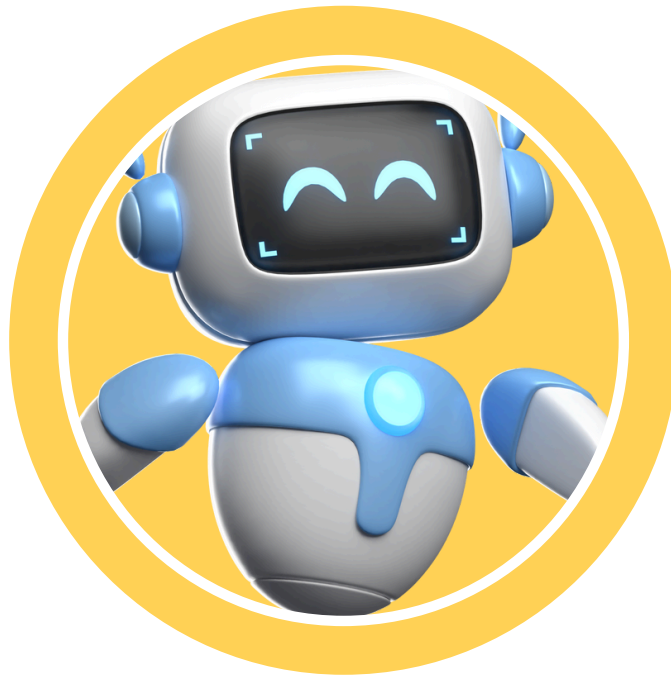




STMIK KAPUTAMA

PRAKTIKUM PENGANTAR PEMROGRAMAN BERGERAK



Disusun Oleh:

Kristina Annatasia Br Sitepu, M.Kom
Ratih Puspadini, ST., M.Kom

PERTEMUAN 2

MEMBUAT APLIKASI DAN MENJALANKANNYA

2.1 TUJUAN PEMBELAJARAN :

- a. Mahasiswa mampu membuat aplikasi sederhana dengan desain stkitard yang disediakan dan menjalankan aplikasi di emulator maupun di perangkat mobile (keluaran)

2.2 ALAT DAN BAHAN :

- a. Laptop/PC
- b. Android Studio
- c. Java Development Kit (JDK)
- d. Driver ADB
- e. Koneksi Internet Stabil

2.3 MATERI

A. Komponen UI (User Interface)

Komponen User Interface (UI) adalah elemen-elemen yang digunakan untuk membuat tampilan antarmuka pada aplikasi Android. Android menyediakan berbagai komponen UI untuk membantu pengembang dalam membangun pengalaman pengguna yang baik. Berikut adalah beberapa komponen utama dalam UI Android:

1) View

- TextView : Digunakan untuk menampilkan teks statis.
- ImageView : Untuk menampilkan gambar.
- Button : Untuk menangani interaksi pengguna seperti klik.
- EditText : Digunakan untuk input teks dari pengguna.
- CheckBox : Untuk membuat pilihan boolean (ya/tidak).
- RadioButton : Untuk membuat pilihan tunggal dari beberapa opsi.
- ProgressBar : Untuk menunjukkan status proses (loading).

2) Layout

- LinearLayout : Mengatur tampilan anak-anaknya dalam satu baris vertikal atau horizontal.
- RelativeLayout : Mengatur posisi komponen berdasarkan hubungan antar komponen.

ConstraintLayout: Memungkinkan pengaturan posisi lebih fleksibel dengan constraint (batasan).

FrameLayout : Umumnya digunakan untuk menumpuk elemen UI di atas satu sama lain.

3) RecyclerView

RecyclerView adalah komponen yang digunakan untuk menampilkan daftar atau koleksi data dalam jumlah besar dengan cara yang efisien. Ini adalah versi yang lebih fleksibel dan canggih dari ListView.

4) Toolbar

Toolbar adalah bar di bagian atas aplikasi yang biasanya berisi judul, tombol navigasi, dan opsi menu.

5) Floating Action Button (FAB)

FAB adalah tombol melayang yang umumnya digunakan untuk tindakan utama di layar.

B. Package, Project Structure, AndroidManifest.xml

1) Package

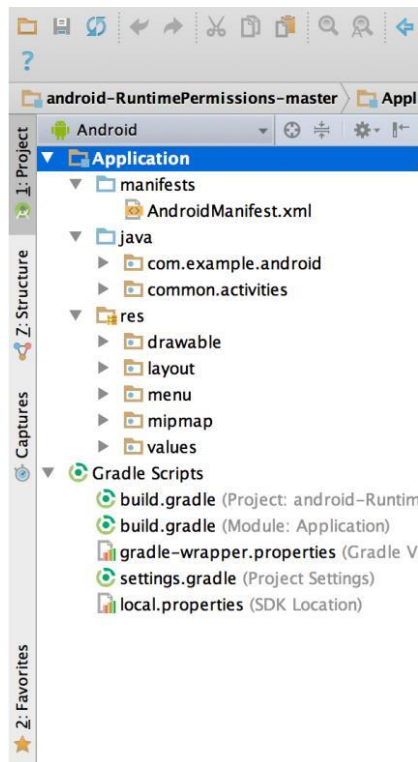
Package adalah nama unik yang mengidentifikasi aplikasi Android. Nama package sering mengikuti format domain terbalik (misalnya com.example.app). Package ini digunakan sebagai penanda unik aplikasi di Play Store dan perangkat Android.

2) Project Structure Android

Struktur proyek Android terdiri dari beberapa folder dan file penting:

- src/ : Berisi kode sumber aplikasi. Terutama ada dua sub-folder utama:
 - ✓ main/ : Menyimpan semua kode sumber, file XML (layout, resources), dan manifest aplikasi.
 - ✓ java/ : Menyimpan file-file kode dalam bahasa Java atau Kotlin.
- res/ : Folder ini menyimpan sumber daya (resource) aplikasi seperti gambar, layout, strings, dan lainnya.
 - ✓ layout/: Menyimpan file XML yang mengatur tata letak UI aplikasi.
 - ✓ drawable/: Menyimpan file gambar atau grafik.
 - ✓ values/: Menyimpan resource seperti string, warna, ukuran, dan gaya.

- AndroidManifest.xml: File konfigurasi utama yang mendeskripsikan aplikasi. Ini berisi informasi seperti :
 - ✓ Nama package.
 - ✓ Izin (permissions) yang dibutuhkan aplikasi.
 - ✓ Aktivitas utama dan aktivitas lain yang didefinisikan dalam aplikasi.
 - ✓ Versi minimum SDK dan target SDK.
 - ✓ Komponen utama seperti services, broadcast receivers, dan lainnya.



Gambar 1. File Project pada Tampilan / View Project Android.

3) AndroidManifest.xml

File ini adalah komponen penting dalam setiap aplikasi Android. AndroidManifest.xml menjelaskan setiap aspek penting aplikasi kepada sistem Android. Contoh isi AndroidManifest.xml:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.app">

  <!-- Izin aplikasi -->
  <uses-permission android:name="android.permission.INTERNET"/>
```

```

<!-- Informasi aplikasi -->
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportsRtl="true"
android:theme="@style/AppTheme">

<!-- Aktivitas utama -->
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
</application>

</manifest>

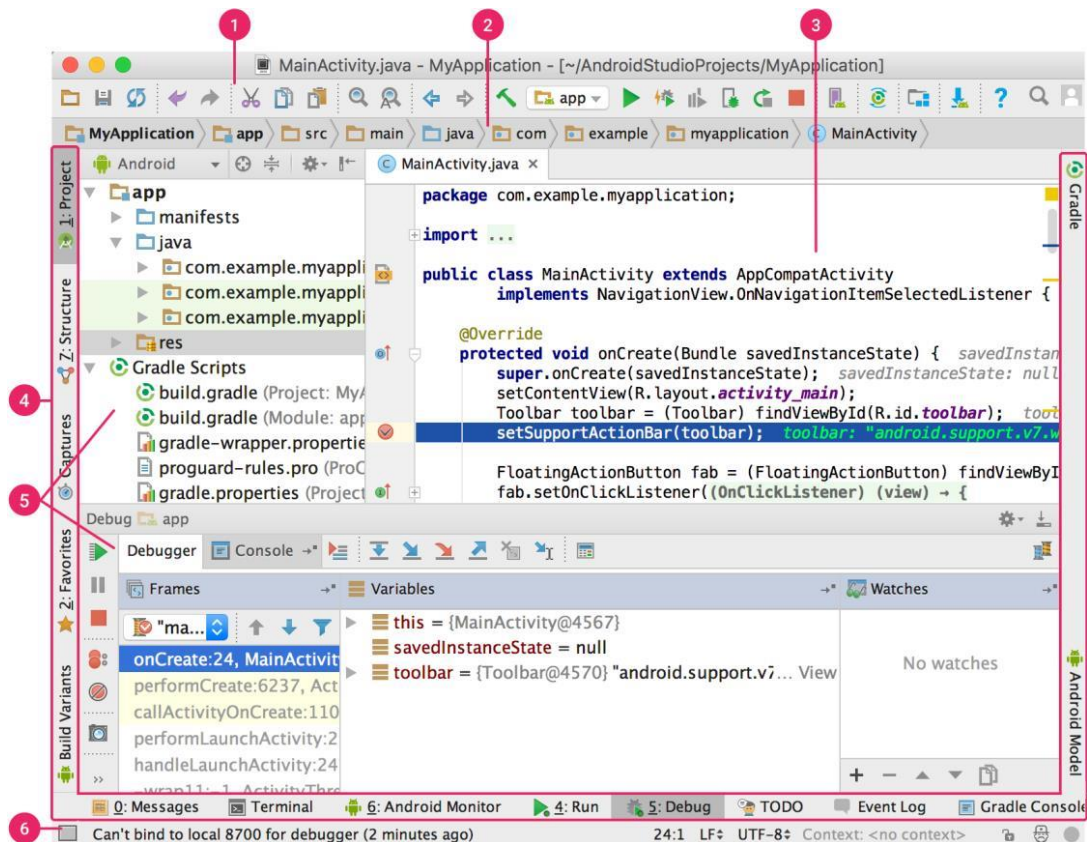
```

C. Antarmuka Pengguna Android Studio

Jendela Utama / Main Window Android Studio terdiri dari beberapa area yang ditampilkan di Gambar 2, antara lain :

- 1) Toolbar : Menjalankan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
- 2) Navigation bar : Membuka project Anda dan membuka file untuk diedit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela Project.
- 3) Editor window : 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) Tool window bar : Menggunakan tombol di bagian luar jendela IDE untuk meluaskan atau menciutkan setiap jendela alat.
- 5) Tool windows : Mengakses tugas tertentu seperti pengelolaan project, penelusuran, control 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.



Gambar 2. Jendela Utama / Main Window Android Studio.


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.

D. Tool Windows Android Studio

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

Buka Tool Windows dengan cara, sebagai berikut :

- 1) 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.

- 2) 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.
- 3) Untuk menampilkan atau menyembunyikan seluruh kolom jendela alat, klik ikon jendela  di pojok kiri bawah jendela Android Studio.
- 4) Untuk menemukan jendela alat tertentu, arahkan kursor ke atas ikon jendela dan pilih jendela alat tersebut dari menu.

Anda juga bisa menggunakan pintasan keyboard / Keyboard Shortcuts untuk membuka jendela alat / Tool Windows. Tabel 1 - mencantumkan pintasan untuk jendela alat yang paling umum.

Tabel 1. Keyboard Shortcuts untuk Tool Windows

Tool Window	Windows and Linux	macOS
Project	Alt+1	Command+1
Version Control	Alt+9	Command+9
Run	Shift+F10	Control+R
Debug	Shift+F9	Control+D
Logcat	Alt+6	Command+6
Return to Editor	Esc	Esc
Hide All Tool Windows	Control+Shift+F12	Command+Shift+F12

E. Sistem Build Gradle Android Studio

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 , antara lain untuk:

- 1) Menyesuaikan, mengonfigurasi, dan memperluas proses build.
- 2) Membuat banyak APK untuk aplikasi Anda dengan berbagai fitur yang menggunakan project dan modul yang sama.
- 3) 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.

F. Varian Build Android Studio

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.

G. Konsep Activity

Activity adalah komponen dasar aplikasi Android yang mewakili satu layar dengan antarmuka pengguna. Setiap kali pengguna berinteraksi dengan aplikasi, mereka berinteraksi dengan satu atau lebih aktivitas. Setiap aplikasi Android minimal memiliki satu aktivitas, yang biasanya merupakan entry point atau layar pertama yang dilihat oleh pengguna.

1. Cara Kerja Activity

Activity terdiri dari dua komponen utama:

- Kode Java/Kotlin : Mengatur logika dan interaksi di balik layar.
- Layout XML : Menentukan tampilan UI dari aktivitas.

Contoh struktur dasar Activity:

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


Penjelasan:

- **public class MainActivity extends AppCompatActivity:** Di Java, deklarasi kelas MainActivity mewarisi (extends) dari AppCompatActivity.
- **@Override:** Anotasi ini menandakan bahwa kita sedang meng-override metode dari superclass (dalam hal ini, onCreate dari AppCompatActivity).
- **onCreate(Bundle savedInstanceState):** Metode ini dipanggil ketika aktivitas pertama kali dibuat.
- **setContentView(R.layout.activity_main):** Ini digunakan untuk mengatur layout tampilan dari file XML activity_main.

2. Jenis-Jenis Activity

- Single Activity : Aplikasi yang hanya memiliki satu aktivitas.
- Multiple Activity : Aplikasi yang terdiri dari banyak aktivitas, di mana pengguna dapat berpindah antar aktivitas dengan intent.

3. Intent dalam Activity

Intent digunakan untuk berkomunikasi antar komponen aplikasi. Intent dapat digunakan untuk memulai aktivitas lain, mengirim data, atau memulai layanan latar belakang.

Contoh :

```
Intent intent = new Intent(MainActivity.this, SecondActivity.class);
startActivity(intent);
```

Penjelasan:

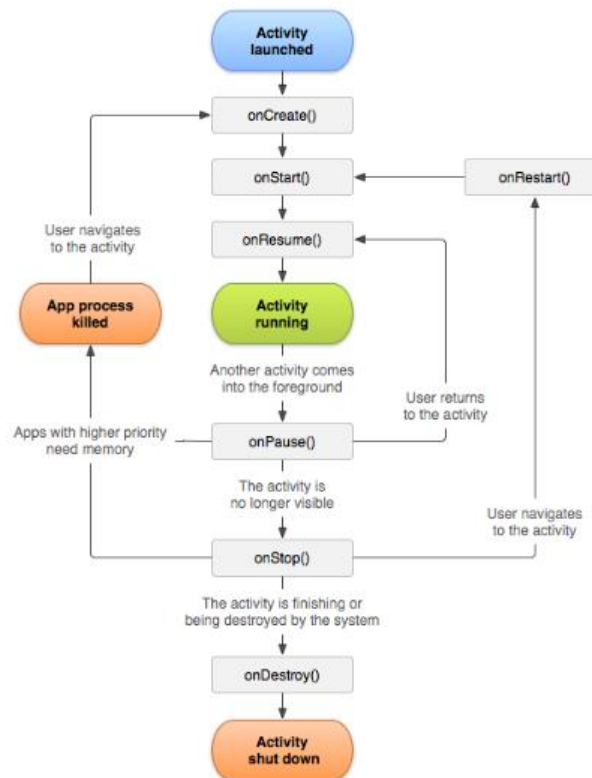
- **Intent intent = new Intent(MainActivity.this, SecondActivity.class); :** Membuat objek Intent baru di Java untuk memulai SecondActivity.
 - MainActivity.this digunakan untuk merujuk ke konteks MainActivity.
 - SecondActivity.class adalah referensi ke kelas SecondActivity yang ingin kita mulai.
- **startActivity(intent); :** Memulai SecondActivity dengan menggunakan intent yang telah dibuat.

H. Siklus Hidup Activity

Siklus hidup (lifecycle) dari sebuah activity mengacu pada bagaimana Android mengelola status dan keberadaan aktivitas selama interaksi dengan pengguna. Activity dapat berada dalam berbagai kondisi (states) tergantung pada interaksi pengguna dan sistem Android. Berikut adalah beberapa metode siklus hidup utama dalam sebuah aktivitas :

1. **onCreate()**
 - Dipanggil saat aktivitas pertama kali dibuat. Ini adalah tempat di mana Kita melakukan semua inisialisasi seperti membuat layout UI dan mempersiapkan data yang dibutuhkan.
2. **onStart()**
 - Dipanggil saat aktivitas akan menjadi terlihat oleh pengguna. Aktivitas mulai masuk ke "forefront".
3. **onResume()**
 - Dipanggil saat aktivitas mulai berinteraksi dengan pengguna. Ini adalah kondisi di mana aktivitas berada di layar dan pengguna mulai berinteraksi langsung dengan aplikasi.
4. **onPause()**
 - Dipanggil saat sebagian atau seluruh aktivitas menjadi tidak terlihat (misalnya, pengguna membuka aplikasi lain). Di sini, Kita sebaiknya menyimpan data sementara dan menghentikan aktivitas yang berat.
5. **onStop()**
 - Dipanggil saat aktivitas tidak lagi terlihat oleh pengguna. Di sini, Kita bisa melepaskan resource yang tidak digunakan.
6. **onDestroy()**
 - Dipanggil saat aktivitas dihapus dari memori, baik karena aplikasi ditutup atau sistem memori membutuhkan ruang.
7. **onRestart()**
 - Dipanggil saat aktivitas berhenti, lalu dimulai kembali (misalnya saat pengguna membuka kembali aplikasi dari background).

Diagram Siklus Hidup Activity:

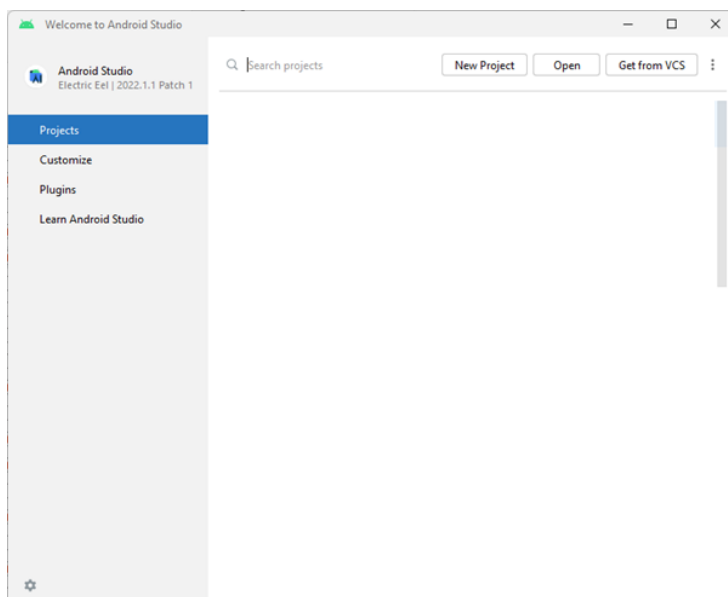


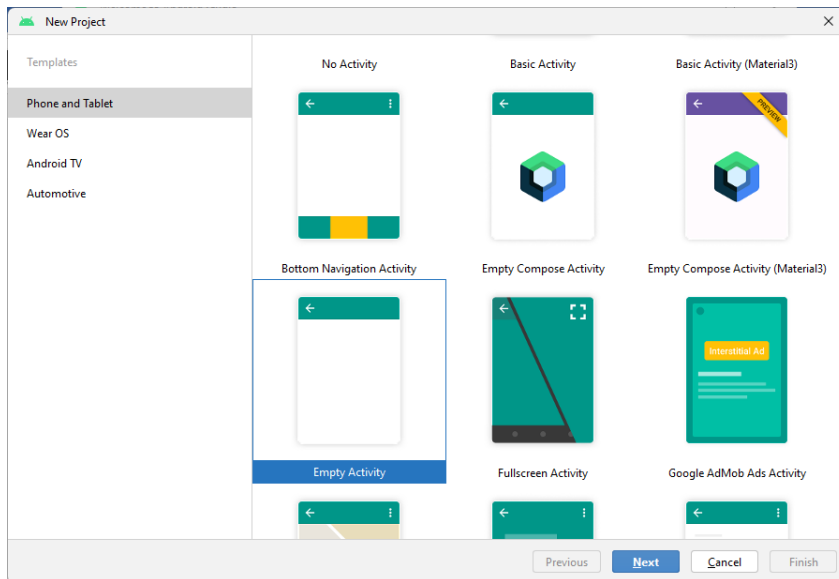
Gambar 3. Siklus hidup Activity

Ini menggambarkan bagaimana sebuah aktivitas berpindah dari satu state ke state lainnya selama siklus hidupnya.

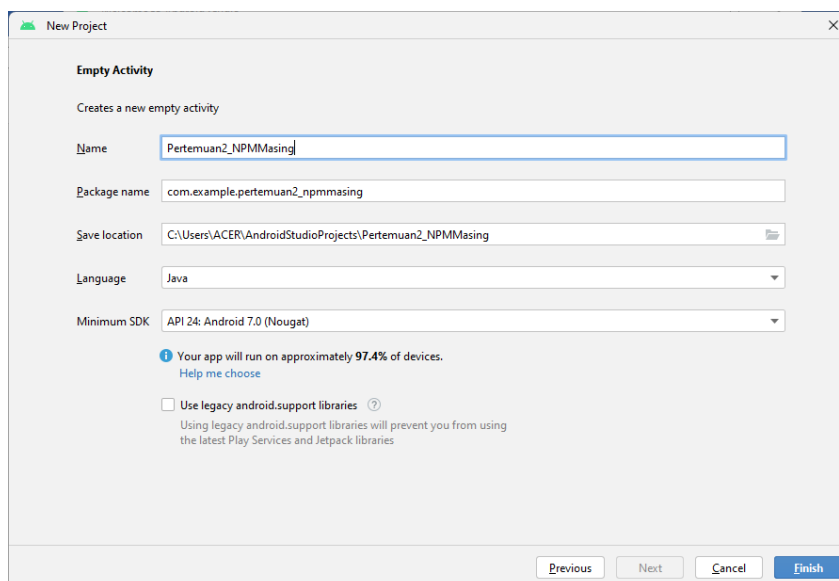
2.4 PRAKTIKUM

1. Buka aplikasi android studio, klik New Project → Empty Activity → Next

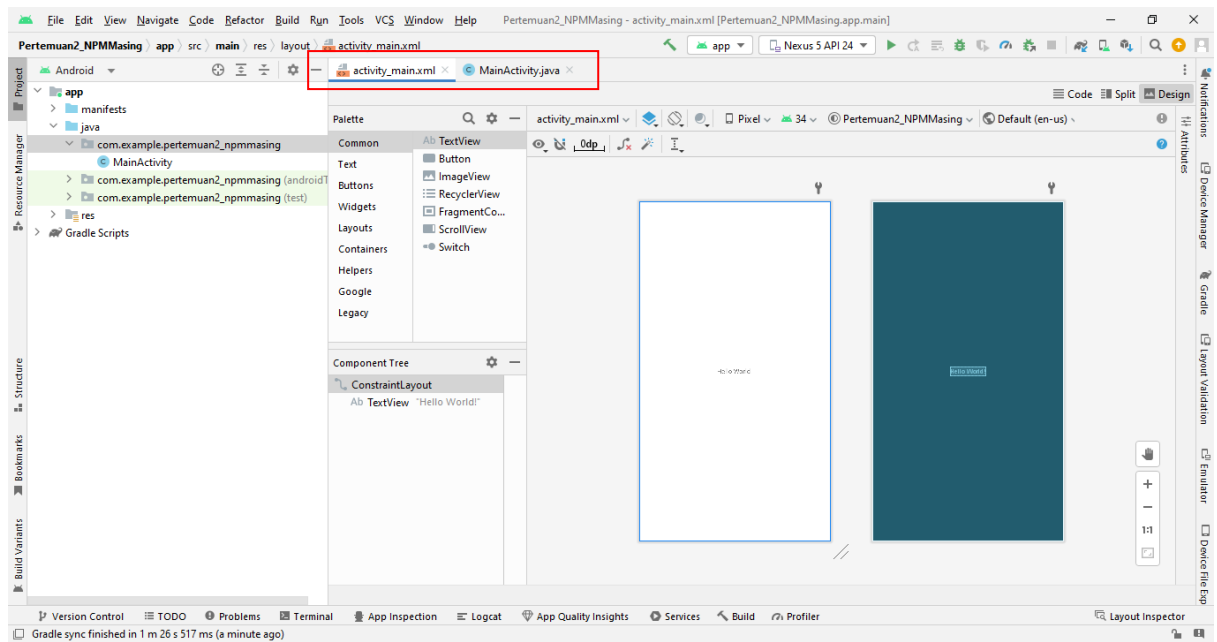




2. Masukkan nama project (nama project yaitu Pertemuan2_NPMmasing-masing), maka package name akan ikut menyesuaikan. (Catatan: jangan mengubah nama di Package name). Jika ingin menempatkan project di folder lainnya, ubah lokasi Save Locationnya. Pilih language Java. Pilih versi Android yang akan didukung aplikasi di kolom minimum SDK. Klik Finish.



3. Kemudian akan muncul lembar kerja pada android studio sebagai berikut :



Keterangan :

MainActivity.java : Kode logika aplikasi.

activity_main.xml : Layout XML untuk antarmuka pengguna.

4. Pada bagian MainActivity.java, ubah kode program menjadi :

```

package com.example.pertemuan2_npmmasing;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    // Deklarasi variabel untuk menampilkan nama dan NPM
    TextView textNama, textNPM;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Inisialisasi variabel TextView
        textNama = findViewById(R.id.textNama);
        textNPM = findViewById(R.id.textNPM);

        // Set nama dan NPM mahasiswa
        String namaMahasiswa = "Nama: Kristina Annatasia Br Sitepu";
        String npmMahasiswa = "NPM: 1234567890";
    }
}

```

```

// Menampilkan data ke TextView
textNama.setText(namaMahasiswa);
textNPM.setText(npmMahasiswa);
}
}

```

5. Ketikkan kode berikut pada bagian activity_main.xml

```

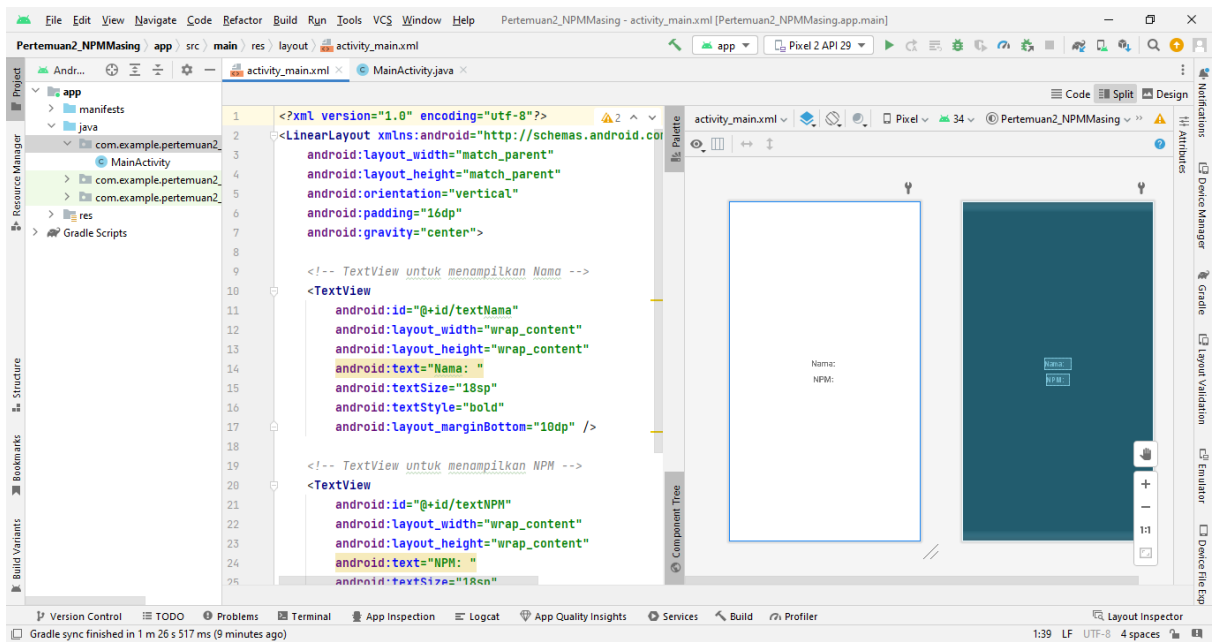
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="16dp"
android:gravity="center">

<!-- TextView untuk menampilkan Nama -->
<TextView
android:id="@+id/textNama"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Nama: "
android:textSize="18sp"
android:textStyle="bold"
android:layout_marginBottom="10dp" />

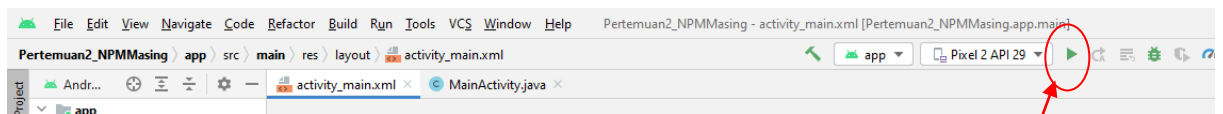
<!-- TextView untuk menampilkan NPM -->
<TextView
android:id="@+id/textNPM"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="NPM: "
android:textSize="18sp"
android:textStyle="bold" />

</LinearLayout>

```



6. Untuk menjalankan aplikasi, dapat dilakukan dengan dua cara, pertama pada emulator, kedua pada perangkat mobile.
 - a. Untuk menjalankan Virtual Device, klik pada icon play.



Tunggu beberapa saat maka akan muncul output sebagai berikut :



b. Selain menggunakan virtual device dalam melakukan debug, kita dapat menggunakan physical device.

Bila kita hendak melakukan run atau debugging, lebih baik kita menjalankannya pada peranti smartphone asli. Running dengan menggunakan peranti memiliki beberapa kelebihan jika dibandingkan dengan emulator yaitu :

- Lebih cepat;
- Fitur seperti geo-location, push notif bisa digunakan;
- Bisa mengetahui daya serap baterai terhadap aplikasi;
- Lebih mudah.

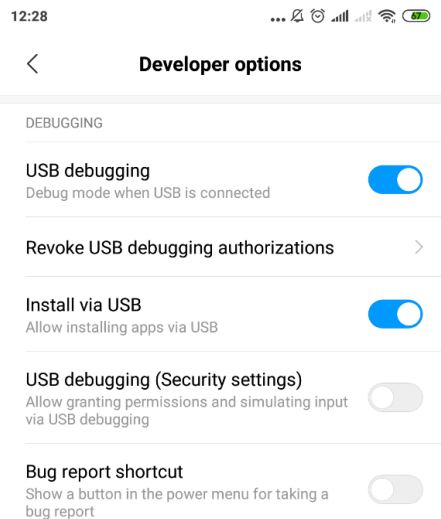
Dengan menggunakan peranti smartphone asli, kita dapat memastikan bahwa aplikasi kita berjalan dengan wajar ketika sudah sampai di tangan pengguna. Kendala dari pendekatan ini adalah beragamnya model peranti yang ada di pasaran.

Tampilan dari langkah berikut bisa dipastikan akan berbeda dengan peranti yang dipakai. Akan tetapi secara garis besar langkahnya akan sama. Adapun Langkah-langkah tersebut yaitu :

- 1) Pastikan peranti yang akan dipakai sesuai dengan target SDK atau paling tidak mendukung versi SDK terendah yang digunakan aplikasi.
- 2) Buka Pengaturan/**Setting** dan masuk ke dalam menu Tentang telepon/**About**. Pada halaman menu ini, kita perlu menemukan informasi tentang Nomor Build/**Build number**.

Berikut persiapan yang harus dilakukan :

- 1) Siapkan smartphone dan kabel data. Smartphone dengan OS Mobile Android, sedangkan kabel data adalah kabel yang dapat membaca data dari smartphone ke laptop (karena beberapa kasus terdapat kabel data yang hanya dapat digunakan untuk mengisi power atau baterai).
- 2) Penggunaan physical device dalam melakukan debug memerlukan beberapa pengaturan pada smartphone yaitu mengaktifkan developer option pada smartphone. Silakan mencari referensi untuk mengaktifkan developer option dari handphone yang anda pakai.
- 3) Kemudian aktifkan USB Debugging sampai pada bagian USB debugging terlihat aktif.



- 4) Pada saat akan menjalankan program pastikan device yang dipilih adalah perangkat smartphone kita.

2.5 TUGAS

1. Demokan project tersebut hingga muncul di emulator!
2. Tugas dikumpul dalam format pdf dengan nama file : NPM_Tugaske?
3. Didalam file tersebut dituliskan :
 - NPM
 - Nama Lengkap
 - Program Studi
 - Kelas