

## **MODUL 3**

### **PENGENALAN APLIKASI ANDROID**



#### **CAPAIAN PEMBELAJARAN**

---

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



#### **KEBUTUHAN ALAT/BAHAN/SOFTWARE**

---

1. Android Studio 3.5
2. Handphone Android versi 7.0 (Nougat)
3. Kabel data USB.
4. Driver ADB.



#### **DASAR TEORI**

---

#### **Instalasi Android Studio**

Android Studio menyediakan IDE lengkap, termasuk editor kode canggih dan template aplikasi. Ini juga berisi alat untuk pengembangan, debugging, pengujian, dan kinerja yang membuatnya lebih cepat dan lebih mudah untuk mengembangkan aplikasi. Anda dapat menggunakan Android Studio untuk menguji aplikasi Anda dengan sejumlah besar emulator yang telah dikonfigurasi sebelumnya, atau pada perangkat seluler Anda sendiri. Anda juga dapat membuat aplikasi produksi dan menerbitkan aplikasi di Google Play Store.

Android Studio tersedia untuk komputer yang menjalankan Windows atau Linux, dan untuk Mac yang menjalankan macOS. OpenJDK (Java Development Kit) terbaru dibundel dengan Android Studio.

Instalasi serupa untuk semua platform.

1. Buka halaman unduh Android Studio dan ikuti instruksi untuk mengunduh dan menginstal Android Studio.
2. Terima konfigurasi default untuk semua langkah, dan pastikan semua komponen dipilih untuk instalasi.
3. Setelah instalasi selesai, panduan setup akan mengunduh dan menginstal komponen tambahan, termasuk Android SDK. Bersabarlah, karena proses ini mungkin memakan waktu, tergantung pada kecepatan internet Anda.
4. Ketika instalasi selesai, Android Studio dimulai, dan Anda siap untuk membuat proyek pertama Anda.

## **Membuat Aplikasi Android**

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.

#### **Apa yang akan Anda pelajari**

- Cara membangun aplikasi Android dasar di Android Studio.
- Cara membuat proyek Android dari template.
- Cara menemukan komponen utama dari proyek Android.
- Cara menjalankan aplikasi Android pada emulator atau perangkat fisik.

#### **Apa yang akan Anda lakukan**

- Buat proyek Android baru dan aplikasi default bernama HelloWorld.
- Buat emulator (perangkat virtual) sehingga Anda dapat menjalankan aplikasi di komputer.
- Jalankan aplikasi HelloWorld di perangkat virtual dan fisik.
- Jelajahi tata letak proyek.
- Jelajahi file AndroidManifest.xml.

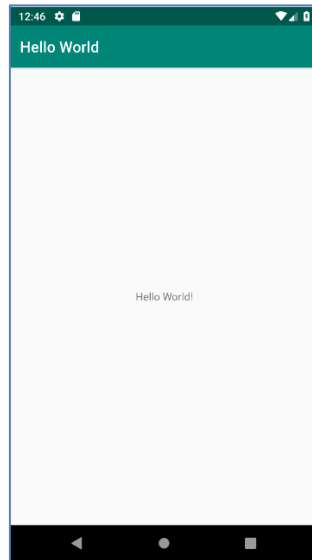


## **PRAKTIK**

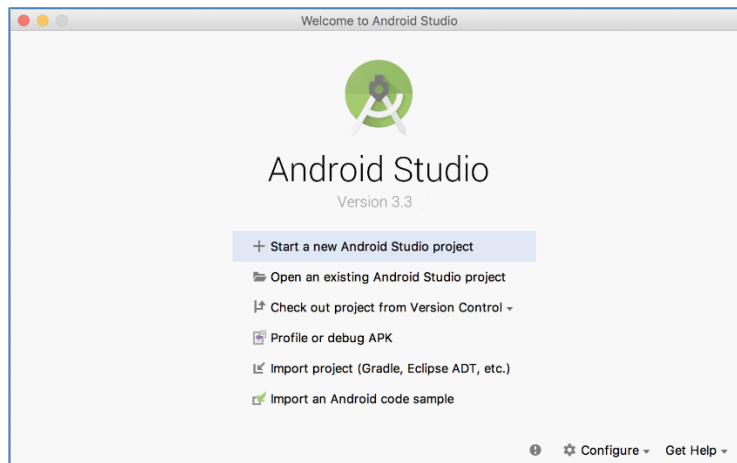
---

### **Praktik 1. Menciptakan Proyek HelloWorld**

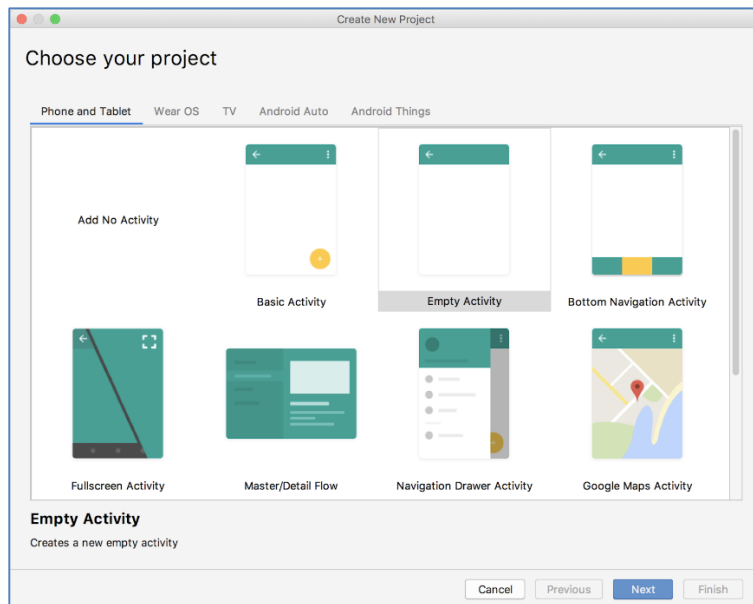
Anda akan membuat aplikasi **HelloWorld** yang menampilkan string "Hello World" pada layar perangkat virtual Android atau perangkat fisik. Begini tampilannya:



1. Buat proyek HelloWorld
  - a. Buka Android Studio.
  - b. Dalam dialog **Welcome to Android Studio**, klik **Start a new Android Studio project**.

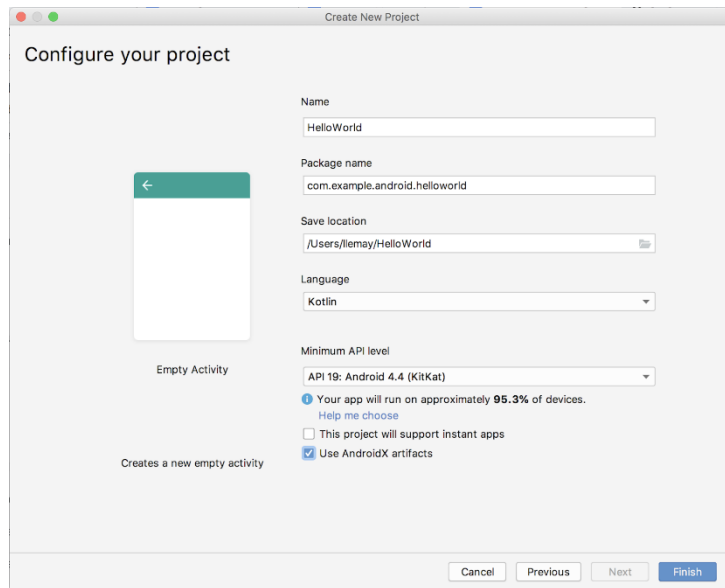


- c. Dialog **Choose your project** muncul. Pilih **Empty Activity** seperti ditunjukkan di bawah, dan klik **Next**.



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**.

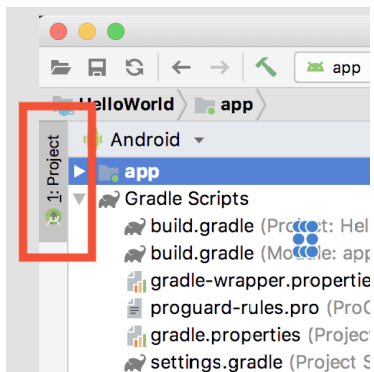


- e. Gunakan **android.example.com** default untuk Package name, atau buat domain unik lainnya. Nilai ini plus nama aplikasi adalah nama paket untuk aplikasi Anda. Jika Anda tidak berencana untuk menerbitkan aplikasi Anda, terima defaultnya. Anda dapat mengubah nama paket aplikasi Anda nanti, tetapi 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.

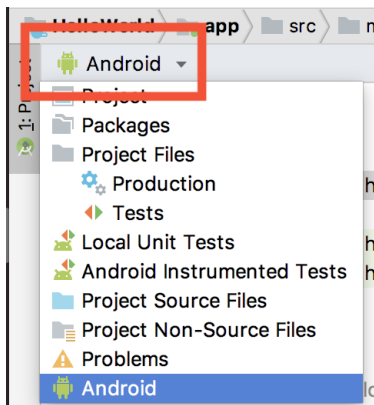
Android Studio sekarang membuat proyek Anda, yang bisa memakan waktu. Anda seharusnya tidak mendapatkan kesalahan. Jika Anda mendapat peringatan, abaikan saja.

## **Praktik 2. Explore Android Studio**

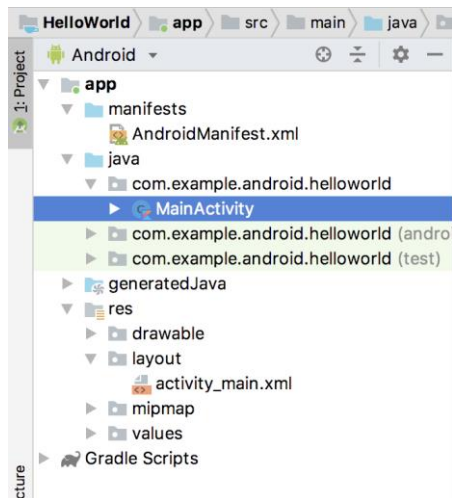
1. Pilih tab **Project**. Tab Project berada di kolom tab vertikal di sisi kiri jendela Android Studio



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.



3. Di panel **Proyek > Android**, perluas folder **app**. Di dalam folder **app** ada empat subfolder: **manifest**, **java**, **generateJava**, 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.

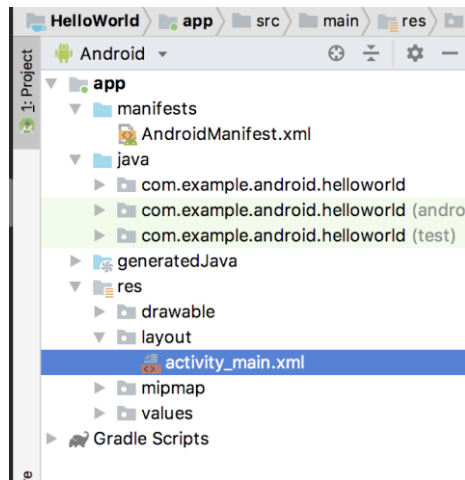
5. Perhatikan folder **generatedJava**. Folder ini berisi file yang dihasilkan Android Studio saat membangun aplikasi. Jangan edit apa pun di folder ini, karena perubahan Anda mungkin ditimpa ketika Anda membangun kembali aplikasi. Tetapi penting untuk mengetahui tentang folder ini ketika Anda perlu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

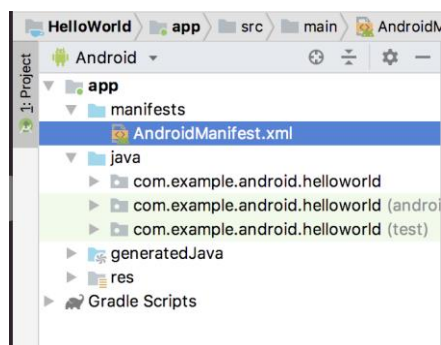
7. Di dalam folder **res**, perluas folder **layout** untuk melihat file `activity_main.xml`.





**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"/>

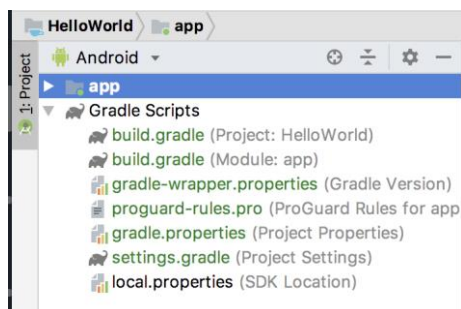
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>

```

10. Perhatikan elemen `<intent-filter>` di dalam `<activity>`. Elemen `<action>` dan `<category>` dalam filter maksud ini memberi tahu Android tempat memulai aplikasi ketika pengguna mengklik ikon peluncur.

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**

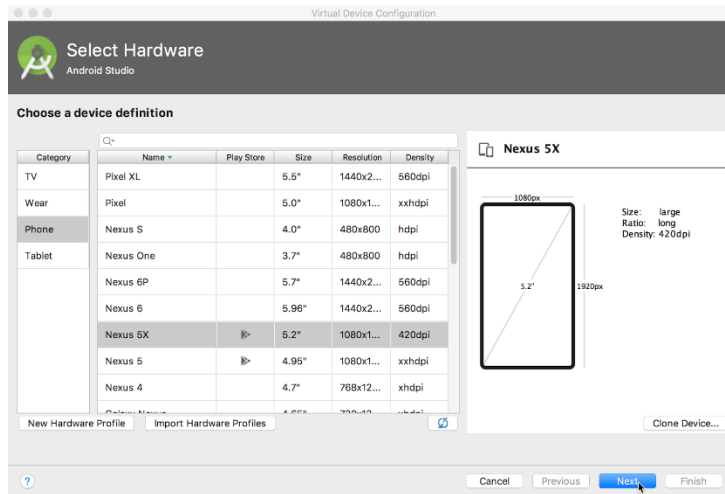


di toolbar. Dialog **Your Virtual Devices** muncul. Jika Anda sudah membuat perangkat virtual, dialog menunjukkannya (seperti yang ditunjukkan pada gambar di bawah), Jika tidak, Anda melihat daftar kosong.

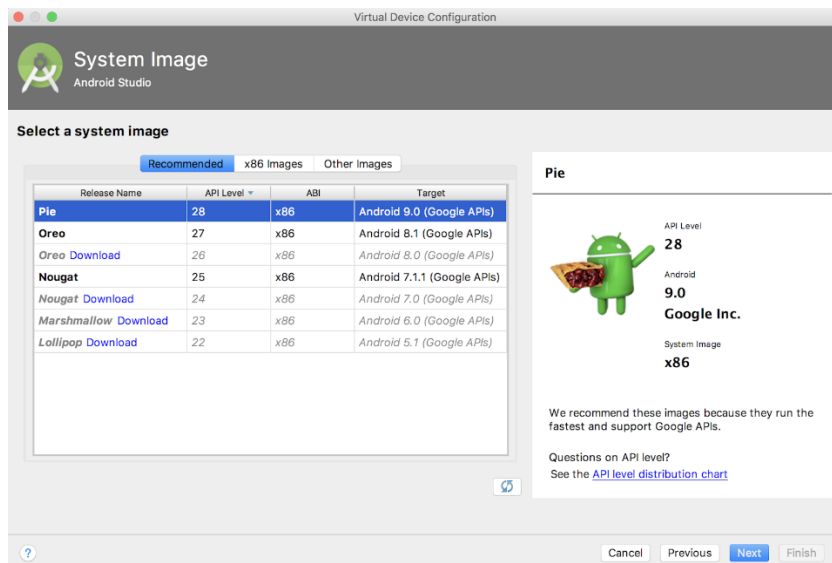
| Type | Name                  | Play Store | Resolution          | API | Target                  | CPU/ABI | Size on Disk | Actions |
|------|-----------------------|------------|---------------------|-----|-------------------------|---------|--------------|---------|
|      | Nexus 4 API 19        |            | 768 × 1280: xhdpi   | 19  | Android 4.4 (Google ... | x86     | 1 GB         |         |
|      | Nexus 5 API 23        |            | 1080 × 1920: xxhdpi | 23  | Android 6.0 (Google ... | x86...  | 2 GB         |         |
|      | Nexus 5 API 25 Nou... |            | 1080 × 1920: xxhdpi | 25  | Android 7.1.1 (Googl... | x86...  | 2 GB         |         |
|      | Nexus 5X O API 26     |            | 1080 × 1920: 420dpi | 26  | Android 8.0 (Google ... | x86     | 2 GB         |         |
|      | Nexus 7 2012 API 16   |            | 800 × 1280: tvdpi   | 16  | Android 4.1             | x86     | 4 GB         |         |
|      | Nexus 7 API 23        |            | 1200 × 1920: xhdpi  | 23  | Android 6.0 (Google ... | x86...  | 4 GB         |         |
|      | Nexus 9 API 23        |            | 2048 × 1536: xhdpi  | 23  | Android 6.0 (Google ... | x86...  | 2 GB         |         |
|      | Nexus One API 16      |            | 480 × 800: hdpi     | 16  | Android 4.1             | x86     | 4 GB         |         |

At the bottom left, there is a button: + Create Virtual Device...

2. Klik **+Create Virtual Device** pada bagian kiri bawah dialog. 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**).

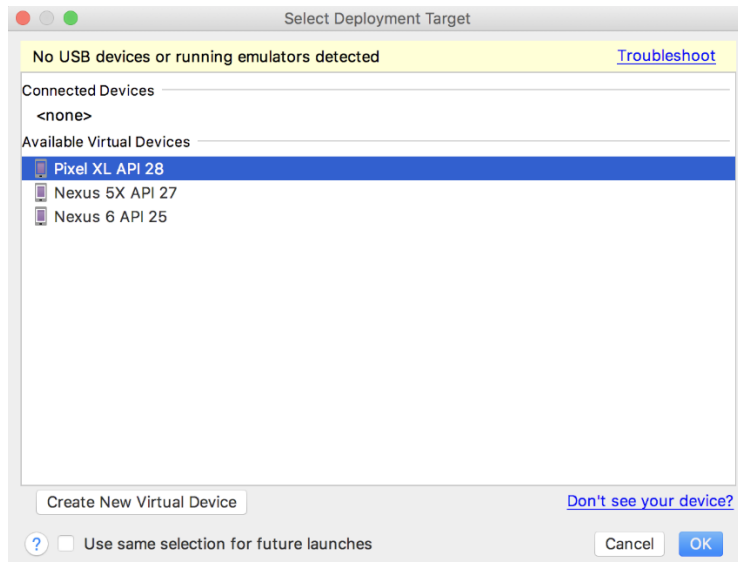


- Pilih perangkat seperti **Nexus 5x** atau **Pixel XL**, dan klik **Next**. Dialog **System Image** muncul.
- Klik tab **Recommended**, dan pilih versi sistem Android mana yang akan dijalankan pada perangkat virtual (seperti **Pie**).



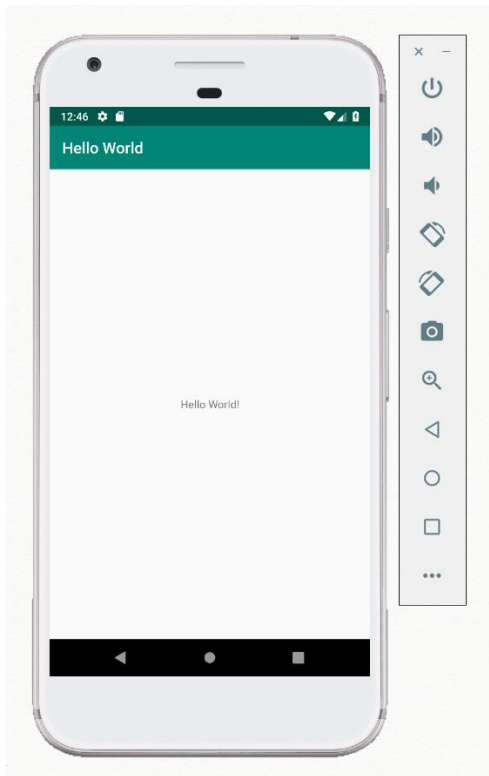
- Setelah Anda memilih system image, klik **Next**. Dialog **Android Virtual Device (AVD)** terbuka. Periksa konfigurasi Anda dan klik **Finish**.
- Di Android Studio, pilih **Run > Run app** atau klik ikon **Run** di toolbar. Dialog **Select Deployment Target** muncul dan memperingatkan Anda bahwa tidak ada perangkat yang tersedia. Anda melihat peringatan ini jika Anda tidak memiliki perangkat fisik yang terhubung ke komputer pengembangan Anda, atau jika Anda belum meluncurkan perangkat virtual.

7. Pada dialog **Select Deployment Target**, dibawah **Available Virtual Devices**, pilih virtual device yyang Anda ciptakan. Klik **OK**.



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

Anda seharusnya melihat aplikasi HelloWorld seperti yang ditunjukkan pada 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 Android 4.2 (Jellybean) dan lebih tinggi, pengaturan opsi Pengembang disembunyikan secara default. Untuk menampilkan opsi pengembang dan mengaktifkan debugging USB:

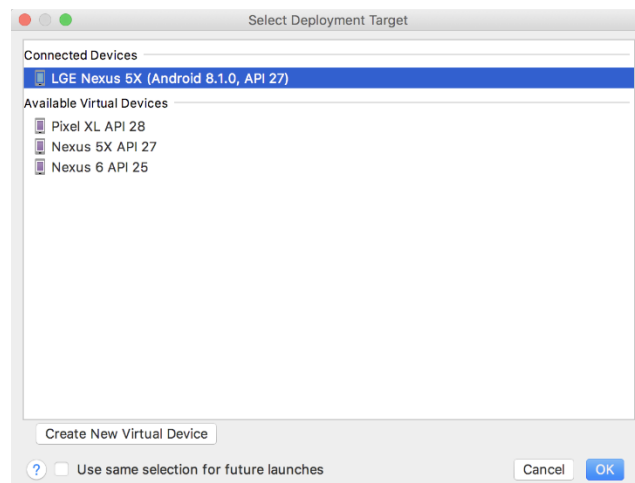
1. Di perangkat Anda, buka **Pengaturan**, cari **Tentang telepon**, ketuk **Tentang telepon**, dan ketuk **nomor bangun (Build number)** tujuh kali.
2. Kembali ke halaman sebelumnya (**Pengaturan / Sistem**). **Opsi pengembang** muncul dalam daftar. Ketuk **Opsi pengembang**.
3. Pilih **USB debugging**.

##### **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. Pilih opsi **Always allow** untuk mengingat komputer ini. Ketuk **OK**.
- c. Di komputer Anda, di toolbar Android Studio, klik tombol **Run** . Dialog **Select Deployment Target** terbuka dengan daftar emulator yang tersedia dan perangkat yang terhubung. Anda harus melihat perangkat fisik Anda bersama dengan emulator apa pun.



- d. Pilih perangkat Anda, dan klik **OK**. Android Studio menginstal aplikasi pada perangkat Anda dan menjalankannya.

### Penyelesaian masalah

Jika Android Studio tidak mengenali perangkat Anda, coba yang berikut ini:

1. Cabut kabel USB dan pasang kembali.
2. Mulai ulang Android Studio.

Jika komputer Anda masih tidak menemukan perangkat atau menyatakannya "tidak sah," ikuti langkah-langkah ini:

1. Lepaskan koneksi kabel USB.
2. Di perangkat, buka opsi Pengembang di aplikasi Pengaturan.
3. Ketuk Cabut otorisasi debugging USB.
4. Hubungkan kembali perangkat ke komputer Anda.
5. Saat diminta, berikan otorisasi.

Anda mungkin perlu menginstal driver USB yang sesuai untuk perangkat Anda.



## LATIHAN

---

1. Buat proyek baru di Android Studio.
2. Ubah salam "Hello World" menjadi "Selamat Ulang Tahun" dan nama seseorang yang baru saja berulang tahun.



## TUGAS

---

1. Install Android Studio pada perangkat komputer anda masing-masing di rumah.
2. Buatlah proyek baru dengan Android Studio, jalankan aplikasi anda pada perangkat virtual maupun perangkat fisik.





## REFERENSI

---

1. <https://codelabs.developers.google.com/android-kotlin-fundamentals/>
2. <https://codelabs.developers.google.com/codelabs/kotlin-android-training-install-studio/index.html?index=..%2F..android-kotlin-fundamentals#1>
3. <https://codelabs.developers.google.com/codelabs/kotlin-android-training-get-started/#0>
4. <https://developer.android.com/studio/run/device>