

# 1.1: Memasang Android Studio dan Menjalankan Hello World

## Daftar Isi:

- Yang harus sudah Anda KETAHUI
- Yang akan Anda BUTUHKAN
- Yang akan Anda PELAJARI
- Yang akan Anda LAKUKAN
- Ringkasan Aplikasi
- Tugas 1. Memasang Android Studio
- Tugas 2: Membuat aplikasi "Hello World"
- Tugas 3: Menjelajahi struktur proyek dan layout
- Tugas 4: Membuat perangkat virtual
- Tugas 5: Menjalankan aplikasi Anda di emulator
- Tugas 6. Menambahkan pernyataan log ke aplikasi Anda
- Tugas 7: Menjelajahi file AndroidManifest.xml
- Tugas 8. Menjelajahi file build.gradle
- Tugas 9. Menjalankan aplikasi Anda di perangkat
- Tantangan penyusunan kode
- Rangkuman
- Konsep terkait
- Ketahui selengkapnya

Selamat datang di latihan praktis. Anda akan belajar:

- Memasang Android Studio, lingkungan development Android.
- Mempelajari proses development Android.
- Membuat dan menjalankan aplikasi Android Hello World pertama di emulator dan perangkat fisik.
- Menambahkan pencatatan log ke aplikasi Anda untuk men-debug.

## Yang harus sudah Anda KETAHUI

Untuk praktik ini Anda harus bisa:

- Memahami proses development perangkat lunak umum untuk aplikasi berorientasi objek menggunakan IDE (Integrated Development Environment).
- Memiliki pengalaman pemrograman berorientasi objek paling sedikit 1-3 tahun, beberapa di antaranya fokus pada bahasa pemrograman Java. (Praktik ini tidak akan menjelaskan pemrograman berorientasi objek atau bahasa Java.)

## Yang akan Anda BUTUHKAN

Untuk praktik ini, Anda akan membutuhkan:

- Komputer Mac, Windows, atau Linux. Lihat di bagian bawah [laman unduhan Android Studio](#) untuk [persyaratan sistem](#) terbaru.
- Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.

## Yang akan Anda PELAJARI

Anda akan belajar:

- Memasang dan menggunakan Android IDE.
- Memahami proses development untuk membangun aplikasi Android.
- Membuat proyek Android dari template aplikasi dasar.

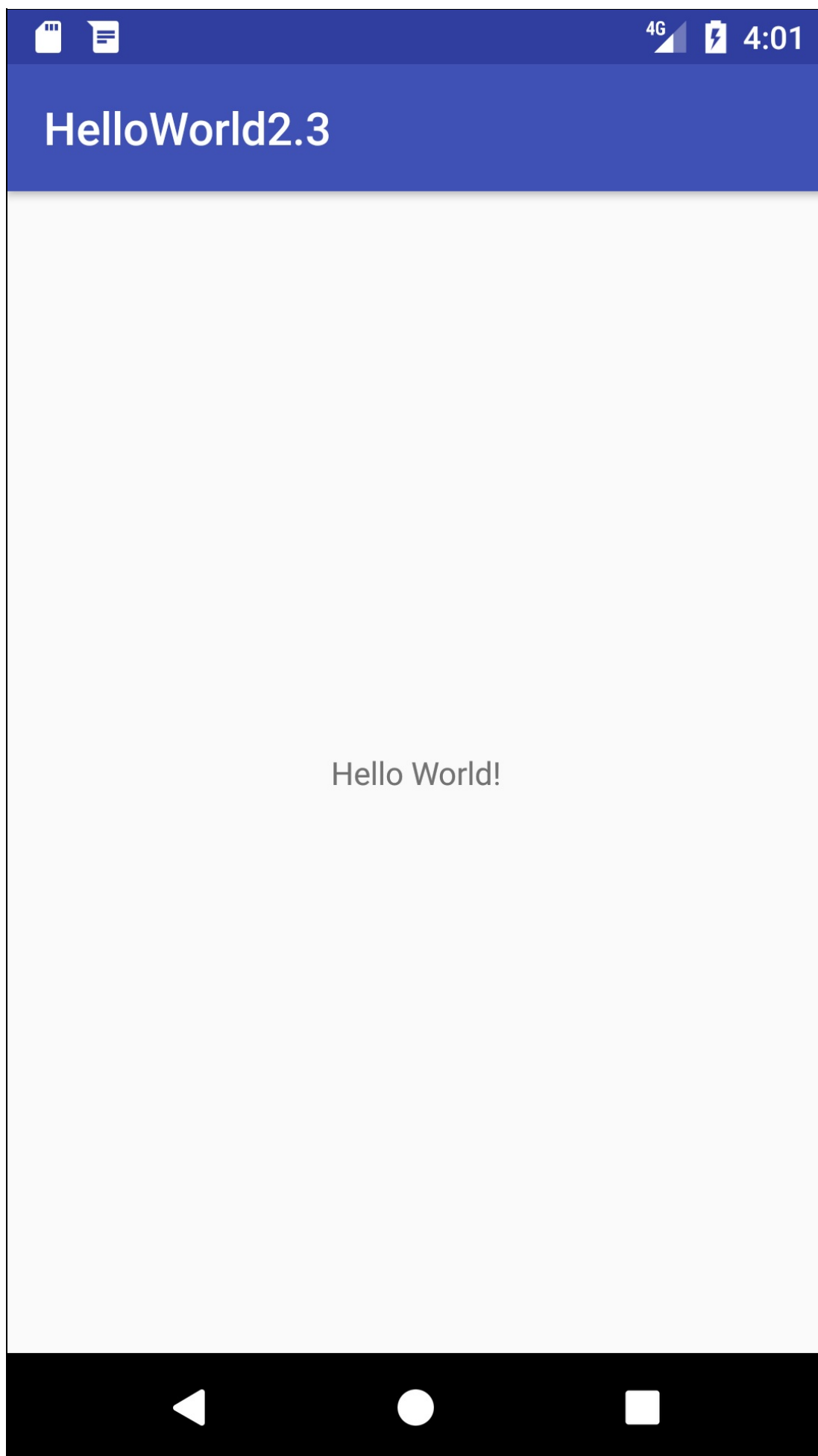
## Yang akan Anda LAKUKAN

- Memasang lingkungan development Android Studio.
- Membuat emulator (perangkat virtual) untuk menjalankan aplikasi Anda di komputer.
- Membuat dan menjalankan aplikasi Hello World di perangkat virtual dan fisik.
- Menjelajahi layout proyek.
- Membuat dan menampilkan pernyataan log dari aplikasi Anda.
- Menjelajahi file AndroidManifest.xml.

## Ringkasan Aplikasi

Setelah berhasil memasang Android Studio IDE, Anda akan membuat proyek Android baru untuk aplikasi 'Hello World' dari sebuah template. Aplikasi sederhana ini menampilkan string "Hello World" di layar perangkat Android virtual atau fisik.

Berikut tampilan aplikasi yang sudah selesai:



# Tugas 1. Memasang Android Studio

Android Studio adalah IDE Google untuk aplikasi Android. Android Studio menyediakan editor kode tingkat lanjut dan serangkaian template aplikasi. Selain itu, Android Studio juga memiliki alat untuk development, men-debug, menguji, dan meningkatkan kinerja yang membuat pengembangan aplikasi lebih cepat dan mudah. Anda bisa menguji aplikasi dengan berbagai macam emulator yang telah dikonfigurasi sebelumnya atau di perangkat seluler Anda sendiri, dan membangun APK produksi untuk publikasi.

**Catatan:** Android Studio masih terus dikembangkan. Untuk informasi terbaru tentang persyaratan sistem dan petunjuk pemasangan, lihat dokumentasinya di [developer.android.com](https://developer.android.com).

Untuk menyiapkan dan mengoperasikan Android Studio:

- Anda mungkin perlu memasang Java Development Kit - Java 7 atau versi yang lebih baik.
- Memasang Android Studio

Android Studio dapat digunakan di komputer Windows, Mac, dan Linux. Pemasangannya sama untuk semua platform. Perbedaanannya akan dijelaskan di bawah ini.

## 1.1. Memasang Java Development Kit

1. Buka jendela terminal di komputer.
2. Ketikkan `java -version`

Output termasuk baris:

```
Java(TM) SE Runtime Environment (build1.X.0_05-b13)
```

**X** adalah nomor versi yang perlu dilihat.

- Jika nomornya 7 atau lebih besar, Anda bisa lanjut memasang Android Studio.
- Jika versi Java SE di bawah 7 atau belum dipasang, Anda perlu memasang versi terbaru paket development Java SE sebelum memasang Android Studio.

Untuk mengunduh Java Standard Edition (JSE) Development Kit (JDK):

1. Buka [laman unduhan Oracle Java SE](#).
2. Klik ikon Java SE Downloads untuk membuka [laman Java SE Development Kit 8 Downloads](#).
3. Di kotak paket Java SE Development terbaru, Anda perlu menerima Perjanjian Lisensi agar dapat melanjutkan. Lalu unduh versi yang sesuai untuk komputer yang Anda gunakan untuk mengembangkan.

**Penting:** Jangan masuk ke demo dan contoh (menunya terlihat sangat mirip, jadi pastikan Anda membaca judul di atas).

4. Memasang paket development. Setelah pemasangan JDK selesai, yang seharusnya hanya memakan waktu beberapa menit, Anda bisa mengonfirmasi bahwa pemasangan sudah benar dengan memeriksa versi Java dari baris perintah.
5. Buka jendela terminal dan ketikkan `java -version` lagi untuk memverifikasi bahwa pemasangan telah berhasil.
6. Setel variabel lingkungan `JAVA_HOME` ke direktori pemasangan JDK.

**Windows:**

1. Setel `JAVA_HOME` ke lokasi pemasangan.
2. Start > Control Panel > System > Advanced System Settings > Environment Variables System Variables > New
  - Nama variabel: `JAVA_HOME`
  - Nilai variabel: `C:\Program Files\Java\jdk1.7.0_80` (atau apa pun versi pemasangan Anda)
3. Jika variabel sudah ada, perbarui dengan versi JDK ini.
4. Verifikasi variabel `JAVA_HOME` dari terminal cmd.exe: `echo %JAVA_HOME%`

Lihat juga: [https://docs.oracle.com/cd/E19182-01/820-7851/inst\\_cli\\_jdk\\_javahome\\_t/](https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/)

**Mac:**

1. Buka Terminal.
2. Konfirmasi bahwa Anda memiliki JDK dengan mengetik "which java".
3. Periksa bahwa Anda memiliki versi Java yang diperlukan dengan mengetik "java -version".
4. Setel JAVA\_HOME menggunakan perintah ini di Terminal: `eksport JAVA_HOME=$(which java)`
5. masukkan `echo $JAVA_HOME` untuk mengonfirmasi jalurnya.

**Linux:**

Lihat: [https://docs.oracle.com/cd/E19182-01/820-7851/inst\\_cli\\_jdk\\_javahome\\_t/](https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/)

**Penting:** Jangan pasang Android Studio sampai setelah Java JDK dipasang. Tanpa salinan Java yang berfungsi dengan baik, sisa proses tidak akan bekerja. Jika Anda tidak dapat mengunduh, telusuri pesan kesalahan dan cari solusinya secara online.

Pemecahan Masalah Dasar:

- Tidak ada ikon UI, Control Panel, atau Startup yang terkait dengan JDK.
- Verifikasi bahwa Anda telah memasang JDK dengan benar dengan membuka direktori tempat Anda memasangnya. Untuk mengidentifikasi tempat JDK, lihat variabel PATH dan/atau cari direktori "jdk" atau "java" atau "javac" yang dapat dijalankan di komputer Anda.

## 1.2. Memasang Android Studio

1. Buka [situs developer Android](#) dan ikuti petunjuk untuk mengunduh dan [memasang Android Studio](#).
  - Terima konfigurasi default untuk semua langkah.
  - Pastikan semua komponen dipilih untuk pemasangan.
2. Setelah selesai memasang, Wizard Persiapan akan mengunduh dan memasang beberapa komponen tambahan. Bersabarlah karena ini mungkin memakan waktu bergantung kecepatan internet Anda, dan beberapa langkah mungkin tampak berulang.
3. Setelah unduhan selesai, Android Studio akan dimulai, dan Anda siap membuat proyek pertama.

**Pemecahan masalah:** Jika menemui masalah dengan pemasangan, periksa dokumentasi terbaru, forum pemrograman, atau minta bantuan instruktur Anda.

## Tugas 2: Membuat aplikasi "Hello World"

Dalam tugas ini, Anda akan mengimplementasikan aplikasi "Hello World" untuk memverifikasi bahwa Android Studio sudah dipasang dengan benar dan mempelajari dasar-dasar pengembangan dengan Android Studio.

### 2.1 Membuat aplikasi "Hello World"

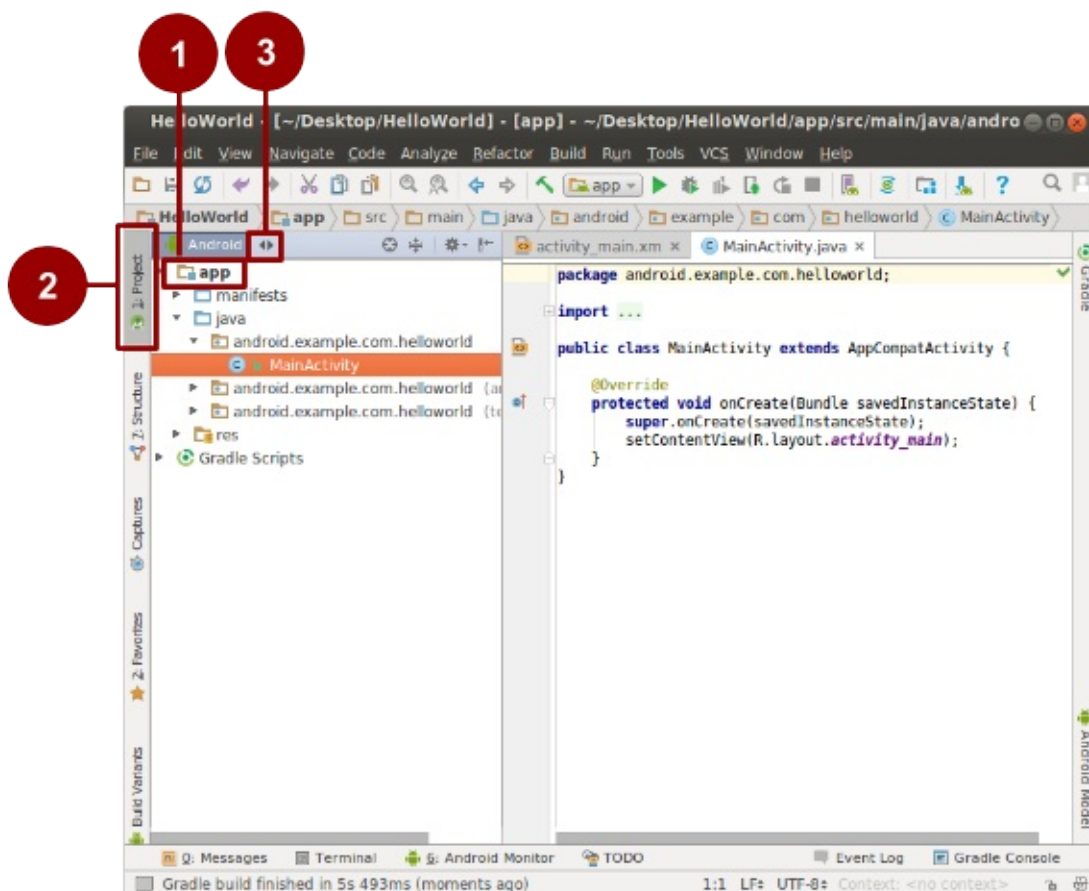
1. Buka Android Studio jika belum terbuka.
2. Di jendela utama **Welcome to Android Studio**, klik "Start a new Android Studio project".
3. Di jendela **New Project**, berikan aplikasi Anda **Application Name**, seperti "Hello World".
4. Verifikasi lokasi Proyek, atau pilih direktori yang berbeda untuk menyimpan proyek.
5. Pilih **Company Domain** yang unik.
  - Aplikasi yang dipublikasikan di Google Play Store harus memiliki nama paket yang unik. Karena domain unik, mengawali nama aplikasi dengan nama atau nama domain perusahaan Anda akan menghasilkan nama paket yang unik.
  - Jika tidak berencana mempublikasikan aplikasi, Anda bisa menerima domain contoh default. Perhatikan bahwa mengubah nama paket aplikasi di kemudian hari berarti melakukan pekerjaan ekstra.
6. Verifikasi bahwa **Project Location** default adalah tempat Anda menyimpan aplikasi Hello World dan proyek Android Studio lainnya, atau ubah lokasi ke direktori yang diinginkan. Klik Next.
7. Di layar **Target Android Devices**, "Phone and Tablet" harus dipilih. Dan Anda harus memastikan bahwa API 15: Android 4.0.3 IceCreamSandwich disetel sebagai **Minimum SDK**. (Perbaiki jika perlu.)

- Saat buku ini ditulis, memilih tingkat API ini menjadikan aplikasi "Hello World" kompatibel dengan 97% perangkat Android yang aktif di Google Play store.
  - Ini adalah setelan yang digunakan oleh contoh di buku ini.
8. Klik **Next**.
  9. Jika proyek Anda memerlukan komponen tambahan untuk SDK target terpilih, Android Studio akan memasangnya secara otomatis. Klik **Next**.
  10. Jendela **Customize the Activity**. Setiap aplikasi membutuhkan paling tidak satu aktivitas. Aktivitas mewakili satu layar dengan satu antarmuka pengguna dan Android Studio menyediakan template untuk membantu Anda memulai. Untuk proyek Hello World, pilih template termudah (saat ini ditulis, template proyek "Empty Activity" adalah template paling sederhana) yang tersedia.
  11. Adalah praktik yang biasa untuk menyebut aktivitas utama Anda MainActivity. Ini bukan persyaratan.
  12. Pastikan kotak **Generate Layout file** dicentang (jika terlihat).
  13. Pastikan kotak **Backwards Compatibility (App Compat)** dicentang.
  14. Biarkan **Layout Name** sebagai activity\_main. Layout biasanya diberi nama sesuai dengan aktivitasnya. Terima defaultnya dan klik **Finish**.

Setelah langkah-langkah ini, Android Studio:

- Membuat folder untuk Proyek Android Studio Anda.
- Bangun proyek Anda dengan [Gradle](#) (ini mungkin memerlukan beberapa waktu). Android Studio menggunakan Gradle sebagai sistem build-nya. Lihat laman developer [Konfigurasi build](#) untuk informasi selengkapnya.
- Membuka editor kode dengan proyek Anda.
- Menampilkan tip hari ini.
  - Android Studio menawarkan banyak pintasan keyboard, dan membaca tip adalah cara yang bagus untuk mempelajarinya dari waktu ke waktu.

Jendela Android Studio harus terlihat mirip dengan diagram berikut:



Anda bisa melihat hierarki file untuk aplikasi Anda dengan beberapa cara.

1. Klik di folder Hello World folder untuk meluaskan hierarki file (1),
2. Klik di **Project (2)**.
3. Klik di menu **Android(3)**.
4. Jelajahi opsi tampilan yang berbeda untuk proyek Anda.

**Catatan:** Buku ini menggunakan tampilan Android dari file proyek, kecuali ditentukan lain.

## Tugas 3: Jelajahi struktur proyek

Dalam praktik ini, Anda akan menjelajahi cara file proyek disusun di Android Studio.

Langkah-langkah ini mengasumsikan bahwa proyek Hello World Anda dimulai seperti yang ditampilkan pada diagram di atas.

### 3.1 Menjelajahi struktur proyek dan layout

Di tampilan Project > Android tugas sebelumnya, ada tiga folder level teratas di bawah folder **app** Anda: **manifests**, **java**, dan **res**.

1. Luaskan folder **manifests**.

Folder ini berisi **AndroidManifest.xml**. File ini menjelaskan semua komponen aplikasi Android Anda dan dibaca oleh sistem waktu proses Android saat program dijalankan.

2. Luaskan folder **java**. Semua file bahasa Java dikelola di folder ini. Folder **java** berisi tiga subfolder:
  - **com.example.hello.helloworld (atau nama domain yang telah Anda tetapkan)**: Semua file untuk paket ada di folder yang bernama sama dengan paket tersebut. Untuk aplikasi Hello World, ada satu paket dan paket berisi MainActivity.java (ekstensi file dapat dihilangkan di tampilan Proyek).
  - **com.example.hello.helloworld(androidTest)**: Folder ini untuk pengujian berinstrumen Anda, dan memulai dengan file pengujian kerangka.
  - **com.example.hello.helloworld(test)**: Folder ini untuk pengujian unit dan memulai dengan file pengujian unit kerangka yang secara otomatis dibuat.
3. Luaskan folder **res**. Folder ini berisi semua sumber daya untuk aplikasi Anda, termasuk gambar, file layout, string, ikon, dan penataan gaya. Folder ini berisi subfolder berikut:
  - **drawable**. Simpan semua gambar aplikasi Anda di folder ini.
  - **layout**. Setiap aktivitas memiliki paling tidak satu file layout yang menjelaskan UI di XML. Untuk Hello World, folder ini berisi activity\_main.xml.
  - **mipmap**. Simpan ikon peluncur Anda di folder ini. Ada subfolder untuk setiap kepadatan layar yang didukung. Android menggunakan kepadatan layar, yaitu, jumlah piksel per inci untuk menentukan resolusi gambar yang diperlukan. Android mengelompokkan kepadatan layar yang sebenarnya ke dalam kepadatan umum seperti medium (mdpi), tinggi (hdpi), atau ekstra-ekstra-ekstra-tinggi (xxxhdpi). Folder ic\_launcher.png berisi ikon peluncur default untuk semua kepadatan yang didukung oleh aplikasi Anda.
  - **values**. Sebagai ganti melakukan hardcode nilai-nilai seperti string, dimensi, dan warna di file XML dan Java, praktik terbaiknya adalah mendefinisikannya menurut file nilai. Ini membuatnya lebih mudah diubah dan konsisten di seluruh aplikasi Anda.
4. Luaskan subfolder **values** dalam folder res. Folder ini berisi subfolder berikut:
  - **colors.xml**. Tunjukkan warna default untuk tema pilihan Anda, dan Anda bisa menambahkan warna sendiri atau mengubahnya berdasarkan persyaratan aplikasi Anda.
  - **dimens.xml**. Simpan ukuran tampilan dan objek untuk resolusi yang berbeda.
  - **strings.xml**. Buat sumber daya untuk semua string Anda. Ini memudahkan penerjemahan string ke bahasa lain.
  - **styles.xml**. Semua gaya untuk aplikasi dan tema Anda ada di sini. Gaya membantu memberikan aplikasi Anda tampilan yang konsisten untuk semua elemen UI.

### 3.2 Sistem pembangunan Gradle

Android Studio menggunakan [Gradle](#) sebagai sistem pembangunan. Seiring dengan kemajuan Anda sepanjang praktik ini, Anda akan mempelajari lebih banyak tentang gradle dan apa yang dibutuhkan untuk membangun dan menjalankan aplikasi Anda.

1. Luaskan folder *\*Gradle Scripts*. Folder ini berisi semua file yang dibutuhkan oleh sistem pembangunan.
2. Cari file **build.gradle(Module:app)**. Saat Anda menambahkan dependensi khusus aplikasi, seperti menggunakan pustaka tambahan, dependensi masuk ke dalam file ini.

## Tugas 4: Membuat perangkat virtual (emulator)

Pada tugas ini, Anda akan menggunakan [Android Virtual Device \(AVD\) manager](#) untuk membuat perangkat virtual atau emulator yang menyimulasikan konfigurasi untuk jenis perangkat Android tertentu.


Menggunakan AVD Manager, Anda mendefinisikan karakteristik perangkat keras sebuah perangkat dan API level-nya, serta menyimpannya sebagai konfigurasi perangkat virtual.

Saat Anda memulai emulator Android, emulator membaca konfigurasi yang ditetapkan dan membuat perangkat emulasi yang bertindak persis seperti versi fisik perangkat tersebut, namun berada di komputer Anda.

**Mengapa:** Dengan perangkat virtual, Anda bisa menguji aplikasi di perangkat yang berbeda (tablet, ponsel) dengan API level berbeda untuk memastikan aplikasi terlihat bagus dan berfungsi untuk sebagian besar pengguna. Anda tidak perlu bergantung pada perangkat fisik yang tersedia untuk development aplikasi.

### 4.1 Membuat perangkat virtual

Untuk menjalankan emulator di komputer, Anda perlu membuat konfigurasi yang menjelaskan perangkat virtual.

1. Di Android Studio, pilih Tools > Android > AVD Manager atau klik ikon AVD Manager  di bilah alat.
2. Klik **+Create Virtual Device...** (Jika Anda telah membuat perangkat virtual sebelumnya, jendela akan menunjukkan semua perangkat yang ada dan tombolnya ada di bagian bawah.)

Layar Select Hardware muncul menunjukkan daftar perangkat keras yang telah dikonfigurasi sebelumnya. Untuk setiap perangkat, tabel menunjukkan ukuran tampilan (Size), resolusi layar dalam piksel (Resolution), dan kepadatan piksel (Density).

Untuk perangkat Nexus 5, kepadatan pikselnya adalah xxhdpi, yang berarti aplikasi Anda menggunakan ikon peluncur di folder xxhdpi dari folder mipmap. Aplikasi Anda juga akan menggunakan layout dan drawable dari folder yang didefinisikan untuk kepadatan itu juga.

3. Pilih perangkat keras Nexus 5 dan klik **Next**.
4. Pada layar **System Image**, dari tab **Recommended**, pilih versi sistem Android yang akan dijalankan pada perangkat virtual. Anda bisa memilih gambar sistem terbaru.

Ada lebih banyak versi yang tersedia dari yang ditunjukkan di tab **Recommended**. Lihat tab **x86 Images** dan **Other Images** untuk melihatnya.


5. Jika tautan **Download** terlihat di samping versi gambar sistem, berarti versi tersebut belum terpasang dan Anda perlu mengunduhnya. Jika perlu, klik tautan untuk mulai mengunduh dan klik **Finish** setelah selesai.
6. Pada layar **System Image**, pilih gambar sistem dan klik **Next**.
7. Verifikasi konfigurasi Anda dan klik **Finish**. (Jika jendela AVD Manager **Your Android Devices** tetap terbuka, Anda bisa melanjutkan dan menutupnya.)

## Tugas 5: Menjalankan aplikasi Anda di emulator

Pada tugas ini Anda akhirnya akan menjalankan aplikasi Hello World.

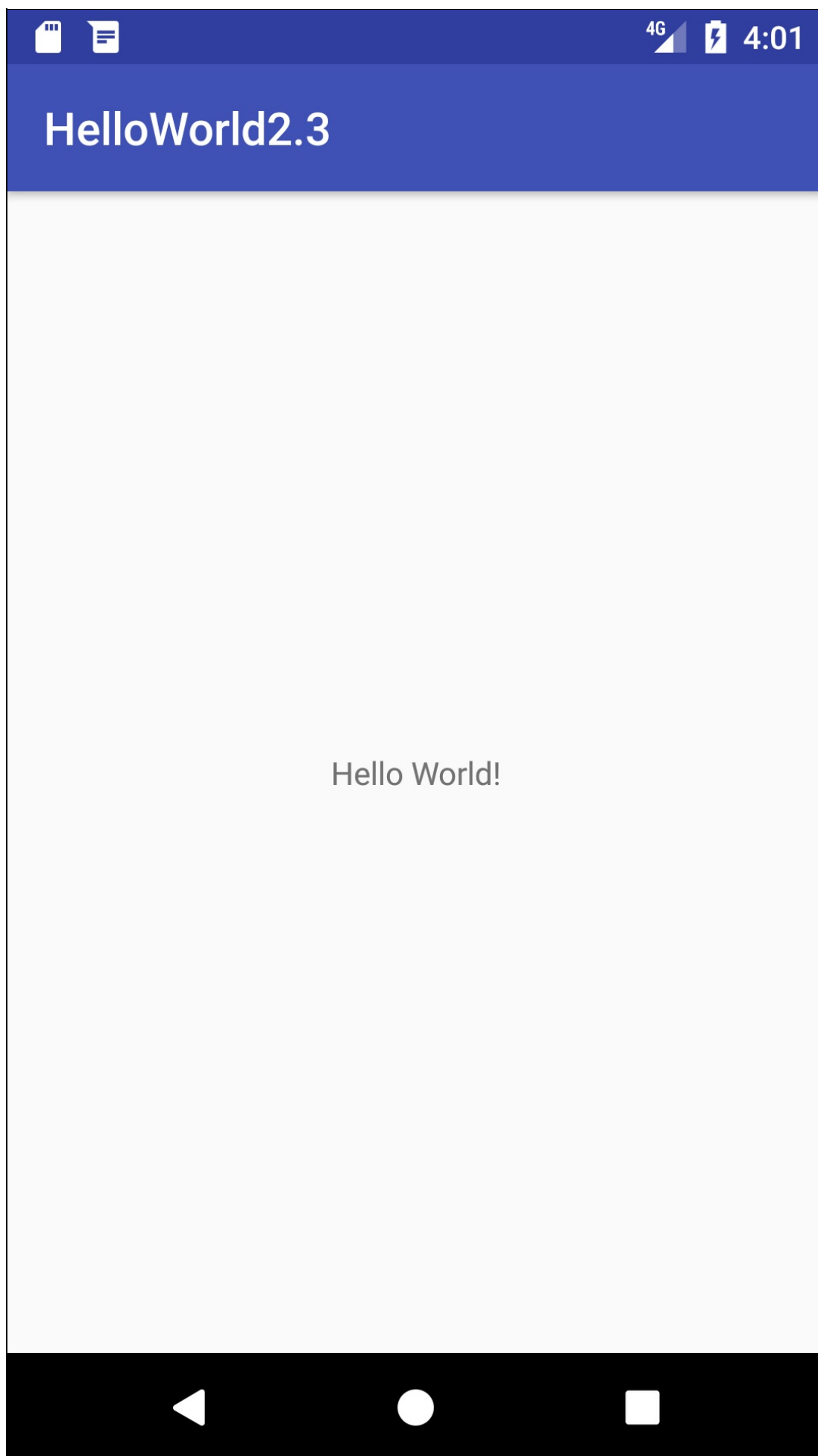


## 5.1 Jalankan aplikasi Anda di emulator

1. Di Android Studio, pilih **Run > Run app** atau klik ikon **Run**  di bilah alat.
2. Di jendela **Select Deployment Target**, di bawah **Available Emulators**, pilih **Nexus 5 API 23** dan klik **OK**.

Emulator dimulai dan di-boot, seperti perangkat fisik. Tergantung kecepatan komputer Anda, ini mungkin akan memerlukan beberapa saat. Aplikasi Anda dibangun, dan setelah emulator siap, Android Studio akan mengunggah aplikasi ke emulator dan menjalankannya.

Anda seharusnya melihat aplikasi Hello World seperti yang ditampilkan di tangkapan layar berikut.



**Catatan:** Saat menguji emulator, praktik terbaiknya adalah dengan memulainya sekali, di awal sesi Anda. Jangan tutup emulator sampai Anda selesai menguji aplikasi, agar aplikasi tidak perlu melalui proses booting lagi.

## Tantangan penyusunan kode

**Catatan:** Semua tantangan penyusunan kode bersifat opsional dan tidak diwajibkan untuk praktik selanjutnya.

**Tantangan:** Anda bisa sepenuhnya menyesuaikan perangkat virtual.

- Pelajari [dokumentasi AVD Manager](#).
- Buat satu atau beberapa perangkat virtual khusus.

Anda mungkin melihat bahwa tidak semua kombinasi perangkat dan versi sistem berfungsi saat menjalankan aplikasi. Ini karena tidak semua gambar sistem dapat berjalan di semua perangkat keras.

## Tugas 6. Menambahkan pernyataan log ke aplikasi Anda

Dalam praktik ini, Anda akan menambahkan pernyataan log ke aplikasi, yang ditampilkan di jendela pencatatan log Android Monitor.

**Mengapa:** Pesan log adalah alat debug bermanfaat yang bisa Anda gunakan untuk memeriksa nilai, jalur eksekusi, dan melaporkan pengecualian.

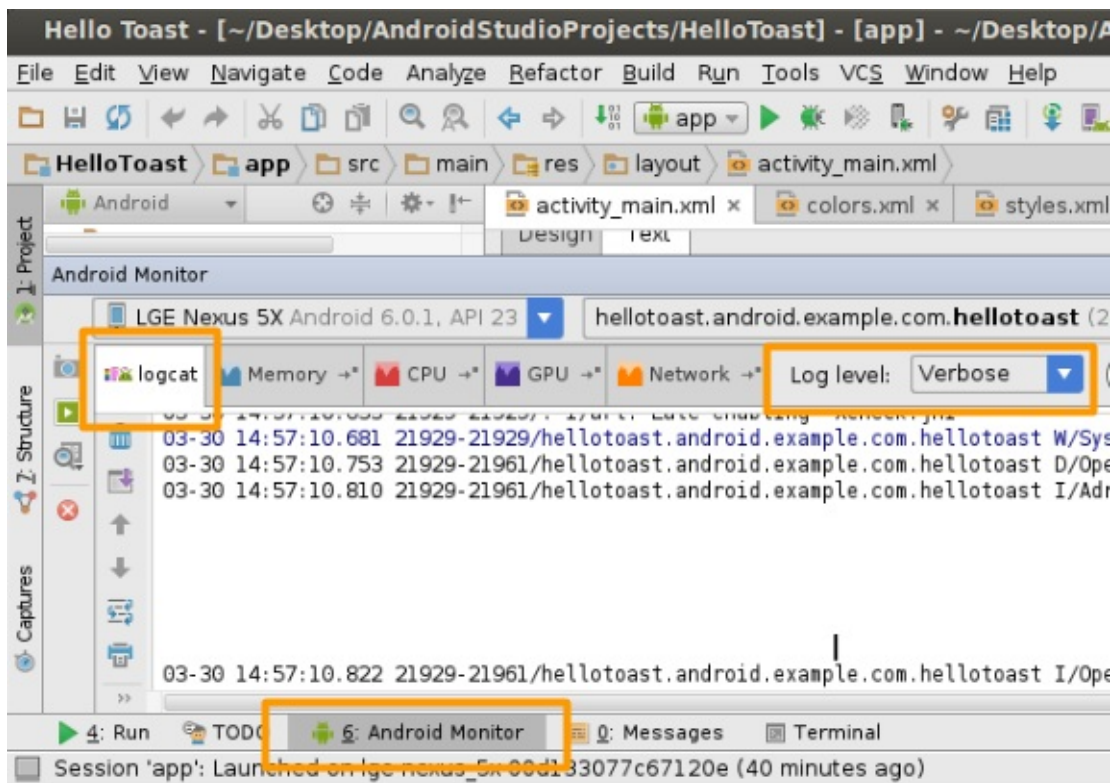
**Android Monitor** menampilkan informasi tentang aplikasi Anda.

1. Klik tombol **Android Monitor** di bawah Android Studio untuk membuka Android Monitor.

Secara default, ini membuka tab **logcat**, yang menampilkan informasi tentang aplikasi Anda saat sedang dijalankan. Jika Anda menambahkan pernyataan log ke aplikasi, pernyataan dicetak di sini juga.

Anda juga bisa memantau Memori, CPU, GPU, dan kinerja jaringan aplikasi dari tab lain Android Monitor. Ini dapat membantu debug dan kinerja menyempurnakan kode.

2. Level log default adalah **Verbose**. Di menu tarik-turun, ubah level log ke **Debug**.



Pernyataan log yang Anda tambahkan ke kode aplikasi mencetak pesan yang ditentukan oleh Anda di tab logcat Android Monitor. Misalnya:

```
Log.d("MainActivity", "Hello World");
```

Bagian dari pesan tersebut adalah:

- Log – [Kelas Log](#). API untuk mengirim pesan log.
- d – Tingkat Log. Digunakan untuk memfilter tampilan pesan di logcat. "d" untuk debug. Tingkat log lainnya adalah "e" untuk error, "w" untuk warning, dan "i" untuk info.
- "MainActivity" – Argumen pertama adalah tag yang dapat digunakan untuk memfilter pesan di logcat. Ini biasanya adalah nama aktivitas asal pesan. Tetapi, Anda dapat menjadikannya apa saja yang berguna untuk men-debug.

Sesuai ketentuan, tag log didefinisikan sebagai konstanta:

```
private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

- "Hello world" – Argumen kedua adalah pesan yang sebenarnya.

## 6.1 Menambahkan pernyataan log ke aplikasi Anda

1. Buka aplikasi Hello World di Android studio, dan buka file MainActivity.
2. **File > Settings > Editor > General > Auto Import** (Mac: **Android Studio > Preferences > Editor > General > Auto Import**). Pilih semua kotak centang dan setel **Insert imports on paste** ke **All**. Impor yang jelas sekarang ditambahkan secara otomatis ke file Anda. Catat bahwa opsi "add unambiguous imports on the fly" penting bagi beberapa fitur Android seperti NumberFormat. Jika tidak dicentang, NumberFormat menampilkan kesalahan. Klik 'Apply' lalu klik tombol 'Ok'.
3. Dalam metode onCreate, tambahkan pernyataan log berikut:

```
Log.d("MainActivity", "Hello World");
```

4. Jika Android Monitor belum dibuka, klik tab Android Monitor di bawah Android Studio untuk membukanya. (Lihat

tangkapan layar.)

5. Pastikan level Log di logcat Android Monitor disetel ke Debug atau Verbose (default).
6. Jalankan aplikasi Anda.

#### Kode Solusi:

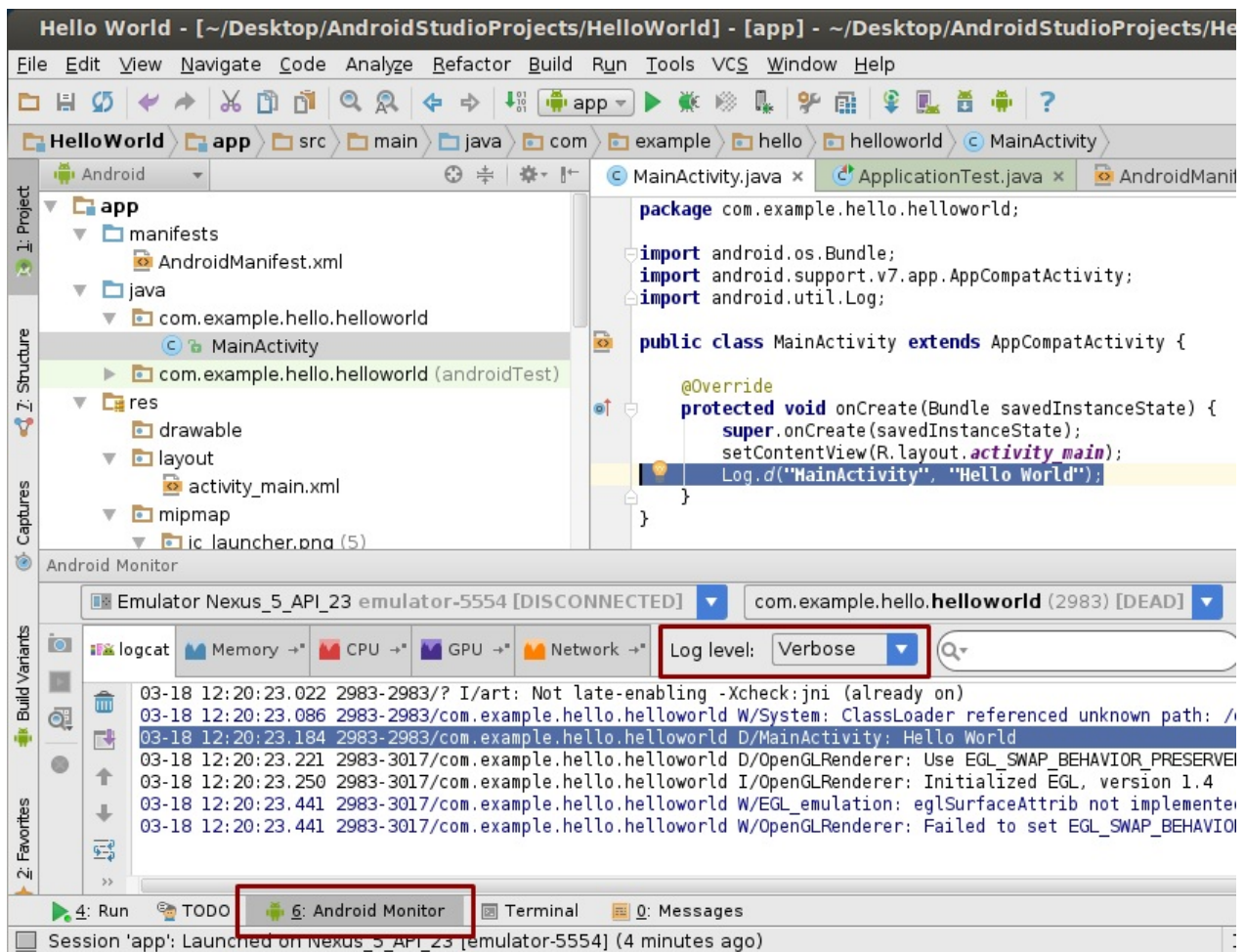
```
package com.example.hello.helloworld;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("MainActivity", "Hello World");
    }
}
```

#### Pesan Log Output

```
03-18 12:20:23.184 2983-2983/com.example.hello.helloworld D/MainActivity: Hello World
```



## Tantangan penyusunan kode

**Catatan:** Semua tantangan penyusunan kode opsional dan bukan prasyarat untuk bab berikutnya.

**Tantangan:** Penggunaan umum kelas Log adalah mencatat log pengecualian Java saat terjadi di program Anda. Ada beberapa metode berguna dalam kelas Log yang dapat digunakan untuk tujuan ini. Gunakan [dokumentasi kelas Log](#) untuk mencari tahu metode apa yang bisa Anda gunakan untuk menyertakan pengecualian dengan pesan log. Lalu, tulis kode di file MainActivity.java untuk memicu dan mencatat pengecualian.

## Tugas 7: Menjelajahi file AndroidManifest.xml

Setiap aplikasi menyertakan file Manifes Android ( `AndroidManifest.xml` ). File manifes berisi informasi penting tentang aplikasi Anda dan menghadirkan informasi ini ke sistem waktu proses Android. Android harus memiliki informasi ini sebelum dapat menjalankan kode aplikasi apa pun.

Dalam praktik ini Anda akan menemukan dan membaca file AndroidManifest.xml untuk aplikasi Hello World.

**Mengapa:** Saat aplikasi menambahkan lebih banyak fungsionalitas dan pengalaman pengguna jadi lebih menarik dan interaktif, file AndroidManifest.xml berisi lebih banyak lagi informasi. Pada pelajaran berikutnya, Anda akan memodifikasi file ini untuk menambahkan fitur dan izin fitur.

### 7.1 Jelajahi file AndroidManifest.xml

1. Buka aplikasi Hello World di Android studio, dan di folder **manifests** folder, buka **AndroidManifest.xml**.
2. Baca file dan pertimbangkan apa yang ditunjukkan oleh setiap kode. Kode di bawah ini dianotasi untuk memberi Anda petunjuk.

**Kode beranotasi:**

```

<!-- XML version and character encoding -->
<?xml version="1.0" encoding="utf-8"?>
<!-- Required starting tag for the manifest -->
<manifest
  <!-- Defines the android namespace. Do not change. -->
  xmlns:android="http://schemas.android.com/apk/res/android"
  <!-- Unique package name of your app. Do not change once app is
    published. -->
    package="com.example.hello.helloworld">
  <!-- Required application tag -->
    <application
      <!-- Allow the application to be backed up and restored. -->
      android:allowBackup="true"
      <!-- Icon for the application as a whole,
        and default icon for application components. -->
      android:icon="@mipmap/ic_launcher"
      <!-- User-readable for the application as a whole,
        and default icon for application components. Notice that Android
        Studio first shows the actual label "Hello World".
        Click on it, and you will see that the code actually refers to a string
        resource. Ctrl-click @string/app_name to see where the resource is
        specified. This will be covered in a later practical . -->
      android:label="@string/app_name"
      <!-- Whether the app is willing to support right-to-left layouts.-->
      android:supportsRtl="true"
      <!-- Default theme for styling all activities. -->
      android:theme="@style/AppTheme">
      <!-- Declares an activity. One is required.
        All activities must be declared,
        otherwise the system cannot see and run them. -->
      <activity
        <!-- Name of the class that implements the activity;
          subclass of Activity. -->
        android:name=".MainActivity">
        <!-- Specifies the intents that this activity can respond to.-->
        <intent-filter>
          <!-- The action and category together determine what
            happens when the activity is launched. -->
          <!-- Start activity as the main entry point.
            Does not receive data. -->
          <action android:name="android.intent.action.MAIN" />
          <!-- Start this activity as a top-level activity in
            the launcher . -->
          <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
    </application>
  </manifest>

```

## Tantangan penyusunan kode

**Catatan:** Semua tantangan penyusunan kode bersifat opsional.

**Tantangan:** Ada banyak elemen lain yang dapat disetel di Android Manifest. Jelajahi [dokumentasi Android Manifest](#) dan pelajari tentang elemen tambahan di Android Manifest.

## Tugas 8. Menjelajahi file build.gradle

Android Studio menggunakan sistem pembangunan bernama Gradle. Gradle melakukan pembangunan bertahap, yang memungkinkan siklus edit-uji yang lebih singkat.

Untuk mengetahui selengkapnya tentang Gradle, buka:

- [Situs Gradle](#)

- Dokumentasi developer [Mengonfigurasi pembangunan](#)
- Telusuri "tutorial gradle" di internet.

Dalam tugas ini, Anda akan menjelajahi file `build.gradle`.

**Mengapa:** Saat menambahkan pustaka baru ke proyek Android, Anda mungkin juga harus memperbarui file **build.gradle**. Mengetahui di mana file tersebut dan struktur dasarnya akan berguna bagi Anda.

## 8.1 Jelajahi file build.gradle(Module.app)

1. Di hierarki proyek, cari **Gradle Scripts** dan luaskan. Ada beberapa file build.gradle. Satu dengan arahan untuk seluruh proyek dan satu untuk setiap modul aplikasi. Modul untuk aplikasi Anda disebut "app". Pada tampilan Proyek, modul ini diwakili oleh folder **app** di tingkat atas tampilan Proyek.
2. Buka **build.gradle (Module.app)**.
3. Baca file dan pelajari apa yang ditunjukkan oleh setiap baris kode.

**Solusi:**

```
// Add Android-specific build tasks
apply plugin: 'com.android.application'
// Configure Android specific build options.
android {
    // Specify the target SDK version for the build.
    compileSdkVersion 23
    // The version of the build tools to use.
    buildToolsVersion "23.0.2"
    // Core settings and entries. Overrides manifest settings!
    defaultConfig {
        applicationId "com.example.hello.helloworld"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    // Controls how app is built and packaged.
    buildTypes {
        // Another common option is debug, which is not signed by default.
        release {
            // Code shrinker. Turn this on for production along with
            // shrinkResources.
            minifyEnabled false
            // Use ProGuard, a Java optimizer.
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
// This is the part you are most likely to change as you start using
// other libraries.
dependencies {
    // Local binary dependency. Include any JAR file inside app/libs.
    compile fileTree(dir: 'libs', include: ['*.jar'])
    // Configuration for unit tests.
    testCompile 'junit:junit:4.12'
    // Remote binary dependency. Specify Maven coordinates of the Support
    // Library needed. Use the SDK Manager to download and install such
    // packages.
    compile 'com.android.support:appcompat-v7:23.2.1'
}
```

- Untuk lebih memahami Gradle baca [Ringkasan Sistem Pembangunan](#) dan dokumentasi [Mengonfigurasi Build Gradle](#).
- Terdapat alat yang dapat membantu Anda [meringkas kode](#), menghapus pustaka/sumber daya tidak penting dan bahkan menyamarkan program untuk mencegah rekayasa balik yang tidak diinginkan.
- Android Studio sendiri menyediakan beberapa fitur yang berguna. Pelajari lebih lanjut tentang alat sumber terbuka yang bermanfaat yang bernama [ProGuard](#).



## Tugas 9. [Opsional] Menjalankan aplikasi Anda di perangkat

Pada tugas terakhir ini, Anda akan menjalankan aplikasi pada perangkat seluler fisik seperti ponsel atau tablet.

**Mengapa:** Pengguna akan menjalankan aplikasi di perangkat fisik. Anda harus selalu menguji aplikasi pada perangkat fisik dan virtual.

Yang Anda butuhkan:

- Perangkat Android seperti ponsel atau tablet.
- Kabel data untuk menghubungkan perangkat Android ke komputer melalui port USB.
- Jika menggunakan OS Linux atau Windows, Anda mungkin perlu melakukan langkah tambahan untuk menjalankannya di perangkat keras. Periksa dokumentasi [Menggunakan Perangkat Keras](#). Pada Windows, Anda mungkin perlu memasang driver USB yang sesuai untuk perangkat. Lihat [Driver US OEM](#).


### Tugas 9.1. [Opsional] Menjalankan aplikasi Anda di perangkat

Untuk mengizinkan Android Studio berkomunikasi dengan perangkat, Anda harus mengaktifkan USB Debugging di perangkat Android. Ini diaktifkan di setelan Developer options perangkat. Perlu diingat bahwa ini tidak sama dengan melakukan rooting perangkat.

Pada Android 4.2 dan yang lebih tinggi, layar Developer options disembunyikan secara default. Untuk menampilkan Developer options dan mengaktifkan USB Debugging:

1. Pada perangkat, buka **Settings > About phone** dan ketuk **Build number** tujuh kali.
2. Kembali ke layar sebelumnya (**Settings**). **Developer options** muncul di bawah daftar. Klik **Developer options**.
3. Pilih **USB Debugging**.

Sekarang Anda dapat menghubungkan perangkat dan menjalankan aplikasi dari Android Studio.

1. Hubungkan perangkat ke mesin development dengan kabel USB.
2. Di Android Studio, di bawah jendela, klik tab Android Monitor. Anda seharusnya melihat perangkat terdaftar di menu tarik-turun kiri atas.
3. Klik tombol **Run**  di bilah alat. Jendela **Select Deployment Target** terbuka dengan daftar emulator yang tersedia dan perangkat yang terhubung.
4. Pilih perangkat dan klik **OK**.

Android Studio seharusnya memasang dan menjalankan aplikasi di perangkat Anda.

### Pemecahan Masalah

Jika Android Studio tidak mengenali perangkat, coba langkah berikut:

- Cabut dan colokkan kembali perangkat.
- Mulai ulang Android Studio.
- Jika komputer masih tidak mengenali perangkat dan mendeklarasikannya "unauthorized":
  1. Cabut perangkat.
  2. Pada perangkat, buka Settings->Developer Options.
  3. Ketuk **Revoke USB Debugging authorizations**.
  4. Hubungkan kembali perangkat ke komputer.
  5. Saat diminta, berikan otorisasi.
- Anda mungkin perlu memasang driver USB yang sesuai untuk perangkat. Baca [dokumentasi Menggunakan Perangkat Keras](#).
- Periksa dokumentasi terbaru, forum pemrograman, atau minta bantuan instruktur Anda.

## Tantangan penyusunan kode

**Catatan:** Semua tantangan penyusunan kode bersifat opsional.

**Tantangan:** Sekarang karena Anda sudah siap dan mengenal alur kerja development, lakukan hal berikut:

1. Buat proyek baru di Android Studio.
2. Ganti sapaan dengan "Happy Birthday to " dan seseorang yang baru saja berulang tahun.
3. Ganti latar belakang aplikasi menggunakan gambar bertema ulang tahun.
4. Ambil tangkapan layar dari aplikasi yang telah selesai dan kirimkan lewat email ke seseorang yang ulang tahunnya Anda lupakan.

## Rangkuman

Pada bab ini, Anda telah mempelajari cara:

- Memasang Android Studio
  - Mendapatkan pemahaman dasar tentang alur kerja development setelah membukanya di Android Studio.
  - Mendapatkan pemahaman dasar tentang struktur aplikasi Android di lingkungan pembangunannya.
  - Mendapatkan pemahaman dasar tentang Manifes Android dan kegunaannya.
  - Menambahkan pernyataan log ke kode yang memberi Anda alat dasar untuk proses debug.
- Menerapkan aplikasi Hello World pada emulator Android dan [opsional] perangkat seluler.