

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 2**



**ANDROID BASICS IN KOTLIN**

**Oleh:**

**Salsabila Syifa**

**NIM. 2010817320004**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS LAMBUNG MANGKURAT  
MARET 2022**

**LEMBAR PENGESAHAN**  
**LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE**  
**MODUL 2**

Laporan Praktikum Pemrograman Mobile Modul 2. Android Basics in Kotlin ini disusun sebagai syarat lulus mata kuliah Praktikum Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Salsabila Syifa  
NIM : 20101817320004

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Rezi Rahadianor  
NIM. 1810817210019

Andreyan Rizky Baskara, S.Kom., M.Kom.  
NIP. 19930703 201903 1 011

## DAFTAR ISI

LEMBAR PENGESAHAN .....	2
DAFTAR ISI .....	3
DAFTAR GAMBAR.....	4
SOAL 1.....	5
A.    Source Code .....	6
B.    Output Program.....	9
C.    Pembahasan.....	9
D.    Tautan Git .....	10

## DAFTAR GAMBAR

Gambar 1. Tampilan Awal Aplikasi.....	5
Gambar 2. Tampilan Dadu setelah Di Roll .....	5
Gambar 3. Tampilan Roll Dadu Double.....	6
Gambar 4. Screenshoot Hasil Jawaban Soal 1: Tampilan Awal Aplikasi.....	9
Gambar 5. Screenshoot Hasil Jawaban Soal 1: Tampilan Dadu setelah Di Roll .....	9
Gambar 6. Screenshoot Hasil Jawaban Soal 1: Tampilan Roll Dadu Double.....	9

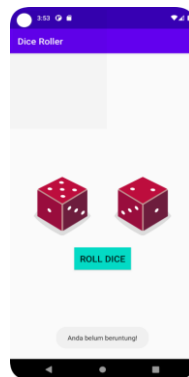
## SOAL 1

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.

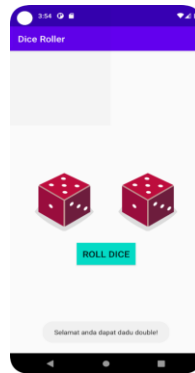


Gambar 2. Tampilan Dadu setelah Di Roll

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

Kotlin

```
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 class MainActivity : AppCompatActivity() {
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14
15         val rollButton: Button =
16             findViewById(R.id.button)
17
18         rollButton.setOnClickListener {acakDadu()}
19
20         printDadu()
21     }
22
23     private fun printDadu() {
24         val diceImg1: ImageView =
25             findViewById(R.id.imageView)
26         diceImg1.setImageResource(R.drawable.dice_0)
```

```

26         val diceImg2: ImageView =
findViewById(R.id.imageView2)
27         diceImg2.setImageResource(R.drawable.dice_0)
28     }
29
30     private fun acakDadu() {
31         val dadu1 = Dadu(6)
32         val acakDadu1 = dadu1.acak()
33         val gambarDadu1: ImageView =
findViewById(R.id.imageView)
34         val drawableResource1 = when (acakDadu1) {
35             1 -> R.drawable.dice_1
36             2 -> R.drawable.dice_2
37             3 -> R.drawable.dice_3
38             4 -> R.drawable.dice_4
39             5 -> R.drawable.dice_5
40             else -> R.drawable.dice_6
41         }
42
43         gambarDadu1.setImageResource(drawableResource1)
44         gambarDadu1.contentDescription =
acakDadu1.toString()
45
46         val dadu2 = Dadu(6)
47         val acakDadu2 = dadu2.acak()
48         val gambarDadu2: ImageView =
findViewById(R.id.imageView2)
49         val drawableResource2 = when (acakDadu2) {
50             1 -> R.drawable.dice_1
51             2 -> R.drawable.dice_2
52             3 -> R.drawable.dice_3
53             4 -> R.drawable.dice_4
54             5 -> R.drawable.dice_5
55             else -> R.drawable.dice_6
56         }
57
58         gambarDadu2.setImageResource(drawableResource2)
59         gambarDadu2.contentDescription =
acakDadu2.toString()
60
61         if (acakDadu1==acakDadu2){
62             Toast.makeText(this, "Selamat anda dapat dadu
double!", Toast.LENGTH_SHORT).show()
63         } else {
64             Toast.makeText(this, "Anda belum beruntung!",
Toast.LENGTH_SHORT).show()
65         }

```

66	}
67	}
68	
69	class Dadu(private val angkaDadu: Int) {
70	fun acak(): Int {
71	return (1..angkaDadu).random()
72	}
73	}
74	
XML (Extensible Markup Language)	
1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
3	
4	xmlns:android="http://schemas.android.com/apk/res/android
5	xmlns:app="http://schemas.android.com/apk/res-auto"
6	xmlns:tools="http://schemas.android.com/tools"
7	android:layout_width="match_parent"
8	android:layout_height="match_parent"
9	tools:context=".MainActivity">
10	
11	<ImageView
12	android:id="@+id/imageView2"
13	android:layout_width="160dp"
14	android:layout_height="200dp"
15	android:contentDescription="@string/dice2"
16	app:layout_constraintBottom_toBottomOf="parent"
17	app:layout_constraintEnd_toEndOf="parent"
18	app:layout_constraintHorizontal_bias="0.936"
19	app:layout_constraintStart_toStartOf="parent"
20	app:layout_constraintTop_toTopOf="parent"
21	app:layout_constraintVertical_bias="0.499"
22	tools:srcCompat="@drawable/dice_2" />
23	
24	<ImageView
25	android:id="@+id/imageView"
26	android:layout_width="160dp"
27	android:layout_height="200dp"
28	android:contentDescription="@string/dice"
29	app:layout_constraintBottom_toBottomOf="parent"
30	app:layout_constraintEnd_toEndOf="parent"
31	app:layout_constraintHorizontal_bias="0.063"
32	app:layout_constraintStart_toStartOf="parent"
33	app:layout_constraintTop_toTopOf="parent"
34	app:layout_constraintVertical_bias="0.499"
35	tools:srcCompat="@drawable/dice_1" />
36	

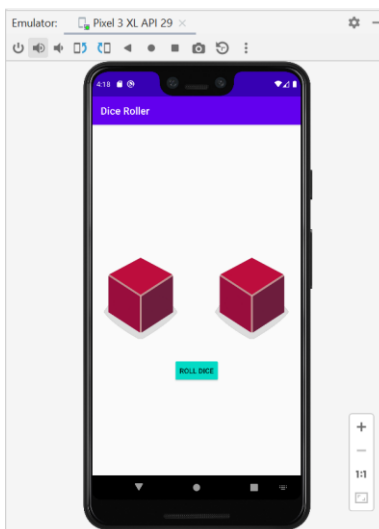


```

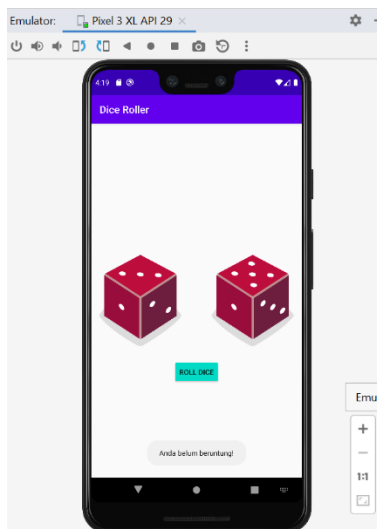
37     <Button
38         android:id="@+id/button"
39         android:layout_width="wrap_content"
40         android:layout_height="wrap_content"
41         android:backgroundTint="@color/teal_200"
42         android:layout_marginTop="16dp"
43         android:text="@string/roll_dice"
44         android:textColor="@color/black"
45         app:layout_constraintEnd_toEndOf="parent"
46         app:layout_constraintStart_toStartOf="parent"
47
48     app:layout_constraintTop_toBottomOf="@+id/imageView"
49         tools:targetApi="lollipop" />
50 </androidx.constraintlayout.widget.ConstraintLayout>

```

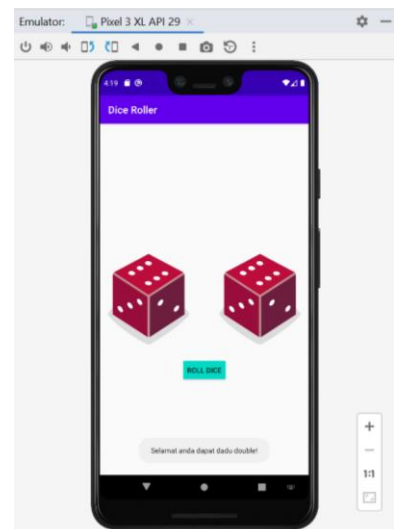
## B. Output Program



Gambar 4. Screenshoot Hasil Jawaban Soal 1: Tampilan Awal Aplikasi



Gambar 5. Screenshoot Hasil Jawaban Soal 1: Tampilan Dadu setelah Di Roll



Gambar 6. Screenshoot Hasil Jawaban Soal 1: Tampilan Roll Dadu Double

## C. Pembahasan

### Kotlin

- Pada baris [9] Aktivitas tersebut memungkinkan pengguna melempar dadu dan melihat hasilnya di layar.
- Pada baris [11] Metode ini dipanggil saat aktivitas pada baris [9] dibuat. Dalam metode ini, terdapat beberapa widget yang digunakan didalamnya yaitu
  - Baris [15] digunakan untuk menemukan tombol di layout .xml

- Baris [17] digunakan untuk menyetel klik di tombol agar melemparkan dadu saat pengguna mengetuk tombol
- Baris [19] menampilkan dadu kosong saat aplikasi dimulai
- Pada baris [30] dibuat fungsi untuk mengacak dadu dan memperbarui layar dengan hasilnya
- Baris [31 dan [32] dibuat objek Dadu baru (yang pertama) dengan 6 sisi dan objek untuk mengacaknya dengan fungsi acak pada kelas dadu
- Baris [33] digunakan untuk menemukan ImageView di layout .xml
- Baris [34] digunakan untuk mengunggah gambar mana yang akan digunakan berdasarkan lemparan dadu
- Baris [43] digunakan untuk memperbarui ImageView dengan gambar yang dipanggil pada baris [34]
- Pada baris [46] sampai [48] dibuat objek dadu kedua yang inisiasi dan deklarasinya sama seperti objek dadu sebelumnya
- Pada baris [61] sampai [62] merupakan suatu kondisional statement (if) yang digunakan untuk menampilkan pesan “Selamat anda dapat dadu double!” apabila user mendapatkan nilai dadu yang sama antara Dadu 1 dan Dadu 2 atau nilai double
- Pada baris [63] sampai [64] merupakan suatu kondisional statement (else) yang digunakan untuk menampilkan pesan “Anda belum beruntung!” apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2
- Pada baris 69 merupakan kelas Dadu dengan jumlah sisi yang tetap.
- Pada baris 70 dibuat fungsi untuk melakukan lemparan dadu acak dan mengembalikan hasilnya.

#### ***XML (Extensible Markup Language)***

- Ada beberapa tools yang digunakan dalam layout UI pada program ini yaitu imageview dan button
- Pada bagian awal dan akhir code terdapat tools berupa ConstraintLayout yang memungkinkan kita untuk membuat tata letak yang kompleks dan besar dengan hierarki tampilan datar (tidak ada kelompok tampilan bertingkat). ConstraintLayout mirip dengan RelativeLayout yang semua tampilannya diletakkan sesuai dengan hubungan antara tampilan yang setara dan tata letak induk, tetapi lebih fleksibel dari RelativeLayout dan lebih mudah digunakan dengan Layout Editor Android Studio.

#### **D. Tautan Git**

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/salsabilaSyifa/praktikummobile2/tree/main/modul2>