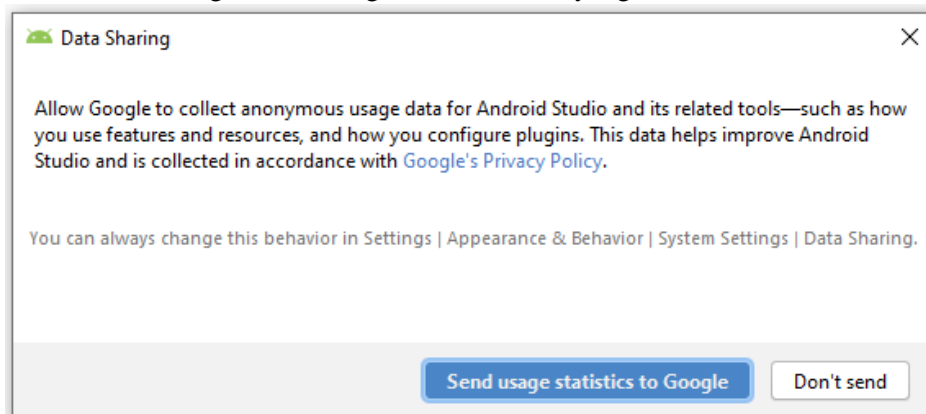
7. Setup Android Studio berhasil diinstall



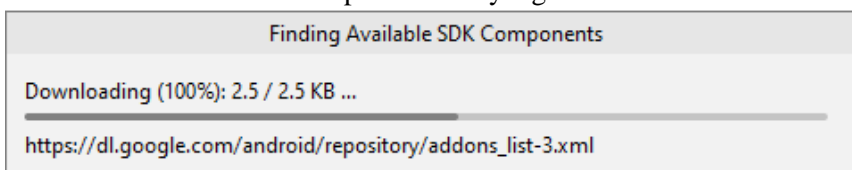
8. Buka Android Studio. karena baru pertama menginstall, maka klik pilihan kedua yaitu Do not import settings



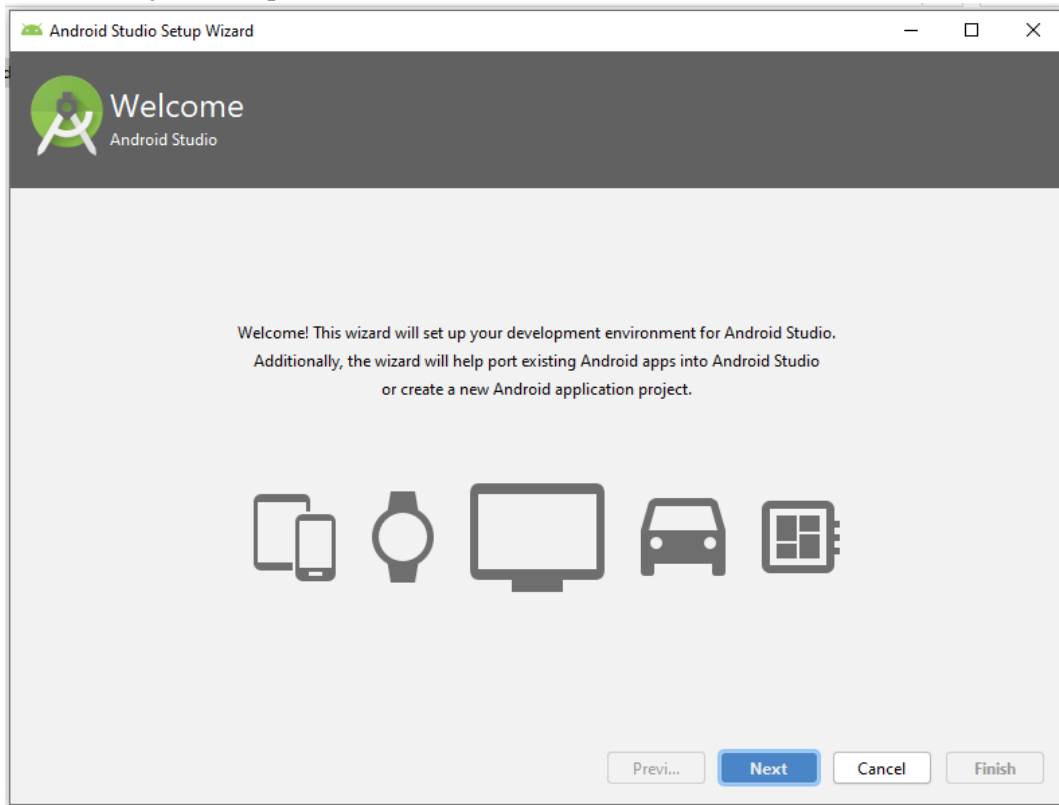
9. Setelah itu tentang Data sharing, bebas memilih yang mana



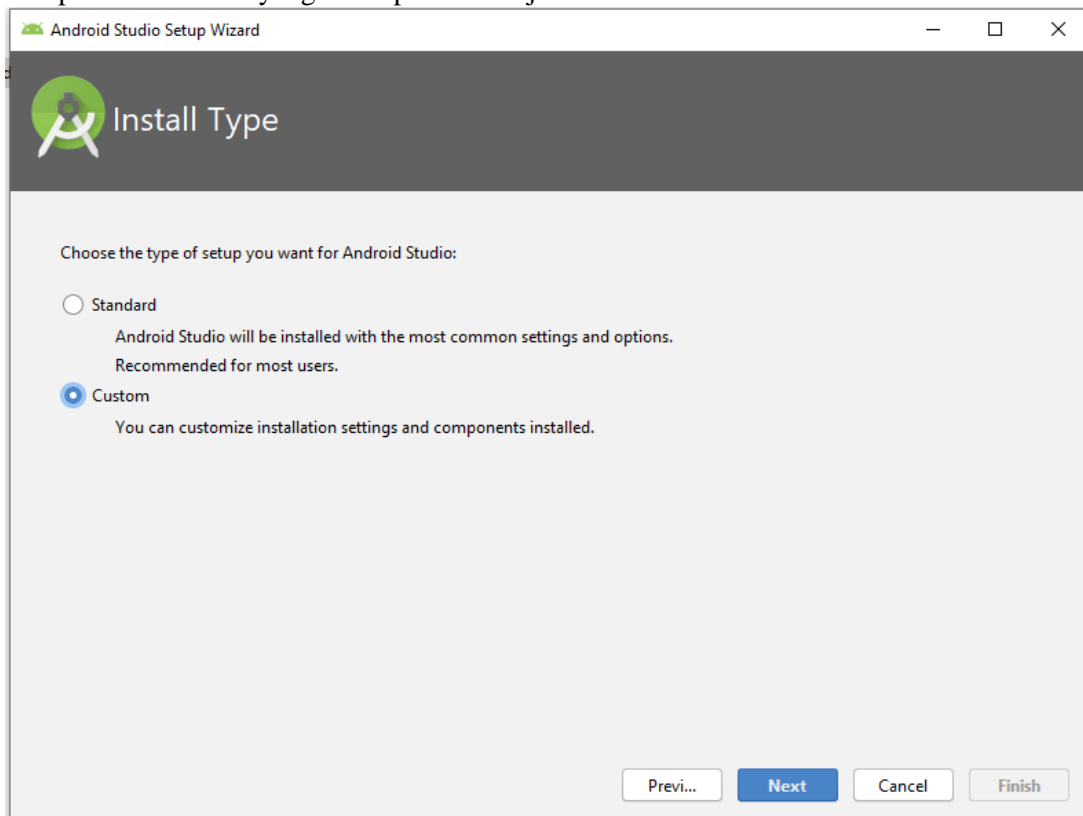
10. Android studio mencari komponen SDK yang tersedia



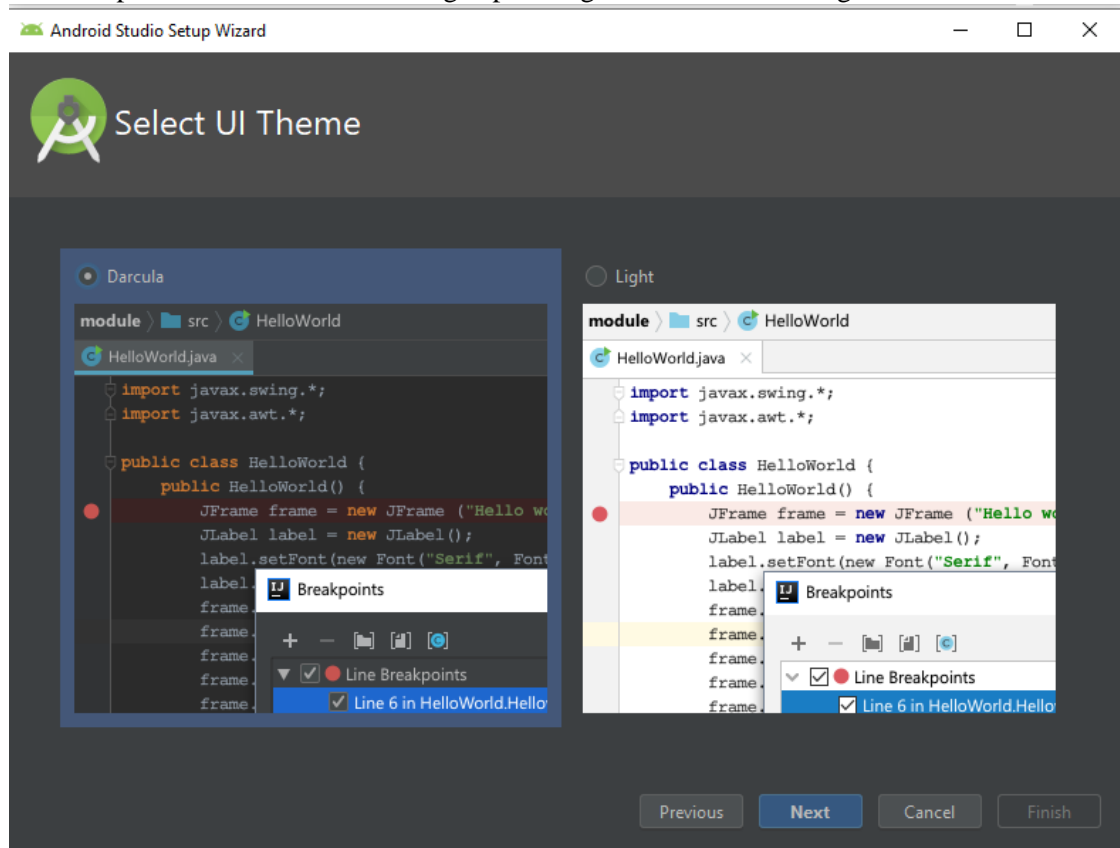
11. Lanjut ke Setup Wizard. Pertama akan muncul halaman seperti pada gambar di bawah ini. Klik Next untuk melanjutkan ke proses instalasi.



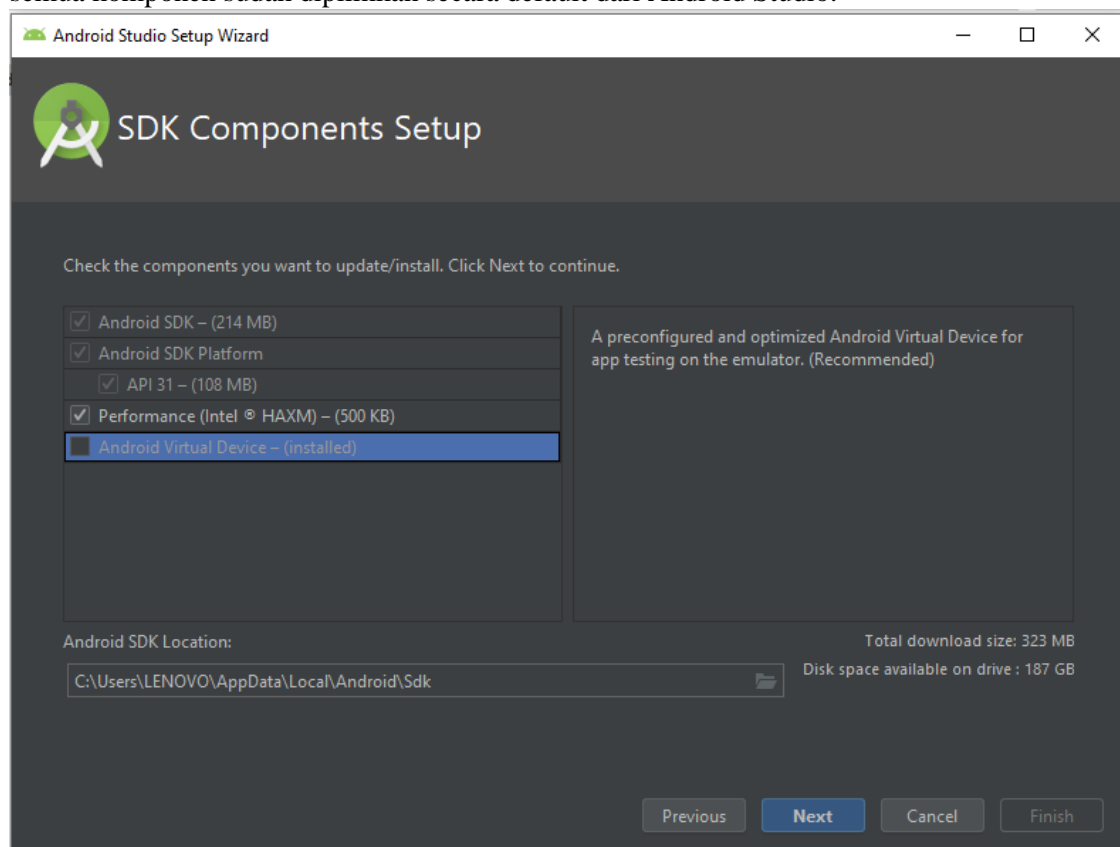
12. Pilih tipe instalasi. Ada dua pilihan, yaitu:
Pilihan pertama adalah **Standard**, dimana akan mendapatkan default pengaturan dan instalasi tambahan dari Android Studio. Pilihan kedua adalah **Custom** yaitu kita bisa memilih pengaturan dan komponen tambahan yang Anda perlukan saja.



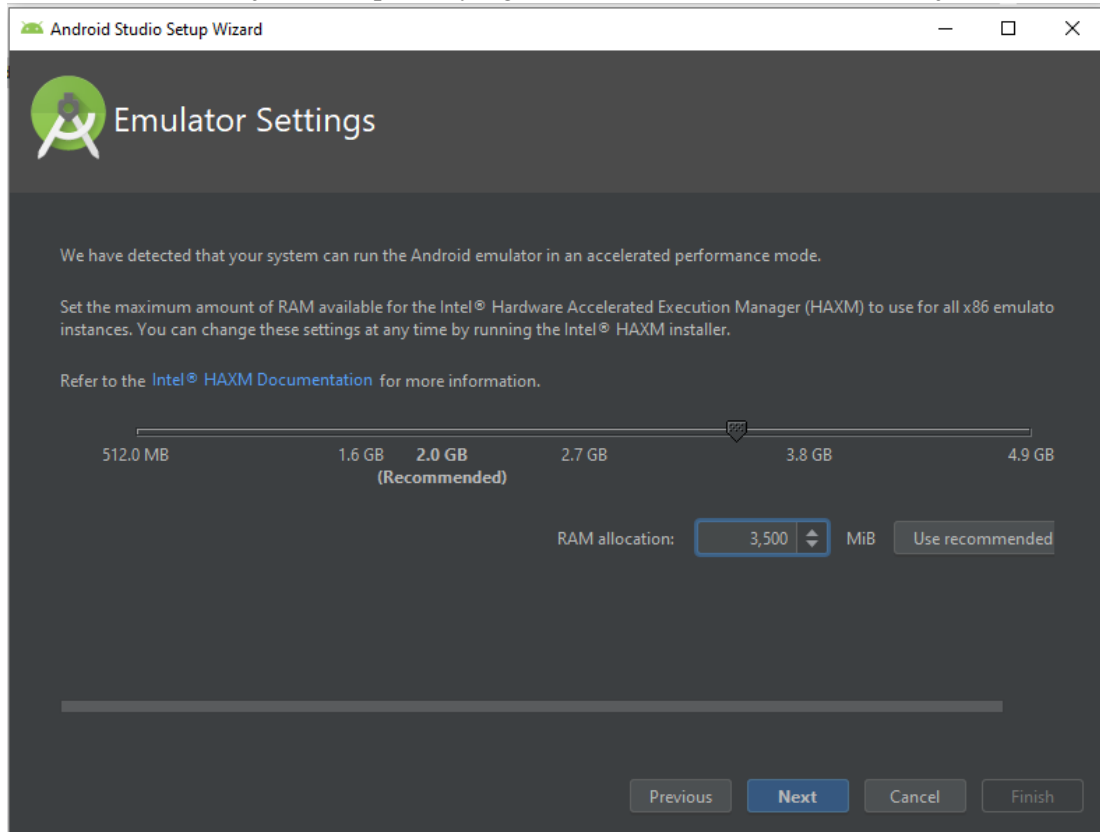
13. Pilih tampilan. Dracula untuk warna gelap dan light untuk warna terang



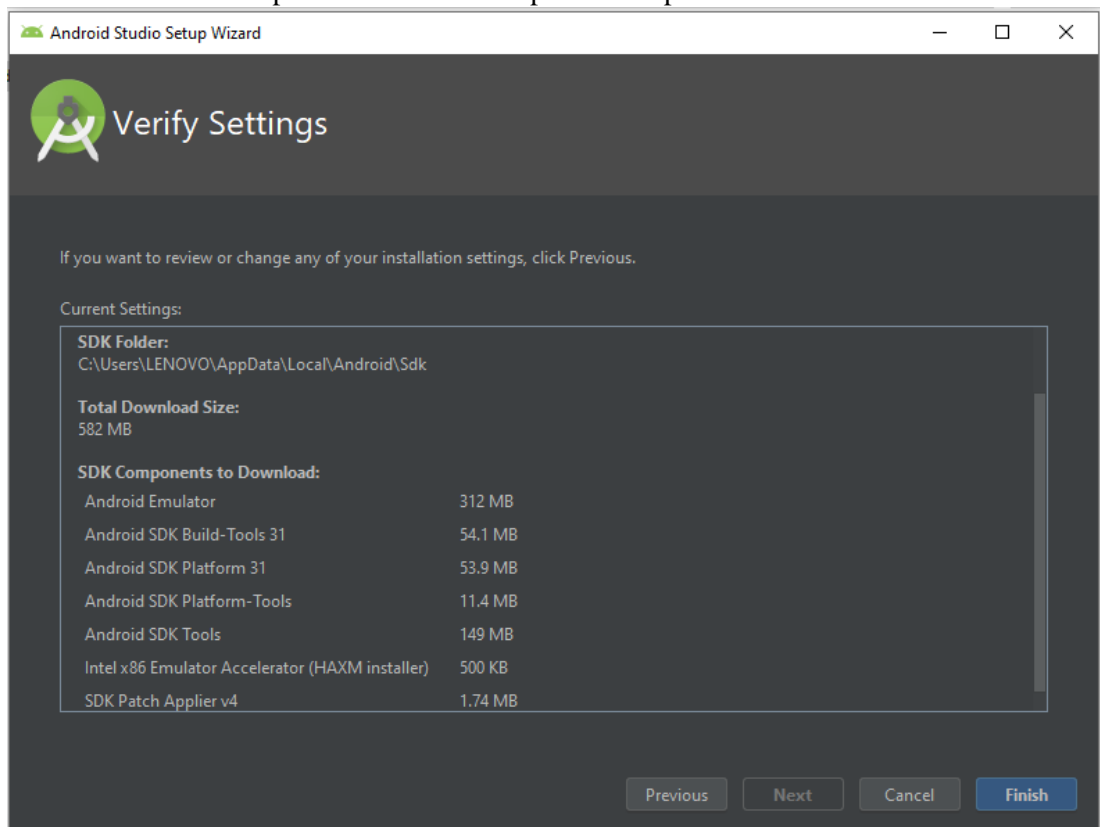
14. Langkah pemilihan komponen SDK ini hanya akan muncul kalau kita memilih tipe instalasi Custom. Jika memilih tipe instalasi Standard, kita tidak akan mendapatkan pilihan komponen SDK karena semua komponen sudah dipilihkan secara default dari Android Studio.



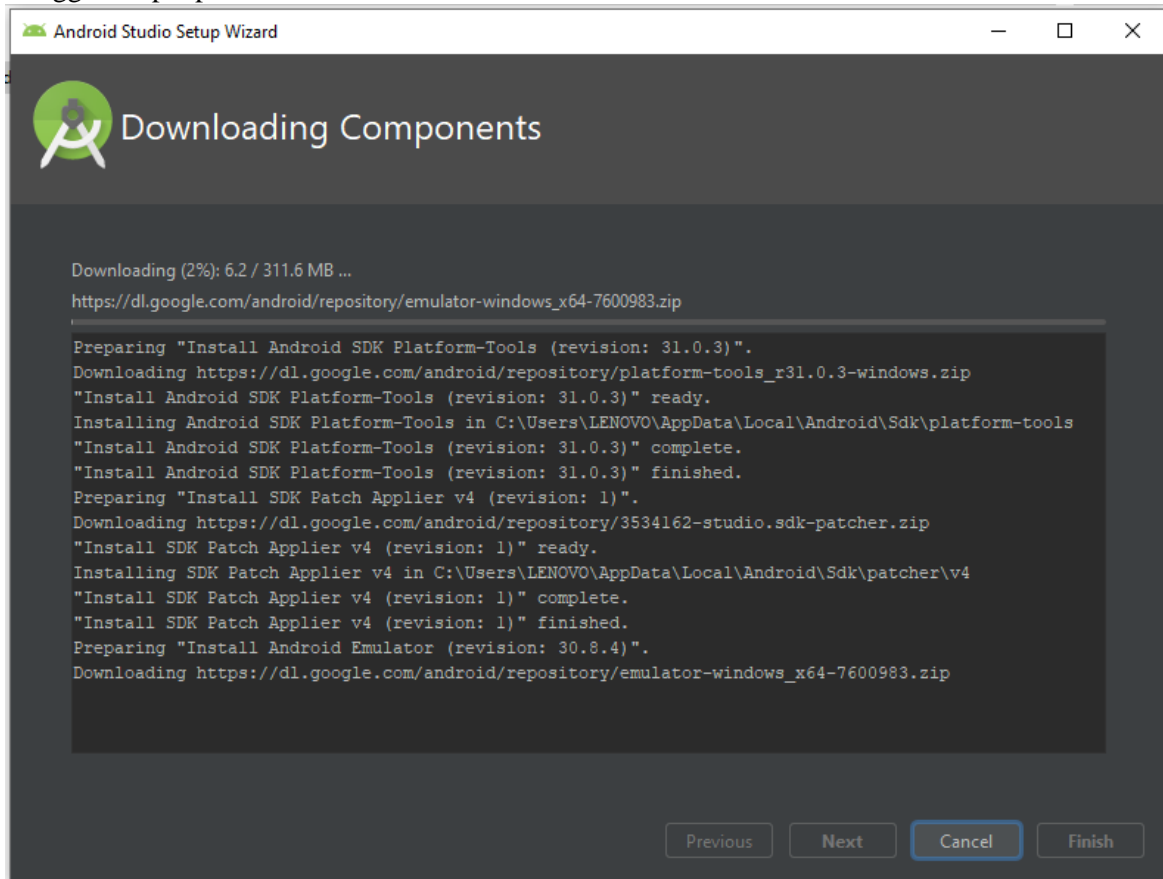
15. Selanjutnya, tentukan RAM. Gunakan RAM minimal 3GB agar tidak memperlambat proses running Emulator ketika menjalankan aplikasi yang telah dbuat. Klik Next untuk melanjutkan instalasi.



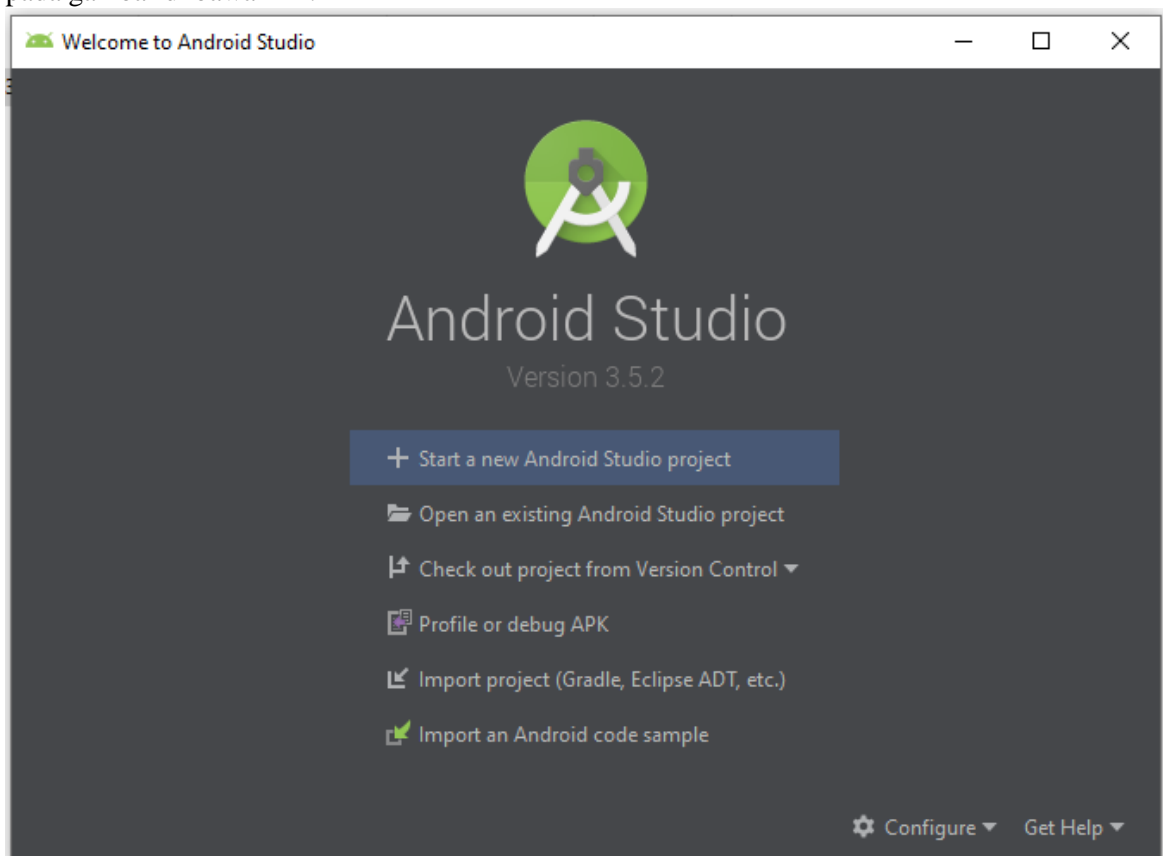
16. Kemudian, kita diberikan informasi mengenai komponen tambahan SDK beserta ukuran filenya. Klik Finish untuk memulai proses download komponen-komponen tersebut.



17. Tunggu sampai proses download SDK selesai

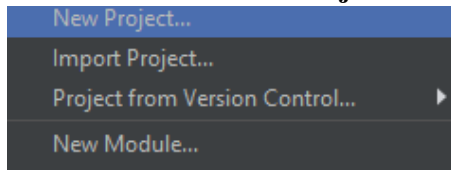


18. Setelah proses download dan install selesai, akan muncul tampilan launcher Android Studio seperti pada gambar di bawah ini.

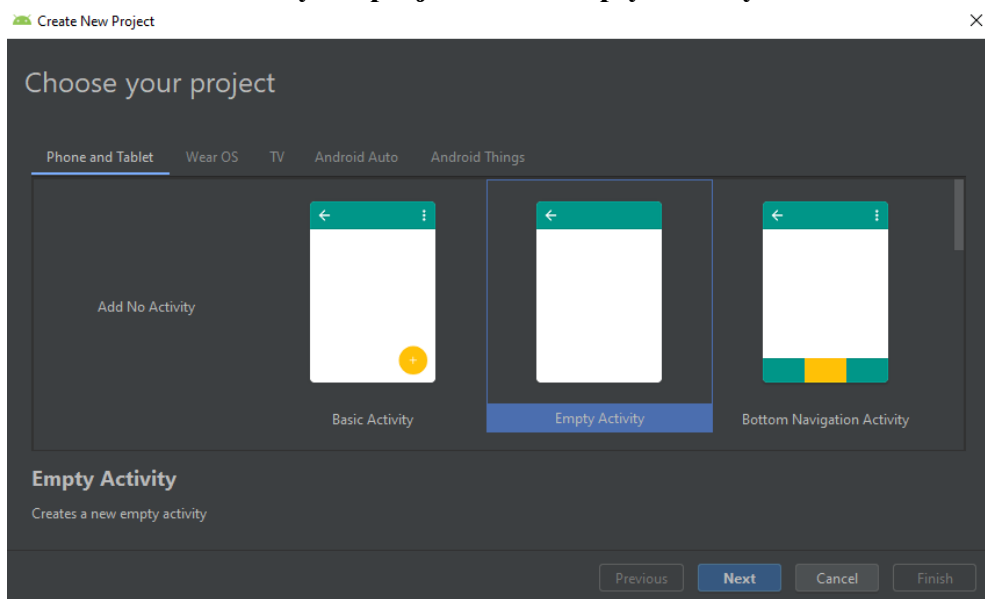


Tugas 2. Buatlah proyek baru dengan Android Studio, jalankan aplikasi anda pada perangkat virtual maupun perangkat fisik.

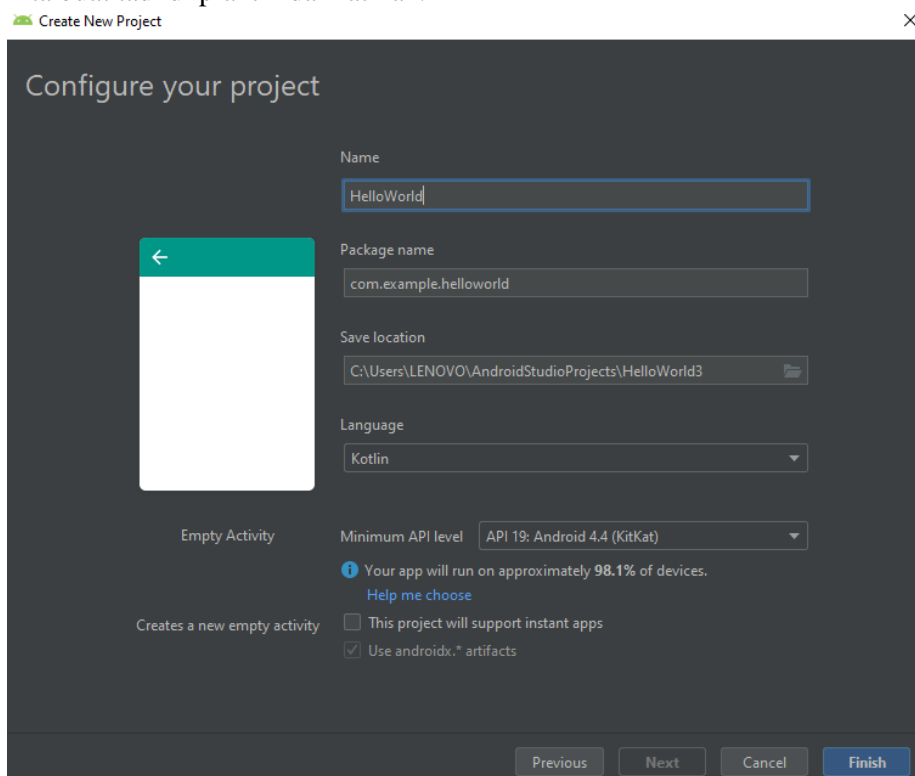
1. Klik **File > New > New Project** untuk membuat project baru



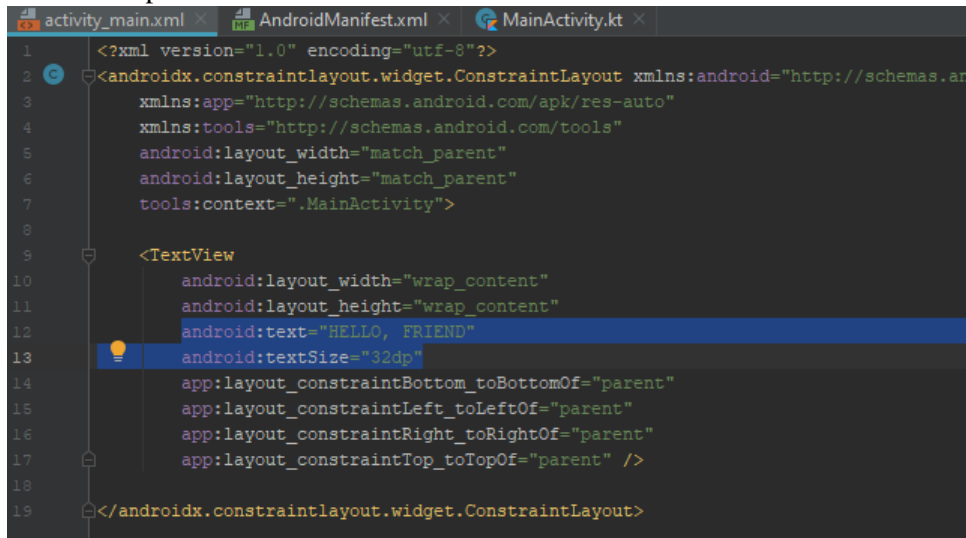
2. muncul menu **Choose your project**. Pilih **Empty Activity**. Lalu klik **Next**



3. konfigurasi project yang akan dibuat. Beri nama “HelloWorld” lagi. Hal tersebut tidak masalah karena otomatis project akan ditempatkan di folder yang berbeda dari dua project HelloWorld yang sudah kita buat tadi di praktik dan latihan.

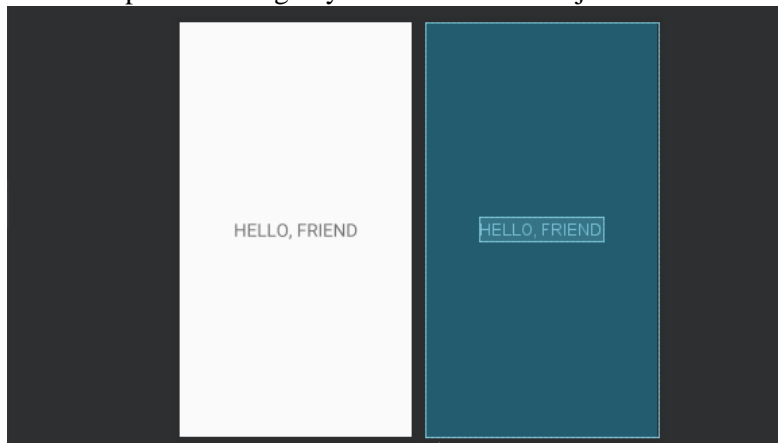


4. Tunggu proses gradle sampai selesai. Project baru sudah jadi.
5. Langsung ke **activity_main.xml**. kita menulis pesan “HELLO, FRIEND” dan mengatur besar font sebesar 32dp

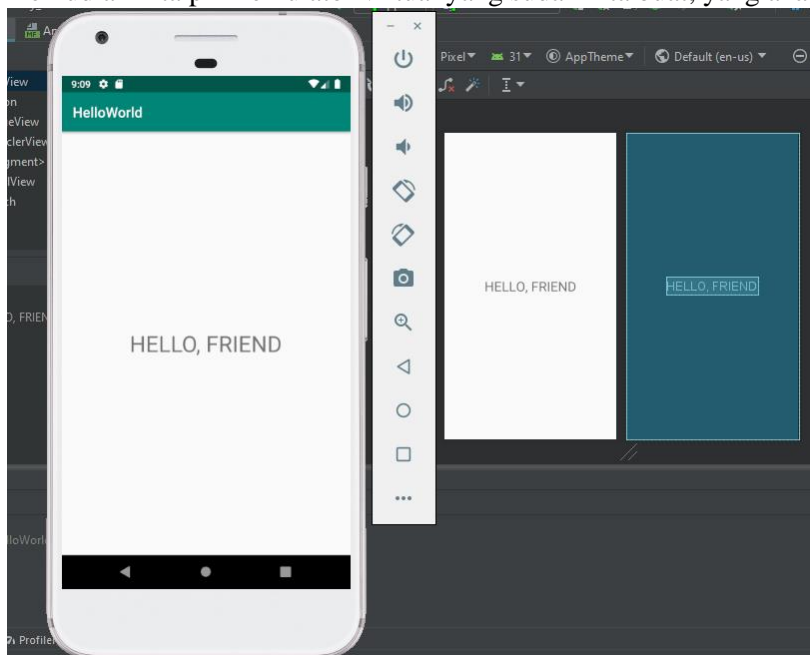


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".MainActivity">
8
9   <TextView
10     android:layout_width="wrap_content"
11     android:layout_height="wrap_content"
12     android:text="HELLO, FRIEND"
13     android:textSize="32dp"
14     app:layout_constraintBottom_toBottomOf="parent"
15     app:layout_constraintLeft_toLeftOf="parent"
16     app:layout_constraintRight_toRightOf="parent"
17     app:layout_constraintTop_toTopOf="parent" />
18
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

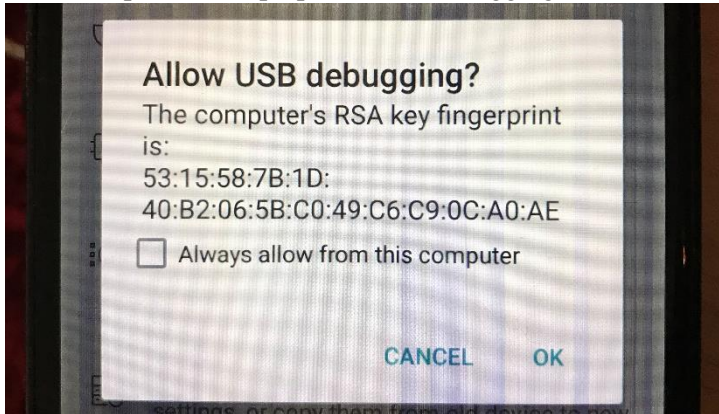
6. Kita lihat preview design nya dulu sebelum kita jalankan di emulator



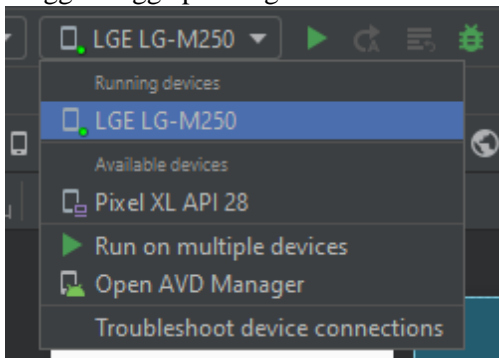
7. Kemudian kita pilih emulator virtual yang sudah kita buat, yang akan menghasilkan:



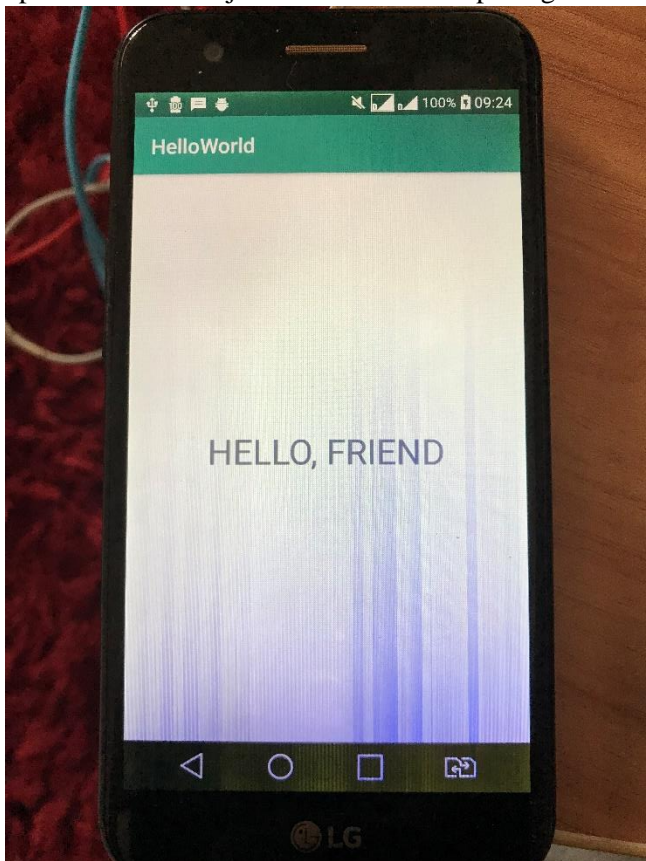
8. Selanjutnya kita coba aplikasi tersebut pada perangkat fisik. Dimulai dengan mencolokkan kabel USB dari smartphone ke laptop dan USB debugging



9. ke toolbar lalu pilih LGE LG-M250. Kemudian klik tombol run di sebelah kanan nya. Kemudian tunggu hingga proses gradle selesai.



10. aplikasi berhasil dijalankan di emulator perangkat smartphone



KESIMPULAN

Di pertemuan kali ini, saya mendapatkan banyak sekali ilmu tentang pengenalan aplikasi android studio. Saya mampu memenuhi tujuan dibuatnya modul ini yaitu mampu memahami dan mengetahui tentang aplikasi Android, cara penginstallan Android Studio, debug dengan virtual device dan debug dengan physical device, dan mampu memahami tentang Pengembangan Aplikasi Android. Seluruh project pada praktik, latihan, dan tugas di pertemuan ke-3 ini memiliki inti dan pola yang sama yaitu: membuat project -> menjalankan aplikasi pada virtual device -> menjalankan aplikasi pada perangkat keras (smartphone). Dengan begitu, setiap mahasiswa mendapat pembelajaran dan pengalaman penting tentang development aplikasi android dasar di Android Studio.

Terima Kasih