

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 1**



**ANDROID BASIC WITH KOTLIN**

**OLEH:**

**ZAHRA NABILA**

**NIM.2310817320007**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS LAMBUNG MANGKURAT  
APRIL 2025**

**LEMBAR PENGESAHAN**  
**LAPORAN PRAKTIKUM PEMROGRAMAN I**  
**MODUL 1**

Laporan Praktikum Pemrograman Mobile Modul 1: Android Basic with Kotlin ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Zahra Nabila  
NIM : 2310817320007

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar  
NIM. 2210817210026

Muti`a Maulida S.Kom M.T.I  
NIP. 19881027 201903 20 13

## DAFTAR ISI

LEMBAR PENGESAHAN .....	2
DAFTAR ISI .....	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL .....	5
<b>SOAL 1</b> .....	6
A. Source Code.....	8
B. Output Program .....	15
C. Pembahasan .....	15
D. Tautan Git.....	19

## DAFTAR GAMBAR

Gambar 1. Tampilan Awal Aplikasi .....	6
Gambar 2. Tampilan Dadu Setelah Di Roll .....	7
Gambar 3. Tampilan Roll Dadu Double .....	8
Gambar 4. Screenshot Hasil Jawaban Soal 1 .....	15
Gambar 5. Screenshot Hasil Jawaban Soal 1 .....	15
Gambar 6. Screenshot Hasil Jawaban Soal 1 .....	15

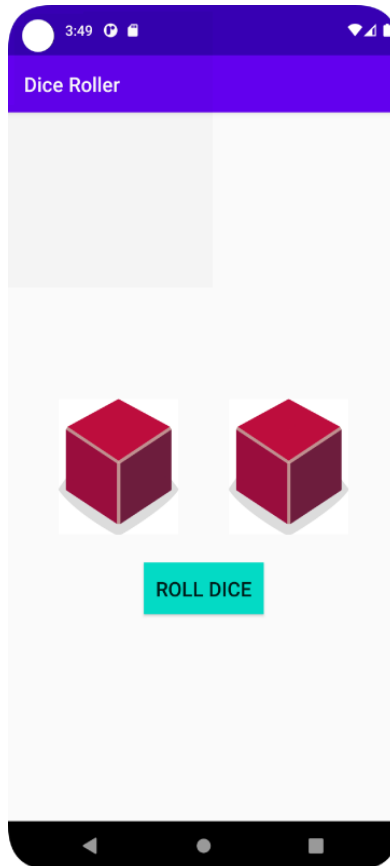
## DAFTAR TABEL

Table 1. Source Code Jawaban Soal 1 .....	11
Table 2. Source Code Jawaban Soal 1 .....	12
Table 3. Source Code Jawaban Soal 1 .....	14

## SOAL 1

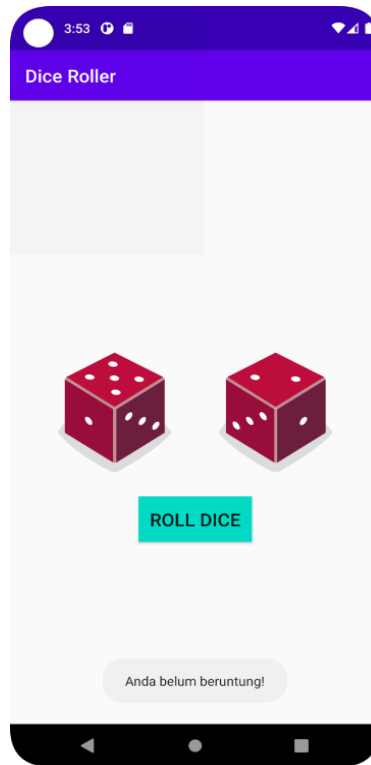
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

1. Tampilan awal aplikasi setelah dijalankan akan menampilkan 2 buah dadu kosong seperti dapat dilihat pada Gambar 1.



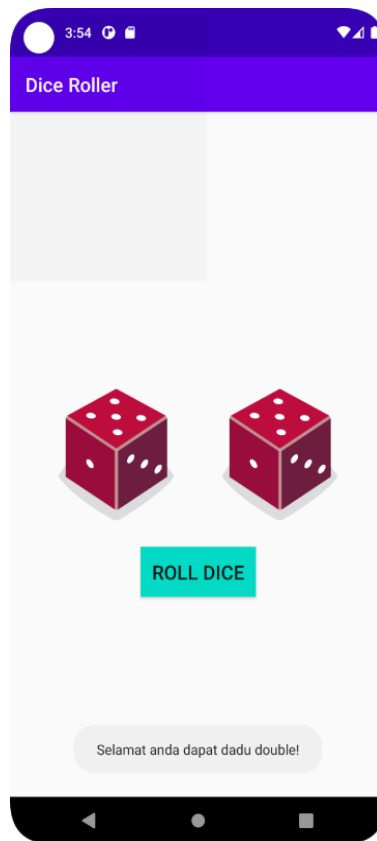
*Gambar 1. Tampilan Awal Aplikasi*

2. Setelah user menekan tombol “Roll Dice” maka masing-masing dadu akan memunculkan sisi dadu masing-masing dengan angka antara 1 s/d 6. Apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2 maka akan menampilkan pesan “Anda belum beruntung!” seperti dapat dilihat pada Gambar 2.



*Gambar 2. Tampilan Dadu Setelah Di Roll*

3. Apabila user mendapatkan nilai dadu yang sama antara Dadu 1 dan Dadu 2 atau nilai double, maka aplikasi akan menampilkan pesan “Selamat anda dapat dadu double!” seperti dapat dilihat pada Gambar 3.
4. Upload aplikasi yang telah anda buat kedalam repository github ke dalam **folder Module 2 dalam bentuk project**. Jangan lupa untuk melakukan **Clean Project** sebelum mengupload pekerjaan anda pada repo.
5. Untuk gambar dadu dapat didownload pada link berikut:  
[https://drive.google.com/u/0/uc?id=147HT2lIH5qin3z5ta7H9y2N\\_5OMW81Ll&export=download](https://drive.google.com/u/0/uc?id=147HT2lIH5qin3z5ta7H9y2N_5OMW81Ll&export=download)



Gambar 3. Tampilan Roll Dadu Double

## A. Source Code

### File MainActivity.kt

```

1  package com.example.dice_roll
2
3  import android.os.Bundle
4  import android.widget.Button
5  import android.widget.ImageView
6  import android.widget.Toast
7  import androidx.activity.enableEdgeToEdge
8  import androidx.appcompat.app.AppCompatActivity
9  import androidx.core.view.ViewCompat
10 import androidx.core.view.WindowInsetsCompat
11
12 class MainActivity : AppCompatActivity() {
13     override fun onCreate(savedInstanceState: Bundle?) {

```



```

14         super.onCreate(savedInstanceState)
15         enableEdgeToEdge()
16         setContentView(R.layout.activity_main)
17
18         val rollBtn: Button = findViewById(R.id.button)
19         val diceImgFirst: ImageView =
20             findViewById(R.id.dice1)
21         val diceImgSecond: ImageView =
22             findViewById(R.id.dice2)
23         diceImgFirst.setImageResource(R.drawable.empty_dice)
24         diceImgSecond.setImageResource(R.drawable.empty_dice)
25
26         rollBtn.setOnClickListener {
27             rollDice()
28         }
29
30         class Dice(private val jumlahSisi: Int) {
31             fun roll(): Int {
32                 return (1..jumlahSisi).random()
33             }
34         }
35
36         private fun rollDice() {
37             val dice1 = Dice(6)
38             val dice2 = Dice(6)
39             val diceRoll1 = dice1.roll()
40             val diceRoll2 = dice2.roll()
41
42             val diceImgFirst: ImageView =

```

	<code>findViewById(R.id.dice1)</code>
43	<code>val diceImgSecond: ImageView =</code>
	<code>findViewById(R.id.dice2)</code>
44	
45	<code>val drawableResourceFirst = when (diceRoll1) {</code>
46	<code>1 -&gt; R.drawable.dice_1</code>
47	<code>2 -&gt; R.drawable.dice_2</code>
48	<code>3 -&gt; R.drawable.dice_3</code>
49	<code>4 -&gt; R.drawable.dice_4</code>
50	<code>5 -&gt; R.drawable.dice_5</code>
51	<code>else -&gt; R.drawable.dice_6</code>
52	<code>}</code>
53	
54	<code>val drawableResourceSecond = when (diceRoll2) {</code>
55	<code>1 -&gt; R.drawable.dice_1</code>
56	<code>2 -&gt; R.drawable.dice_2</code>
57	<code>3 -&gt; R.drawable.dice_3</code>
58	<code>4 -&gt; R.drawable.dice_4</code>
59	<code>5 -&gt; R.drawable.dice_5</code>
60	<code>else -&gt; R.drawable.dice_6</code>
61	<code>}</code>
62	
63	<code>diceImgFirst.setImageResource(drawableResourceFirst)</code>
64	<code>diceImgSecond.setImageResource(drawableResourceSecond)</code>
65	
66	<code>diceImgFirst.contentDescription =</code>
	<code>diceRoll1.toString()</code>
67	<code>diceImgSecond.contentDescription =</code>
	<code>diceRoll2.toString()</code>
68	
69	<code>if (diceRoll1 != diceRoll2) {</code>

70	Toast.makeText(this, "Anda belum beruntung", Toast.LENGTH_SHORT).show()
71	} else {
72	Toast.makeText(this, "Selamat anda dapat dadu double", Toast.LENGTH_SHORT).show()
73	}
74	}
75	}

Table 1. Source Code Jawaban Soal 1

File activity_main.xml	
1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3	xmlns:app="http://schemas.android.com/apk/res-auto"
4	xmlns:tools="http://schemas.android.com/tools"
5	android:id="@+id/main"
6	android:layout_width="match_parent"
7	android:layout_height="match_parent"
8	tools:context=".MainActivity">
9	
10	<TextView
11	android:layout_width="match_parent"
12	android:layout_height="wrap_content"
13	android:text="Dice Roller"
14	android:textSize="30sp"
15	app:layout_constraintEnd_toStartOf="parent"
16	app:layout_constraintTop_toTopOf="parent"
17	android:padding="10dp"
18	android:background="#B43757"/>
19	

20	<ImageView
21	android:id="@+id/dice1"
22	android:layout_width="150dp"
23	android:layout_height="150dp"
24	app:layout_constraintBottom_toBottomOf="parent"
25	app:layout_constraintTop_toTopOf="parent"
26	app:layout_constraintStart_toEndOf="@+id/dice2"
27	app:layout_constraintEnd_toEndOf="parent"
28	app:layout_constraintHorizontal_chainStyle="packed"
29	app:srcCompat="@drawable/empty_dice" />
30	
31	<ImageView
32	android:id="@+id/dice2"
33	android:layout_width="150dp"
34	android:layout_height="150dp"
35	app:layout_constraintBottom_toBottomOf="parent"
36	app:layout_constraintTop_toTopOf="parent"
37	app:layout_constraintStart_toStartOf="parent"
38	app:layout_constraintEnd_toStartOf="@+id/dice1"
39	app:srcCompat="@drawable/empty_dice" />
40	
41	<Button
42	android:id="@+id/button"
43	android:layout_width="wrap_content"
44	android:layout_height="wrap_content"
45	android:text="Roll Dice"
46	app:layout_constraintEnd_toEndOf="parent"
47	app:layout_constraintStart_toStartOf="parent"
48	app:layout_constraintTop_toBottomOf="@+id/dice1"
49	/>
50	</androidx.constraintlayout.widget.ConstraintLayout>

Table 2. Source Code Jawaban Soal 1

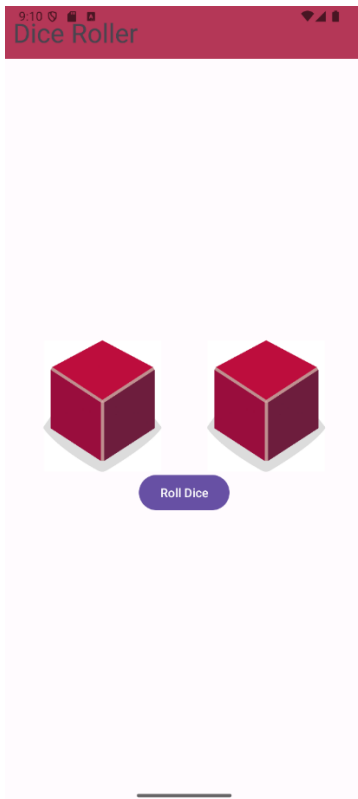
**File build.gradle.kts**

```
1  plugins {
2      alias(libs.plugins.android.application)
3      alias(libs.plugins.kotlin.android)
4  }
5
6  android {
7      namespace = "com.example.dice_roll"
8      compileSdk = 35
9
10     defaultConfig {
11         applicationId = "com.example.dice_roll"
12         minSdk = 24
13         targetSdk = 35
14         versionCode = 1
15         versionName = "1.0"
16
17         testInstrumentationRunner =
18             "androidx.test.runner.AndroidJUnitRunner"
19     }
20     buildFeatures{
21         viewBinding = true
22     }
23
24     buildTypes {
25         release {
26             isMinifyEnabled = false
27             proguardFiles(
28                 getDefaultProguardFile("proguard-android-
```

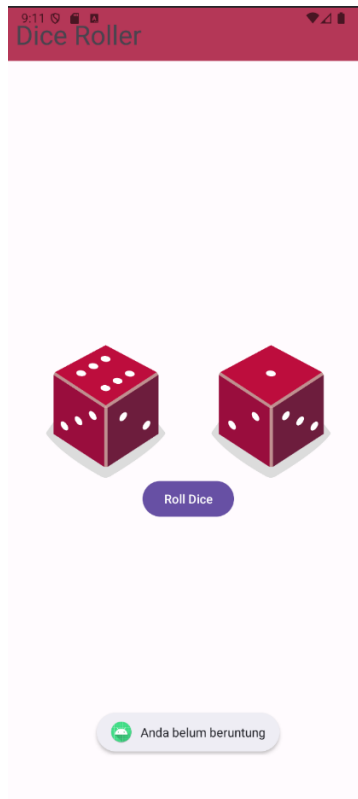
	<code>optimize.txt"),</code>
29	<code>        "proguard-rules.pro"</code>
30	<code>    )</code>
31	<code>    }</code>
32	<code>}</code>
33	<code>compileOptions {</code>
34	<code>    sourceCompatibility = JavaVersion.VERSION_11</code>
35	<code>    targetCompatibility = JavaVersion.VERSION_11</code>
36	<code>}</code>
37	<code>kotlinOptions {</code>
38	<code>    jvmTarget = "11"</code>
39	<code>}</code>
40	<code>}</code>
41	
42	<code>dependencies {</code>
43	
44	<code>    implementation(libs.androidx.core.ktx)</code>
45	<code>    implementation(libs.androidx.appcompat)</code>
46	<code>    implementation(libs.material)</code>
47	<code>    implementation(libs.androidx.activity)</code>
48	<code>    implementation(libs.androidx.constraintlayout)</code>
49	<code>    testImplementation(libs.junit)</code>
50	<code>    androidTestImplementation(libs.androidx.junit)</code>
51	<code>    androidTestImplementation(libs.androidx.espresso.core)</code>
52	<code>}</code>

*Table 3. Source Code Jawaban Soal 1*

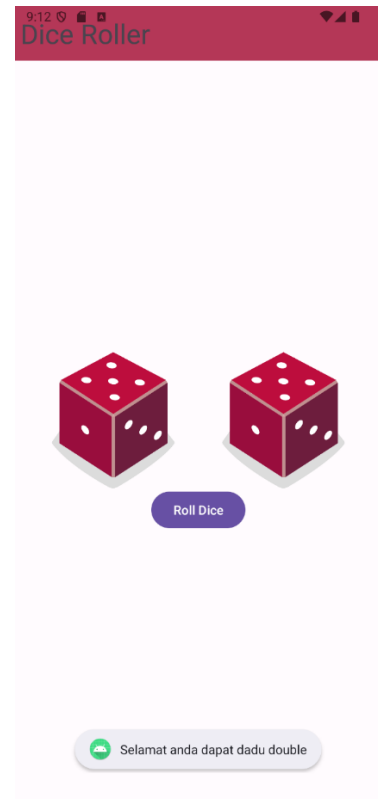
## B. Output Program



Gambar 4. Screenshot Hasil Jawaban Soal 1



Gambar 5. Screenshot Hasil Jawaban Soal 1



Gambar 6. Screenshot Hasil Jawaban Soal 1

## C. Pembahasan

### File MainActivity.kt

Kelas utama bernama **MainActivity** yang mewarisi dari **AppCompatActivity**. Di dalam fungsi **onCreate()**, sistem Android akan menyiapkan tampilan dengan memuat layout dari file **activity\_main.xml** melalui **setContentView()**. Selain itu, fungsi **enableEdgeToEdge()** digunakan untuk membuat tampilan aplikasi menyesuaikan layar penuh. Selanjutnya, program mengambil referensi dari elemen UI yang ada pada layout, yaitu tombol roll **R.id.button** serta dua gambar dadu **R.id.dice1** dan **R.id.dice2**. Kedua gambar dadu tersebut awalnya diatur untuk menampilkan gambar dadu kosong **R.drawable.empty\_dice**, sebagai tampilan awal sebelum tombol ditekan.

Tombol roll diatur agar memiliki sebuah listener, yaitu dengan menggunakan **rollBtn.setOnClickListener**. Listener ini adalah bagian dari program yang akan menunggu dan merespon saat tombol ditekan oleh pengguna. Ketika tombol tersebut ditekan, listener akan memicu fungsi **rollDice()**. Di dalam fungsi **rollDice()**, pertama-tama dibuat dua objek dari kelas **Dice** yang masing-masing mewakili sebuah dadu dengan enam sisi. Kelas **Dice** sendiri adalah kelas khusus yang dideklarasikan di dalam **MainActivity** dan memiliki fungsi **roll()** untuk menghasilkan angka acak dari 1 sampai 6. Nilai acak dari masing-masing dadu disimpan dalam variabel **diceRoll1** dan **diceRoll2**.

Setelah angka acak didapatkan, langkah berikutnya adalah menyesuaikan gambar dadu yang akan ditampilkan di layar. Hal ini dilakukan dengan struktur kontrol **when**, yang berfungsi mirip seperti **if-else** tetapi lebih ringkas. Jika nilai acak adalah 1, maka akan digunakan gambar **R.drawable.dice\_1**, jika nilainya 2 maka **dice\_2**, dan seterusnya hingga nilai 6. Gambar ini kemudian ditampilkan dengan **setImageResource()** pada kedua **ImageView**, **diceImgFirst** dan **diceImgSecond**. Tak hanya itu, program juga mengatur **contentDescription** dari masing-masing gambar sesuai dengan angka dadu yang muncul.

Setelah kedua dadu ditampilkan, aplikasi akan memberikan umpan balik kepada pengguna dalam bentuk notifikasi kecil bernama **Toast**. Jika angka dari kedua dadu berbeda, maka aplikasi akan menampilkan pesan “**Anda belum beruntung**”. Namun, jika kedua dadu menunjukkan angka yang sama, maka akan muncul pesan “**Selamat anda dapat dadu double**”.

File activity_main.xml
file layout untuk tampilan utama aplikasi <i>Dice Roller</i> yang dibuat menggunakan <b>ConstraintLayout</b> sebagai <i>root layout</i> . <b>ConstraintLayout</b> adalah jenis layout fleksibel di Android yang memungkinkan pengaturan posisi elemen UI berdasarkan hubungan ( <i>constraint</i> ) antar elemen, baik terhadap elemen lain maupun terhadap parent-nya. File ini menentukan bagaimana komponen visual seperti teks, gambar dadu, dan tombol akan ditampilkan dan diposisikan pada layar pengguna.



Di bagian atas layout terdapat komponen **TextView** yang berfungsi sebagai judul aplikasi. **TextView** ini memiliki **match\_parent** (lebar penuh), **wrap\_content** (tinggi sesuai konten), dan teks bertuliskan **"Dice Roller"** dengan ukuran huruf **30sp**. Komponen ini diberi padding sebesar **10dp** agar teks tidak terlalu mepet ke pinggir, serta latar belakang berwarna merah muda gelap (**#B43757**). Menggunakan **app:layout\_constraintEnd\_toEndOf="parent"** agar teks bisa benar-benar terpusat secara horizontal.

Berikutnya terdapat dua komponen **ImageView** yang masing-masing digunakan untuk menampilkan gambar dadu. **ImageView** dengan ID **dice1** dan **dice2** masing-masing diberi ukuran tetap sebesar **150dp x 150dp**. Keduanya diatur agar berada di tengah layar secara vertikal dengan menggunakan **app:layout\_constraintTop\_toTopOf="parent"** dan **Bottom\_toBottomOf="parent"**. Posisi horizontal kedua gambar dadu ini diatur saling berdampingan menggunakan constraint: **dice2** berada di kiri **Start\_toStartOf="parent"**, **End\_toStartOf="@+id/dice1"**, sedangkan **dice1** di sebelah kanannya **Start\_toEndOf="@+id/dice2"**, **End\_toEndOf="parent"**. Mereka juga berada dalam sebuah *horizontal chain* dengan gaya **app:layout\_constraintHorizontal\_chainStyle="packed"**, yang membuat keduanya saling rapat dan terpusat di tengah layar.

Di bawah kedua gambar dadu, terdapat komponen **Button** dengan ID **button**. Tombol ini memiliki ukuran lebar dan tinggi yang mengikuti isi teks **wrap\_content** dan berfungsi sebagai pemicu untuk melempar dadu. Tombol diberi teks **"Roll Dice"** dan diposisikan di bawah **dice1** menggunakan constraint **app:layout\_constraintTop\_toBottomOf="@+id/dice1"**, serta disejajarkan secara horizontal di tengah layar dengan menggunakan **Start\_toStartOf="parent"** dan **End\_toEndOf="parent"**.

Kombinasi antara **ConstraintLayout**, **ImageView**, **TextView**, dan **Button** ini menyusun struktur visual dasar aplikasi pelempar dadu secara efisien dan responsif.

**File build.gradle.kts**

Pada file **build.gradle.kts** yang ditampilkan adalah berkas konfigurasi utama untuk proyek Android berbasis Kotlin menggunakan Gradle dengan DSL (*Domain Specific Language*) Kotlin. File ini digunakan untuk mendefinisikan bagaimana proyek akan dibangun, termasuk plugin yang digunakan, versi SDK Android, fitur *build*, dan dependensi *library* yang dibutuhkan dalam pengembangan aplikasi.

Bagian pertama dari file ini adalah deklarasi plugin, di mana proyek ini menggunakan dua plugin utama: **android.application** dan **kotlin.android**. Plugin **android.application** digunakan untuk menandai bahwa proyek ini adalah sebuah aplikasi Android, bukan *library*. Sementara itu, plugin **kotlin.android** menambahkan dukungan Kotlin ke dalam proyek Android, sehingga pengembangan dapat dilakukan menggunakan bahasa Kotlin. Penulisan **alias(libs.plugins...)** menunjukkan bahwa proyek ini menggunakan fitur *Version Catalog*, yang memungkinkan pengelolaan versi plugin dan dependensi secara terpusat melalui file **libs.versions.toml**.

Selanjutnya, blok **android** berisi konfigurasi utama proyek. Properti **namespace** menentukan *namespace* atau ruang nama dasar yang akan digunakan oleh aplikasi, umumnya mengikuti format nama paket. Kemudian **compileSdk** diatur ke **versi 35**, yang berarti aplikasi ini dikompilasi menggunakan **Android API level 35**, memberikan akses terhadap fitur-fitur terbaru dari Android versi tersebut. Di dalam **defaultConfig**, terdapat beberapa pengaturan penting seperti **applicationId** yang menjadi identitas unik aplikasi di perangkat dan di *Google Play Store*. **minSdk** diatur ke 24, artinya aplikasi ini hanya dapat diinstal di perangkat dengan **Android Nougat (7.0) ke atas**. **targetSdk** ditetapkan ke 35 agar aplikasi ini dioptimalkan untuk versi Android terbaru. Selain itu, **versionCode** dan **versionName** digunakan untuk pengelolaan versi aplikasi, di mana **versionCode** bersifat **numerik** dan **dibutuhkan saat proses update di Google Play**, sedangkan **versionName** bersifat **deskriptif** dan **ditampilkan kepada pengguna**. **testInstrumentationRunner** menentukan runner untuk pengujian instrumentasi Android.

Pada bagian **buildFeatures**, fitur **viewBinding** diaktifkan. **ViewBinding** adalah fitur yang memungkinkan pengembang untuk lebih mudah mengakses elemen

antarmuka pengguna (UI) tanpa harus menggunakan **findViewById**, sehingga lebih aman dari kesalahan *runtime* dan meningkatkan produktivitas. Kemudian di blok **buildTypes**, terdapat konfigurasi untuk tipe build **release**, yang biasanya digunakan saat aplikasi akan dipublikasikan. Di sini, **isMinifyEnabled** diatur ke **false**, artinya proses minifikasi kode tidak diaktifkan, yang umum dilakukan pada tahap awal pengembangan atau untuk aplikasi kecil. File **proguardFiles** menunjuk ke konfigurasi *default ProGuard* dan file tambahan **proguard-rules.pro** untuk aturan khusus.

Bagian **compileOptions** dan **kotlinOptions** memastikan bahwa proyek menggunakan Java 11 untuk kompatibilitas sumber dan target *bytecode*, dan juga mengarahkan Kotlin untuk menggunakan JVM target versi 11.

Terakhir, bagian **dependencies** mencantumkan semua *library* yang digunakan dalam aplikasi. Di antaranya adalah **androidx.core.ktx**, **appcompat**, **material**, **activity**, dan **constraintlayout**, yang merupakan *library* penting untuk pengembangan antarmuka dan fungsi dasar aplikasi Android. Untuk pengujian, digunakan **junit** untuk unit *testing* standar, serta **androidx.junit** dan **espresso.core** untuk *instrumented testing* yang berjalan langsung di perangkat atau emulator. Semua dependensi ini juga ditulis menggunakan **libs.xxx**, menunjukkan bahwa proyek mengadopsi pendekatan *Version Catalog* yang lebih modern dan terstruktur dalam pengelolaan dependensi.

#### D. Tautan Git

<https://github.com/biilaaa/Pemrograman-Mobile>