

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Muhammad Azwin Hakim

NIM. 2310817310012

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2024**

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 Azwin Hakim
NIM : 2310817310012

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	13
D. Tautan Git	13

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	11
Gambar 2. Screenshot Hasil Jawaban Soal 1	12
Gambar 3. Screenshot Hasil Jawaban Soal 1	12

DAFTAR TABEL

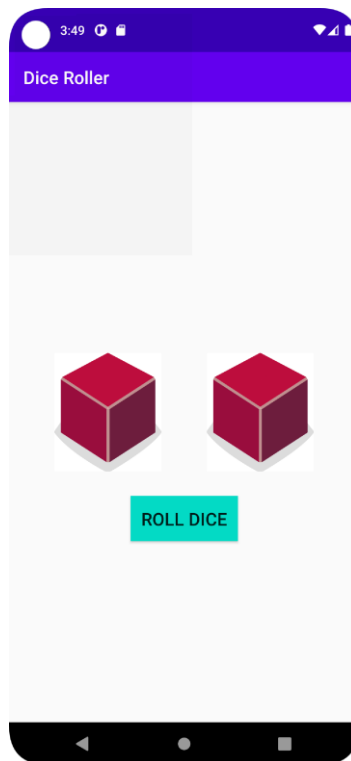
Tabel 1. Source Code Jawaban Soal 1.....	10
Tabel 2. Source Code Jawaban Soal 1.....	11

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.



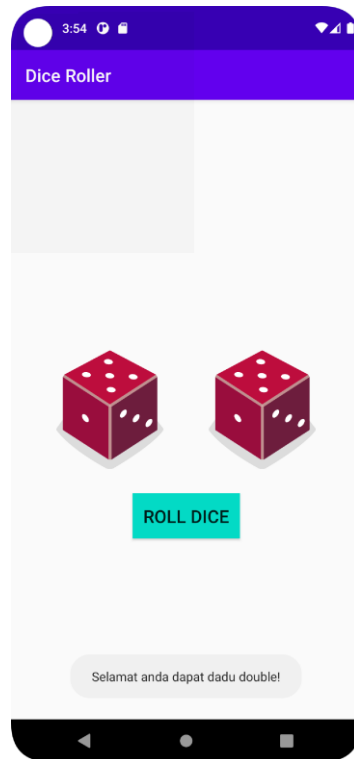
Gambar 1 Tampilan Awal Aplikasi

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_5OMW81LI&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.Toast
5 import androidx.activity.enableEdgeToEdge
6 import androidx.appcompat.app.AppCompatActivity
7 import androidx.core.view.ViewCompat
8 import androidx.core.view.WindowInsetsCompat
9 import com.example.diceroller.databinding.ActivityMainBinding
10
11 class MainActivity : AppCompatActivity() {
12
13     private lateinit var binding: ActivityMainBinding
14
15     private var diceRoll = 0
16     private var diceRollTwo = 0
17
18     override fun onCreate(savedInstanceState: Bundle?) {
19         super.onCreate(savedInstanceState)
20         enableEdgeToEdge()
21
22         binding = ActivityMainBinding.inflate(layoutInflater)
```



```

23         setContentView(binding.root)
24
25         ViewCompat.setOnApplyWindowInsetsListener(binding.main)
26     { v, insets ->
27         val systemBars =
28         insets.getInsets(WindowInsetsCompat.Type.systemBars())
29         v.setPadding(systemBars.left, systemBars.top,
30         systemBars.right, systemBars.bottom)
31         insets
32     }
33
34     binding.button.setOnClickListener { rollDice() }
35
36     if (savedInstanceState != null) {
37         diceRoll = savedInstanceState.getInt("diceRoll", 0)
38         diceRollTwo =
39         savedInstanceState.getInt("diceRollTwo", 0)
40     }
41
42     updateDiceImages()
43 }
44
45 override fun onSaveInstanceState(outState: Bundle) {
46     super.onSaveInstanceState(outState)
47     outState.putInt("diceRoll", diceRoll)
48     outState.putInt("diceRollTwo", diceRollTwo)
49 }
50
51 private fun rollDice() {
52     val dice = Dice(6)
53     val diceTwo = Dice(6)
54
55     diceRoll = dice.roll()
56     diceRollTwo = diceTwo.roll()
57
58     updateDiceImages()
59
60     if (diceRoll == diceRollTwo) {
61         Toast.makeText(this, "Selamat anda dapat dadu
62         double!", Toast.LENGTH_SHORT).show()
63     } else {
64         Toast.makeText(this, "Anda belum beruntung!",
65         Toast.LENGTH_SHORT).show()
66     }
67 }
68
69 private fun updateDiceImages() {
70
71     binding.imageView.setImageResource(getDiceImage(diceRoll))
72     binding.imageView.contentDescription =
73     diceRoll.toString()

```

```

66
67 binding.imageView2.setImageResource(getDiceImage(diceRollTwo))
68     binding.imageView2.contentDescription =
    diceRollTwo.toString()
69 }
70
71     private fun getDiceImage(value: Int): Int {
72         return when (value) {
73             1 -> R.drawable.dice_1
74             2 -> R.drawable.dice_2
75             3 -> R.drawable.dice_3
76             4 -> R.drawable.dice_4
77             5 -> R.drawable.dice_5
78             6 -> R.drawable.dice_6
79             else -> R.drawable.dice_0
80         }
81     }
82 }
83
84 class Dice(private val numSides: Int) {
85     fun roll(): Int = (1..numSides).random()
86 }

```

Tabel 1. Source Code Jawaban Soal 1

2. 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    <Button
11        android:id="@+id/button"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:layout_marginTop="16dp"
15        android:text="Roll Dice"
16        app:layout_constraintEnd_toEndOf="parent"
17        app:layout_constraintStart_toStartOf="parent"
18        app:layout_constraintTop_toBottomOf="@+id/imageView" />
19
20    <ImageView
21        android:id="@+id/imageView"
22        android:layout_width="160dp"
23        android:layout_height="200dp"
24        android:contentDescription="@android:string/no"
25        app:layout_constraintBottom_toBottomOf="parent"

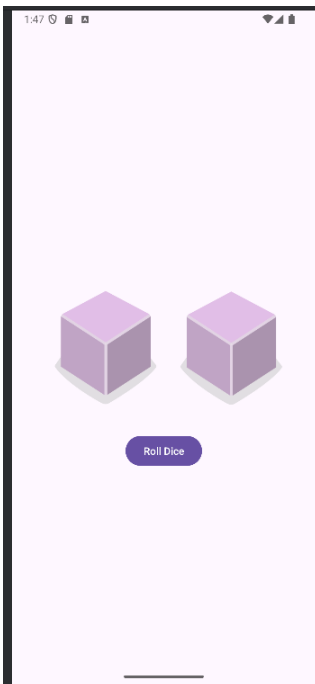
```

26	app:layout_constraintEnd_toEndOf="parent"
27	app:layout_constraintHorizontal_bias="0.187"
28	app:layout_constraintStart_toStartOf="parent"
29	app:layout_constraintTop_toTopOf="parent"
30	app:layout_constraintVertical_bias="0.499"
31	tools:srcCompat="@drawable/dice_0" />
32	
33	<ImageView
34	android:id="@+id/imageView2"
35	android:layout_width="160dp"
36	android:layout_height="200dp"
37	android:contentDescription="@android:string/no"
38	app:layout_constraintBottom_toBottomOf="parent"
39	app:layout_constraintEnd_toEndOf="parent"
40	app:layout_constraintHorizontal_bias="0.864"
41	app:layout_constraintStart_toStartOf="parent"
42	app:layout_constraintTop_toTopOf="parent"
43	tools:srcCompat="@drawable/dice_0" />
44	
45	</androidx.constraintlayout.widget.ConstraintLayout>

Tabel 2. Source Code Jawaban Soal 1

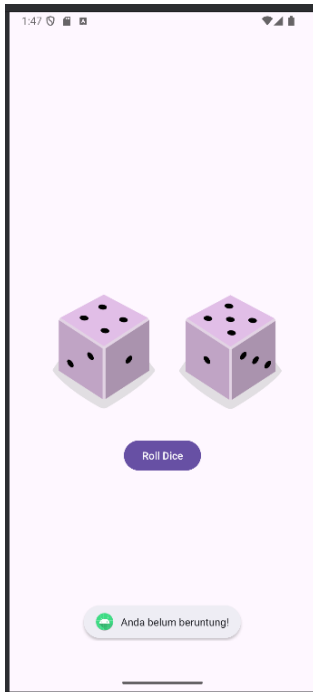
B. Output Program

- Tampilan awal :



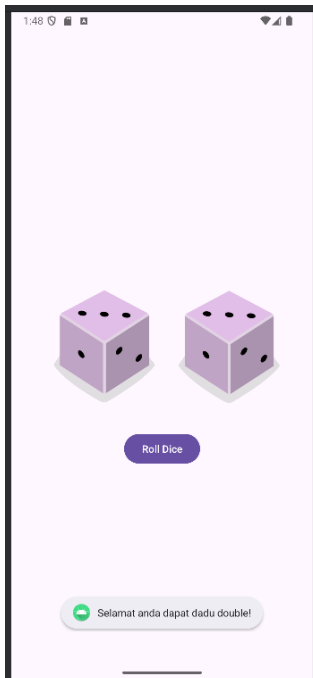
Gambar 1. Screenshot Hasil Jawaban Soal 1

- Tampilan jika nilai dadu berbeda :



Gambar 2. Screenshot Hasil Jawaban Soal 1

- Tampilan jika dadu bernilai sama :



Gambar 3. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt:

Kode pertama adalah file Kotlin MainActivity.kt yang berfungsi sebagai logika utama aplikasi Android untuk melempar dua buah dadu. Aktivitas ini menggunakan ViewBinding, seperti yang ditunjukkan oleh deklarasi "private lateinit var binding: ActivityMainBinding", untuk menghubungkan elemen-elemen di XML dengan kode Kotlin tanpa perlu menggunakan findViewById.

Pada metode onCreate, aplikasi mengatur layout dengan "setContentView(binding.root)", lalu menetapkan listener pada tombol "binding.button.setOnClickListener { rollDice() }" untuk menjalankan fungsi rollDice() saat tombol ditekan. Fungsi rollDice() sendiri menggunakan class Dice yang didefinisikan di bagian bawah kode ("class Dice(private val numSides: Int)") untuk menghasilkan angka acak antara 1 hingga 6 dengan "diceRoll = dice.roll()" dan "diceRollTwo = diceTwo.roll()". Setelah angka acak didapatkan, fungsi updateDiceImages() dipanggil untuk memperbarui gambar dadu sesuai angka yang muncul, menggunakan fungsi getDiceImage(value: Int), yang mengembalikan ID resource drawable berdasarkan nilai dadu. Selain itu, jika kedua dadu menunjukkan angka yang sama ("if (diceRoll == diceRollTwo)"), maka akan muncul toast "Selamat anda dapat dadu double!", jika tidak, toast "Anda belum beruntung!" akan ditampilkan. Untuk menangani rotasi layar, onSaveInstanceState digunakan untuk menyimpan nilai diceRoll dan diceRollTwo agar dapat dipulihkan saat aktivitas dibuat ulang.

2. activity_main.xml

Kode kedua adalah file XML activity_main.xml yang mendefinisikan tampilan antarmuka pengguna menggunakan ConstraintLayout sebagai root layout. Di dalam layout ini terdapat sebuah tombol dengan ID "@+id/button" yang berfungsi sebagai pemicu untuk melempar dadu. Tombol ini ditempatkan secara horizontal di tengah dan vertikal di bawah ImageView pertama melalui atribut app:layout_constraintTop_toBottomOf="@+id/imageView". Dua ImageView digunakan untuk menampilkan gambar dua buah dadu, dengan ID masing-masing "@+id/imageView" dan "@+id/imageView2". Kedua gambar ini ditempatkan secara horizontal di sisi kiri dan kanan layout, yang dikontrol melalui atribut app:layout_constraintHorizontal_bias. Gambar awal yang digunakan untuk kedua dadu adalah @drawable/dice_0, yang merupakan gambar dadu kosong atau posisi awal sebelum dadu dilempar. Layout ini memiliki atribut tools:context=".MainActivity" yang menunjukkan bahwa layout terkait dengan MainActivity.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Omelette719/Pemrograman-Mobile.git>