

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 2**



**Android Basics in Kotlin**

**Oleh:**

**Muhammad Firda Ryanifar**

**NIM. 2010817120007**

**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 Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Muhammad Firda Ryanifar  
NIM : 2010817310003

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
DAFTAR TABEL.....	5
SOAL .....	6
<b>A. Source Code</b> .....	8
<b>B. Output Program</b> .....	11
<b>C. Pembahasan</b> .....	13
TAUTAN GIT .....	15

## DAFTAR GAMBAR

Gambar 1 Tampilan Awal Aplikasi .....	6
Gambar 2 Tampilan Dadu Setelah Di Roll .....	7
Gambar 3 Tampilan Roll Dadu Double .....	8
Gambar 5 Screenshot Hasil Tampilan Awal Aplikasi .....	11
Gambar 6 Screenshot Hasil Tampilan Setelah di Roll.....	12
Gambar 7 Screenshot Hasil Tampilan Roll Dadu Double .....	12

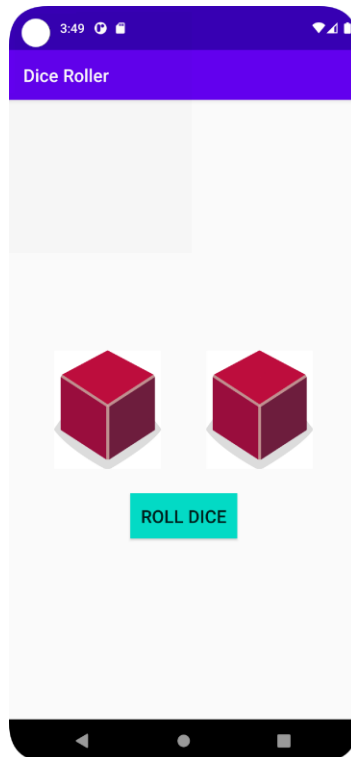
## DAFTAR TABEL

Table 4 Source Code activity_main.xml.....	9
Table 5 Source Code MainActivity.kt .....	11

## SOAL

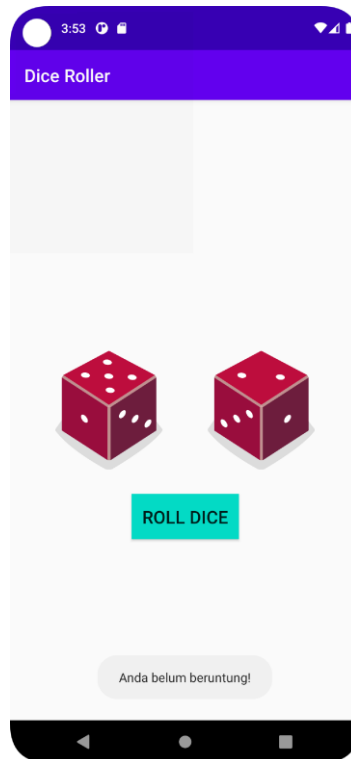
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



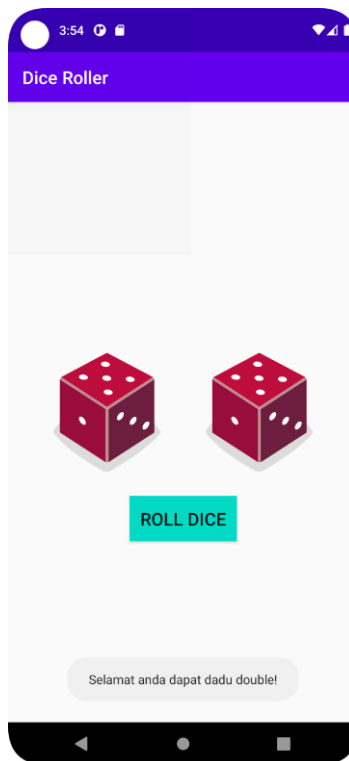
*Gambar 1 Tampilan Awal Aplikasi*

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](https://drive.google.com/u/0/uc?id=147HT2IIH5qin3z5ta7H9y2N_5OMW81Ll&export=download)



*Gambar 3 Tampilan Roll Dadu Double*

## A. Source Code

```

1  <?xml version="1.0" encoding="utf-8"?>
2
3  <!-- Layout of the text-based Dice Roller app -->
4  <androidx.constraintlayout.widget.ConstraintLayout
5
6  xmlns:android="http://schemas.android.com/apk/res/android"
7      xmlns:app="http://schemas.android.com/apk/res-auto"
8      xmlns:tools="http://schemas.android.com/tools"
9      android:layout_width="match_parent"
10     android:layout_height="match_parent"
11     tools:context=".MainActivity">
12
13
14     <Button
15         android:id="@+id/button"
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:layout_marginTop="16dp"
19         android:background="@color/colorAccent"
20         android:text="@string/roll"

```



21	app:layout_constraintEnd_toEndOf="parent"
22	app:layout_constraintStart_toStartOf="parent"
23	app:layout_constraintTop_toBottomOf="@+id/image1"
24	/>
25	
26	
27	<ImageView
28	android:id="@+id/image1"
29	android:layout_width="100dp"
30	android:layout_height="150dp"
31	android:layout_marginStart="30dp"
32	android:layout_marginTop="265dp"
33	app:layout_constraintEnd_toEndOf="parent"
34	app:layout_constraintHorizontal_bias="0.0"
35	app:layout_constraintStart_toEndOf="@+id/image2"
36	app:layout_constraintTop_toTopOf="parent"
37	tools:srcCompat="@drawable/empty_dice" />
38	
39	<ImageView
40	android:id="@+id/image2"
41	android:layout_width="100dp"
42	android:layout_height="150dp"
43	android:layout_marginStart="80dp"
44	android:layout_marginTop="265dp"
45	app:layout_constraintStart_toStartOf="parent"
46	app:layout_constraintTop_toTopOf="parent"
47	tools:srcCompat="@drawable/empty_dice" />
48	
49	
50	</androidx.constraintlayout.widget.ConstraintLayout>

Table 1 Source Code activity\_main.xml

1	
2	package com.example.diceroller
3	
4	import android.os.Bundle
5	import android.widget.Button
6	import android.widget.ImageView
7	import android.widget.Toast
8	import androidx.appcompat.app.AppCompatActivity
9	
10	class MainActivity : AppCompatActivity() {
11	override fun onCreate(savedInstanceState: Bundle?) {
12	super.onCreate(savedInstanceState)
13	setContentView(R.layout.activity_main)
14	

```

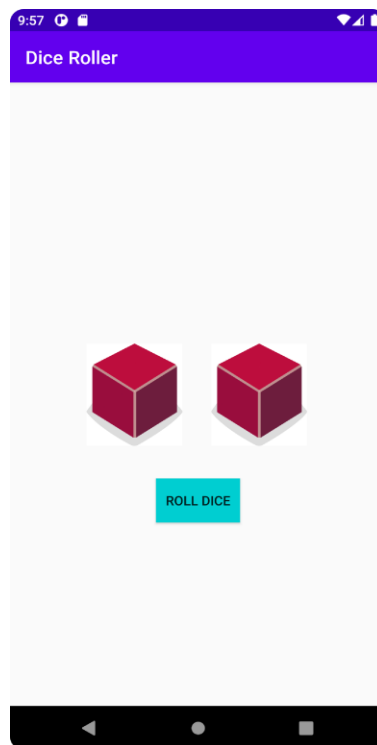
15
16         val rollButton: Button =
17     findViewById(R.id.button)
18
19         rollButton.setOnClickListener { rollDice() }
20
21         val diceImage1: ImageView =
22     findViewById(R.id.image1)
23         val diceImage2: ImageView =
24     findViewById(R.id.image2)
25
26
27     diceImage1.setImageResource(R.drawable.empty_dice)
28
29     diceImage2.setImageResource(R.drawable.empty_dice)
30     }
31
32     private fun rollDice() {
33         val dice = Dice(6)
34         val diceRoll1 = dice.roll()
35         val diceRoll2 = dice.roll()
36
37         val diceImage1: ImageView =
38     findViewById(R.id.image1)
39         val diceImage2: ImageView =
40     findViewById(R.id.image2)
41
42         val drawableResource1 = when (diceRoll1) {
43             1 -> R.drawable.dice_1
44             2 -> R.drawable.dice_2
45             3 -> R.drawable.dice_3
46             4 -> R.drawable.dice_4
47             5 -> R.drawable.dice_5
48             else -> R.drawable.dice_6
49         }
50
51         val drawableResource2 = when (diceRoll2) {
52             1 -> R.drawable.dice_1
53             2 -> R.drawable.dice_2
54             3 -> R.drawable.dice_3
55             4 -> R.drawable.dice_4
56             5 -> R.drawable.dice_5
57             else -> R.drawable.dice_6
58         }
59
60         diceImage1.setImageResource(drawableResource1)
61         diceImage2.setImageResource(drawableResource2)

```

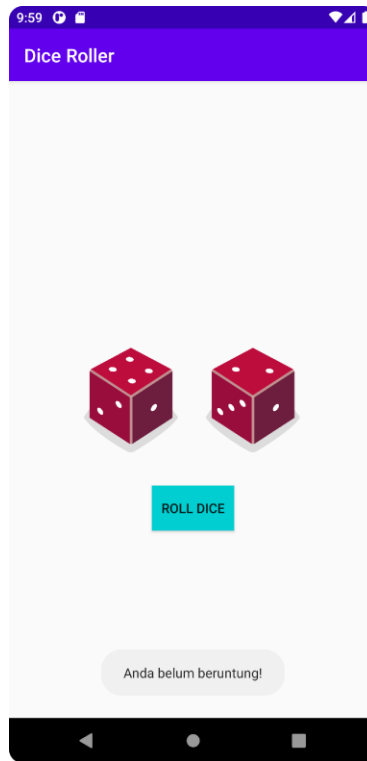
62	diceImage1.contentDescription =
63	diceRoll1.toString()
64	diceImage2.contentDescription =
65	diceRoll2.toString()
66	
67	if (diceRoll1 == diceRoll2) {
68	Toast.makeText(applicationContext, "Selamat
69	anda dapat dadu double!", Toast.LENGTH_LONG).show()
70	} else {
71	Toast.makeText(applicationContext, "Anda
72	belum beruntung!", Toast.LENGTH_LONG).show()
73	
74	}
75	}
76	}
77	
78	class Dice(private val numSides: Int) {
79	fun roll(): Int {
80	return (1..numSides).random()
81	}
82	}

Table 2 Source Code MainActivity.kt

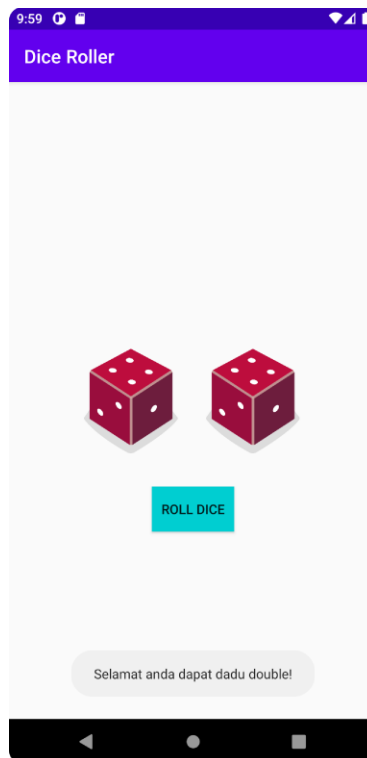
## B. Output Program



Gambar 4 Screenshot Hasil Tampilan Awal Aplikasi



*Gambar 5 Screenshot Hasil Tampilan Setelah di Roll*



*Gambar 6 Screenshot Hasil Tampilan Roll Dadu Double*

### C. Pembahasan

Source Code activity\_main.xml

Diawali baris [1] menunjukkan bahwa jenis file adalah XML. Tag `ConstraintLayout` baris [2] setelah itu `xmlns` adalah singkatan dari namespace XML, dan setiap baris mendefinisikan skema atau kosakata untuk atribut yang terkait dengan kata-kata tersebut. Namespace `android:`, misalnya, menandai atribut yang ditentukan oleh sistem Android. Semua atribut dalam XML tata letak dimulai dengan salah satu namespace tersebut. Pada baris [9] terdapat tag `<button>`. Atribut `android:id` gunanya untuk menginisiasi id pada button tsb. Pada baris [11] dan [12] terdapat atribut untuk mengatur lebar dan tinggi pada widget. Pada baris [13] `android:text` untuk menambahkan text pada widget. Pada baris [21] terdapat tag `<imageView>` untuk `imageView1`. Atribut `android:id` gunanya untuk menginisiasi id pada button tsb. Pada baris [23] dan [24] terdapat atribut untuk mengatur lebar dan tinggi pada `imageView1`. Pada baris [33] terdapat tag `<imageView>` untuk `imageView2`. Atribut `android:id` gunanya untuk menginisiasi id pada button tsb. Pada baris [35] dan [36] terdapat atribut untuk mengatur lebar dan tinggi pada `imageView2`.

Source Code MainActivity.kt

Pertama untuk kode kotlin class `MainActivity` didefinisikan dengan kata kunci `class`. Sistem android akan memanggil metode `onCreate()` dari `MainActivity`. metode `findViewById()` menemukan `Button` dalam tata letak. `R.id.button` adalah ID resource untuk `Button` yang merupakan ID unik untuknya. Kode menyimpan reference ke objek `Button` dalam variabel yang disebut `rollButton`, bukan objek `Button` itu sendiri. Pada baris [23] fungsi `rollDice` untuk menampilkan hasil pelemparan dadu. Pada baris [24] `val dice1 = Dice(6)` untuk membuat dadu dengan 6 sisi. Setelah itu memanggil metode `roll()` lalu hasilnya disimpan pada variable `diceRoll1` pada baris[25]. Pada baris [26] `val diceImage1: ImageView = findViewById(R.id.imageView1)` berfungsi untuk menampilkan gambar `imageView1`. Pada baris [28] `val dice2 = dice(6)` untuk membuat dadu dengan 6 sisi. Setelah itu memanggil metode `roll()` lalu hasilnya disimpan pada variable `diceRoll2` pada baris[29]. Pada baris [30] `val diceImage1: ImageView = findViewById(R.id.imageView2)` berfungsi untuk menampilkan gambar `imageView2`. Pada baris [32] menggunakan logika `If else`, jika `diceRoll1 == diceRoll2` maka output akan mengeluarkan “salamat”. Dan jika tidak maka output akan menghasilkan “maaf anda belum beruntung”. Pada baris ke [40] pada `dadu1` menggunakan logika `when`, jika user mendapat angka 1, maka tampilkan gambar `dice_1`. Jika user mendapat angka 2, maka tampilkan gambar `dice_2`. Jika user mendapat angka 3, maka tampilkan gambar `dice_3`. Jika user mendapatkan angka 4, maka tampilkan gambar `dice_4`. Jika user dapat angka 5, maka tampilkan gambar `dice_5`. `Else` tampilkan gambar `dice_6`. Pada baris ke [48] pada `dadu2` menggunakan logika `when`, jika user mendapat angka 1, maka tampilkan gambar `dice_1`. Jika user mendapat angka 2, maka tampilkan gambar `dice_2`. Jika user mendapat angka 3, maka tampilkan gambar `dice_3`. Jika user mendapatkan angka 4, maka tampilkan gambar `dice_4`. Jika user dapat angka 5, maka

tampilkan gambar dice\_5. Else tampilkan gambar dice\_6. Pada baris [56] kita memanggil `diceImage1.setImageResource(drawableResource1)` untuk menyimpan ID resource yang digunakan. Pada baris [57] `diceImage1.contentDescription = diceRoll.toString()` kita mendeskripsikan konten sesuai pada `imageView`. Pada baris [59] kita memanggil `diceImage2.setImageResource(drawableResource2)` untuk menyimpan ID resource yang digunakan. Pada baris [60] `diceImage2.contentDescription = diceRoll.toString()` kita mendeskripsikan konten sesuai pada `imageView`. Pada baris [64] membuat class `Dice` dengan menggunakan metode `roll()`. `numSides` sebagai `private` akan membuatnya hanya dapat diakses dalam class `Dice`. Karena satu-satunya kode yang akan menggunakan `numSides` ada di dalam class `Dice`, tidak masalah untuk membuat argumen ini `private` untuk class `Dice`.

## TAUTAN GIT

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Ryanifar/praktikumMobile2/tree/main/modul2>