

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Muhammad Aufa Fitrianda

NIM. 2310817210013

**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 : Muhammad Aufa Fitrianda
NIM : 2310817210013

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Raka Azwar
NIM. 2210817210012

Ir. Eka Setya Wijaya, S.T., M.Kom
NIP. 198205082008011010

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code.....	8
B. Output Program	12
C. Pembahasan	13
D. Tautan Git	16

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	11
---	----

DAFTAR TABEL

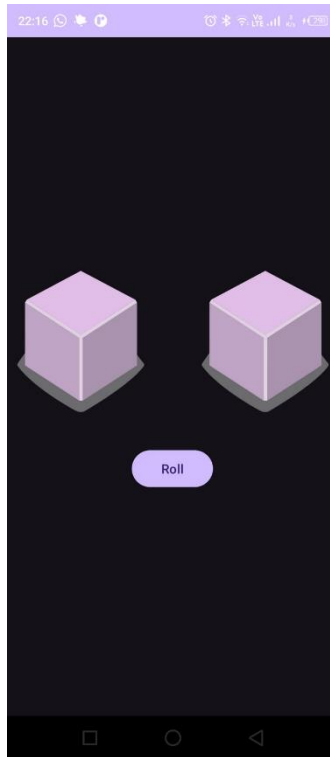
Tabel 1. Source Code Jawaban Soal 1	10
Tabel 2. Source Code Jawaban Soal 1	10

SOAL 1

Soal Praktikum:

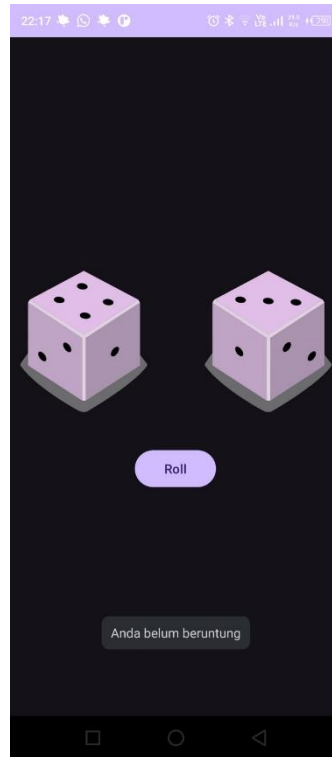
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



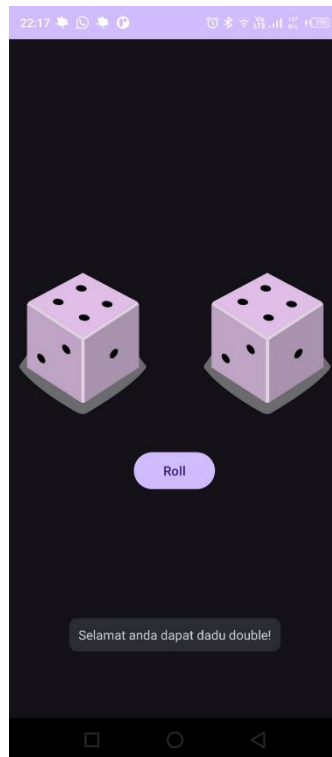
Gambar 1 Tampilan Awal Aplikasi

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.



Gambar 3 Tampilan Roll Dadu Double

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

A. Source Code

MainActivity.kt

```
1 package com.example.diceroller
2
3 import android.os.Bundle
4 import android.widget.Button
5 import android.widget.ImageView
6 import android.widget.Toast
7 import androidx.appcompat.app.AppCompatActivity
8
```



```

9
10 class MainActivity : AppCompatActivity() {
11     private var lastDiceRoll: Int? = null
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_main)
15
16         val rollButton: Button = findViewById(R.id.button)
17         rollButton.setOnClickListener { rollDice(); rollDice2()
18     }
19     }
20
21
22     private fun rollDice() {
23         // Create new Dice object with 6 sides and roll the
dice
24         val dice = Dice(6)
25         val diceRoll = dice.roll()
26
27         // Find the ImageView in the layout
28         val diceImage: ImageView = findViewById(R.id.imageView)
29
30         // Determine which drawable resource ID to use based on
the dice roll
31         val drawableResource = when (diceRoll) {
32             1 -> R.drawable.dice_1
33             2 -> R.drawable.dice_2
34             3 -> R.drawable.dice_3
35             4 -> R.drawable.dice_4
36             5 -> R.drawable.dice_5
37             else -> R.drawable.dice_6
38         }
39
40         // Update the ImageView with the correct drawable
resource ID
41         diceImage.setImageResource(drawableResource)
42
43         // Update the content description
44         diceImage.contentDescription = diceRoll.toString()
45
46         // Save the result of the dice roll
47         lastDiceRoll = diceRoll
48     }
49
50     private fun rollDice2() {
51         // Create new Dice object with 6 sides and roll the
dice
52         val dice = Dice(6)
53         val diceRoll = dice.roll()
54
55         // Find the ImageView in the layout
56         val diceImage: ImageView =

```

57	findViewById(R.id.imageView2)
58	
59	// Determine which drawable resource ID to use based on the dice roll
60	val drawableResource = when (diceRoll) {
61	1 -> R.drawable.dice_1
62	2 -> R.drawable.dice_2
63	3 -> R.drawable.dice_3
64	4 -> R.drawable.dice_4
65	5 -> R.drawable.dice_5
66	else -> R.drawable.dice_6
67	}
68	
69	// Update the ImageView with the correct drawable resource ID
70	diceImage.setImageResource(drawableResource)
71	
72	// Update the content description
73	diceImage.contentDescription = diceRoll.toString()
74	
75	// Check if both dice rolls are the same or equal to 6
76	if (diceRoll == lastDiceRoll) {
77	Toast.makeText(this, "Selamat anda dapat dadu double!", Toast.LENGTH_SHORT).show()
78	}
79	else{
80	Toast.makeText(this, "Anda belum beruntung", Toast.LENGTH_SHORT).show()
81	}
82	}
83	
84	class Dice(private val numSides: Int) {
85	
86	fun roll(): Int {
87	return (1..numSides).random()
88	}
89	}
90	}

Tabel 1. Source Code Jawaban Soal 1

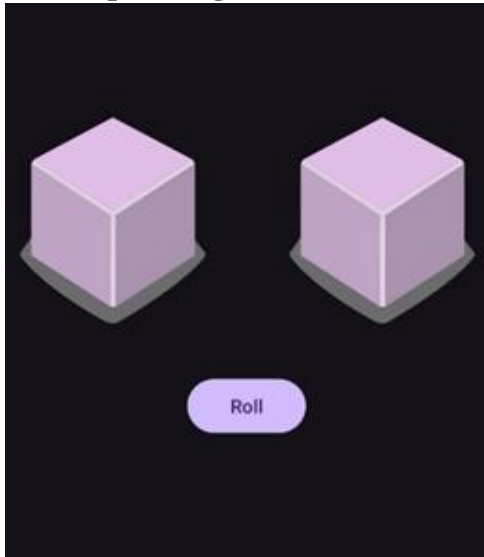
activity_main.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
3	xmlns:android="http://schemas.android.com/apk/res/android"
4	xmlns:app="http://schemas.android.com/apk/res-auto"
5	xmlns:tools="http://schemas.android.com/tools"
6	android:id="@+id/main"
7	android:layout_width="match_parent"
8	android:layout_height="match_parent"
9	tools:context=".MainActivity"

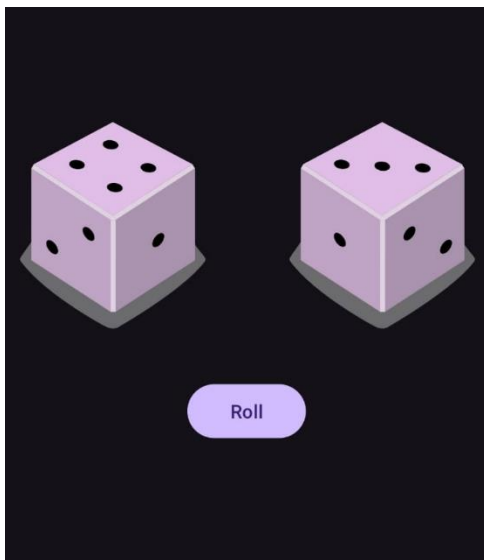
10	tools:layout_editor_absoluteX="-1dp"
11	tools:layout_editor_absoluteY="62dp">
12	
13	<Button
14	android:id="@+id/button"
15	android:layout_width="wrap_content"
16	android:layout_height="wrap_content"
17	android:layout_marginBottom="244dp"
18	android:text="@string/roll"
19	app:layout_constraintBottom_toBottomOf="parent"
20	app:layout_constraintEnd_toEndOf="parent"
21	app:layout_constraintHorizontal_bias="0.498"
22	app:layout_constraintStart_toStartOf="parent" />
23	
24	<ImageView
25	android:id="@+id/imageView"
26	android:layout_width="160dp"
27	android:layout_height="200dp"
28	android:layout_marginEnd="200dp"
29	app:layout_constraintBottom_toBottomOf="parent"
30	app:layout_constraintEnd_toEndOf="parent"
31	app:layout_constraintHorizontal_bias="0.49"
32	app:layout_constraintStart_toStartOf="parent"
33	app:layout_constraintTop_toTopOf="parent"
34	app:layout_constraintVertical_bias="0.429"
35	app:srcCompat="@drawable/dice_0" />
36	
37	<ImageView
38	android:id="@+id/imageView2"
39	android:layout_width="160dp"
40	android:layout_height="200dp"
41	android:layout_marginStart="200dp"
42	app:layout_constraintBottom_toBottomOf="parent"
43	app:layout_constraintEnd_toEndOf="parent"
44	app:layout_constraintHorizontal_bias="0.49"
45	app:layout_constraintStart_toStartOf="parent"
46	app:layout_constraintTop_toTopOf="parent"
47	app:layout_constraintVertical_bias="0.429"
48	app:srcCompat="@drawable/dice_0" />
49	
50	</androidx.constraintlayout.widget.ConstraintLayout>

Tabel 2. Source Code Jawaban Soal 1

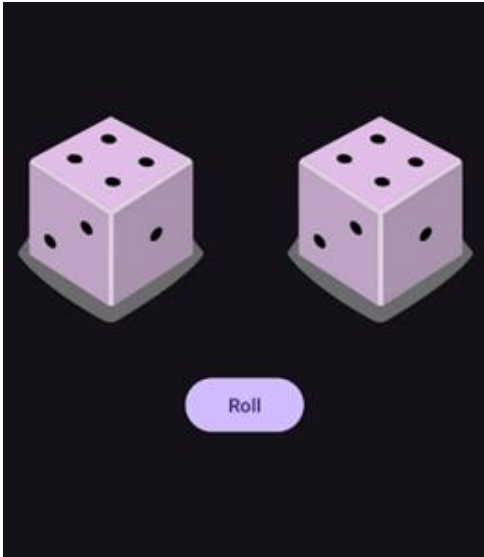
B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1



Gambar 1. Screenshot Hasil Jawaban Soal 1



Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

MainActivity.kt:

Baris 1 terdapat deklarasi package `com.example.diceroller`, yang digunakan untuk mengorganisir file ini dalam struktur aplikasi.

Baris 2–6 terdapat perintah import untuk mengambil kelas-kelas dari Android SDK yang digunakan dalam aplikasi, seperti `Button`, `ImageView`, `Toast`, dan `AppCompatActivity`.

Baris 8 terdapat deklarasi kelas `MainActivity` yang merupakan turunan dari `AppCompatActivity`, menandakan bahwa ini adalah activity utama aplikasi.

Baris 9 terdapat deklarasi variabel `lastDiceRoll` bertipe `Int?` untuk menyimpan hasil lemparan dadu pertama.

Baris 10–13 terdapat method `onCreate()` yang dijalankan saat activity pertama kali dibuat. Di dalamnya:

- `super.onCreate(savedInstanceState)` menjalankan logika default superclass.
- `setContentView(R.layout.activity_main)` menghubungkan file Kotlin ini dengan layout XML.

Baris 15 terdapat pemanggilan tombol dengan ID `button` dari layout yang disimpan ke variabel `rollButton`.

Baris 16 terdapat listener yang membuat tombol menjalankan dua fungsi saat ditekan, yaitu `rollDice()` dan `rollDice2()`.

Baris 19 terdapat deklarasi fungsi `rollDice()` untuk menangani lemparan dadu pertama.

Baris 20 terdapat pembuatan objek `Dice` dengan 6 sisi.

Baris 21 terdapat pemanggilan fungsi `roll()` dari objek `Dice` untuk mendapatkan angka acak 1–6.

Baris 23 terdapat pengambilan `ImageView` pertama dari layout menggunakan ID `imageView`.

Baris 25–31 terdapat pernyataan `when` yang mencocokkan angka hasil roll dengan gambar dadu yang sesuai dari `R.drawable`.

Baris 33 terdapat perintah untuk mengatur gambar yang ditampilkan di `ImageView` berdasarkan hasil dadu.

Baris 35 terdapat pengaturan `contentDescription` untuk aksesibilitas dengan isi angka hasil roll.

Baris 37 terdapat penyimpanan hasil roll ke variabel `lastDiceRoll` untuk digunakan pada fungsi `rollDice2()`.

Baris 40 terdapat deklarasi fungsi `rollDice2()` untuk menangani lemparan dadu kedua.

Baris 41–42 terdapat pembuatan objek `Dice` dan pemanggilan fungsi `roll()` untuk mendapatkan hasil acak dadu kedua.

Baris 44 terdapat pengambilan `ImageView` kedua dari layout menggunakan ID `imageView2`.

Baris 46–52 terdapat struktur `when` yang mencocokkan hasil dadu kedua dengan gambar yang sesuai.

Baris 54 terdapat perintah untuk mengubah gambar `ImageView` kedua sesuai hasil roll.

Baris 56 terdapat pengaturan `contentDescription` untuk gambar dadu kedua.

Baris 58 terdapat logika `if` yang membandingkan hasil dadu kedua dengan nilai `lastDiceRoll`.

Baris 59 terdapat `Toast` untuk menampilkan pesan “Selamat” jika kedua dadu bernilai sama.

Baris 61–62 terdapat `else block` yang menampilkan pesan “Anda belum beruntung” jika hasil dadu berbeda.

Baris 65 terdapat deklarasi kelas `Dice` yang menerima parameter `numSides` untuk menentukan jumlah sisi dadu.

Baris 67 terdapat fungsi `roll()` yang menghasilkan angka acak antara 1 sampai jumlah sisi dengan `(1..numSides).random()`.

Baris 68–70 terdapat penutup fungsi dan kelas.

activity_main.xml:

Baris 1 terdapat deklarasi XML standar yang menyatakan bahwa file ini menggunakan versi XML 1.0 dengan encoding UTF-8.

Baris 2 terdapat tag pembuka `<ConstraintLayout>` yang menjadi root layout dari tampilan, menggunakan `androidx.constraintlayout.widget.ConstraintLayout`.

Baris 3–5 terdapat deklarasi namespace (`xmlns`) yang diperlukan agar atribut XML seperti `android:`, `app:`, dan `tools:` dapat dikenali oleh Android Studio.

Baris 6 terdapat atribut `android:id="@+id/main"` untuk memberikan ID pada layout utama.

Baris 7–8 terdapat pengaturan ukuran layout root menjadi `match_parent`, artinya layout akan memenuhi seluruh lebar dan tinggi layar.

Baris 9 terdapat atribut `tools:context=".MainActivity"` untuk memberi tahu Android Studio bahwa layout ini digunakan oleh `MainActivity`.

Baris 10–11 terdapat atribut `tools:layout_editor_absoluteX` dan `tools:layout_editor_absoluteY` yang hanya digunakan di editor visual Android Studio, bukan saat runtime.

Baris 13 terdapat tag pembuka `<Button>` untuk mendefinisikan tombol.

Baris 14 terdapat ID tombol `@+id/button` agar bisa dikenali di Kotlin melalui `findViewById()`.

Baris 15–16 terdapat pengaturan ukuran tombol menjadi `wrap_content`, menyesuaikan ukuran berdasarkan isi teks.

Baris 17 terdapat margin bawah sebesar 244dp untuk memberi jarak dari bawah layout.

Baris 18 terdapat teks tombol yang diambil dari string resource `@string/roll`.

Baris 19–22 terdapat constraint untuk menyatukan tombol ke batas bawah, kiri, dan kanan parent layout agar posisinya berada di tengah bawah layar.

Baris 23 terdapat pengaturan `layout_constraintHorizontal_bias="0.498"` untuk sedikit menggeser posisi tombol ke tengah (bias ke kiri-kanan).

Baris 24 adalah penutup tag <Button>.

Baris 26 terdapat tag <ImageView> untuk menampilkan gambar dadu pertama.

Baris 27 terdapat ID @+id/imageView agar bisa diakses dari kode Kotlin.

Baris 28–29 mengatur ukuran gambar dadu menjadi 160dp x 200dp.

Baris 30 terdapat margin ke kanan sebesar 200dp agar dadu ini bergeser ke kiri dari tengah.

Baris 31–35 terdapat constraint agar gambar berada di tengah layout secara vertikal dan horizontal.

Baris 36 terdapat pengaturan layout_constraintVertical_bias="0.429" untuk sedikit menggeser gambar ke atas.

Baris 37 terdapat atribut app:srcCompat="@drawable/dice_0" untuk menentukan gambar default awal yang ditampilkan (dadu kosong).

Baris 38 adalah penutup tag <ImageView>.

Baris 40 terdapat tag <ImageView> untuk menampilkan gambar dadu kedua.

Baris 41 terdapat ID @+id/imageView2 untuk diakses dari Kotlin.

Baris 42–43 mengatur ukuran gambar menjadi sama seperti dadu pertama.

Baris 44 terdapat margin kiri sebesar 200dp untuk menggeser gambar ke kanan.

Baris 45–49 terdapat constraint agar posisi gambar berada di tengah layout secara vertikal dan horizontal.

Baris 50 terdapat pengaturan layout_constraintVertical_bias="0.429" agar gambar sejajar dengan dadu pertama.

Baris 51 terdapat app:srcCompat="@drawable/dice_0" untuk gambar default kedua (dadu kosong).

Baris 52 adalah penutup tag <ImageView>.

Baris 54 adalah penutup tag ConstraintLayout.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/AuFaMiReDo/PrakMobile/tree/main/Modul%201>