

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Abdurrahman Gilang Harjuna

NIM. 2310817110004

**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 : Abdurrahman Gilang Harjuna
NIM : 2310817110004

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
SOAL 1	6
A. Source Code.....	8
B. Output Program	11
C. Pembahasan	11
D. Tautan Git	12

DAFTAR GAMBAR

Tabel 1. Source Code Jawaban Soal 1.....	9
--	---

DAFTAR TABEL

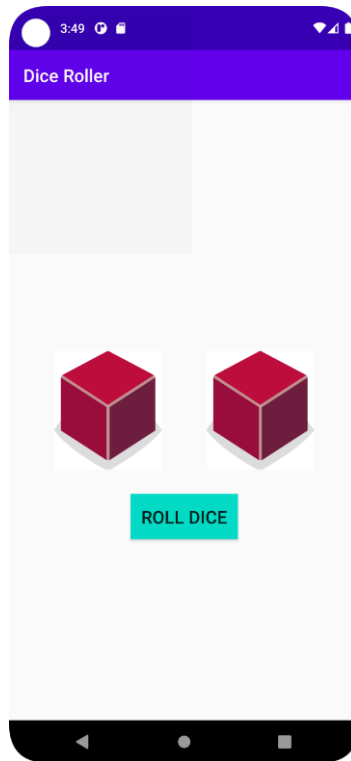
Table 1. Source Code Jawaban Soal 1.....	9
Table 2. Source Code Jawaban Soal 1.....	10

SOAL 1

Soal Praktikum:

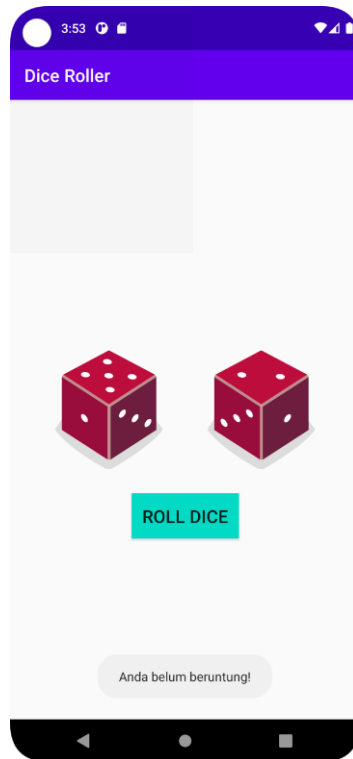
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



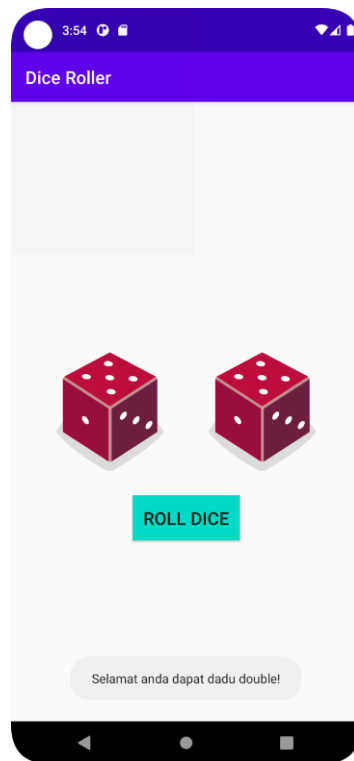
Gambar 1 Tampilan Awal Aplikasi

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.
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=147HT2IIH5qin3z5ta7H9y2N_5OMW81Ll&export=download



Gambar 3 Tampilan Roll Dadu Double

A. Source Code

1. MainActivity.kt

```
1 package com.example.diceroller
2
3 import android.os.Bundle
4 import android.widget.Button
5 import androidx.appcompat.app.AppCompatActivity
6 import android.widget.ImageView
7 import android.widget.Toast
8
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         val diceImage01: ImageView = findViewById(R.id.Dice1)
15         diceImage01.setImageResource(R.drawable.dice_0)
16
17         val diceImage02: ImageView = findViewById(R.id.Dice2)
18         diceImage02.setImageResource(R.drawable.dice_0)
19
20         val rollButton: Button = findViewById(R.id.Buttonn)
21         rollButton.setOnClickListener { diceRoller() }
```


	<pre> private fun diceRoller() { val dice1 = Dice(6) val diceRoll1 = dice1.roll() val diceImage1: ImageView = findViewById(R.id.Dice1) when (diceRoll1) { 1 -> diceImage1.setImageResource(R.drawable.dice_1) 2 -> diceImage1.setImageResource(R.drawable.dice_2) 3 -> diceImage1.setImageResource(R.drawable.dice_3) 4 -> diceImage1.setImageResource(R.drawable.dice_4) 5 -> diceImage1.setImageResource(R.drawable.dice_5) 6 -> diceImage1.setImageResource(R.drawable.dice_6) } val dice2 = Dice(6) val diceRoll2 = dice2.roll() val diceImage2: ImageView = findViewById(R.id.Dice2) when (diceRoll2) { 1 -> diceImage2.setImageResource(R.drawable.dice_1) 2 -> diceImage2.setImageResource(R.drawable.dice_2) 3 -> diceImage2.setImageResource(R.drawable.dice_3) 4 -> diceImage2.setImageResource(R.drawable.dice_4) 5 -> diceImage2.setImageResource(R.drawable.dice_5) 6 -> diceImage2.setImageResource(R.drawable.dice_6) } if (diceRoll1 == diceRoll2) { val toast = Toast.makeText(this, "Selamat anda dapat dadu double!", Toast.LENGTH_SHORT) toast.show() } else { val toast = Toast.makeText(this, "Anda belum beruntung!", Toast.LENGTH_SHORT) toast.show() } } class Dice(private val numSides: Int) { fun roll(): Int { return (1..numSides).random() } } </pre>
--	---

Table 1. Source Code Jawaban Soal 1

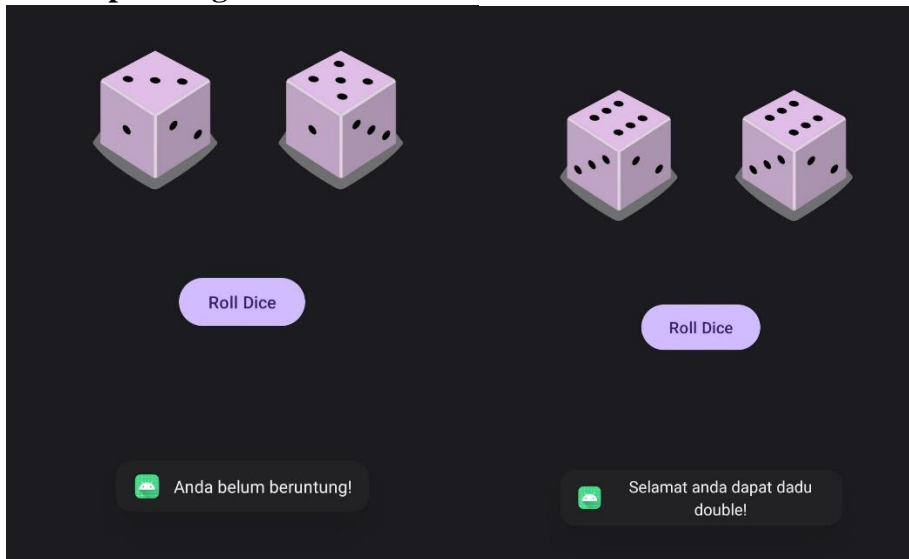
2. activity_main.xml

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"
	android:layout_height="match_parent"

8	<code>tools:context=".MainActivity"></code>
9	
10	<code><Button</code>
11	<code> android:id="@+id/Buttonn"</code>
	<code> android:layout_width="wrap_content"</code>
	<code> android:layout_height="wrap_content"</code>
	<code> android:layout_margin="52dp"</code>
	<code> android:text="Roll Dice"</code>
	<code> app:layout_constraintEnd_toEndOf="parent"</code>
	<code> app:layout_constraintStart_toStartOf="parent"</code>
	<code> app:layout_constraintTop_toBottomOf="@+id/Dice1"</code>
	<code> app:layout_constraintHorizontal_bias="0.498"</code>
	<code> tools:ignore="UnknownId" /></code>
	<code><ImageView</code>
	<code> android:id="@+id/Dice1"</code>
	<code> android:layout_width="150dp"</code>
	<code> android:layout_height="150dp"</code>
	<code> app:layout_constraintBottom_toBottomOf="parent"</code>
	<code> app:layout_constraintEnd_toEndOf="parent"</code>
	<code> app:layout_constraintHorizontal_bias="0.201"</code>
	<code> app:layout_constraintStart_toStartOf="parent"</code>
	<code> app:layout_constraintTop_toTopOf="parent"</code>
	<code> app:layout_constraintVertical_bias="0.449"</code>
	<code> app:srcCompat="@drawable/dice_1"</code>
	<code> tools:layout_editor_absoluteX="83dp"</code>
	<code> tools:layout_editor_absoluteY="268dp" /></code>
	<code><ImageView</code>
	<code> android:id="@+id/Dice2"</code>
	<code> android:layout_width="150dp"</code>
	<code> android:layout_height="150dp"</code>
	<code> app:layout_constraintBottom_toBottomOf="parent"</code>
	<code> app:layout_constraintEnd_toEndOf="parent"</code>
	<code> app:layout_constraintHorizontal_bias="0.837"</code>
	<code> app:layout_constraintStart_toStartOf="parent"</code>
	<code> app:layout_constraintTop_toTopOf="parent"</code>
	<code> app:layout_constraintVertical_bias="0.449"</code>
	<code> app:srcCompat="@drawable/dice_2" /></code>
	<code></androidx.constraintlayout.widget.ConstraintLayout></code>

Table 2. Source Code Jawaban Soal 1

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt:

MainActivity adalah activity utama dari aplikasi. Turunan dari AppCompatActivity, yang merupakan base class umum untuk activity modern di android. onCreate() dipanggil saat activity pertama dibuat. setContentView(R.layout.activity_main) menghubungkan file XML sebagai tampiln utama aplikasi.

```
val diceImage01: ImageView = findViewById(R.id.Dice1)
diceImage01.setImageResource(R.drawable.dice_0) dan val diceImage02: ImageView =
findViewById(R.id.Dice2) diceImage02.setImageResource(R.drawable.dice_0) kedua kode
ini mengamil dua ImgeView dari layout (dadu 1 dan 2). Gambar diset ke dice_0, artinya
sebelum tombol ditekan, gambar dadu kosong.
```

```
val rollButton: Button = findViewById(R.id.Buttonnn) rollButton.setOnClickListener {
diceRoller() } menghubungkan tombol dari layout dengan kode, saat tombol diklik akan
menjalankan fungsi diceRoller().private fun diceRoller() { ... } val dice1 = Dice(6) val
diceRoll1 = dice1.roll() membuat dice dengan 6 sisi, roll() menghasilkan angka acak antara
1-6.
```

```
val diceImage1: ImageView = findViewById(R.id.Dice1) when (diceRoll1) { 1 ->
diceImage1.setImageResource(R.drawable.dice_1) ...} Berdasarkan hasil roll, gambar dadu
diubah sesuai angka (dice_1.png sampai dice_6.png). val dice2 = Dice(6) val diceRoll2 =
dice2.roll() dan val diceImage2: ImageView = findViewById(R.id.Dice2) when (diceRoll2)
{ ... } sama seperti dadu pertama tapi ini untuk dadu kedua.
```

Lanjut ke toast, menampilkan pesan kecil di layar, disini cek apakah kedua dadu sama atau tidak. Jika sama berarti dapat dadu double, jika beda maka belum beruntung. class Dice(private val numSides: Int) { ini adalah class custom dadu, numSides ini jumlah sisi dadu berupa angka. fun roll(): Int { return (1..numSides).random() } } Fungsi roll() menghasilkan angka acak dari 1 sampai numSides yaitu 6.

2. activity_main.xml

<androidx.constraintlayout.widget.ConstraintLayout> jenis layout yang memungkinkan untuk mengatur setiap elemen dengan menghubungkan ke elemen lain atau ke parent-nya. Ini root layout artinya semua elemen UI di dalamnya akan menggunakan constraint untuk pengaturan posisinya.

ImageView Dice 1 ini adalah gambar dadu pertama. @+id/Dice1 ini Id untuk mengakses dari kode kotlin. layout_width, layout_height di set ke 150dp, ukuran tetap. app:srcCompat="@drawable/dice_1" menampilkan gambar dadu (dice_1.png dari folder res/drawable). Sisa constraint untuk mengatur posisi agar bisa di tengah bagian kiri.

ImageView Dice 2 ini adalah dadu kedua. @+id/Dice2 sebagai id sisanya sama seperti dadu pertama. Untuk posisi diatur dengan constraint agar berada di tengah bagian kanan.

Button roll dice dengan id Buttonn, tombol teks Roll Dice. Diberi margin 52dp sekelilingnya. Posisinya diletakkan dibawah dice 1 karena Top_toBottomOf="@+id/Dice1" dan diatur lagi dengan Start_toStartOf="parent", End_toEndOf="parent" dan Horizontal_bias="0.498" agar bisa berada di tengah antara dice 1 dan 2.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Specter735/Pemrograman-Mobile.git>