Android Studio

Praktikum Pemrograman Mobile - 02

Wanda Gusdya

2023

Pendahuluan

Android Studio adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android. Berbasis editor kode dan alat developer yang andal dari IntelliJ IDEA , Android Studio menawarkan lebih banyak fitur yang mampu meningkatkan produktivitas Anda saat mem-build aplikasi Android, seperti:

- Sistem build berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android
- Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
- Framework dan alat pengujian yang lengkap
- Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- Dukungan C++ dan NDK
- Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine

Struktur project

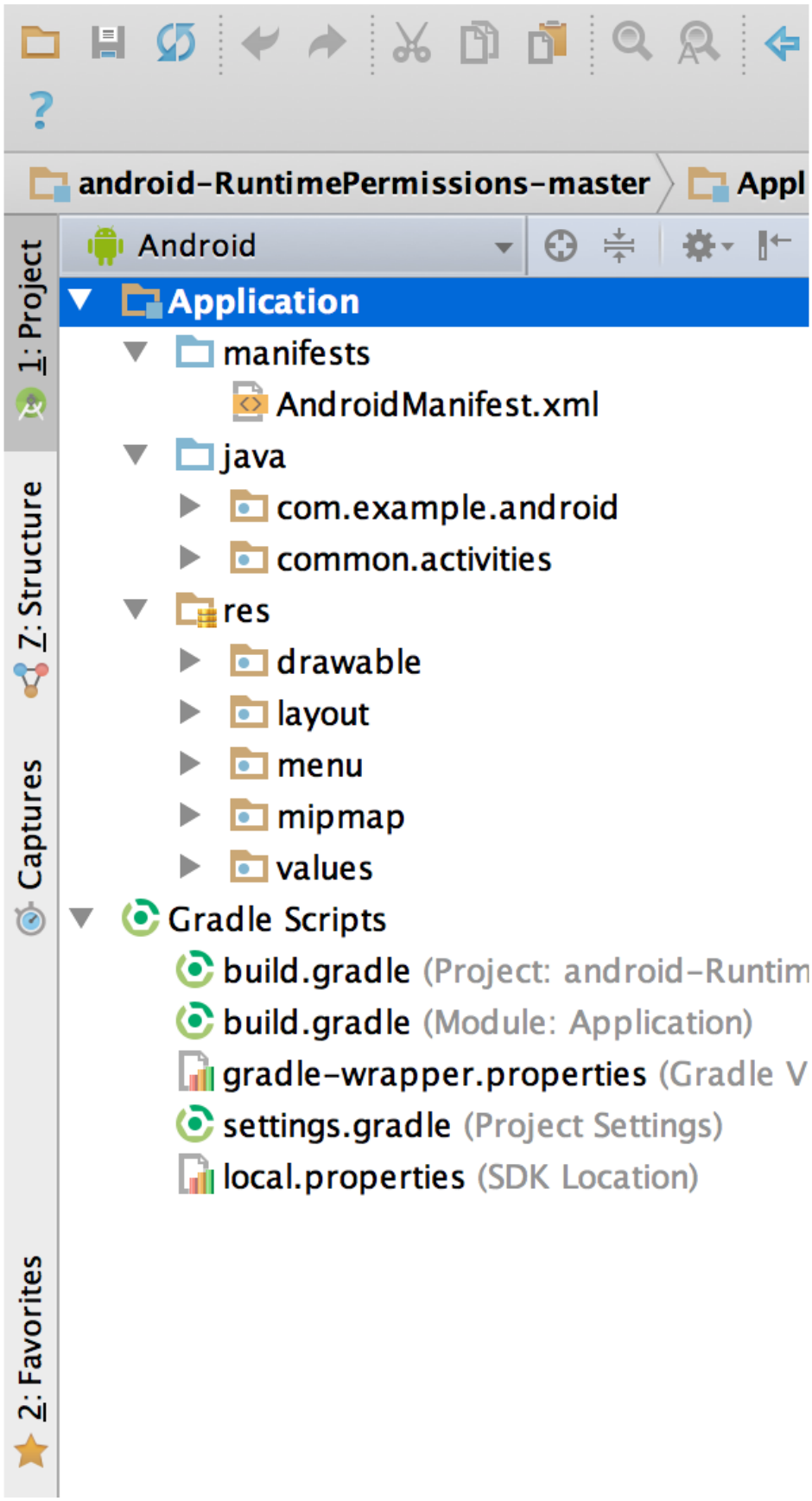
Setiap project di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file resource. Jenis modul meliputi:

- Modul aplikasi Android
- Modul library
- Modul Google App Engine

Secara default, Android Studio menampilkan file project Anda dalam tampilan project Android, seperti yang ditunjukkan pada gambar 1. Tampilan ini disusun menurut modul untuk memberikan akses cepat ke file sumber utama project Anda. Semua file build terlihat di tingkat atas, di bagian Gradle Scripts.

Setiap modul aplikasi berisi folder berikut:

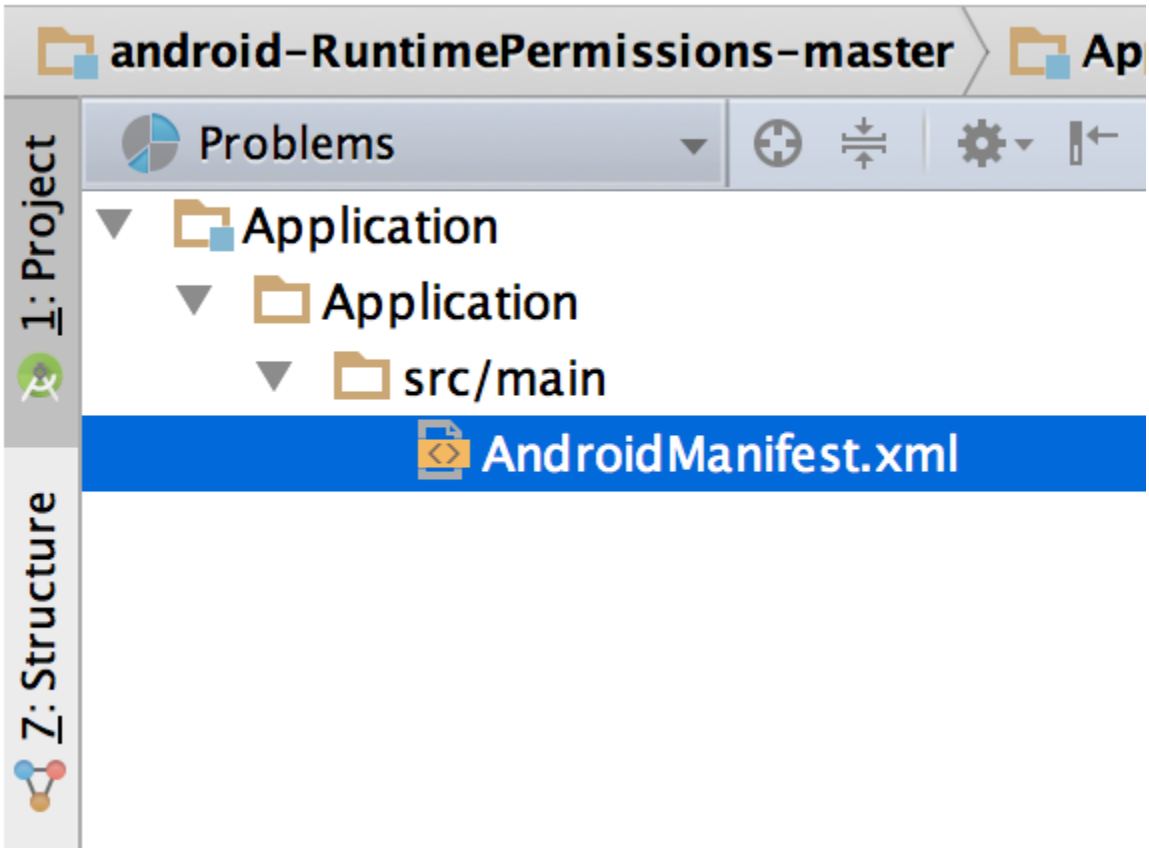
- manifes: Berisi file AndroidManifest.xml.
- java: Berisi file kode sumber Java dan Kotlin, termasuk kode pengujian JUnit.
- res: Berisi semua resource non-kode, seperti tata letak XML, string UI, dan gambar bitmap.


android-RuntimePermissions-master
Appl
Android
Application
manifests
AndroidManifest.xml
java
com.example.android
common.activities
res
drawable
layout
menu
mipmap
values
Gradle Scripts
build.gradle (Project: android-Runtim
build.gradle (Module: Application)
gradle-wrapper.properties (Gradle V
settings.gradle (Project Settings)
local.properties (SDK Location)
1: Project
Z: Structure
Captures
2: Favorites

Gambar 1 File project dalam tampilan project Android.

Struktur project Android pada disk berbeda dengan representasi tersatukan ini. Untuk melihat struktur file project sebenarnya, pilih Project, bukan Android, dari menu Project.

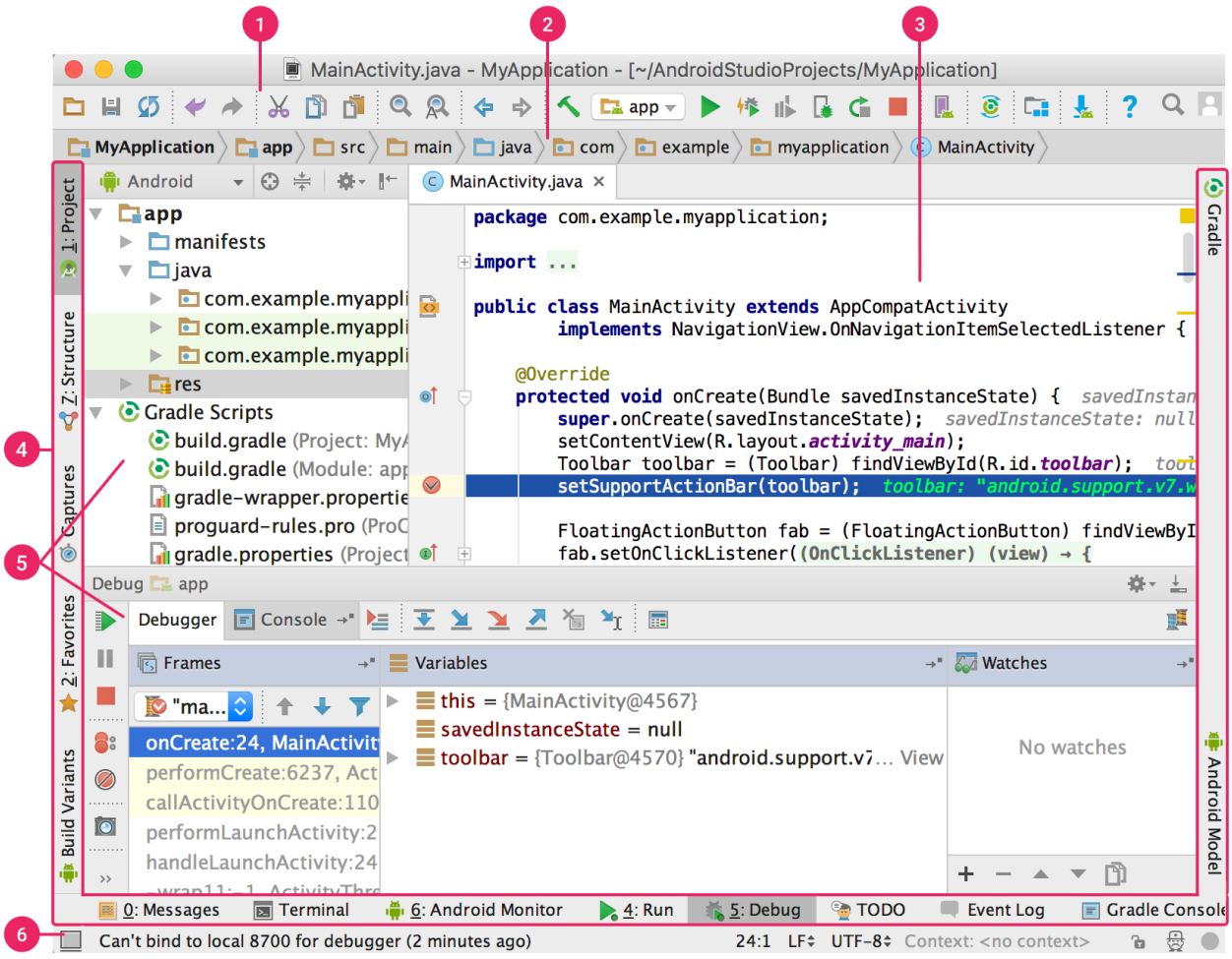
Anda juga dapat menyesuaikan tampilan file project untuk berfokus pada aspek spesifik dari pengembangan aplikasi Anda. Misalnya, pilih tampilan Problems project Anda untuk menampilkan link ke file sumber yang berisi error coding dan sintaksis yang dikenali, seperti tag penutup elemen XML yang tidak ada dalam file tata letak.



Gambar 2 File tata letak yang memiliki masalah di tampilan Problems.

Antarmuka pengguna

Jendela utama Android Studio terdiri dari beberapa area logis yang ditampilkan dalam gambar 3.



Gambar 3 Jendela utama Android Studio.

1. Toolbar: Menjalankan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
2. Menu navigasi: Membuka project Anda dan membuka file untuk diedit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela Project.
3. Jendela editor: Membuat dan mengubah kode. Bergantung pada jenis file yang ada, editor ini dapat berubah. Misalnya, saat menampilkan file tata letak, editor akan menampilkan Layout Editor.
4. Kolom jendela alat: Menggunakan tombol di bagian luar jendela IDE untuk meluaskan atau menciutkan setiap jendela alat.
5. Jendela alat: Mengakses tugas tertentu seperti pengelolaan project, penelusuran, kontrol versi, dan banyak lagi. Anda dapat memperluas dan menciutkan jendela ini.
6. Status bar Menampilkan status project Anda dan IDE itu sendiri, serta semua peringatan atau pesan.

Untuk mengatur jendela utama agar mendapatkan lebih banyak ruang layar, sembunyikan atau pindahkan toolbar dan jendela alat. Anda juga dapat menggunakan pintasan keyboard untuk mengakses sebagian besar fitur IDE.

Untuk menelusuri seluruh kode sumber, database, tindakan, elemen antarmuka pengguna, dan lainnya, lakukan salah satu tindakan berikut:

- Tekan dua kali tombol Shift.
- Klik kaca pembesar yang ada di sudut kanan atas jendela Android Studio.

Tips ini sangat berguna jika, misalnya, Anda mencoba menemukan tindakan IDE tertentu yang Anda lupa cara memicunya.

Jendela alat

Sebagai ganti menggunakan perspektif preset, Android Studio mengikuti konteks Anda dan otomatis menampilkan jendela alat yang relevan saat Anda bekerja. Secara default, jendela alat yang paling umum digunakan disematkan ke kolom jendela alat di tepi jendela aplikasi.

Buka jendela alat dengan cara berikut:

- Untuk memperluas atau menciutkan jendela alat, klik nama alat di kolom jendela alat. Anda juga dapat menarik, menyematkan, melepaskan sematan, memasang, dan melepas jendela alat.
- Untuk kembali ke tata letak default jendela alat saat ini, klik Window > Restore Default Layout. Untuk menyesuaikan tata letak default, klik Window > Store Current Layout as Default.
- Untuk menampilkan atau menyembunyikan seluruh kolom jendela alat, klik ikon jendela di pojok kiri bawah jendela Android Studio.
- Untuk menemukan jendela alat tertentu, arahkan kursor ke atas ikon jendela dan pilih jendela alat tersebut dari menu.

Anda juga bisa menggunakan pintasan keyboard untuk membuka jendela alat. Tabel 1 mencantumkan pintasan untuk jendela alat yang paling umum.

Tabel 1 Pintasan keyboard untuk jendela alat

Jendela alat	**Windows dan Linux**	**macOS**
Project	Alt+1	Command+1
Kontrol Versi	Alt+9	Command+9
Run	Shift+F10	Control+R
Debug	Shift+F9	Control+D
Logcat	Alt+6	Command+6
Kembali ke Editor	Esc	Esc
Menyembunyikan Semua Jendela Alat	Control+Shift+F12	Command+Shift+F12

Untuk menyembunyikan semua toolbar, jendela alat, dan tab editor, klik View > Enter Distraction Free Mode. Untuk keluar dari Distraction Free Mode, klik View > Exit Distraction Free Mode.

Gunakan Speed Search untuk menelusuri dan memfilter di sebagian besar jendela alat di Android Studio. Untuk menggunakan Speed Search, pilih jendela alat, lalu ketik kueri penelusuran Anda.

Pelengkapan kode

Android Studio memiliki tiga jenis pelengkapan kode, yang dapat Anda akses menggunakan pintasan keyboard.

Tabel 2 Pintasan keyboard untuk pelengkapan kode

Jenis	Deskripsi	Windows dan Linux	macOS
Basic Completion	Menampilkan saran dasar untuk variabel, jenis, metode, ekspresi, dan sebagainya. Jika memanggil Basic Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk anggota pribadi dan anggota statis yang tidak diimpor.	Control+Spasi	Control+Spasi
Smart Completion	Menampilkan opsi yang relevan berdasarkan konteks. Smart Completion mengetahui jenis dan alur data yang diharapkan. Jika Anda memanggil Smart Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk chain.	Control+Shift+Spasi	Control+Shift+Spasi
Statement Completion	Melengkapi pernyataan saat ini, seperti menambahkan tanda kurung, tanda kurung siku, tanda kurung kurawal, pemformatan, dan sebagainya yang tidak ada.	Control+Shift+Enter	Command+Shift+Enter

Untuk melakukan perbaikan cepat dan menampilkan tindakan intent, tekan Alt+Enter.

Navigasi

Berikut beberapa tips yang dapat membantu Anda membuka Android Studio.

- Gunakan tindakan Recent Files untuk beralih antar-file yang baru-baru ini diakses:

- Untuk memunculkan tindakan Recent Files, tekan Control+E (Command+E di macOS). Secara default, file yang terakhir diakses akan dipilih. Dengan tindakan ini, Anda juga dapat mengakses jendela alat apa pun melalui kolom sebelah kiri.
- Gunakan tindakan File Structure untuk melihat struktur file saat ini dan membuka bagian file mana pun saat ini dengan cepat:
- Untuk memunculkan tindakan File Structure, tekan Control+F12 (Command+F12 di macOS).
- Gunakan tindakan Navigate to Class untuk menelusuri dan menavigasi ke class tertentu dalam project Anda. Navigate to Class mendukung ekspresi lanjutan, seperti camel humps (yang memungkinkan Anda menelusuri huruf kapital dalam nama camel-case dari suatu elemen), jalur, line navigate to (yang memungkinkan Anda memilih baris tertentu dalam file), middle name matching (yang memungkinkan Anda menelusuri bagian nama class), dan banyak lagi. Jika Anda memanggilnya dua kali berturut-turut, hasil dari class project akan ditampilkan.
- Untuk memunculkan tindakan Navigate to Class, tekan Control+N (Command+O di macOS).
- Gunakan tindakan Navigate to File untuk membuka file atau folder:
- Untuk memunculkan tindakan Navigate to File, tekan Control+Shift+N (Command+Shift+O di macOS). Untuk menelusuri folder, bukan file, tambahkan "/" di akhir ekspresi.
- Gunakan tindakan Navigate to Symbol untuk membuka metode atau kolom berdasarkan nama:
- Untuk memunculkan tindakan Navigate to Symbol, tekan Control+Shift+Alt+N (Command+Option+O di macOS).

Untuk menemukan semua bagian kode yang merujuk ke class, metode, kolom, parameter, atau pernyataan di posisi kursor saat ini, tekan Alt+F7 (Opsi+F7 di macOS).

Gaya dan pemformatan

Saat Anda mengedit, Android Studio otomatis menerapkan pemformatan dan gaya seperti yang ditentukan dalam setelan gaya kode Anda. Anda dapat menyesuaikan setelan gaya kode menurut bahasa pemrograman, termasuk menentukan konvensi untuk tab dan indentasi, spasi, penggabungan, tanda kurung kurawal, dan baris kosong.

Untuk menyesuaikan setelan gaya kode, klik File > Settings > Editor > Code Style (Android Studio > Preferences > Editor > Code Style di macOS.)

Meskipun IDE secara otomatis menerapkan pemformatan selagi Anda bekerja, Anda juga dapat memanggil tindakan Reformat Code secara eksplisit. Untuk memanggil tindakan tersebut, tekan Control+Alt+L (Opt+Command+L di macOS). Untuk mengindentasi semua baris secara otomatis, tekan Control+Alt+I (Control+Option+I di macOS).

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    mActionBar = getSupportActionBar();
        mActionBar.setDisplayHomeAsUpEnabled(true);
```

Gambar 4 Kode setelah pemformatan.

Dasar-dasar kontrol versi

Android Studio mendukung berbagai sistem kontrol versi (VCS), termasuk Git, GitHub, CVS, Mercurial, Subversion, dan Google Cloud Source Repositories.

Setelah mengimpor aplikasi Anda ke Android Studio, gunakan opsi menu VCS pada Android Studio untuk mengaktifkan dukungan VCS bagi sistem kontrol yang diinginkan, membuat repositori, mengimpor file baru ke kontrol versi, dan menjalankan operasi kontrol versi lainnya.

Untuk mengaktifkan dukungan VCS, ikuti langkah-langkah berikut:

1. Dari menu VCS Android Studio, pilih Enable Version Control Integration.
2. Dari menu, pilih VCS untuk dikaitkan dengan root project.
3. Klik OK.

Menu VCS sekarang menampilkan sejumlah opsi kontrol versi berdasarkan sistem yang Anda pilih.

Sistem build Gradle

Android Studio menggunakan Gradle sebagai dasar sistem build, dengan lebih banyak kemampuan khusus Android yang disediakan oleh plugin Android Gradle. Sistem build ini berjalan sebagai alat terintegrasi dari menu Android Studio dan terpisah dari command line. Anda dapat menggunakan fitur-fitur sistem build untuk:

- Menyesuaikan, mengonfigurasi, dan memperluas proses build.
- Membuat banyak APK untuk aplikasi Anda dengan berbagai fitur yang menggunakan project dan modul yang sama.
- Gunakan kembali kode dan resource ke seluruh set sumber.
- Berkat fleksibilitas Gradle, Anda dapat mencapai semua ini tanpa mengubah file sumber inti aplikasi Anda.

File build Android Studio diberi nama build.gradle. File tersebut adalah file teks biasa yang menggunakan sintaksis Groovy untuk mengonfigurasi build dengan elemen yang disediakan oleh plugin Android Gradle. Setiap project memiliki satu file build tingkat atas untuk seluruh project dan file build tingkat modul terpisah untuk setiap modul. Saat Anda mengimpor project yang ada, Android Studio akan otomatis menghasilkan file build yang diperlukan.

Varian build

Sistem build dapat membantu Anda membuat beberapa versi berbeda untuk aplikasi yang sama dari satu project. Hal ini berguna saat Anda menyediakan aplikasi dalam versi gratis dan berbayar atau jika Anda ingin mendistribusikan beberapa APK untuk berbagai konfigurasi perangkat di Google Play.

Dukungan multi-APK

Dukungan multi-APK memungkinkan Anda membuat beberapa APK sekaligus secara efisien berdasarkan kepadatan layar atau ABI. Misalnya, Anda dapat membuat APK aplikasi terpisah untuk kepadatan layar hdpi dan mdpi, sambil tetap mempertimbangkannya sebagai satu varian dan mengizinkannya berbagi setelan APK pengujian, javac, dx, dan ProGuard.

Penyingkatan resource

Penyingkatan resource di Android Studio otomatis menghapus resource yang tidak digunakan dari aplikasi terpaket dan dependensi library Anda. Misalnya, jika aplikasi Anda menggunakan layanan Google Play untuk mengakses fungsi Google Drive, dan saat ini Anda tidak menggunakan Login dengan Google, penyingkatan resource dapat menghapus beragam aset drawable untuk tombol SignInButton.

Mengelola dependensi

Dependensi untuk project Anda ditetapkan berdasarkan nama dalam file build.gradle. Gradle menemukan dependensi dan menyediakannya di build Anda. Anda dapat mendeklarasikan dependensi modul, dependensi biner jarak jauh, dan dependensi biner lokal dalam file build.gradle Anda.

Android Studio mengonfigurasi project untuk menggunakan Maven Central Repository secara default. Konfigurasi ini disertakan dalam file build level atas untuk project.

Alat profil dan debug

Android Studio membantu Anda men-debug dan meningkatkan performa kode, termasuk proses debug inline dan alat analisis performa.

Proses debug inline

Gunakan proses debug inline untuk meningkatkan panduan kode Anda dalam tampilan debugger dengan verifikasi inline untuk referensi, ekspresi, dan nilai variabel.

Informasi proses debug inline meliputi:

- Nilai variabel inline
- Objek yang mereferensikan objek yang dipilih
- Nilai yang ditampilkan metode
- Ekspresi operator dan lambda
- Nilai tooltip

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {  menu: MenuBuilder@4312
    // Inflate our menu from the resources by using the menu inflater.
    getMenuInflater().inflate(R.menu.main, menu);  menu: MenuBuilder@4312

    // It is also possible add items here. Use a generated id from
    // resources (ids.xml) to ensure that all menu ids are distinct.
    MenuItem locationItem = menu.add(0, R.id.menu_location, 0, "Location");
    locationItem.setIcon(R.drawable.ic_action_location);
```

Gambar 5 Nilai variabel inline.

Untuk mengaktifkan proses debug inline, di jendela Debug, klik Settings , lalu pilih Show Values Inline.

Profiler performa

Android Studio menyediakan profiler performa agar Anda dapat melacak penggunaan memori dan CPU aplikasi, menemukan objek yang batal dialokasikan, menemukan kebocoran memori, mengoptimalkan performa grafis, dan menganalisis permintaan jaringan dengan mudah.

Untuk menggunakan profiler performa, sembari aplikasi Anda berjalan di perangkat atau emulator, buka Android Profiler dengan memilih View > Tool Windows > Profiler.

Heap dump

Saat memprofilkan penggunaan memori di Android Studio, Anda dapat sekaligus memulai pembersihan sampah memori dan membuang heap Java ke cuplikan heap dalam file format biner HPROF khusus Android. Penampil HPROF akan menampilkan class, instance setiap class, dan struktur referensi untuk membantu Anda melacak penggunaan memori serta menemukan kebocoran memori.

Memory Profiler

Gunakan Memory Profiler untuk melacak alokasi memori dan melihat di mana objek dialokasikan saat Anda melakukan tindakan tertentu. Alokasi ini membantu Anda mengoptimalkan penggunaan memori dan performa aplikasi dengan menyesuaikan panggilan metode yang terkait dengan tindakan tersebut.

Akses file data

Android SDK Tools, seperti Systrace dan Logcat, menghasilkan data performa dan proses debug untuk analisis aplikasi secara mendetail.

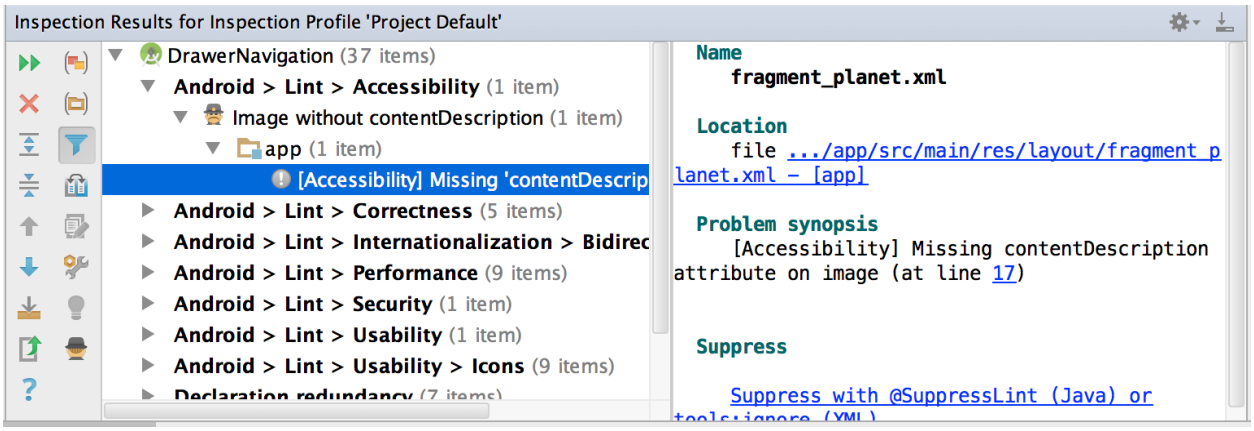
Untuk melihat file data yang dihasilkan:

1. Buka jendela alat Perekaman.
2. Dalam daftar file yang dihasilkan, klik dua kali file untuk melihat data.
3. Klik kanan file HPROF apa pun untuk mengonversinya ke dalam format standar.
4. Cari tahu format file penggunaan RAM Anda.

Pemeriksaan kode

Setiap kali Anda mengompilasi program, Android Studio akan otomatis menjalankan pemeriksaan lint yang telah dikonfigurasi dan pemeriksaan IDE lainnya untuk memudahkan Anda mengidentifikasi serta memperbaiki masalah kualitas struktur kode Anda.

Alat lint memeriksa file sumber project Android untuk menemukan kemungkinan bug dan peningkatan pengoptimalan guna mencapai ketepatan, keamanan, performa, kegunaan, aksesibilitas, serta internasionalisasi.



Gambar 6 Hasil pemeriksaan lint di Android Studio.

Selain pemeriksaan lint, Android Studio juga menjalankan pemeriksaan kode IntelliJ dan memvalidasi anotasi untuk menyederhanakan alur kerja coding Anda.

Anotasi di Android Studio

Android Studio mendukung anotasi variabel, parameter, dan nilai yang dihasilkan untuk membantu Anda mengidentifikasi bug, seperti pengecualian pointer null dan konflik jenis resource.

Android SDK Manager mengemas library Jetpack Annotations di Android Support Repository untuk digunakan dengan Android Studio. Android Studio memvalidasi anotasi yang sudah dikonfigurasi selama pemeriksaan kode.

Pesan log

Saat mem-build dan menjalankan aplikasi dengan Android Studio, Anda dapat melihat output adb dan pesan log perangkat di jendela Logcat.

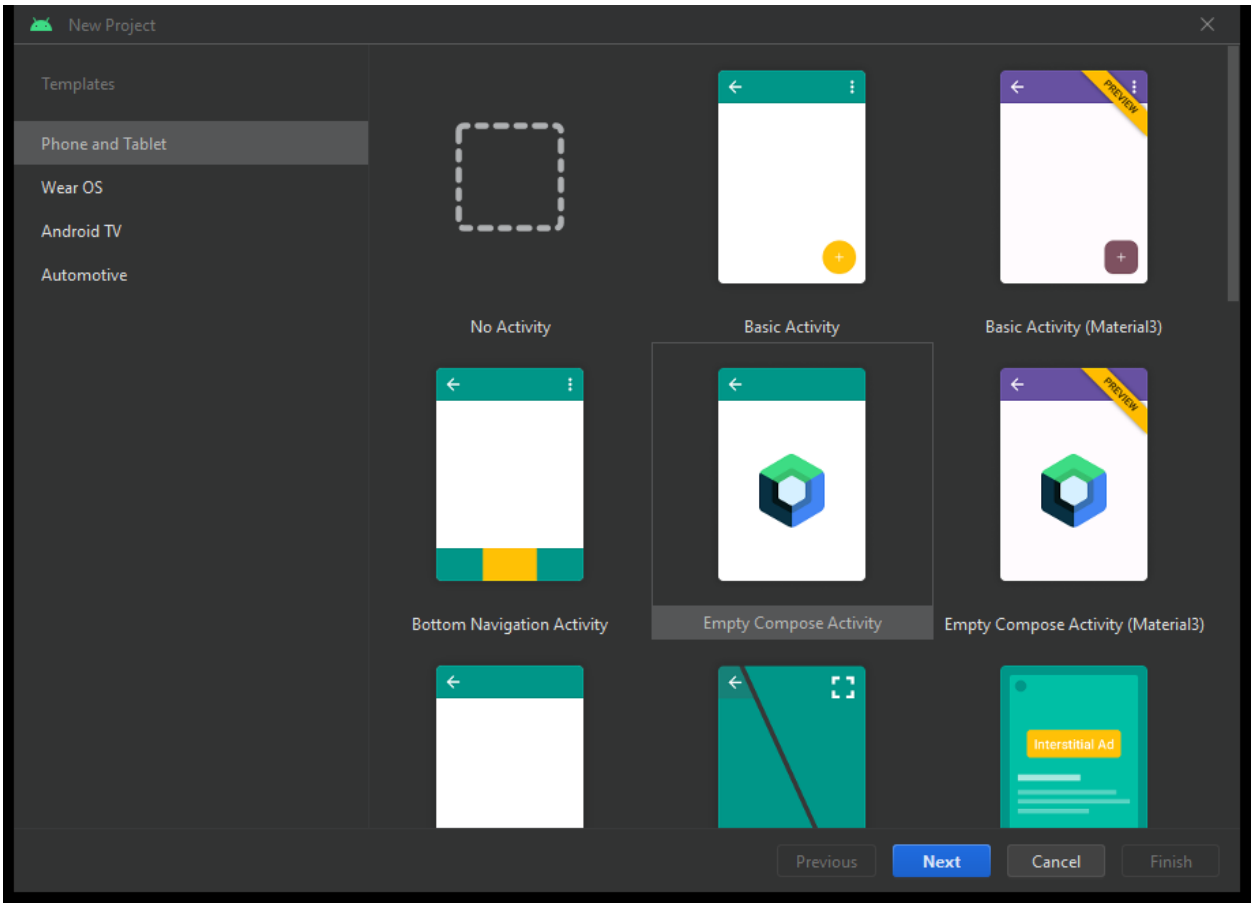
Jetpack Compose

Jetpack Compose adalah toolkit modern untuk mem-build UI Android native. Jetpack Compose menyederhanakan dan mempercepat pengembangan UI di Android dengan kode yang lebih sedikit, alat yang canggih, dan API Kotlin yang intuitif.

Dalam tutorial ini, Anda akan membuat komponen UI sederhana dengan fungsi deklaratif. Anda tidak akan mengedit tata letak XML apa pun atau menggunakan Layout Editor. Sebagai gantinya, Anda akan memanggil fungsi composable untuk menentukan elemen yang diinginkan, dan selebihnya akan ditangani oleh compiler Compose.

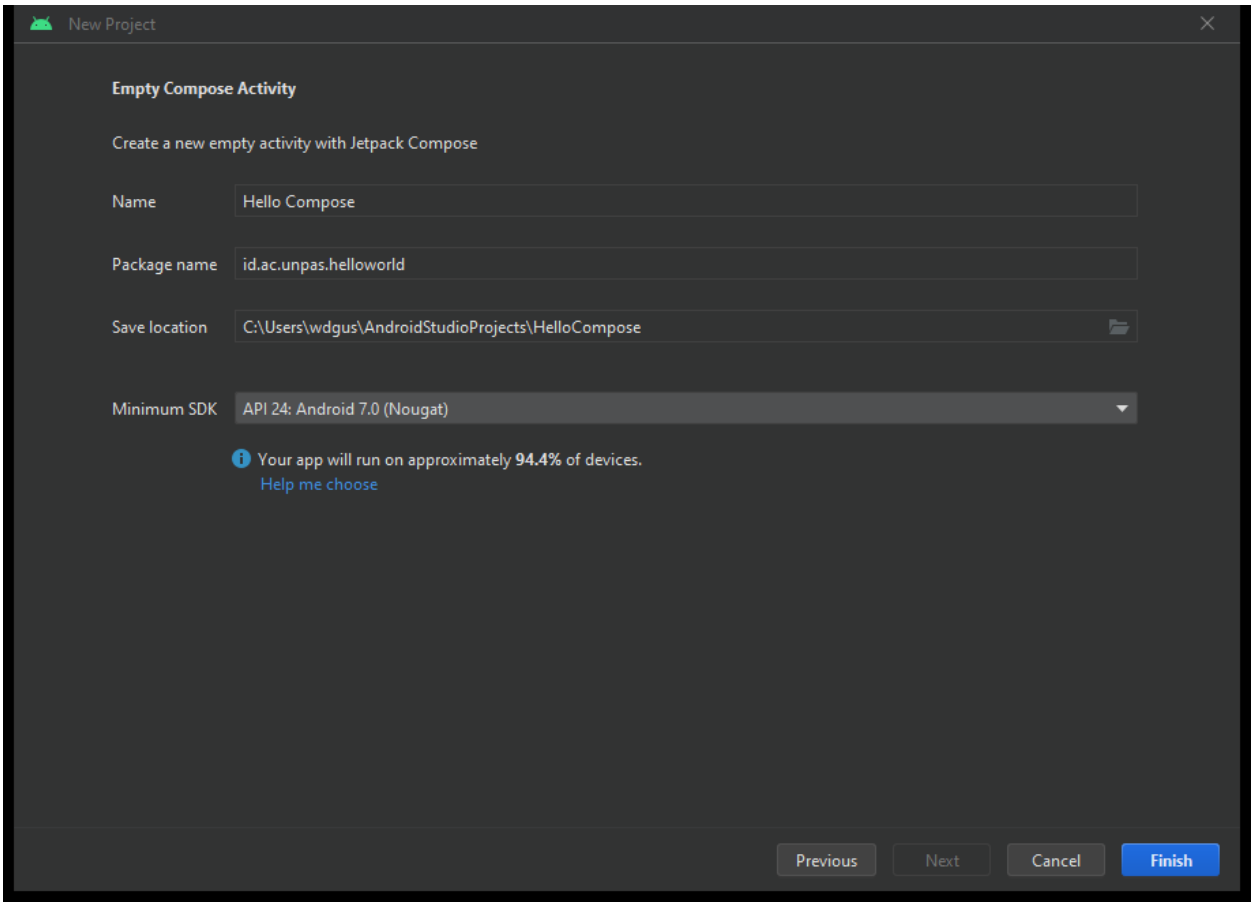
Latihan Hello World

Buatlah project baru untuk Phone and Tablets di Android Studio dengan memilih Empty Compose Activity seperti pada gambar berikut.



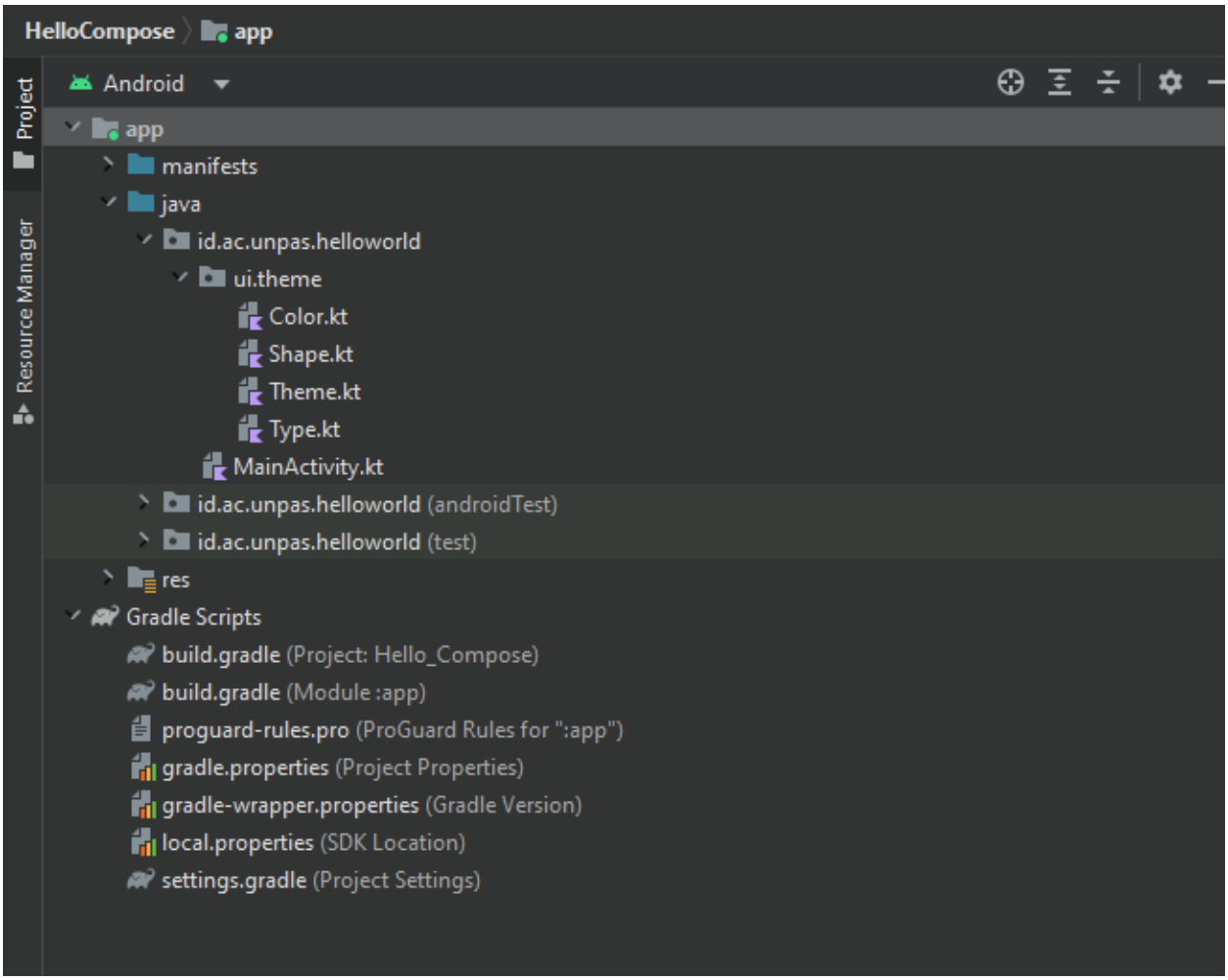
Gambar 7 Konfigurasi New Project

Klik next, kemudian beri nama project dan package pada jendela tersebut.



Gambar 8 Nama project dan package utama

Kemudian klik Finish agar Android Studio mulai membuat project yang Anda inginkan. Berikut ini adalah struktur proyek Anda, Android Studio akan membuatkan file MainActivity beserta file-file yang mendukung tema dan tampilan aplikasi.



Gambar 9 Struktur Proyek HelloCompose

Berikut ini adalah isi kelas MainActivity yang telah dibuat.

```
package id.ac.unpas.helloworld

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import id.ac.unpas.helloworld.ui.theme.HelloComposeTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
```

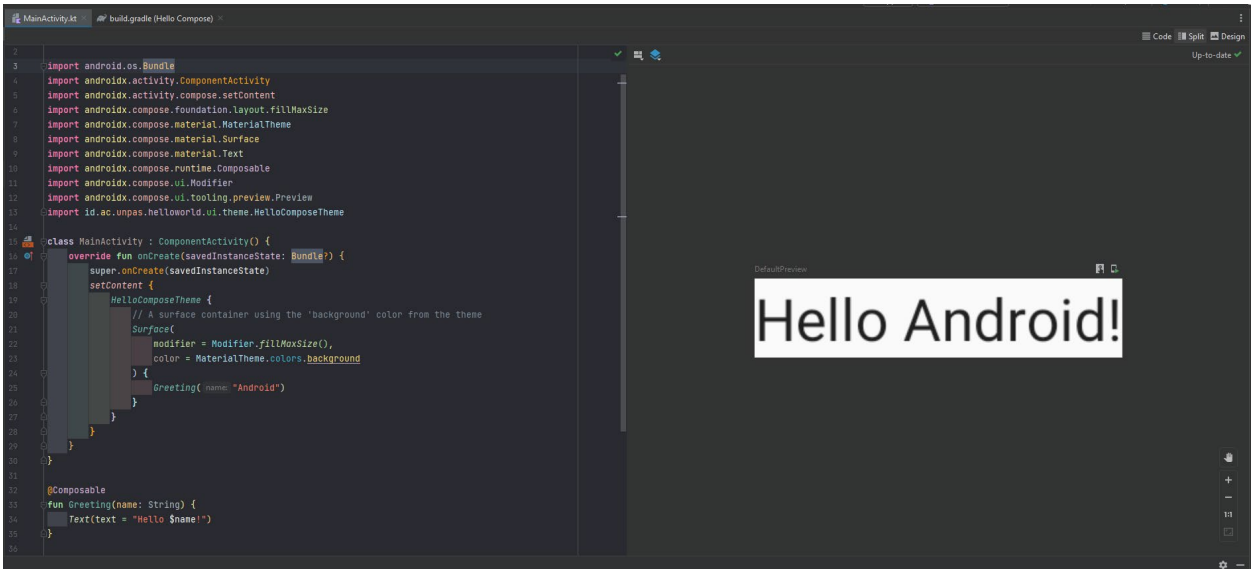
```
        super.onCreate(savedInstanceState)
        setContent {
            HelloComposeTheme {
                // A surface container using the 'background' color from the
theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}

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

@Preview(showBackground = true)
@Composable
fun DefaultPreview() {
    HelloComposeTheme {
        Greeting("Android")
    }
}
```

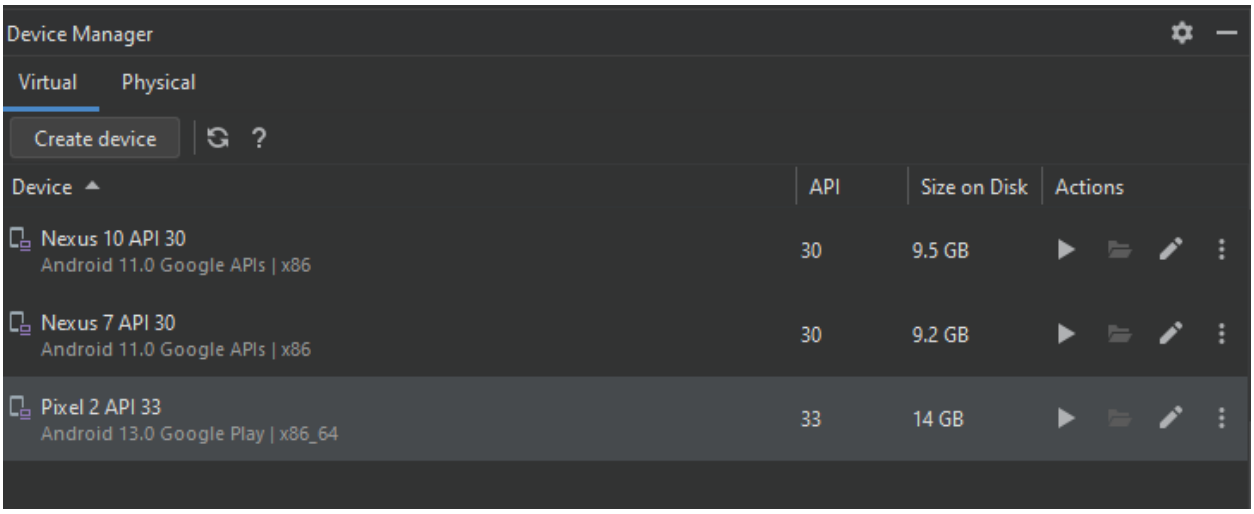
Hal-hal yang perlu diperhatikan diantaranya:

1. Fungsi Composable Greeting (ditandai dengan anotasi @Composable) adalah fungsi untuk membuat konten utama
2. Fungsi DefaultPreview digunakan untuk membuat preview yang dapat ditampilkan langsung di Android Studio
3. Fungsi onCreate adalah fungsi turunan dari kelas ComponentActivity yang akan dijalankan ketika kelas activity dibuat



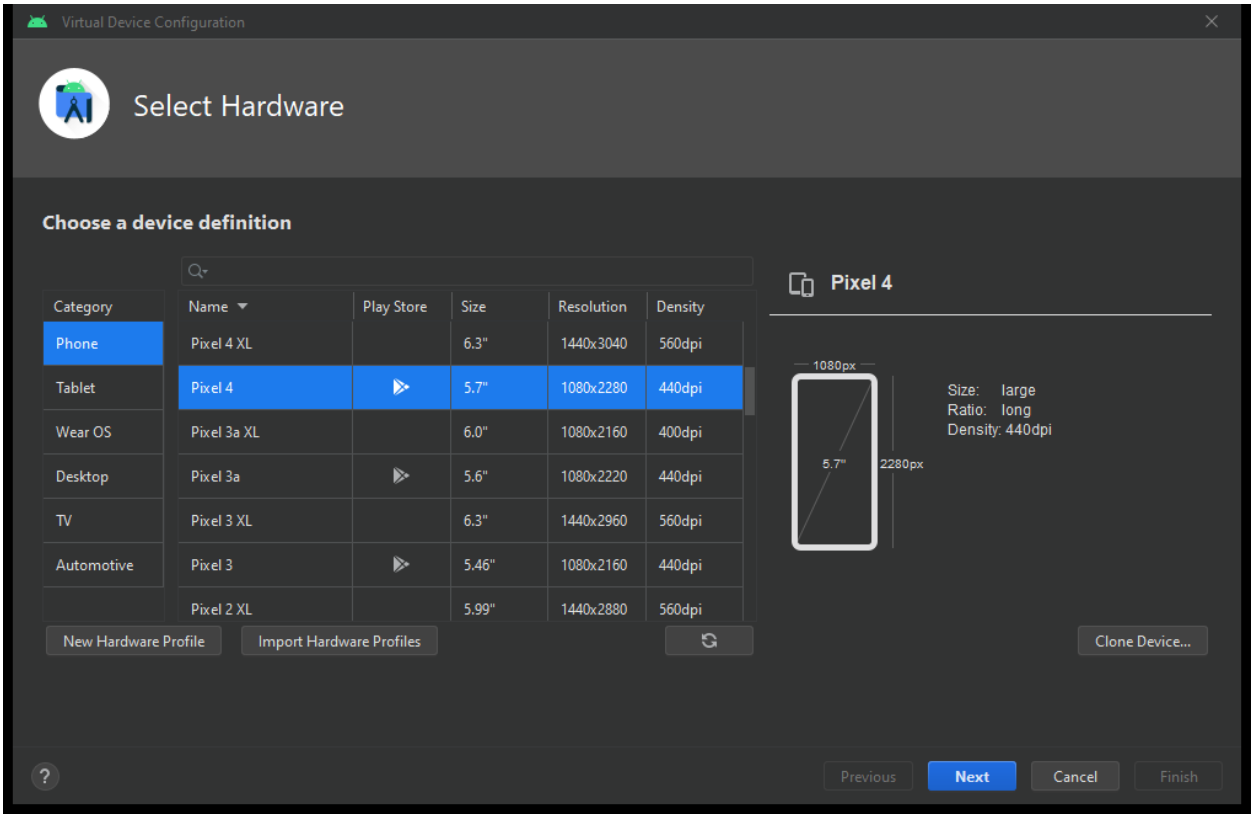
Gambar 10 Preview Tampilan Activity

Untuk menjalankan aplikasi, Anda perlu membuat Android Virtual Device (emulator) atau menghubungkan komputer anda dengan perangkat asli. Untuk membuat Android Virtual Device, bukalah menu Tools > Device Manager. Kotak Device Manager akan muncul di sebelah kanan layar seperti berikut.



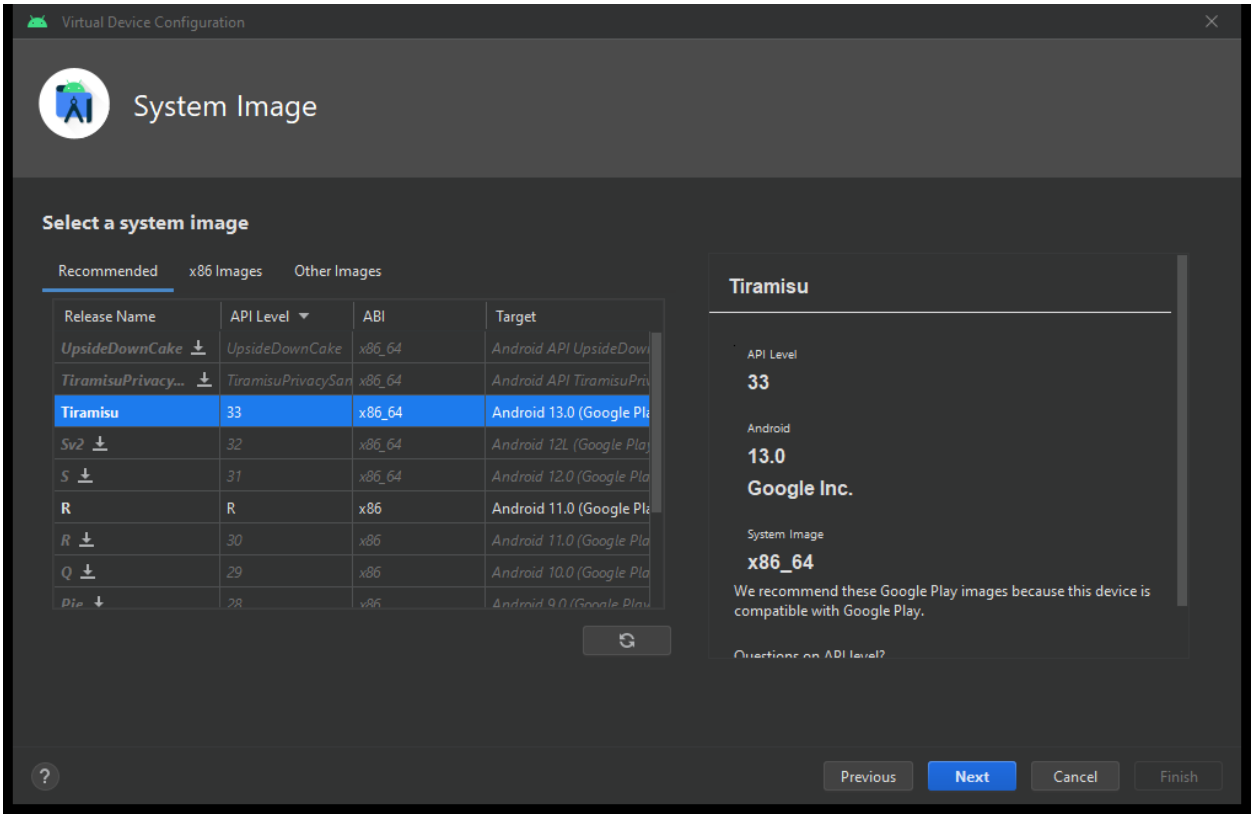
Gambar 11 Device Manager

Klik Create device untuk membuat Android Virtual Device baru.



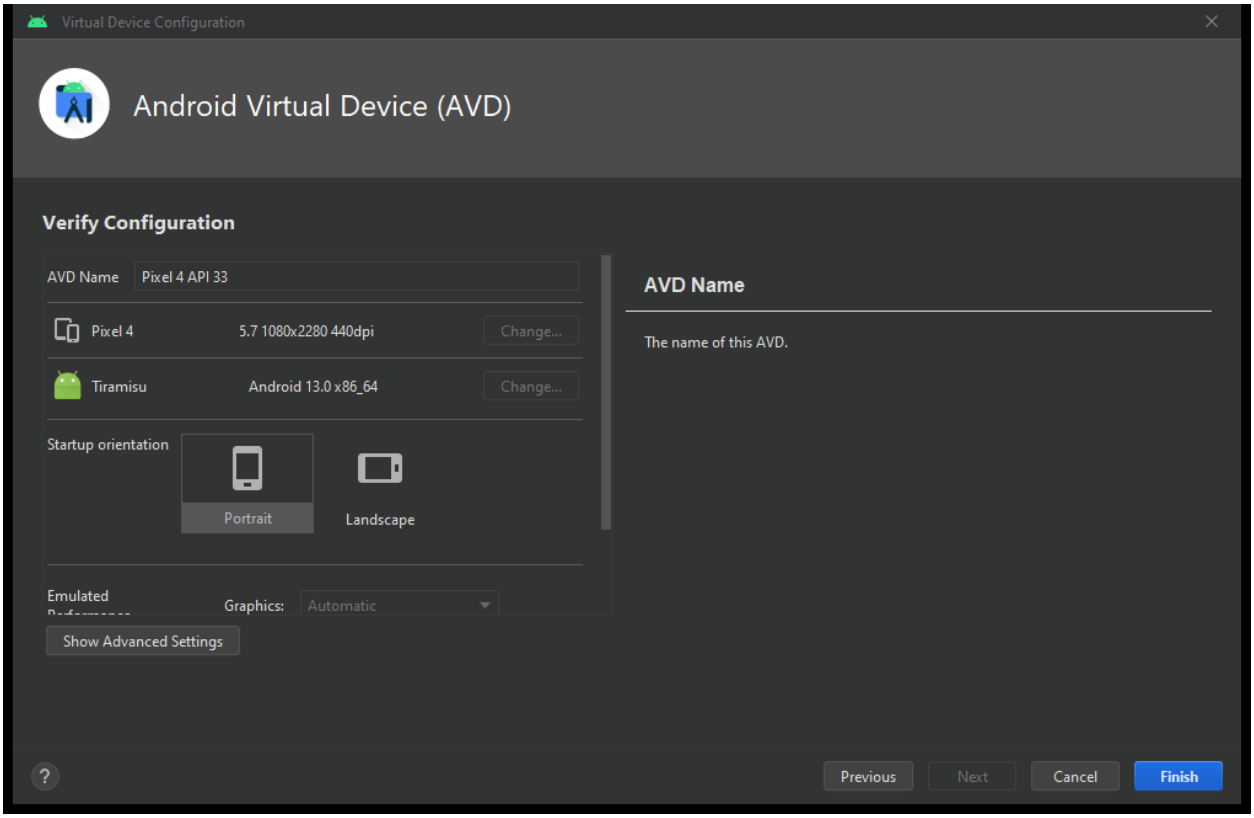
Gambar 12 Pilih Definisi Virtual Device

Pilih Phone pada kolom Category kemudian pilih profil yang cocok untuk project yang Anda buat (misal Pixel 4). Profil dengan ikon Play Store akan memiliki Google Play di dalamnya yang memungkinkan kita untuk menggunakan API dari Google Play. Selanjutnya, pilih versi sistem operasi yang cocok dengan proyek Anda, disarankan untuk memilih versi tertinggi terlebih dahulu, jika diperlukan Anda dapat membuat beberapa virtual device untuk beberapa versi operating sistem.



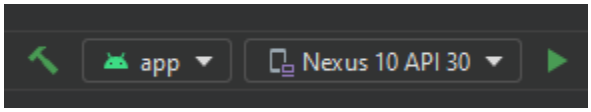
Gambar 13 Pilihan Versi OS

Kemudian, beri nama Android Virtual Device anda dan tekan Finish.



Gambar 14 Verifikasi Konfigurasi

Setelah Android Virtual Device selesai dibuat, anda akan dapat menggunakannya untuk menjalankan aplikasi, terlihat dari pilihan di sebelah kiri tombol Run seperti berikut.



Gambar 15 Tombol Run

Jalankan aplikasi Anda dengan menekan tombol Run tersebut, kemudian perhatikan hasilnya.

Gambar 16 Hasil Run Aplikasi

Lakukan eksplorasi fitur-fitur Android Studio dan kode kotlin yang telah anda ketahui dengan project tersebut.