

LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN MOBILE



Oleh:

Muhammad Firda Ryanifar NIM. 2010817310003

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

LEMBAR PENGESAHAN
LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN MOBILE

Laporan Akhir Praktikum Pemrograman Mobile 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	1
DAFTAR ISI	2
DAFTAR GAMBAR.....	3
DAFTAR TABEL	4
MODUL 1 INPUT, OUTPUT, DAN LOGIKA SEDERHANA	5
SOAL 1	6
A. Source Code	6
B. Output Program	6
C. Pembahasan.....	7
SOAL 2.....	7
A. Source Code	7
B. Output Program	7
C. Pembahasan.....	8
SOAL 3.....	8
A. Source Code	8
B. Output Program	9
C. Pembahasan.....	10
SOAL 4.....	10
A. Source Code	11
B. Output Program	11
C. Pembahasan.....	11
TAUTAN GIT	12
MODUL 2 ANDROID BASICS IN KOTLIN.....	13
SOAL.....	14
A. Source Code	16
B. Output Program	19
C. Pembahasan.....	21
TAUTAN GIT	22

DAFTAR GAMBAR

Gambar 1 Screenshot Hasil Jawaban Soal 1 Modul	6
Gambar 2 Screenshot Hasil Jawaban Soal 2 Modul 1	7
Gambar 3 Screenshot Hasil Jawaban Soal 3 Modul 1	9
Gambar 4 Screenshot Hasil Jawaban Soal 4 Modul 1	11
Gambar 5 Tampilan Awal Aplikasi (Modul 2)	14
Gambar 6 Tampilan Dadu Setelah Di Roll (Modul 2)	15
Gambar 7 Tampilan Roll Dadu Double (Modul 2)	16
Gambar 8 Screenshot Hasil Tampilan Awal Aplikasi (Modul 2)	19
Gambar 9 Screenshot Hasil Tampilan Setelah di Roll (Modul 2)	20
Gambar 10 Screenshot Hasil Tampilan Roll Dadu Double (Modul 2)	20

DAFTAR TABEL

Table 1 Source Code Soal 1 Modul 1Source Code	6
Table 2 Source Code Soal 2 Modul 1	7
Table 3 Source Code Soal 3 Modul 1	8
Table 4 Source Code Soal 4 Modul 1	11
Table 5 Source Code activity_main.xml (Modul 2)	16
Table 6 Source Code MainActivity.kt (Modul 2)	17

MODUL 1
INPUT, OUTPUT, DAN LOGIKA SEDERHANA



Oleh:

Muhammad Firda Ryanifar NIM. 2010817310003

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MARET 2022

SOAL 1

Buatlah sebuah program yang menerima input berupa waktu (Pagi, Siang, Sore, Malam), nama anda, umur anda, dan suhu tubuh anda (boleh random di range 35-37) kemudian mengeluarkan output sebagaimana dicontohkan dibawah ini

Input	Output
Waktu Sekarang: Pagi Nama Anda: Muhammad Alkaff Umur Anda: 35 Suhu Tubuh Anda: 36.4	Selamat Pagi, Muhammad Alkaff. Umur anda 35 Tahun. Suhu Tubuh anda 36.4 derajat Celcius.

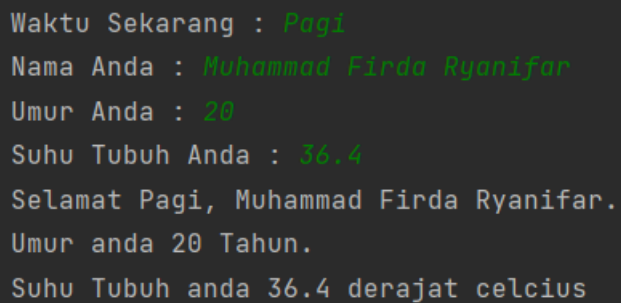
Simpan dengan nama file: PRAK101.kt dan push ke repository kalian masing-masing.

A. Source Code

Table 1 Source Code Soal 1 Modul 1 Source Code

```
1 package modul1
2
3 fun main(args: Array<String>){
4     print("Waktu Sekarang : "); var waktu = readLine()
5     print("Nama Anda : "); var nama = readLine()
6     print("Umur Anda : "); var umur = readLine()
7     print("Suhu Tubuh Anda : "); var suhu = readLine()
8
9     println("Selamat $waktu, $nama. \nUmur anda $umur
10    Tahun.\nSuhu Tubuh anda $suhu derajat celcius")
11 }
```

B. Output Program



```
Waktu Sekarang : Pagi
Nama Anda : Muhammad Firda Ryanifar
Umur Anda : 20
Suhu Tubuh Anda : 36.4
Selamat Pagi, Muhammad Firda Ryanifar.
Umur anda 20 Tahun.
Suhu Tubuh anda 36.4 derajat celcius
```

Gambar 1 Screenshot Hasil Jawaban Soal 1 Modul

C. Pembahasan

Pada baris [1], terdapat `package modul1`, digunakan untuk membuat package bernama `modul1`. Pada baris [3], terdapat `fun main(args: Array<String>)`, digunakan sebagai fungsi utama yang di dalamnya terdapat variabel yang menyimpan argumen. Pada baris [4], terdapat perintah `print`, yang digunakan untuk menampilkan teks ke layar. Pada baris [5], terdapat `var`, yang digunakan sebagai variabel mutable yang artinya variabelnya dapat berubah, selain itu juga terdapat `readLine()`, yang digunakan untuk mengambil inputan dengan tipe string.

SOAL 2

Buatlah sebuah program dengan menerapkan function (fungsi) untuk dapat menerjemahkan fungsi matematika dibawah ini kedalam sebuah kode program.

$$f(x) = 2x^2 + 5x - 8$$

Input	Output
Nilai x = 2	10
Input	Output
Nilai x = 4	44

Simpan dengan nama file: PRAK102.kt dan push ke repository kalian masing-masing.

A. Source Code

Table 2 Source Code Soal 2 Modul 1

1	<code>package modul1</code>
2	<code>import java.util.Scanner;</code>
3	
4	<code>fun matematika(x: Int): Int {</code>
5	<code> var fx = 2*(Math.pow(x.toDouble(), 2.0)</code>
6	<code>.toInt())+(5*x) - 8</code>
7	<code> return fx</code>
8	<code>}</code>
9	
10	<code>fun main(args: Array<String>) {</code>
11	<code> var input = Scanner(System.`in`)</code>
12	<code> print("Nilai x = ")</code>
13	<code> var x: Int = input.nextInt()</code>
14	<code> println(matematika(x))</code>
	<code>}</code>

B. Output Program

<code>"C:\Program Files"</code>	<code>"C:\Program</code>
<code>Nilai x = 2</code>	<code>Nilai x = 4</code>
<code>10</code>	<code>44</code>

Gambar 2 Screenshot Hasil Jawaban Soal 2 Modul 1

C. Pembahasan

Pada baris [4], terdapat `fun matematika(x: Int): Int`, digunakan sebagai fungsi bernama `matematika` memiliki parameter `x` bertipe `integer`. Pada baris [5], terdapat `var fx = 2*(Math.pow(x.toDouble(),2.0).toInt())+(5*x) - 8`, digunakan sebagai perhitungan dari $f(x) = 2x^2 + 5x - 8$, `Math.pow(x.toDouble(),2.0)`, digunakan untuk meningkatkan bilangan dengan tipe data keluarannya `double`, `toInt()`, digunakan untuk mengubah tipe data `double` ke tipe data `integer`. Pada baris [7], terdapat `return fx`, digunakan untuk mengembalikan nilai `fx`.

SOAL 3

Buatlah sebuah program yang dapat menampilkan 5 bilangan kelipatan dari bilangan yang telah diinput dan habis dibagi 2 atau 3.

Input	Output
Input bilangan = 5	10 15 20 30 40
Input	Output
Nilai bilangan = 7	14 21 28 42 56

Simpan dengan nama file: PRAK103.kt dan push ke repository kalian masing-masing.

A. Source Code

Table 3 Source Code Soal 3 Modul 1

```
1 package modul1
2 import java.util.Scanner
3
4 fun main(args: Array<String>) {
5     val input = Scanner(System.`in`)
6     print("Input Bilangan = ")
7     var bilangan = input.nextInt()
8     var loop: Int = 1
9     var hasil: Int = 0
10
11     while(loop < 6 ) {
12         hasil = hasil + bilangan
13         if (hasil % 2 == 0 || hasil % 3 == 0) {
14             print("$hasil ")
15             loop+=1
16         }
17     }
18 }
```

B. Output Program

```
Input Bilangan = 5  
10 15 20 30 40  
Process finished with exit code 0
```

```
Input Bilangan = 7  
14 21 28 42 56
```

Gambar 3 Screenshot Hasil Jawaban Soal 3 Modul 1

C. Pembahasan

Pada baris [2], terdapat `import java.util.Scanner`, digunakan agar dapat memasukkan fungsi scanner pada syntax. Pada baris [5], terdapat `Scanner(System.`in`)`, digunakan untuk membuat objek scanner baru. Pada baris [7], terdapat `input.nextInt()`, digunakan untuk mengambil inputan bertipe integer. Pada baris [8], terdapat `var loop: Int = 1`, digunakan untuk memberi inisialisasi pada variabel loop yang bertipe integer dengan nilai 1. Pada baris [9], terdapat `var hasil: Int = 0`, digunakan untuk menginisialisasi variabel hasil yang bertipe integer dengan nilai 0. Pada baris [11], terdapat `while(loop < 6)`, digunakan sebagai perulangan dimana selama nilai dari loop kurang dari 6, maka jalankan program secara berulang. Pada baris [13], terdapat `hasil = hasil + bilangan`, artinya hasil memiliki nilai hasil itu sendiri + bilangan yang diinputkan. Pada baris [13], terdapat `if (hasil % 2 == 0 || hasil % 3 == 0)`, digunakan sebagai kondisional apabila variabel hasil habis dibagi 2 atau 3, maka akan menampilkan hasilnya. Pada baris [15], terdapat `loop+=1`, dimana perulangan akan terus berjalan loop + 1, apabila loop berjalan sampai kurang dari 6 atau sama dengan 5, maka perulangan akan stop.

SOAL 4

Buatlah sebuah program dengan menerapkan konsep Class dan Object pada Kotlin. Class dan Object yang dibuat disesuaikan dengan ketentuan berikut:

- NIM dengan akhiran 0/1 : Class dan Object dengan Tema **Pekerjaan**
- NIM dengan akhiran 2/3 : Class dan Object dengan Tema **Otomotif**
- NIM dengan akhiran 4/5 : Class dan Object dengan Tema **Olahraga**
- NIM dengan akhiran 6/7 : Class dan Object dengan Tema **Hewan**
- NIM dengan akhiran 8/9 : Class dan Object dengan Tema **Teknologi**

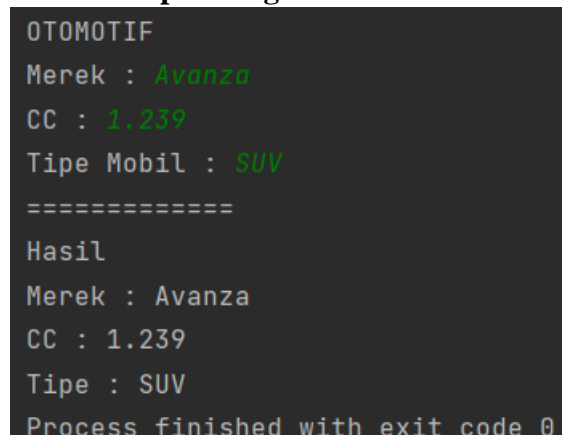
Simpan dengan nama file: PRAK104.kt dan push ke repository kalian masing-masing.

A. Source Code

Table 4 Source Code Soal 4 Modul 1

```
1 package modul1
2
3 class Mobil(merek:String, cc:String, tipe:String){
4     var merek : String = merek
5     var cc : String = cc
6     var tipe : String = tipe
7 }
8
9 fun main(){
10     println("OTOMOTIF")
11     print("Merek : ")
12     val merek = readLine() ?:"Toyota Avanza"
13     print("CC : ")
14     val cc = readLine() ?:"1.239"
15     print("Tipe Mobil : ")
16     val tipe = readLine() ?:"SUV"
17     val call = Mobil(merek,cc,tipe)
18     val hasil = ""
19         =====
20     Hasil
21     Merek : ${call.merek}
22     CC : ${call.cc}
23     Tipe : ${call.tipe}
24     """.trimIndent()
25     print(hasil)
26 }
```

B. Output Program



```
OTOMOTIF
Merek : Avanza
CC : 1.239
Tipe Mobil : SUV
=====
Hasil
Merek : Avanza
CC : 1.239
Tipe : SUV
Process finished with exit code 0
```

Gambar 4 Screenshot Hasil Jawaban Soal 4 Modul 1

C. Pembahasan

Pada baris [3], terdapat `class Mobil(merek:String, cc:String, tipe:String)`, digunakan untuk membuat class yang memiliki paramater merek, cc,

dan tipe yang bertipe String. Pada baris [4]terdapat `var merek : String = merek`, digunakan sebagai property pada class, `val call = Mobil(merek,cc,tipe)`, digunakan untuk membuat objek baru yaitu call dari class Mobil. Pada baris [21], terdapat `${call.merek}`, digunakan untuk memanggil objek nama pada class kucing.

TAUTAN GIT

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Ryanifar/praktikumMobile2/tree/main/modul1>

MODUL 2
ANDROID BASICS IN KOTLIN



Oleh:

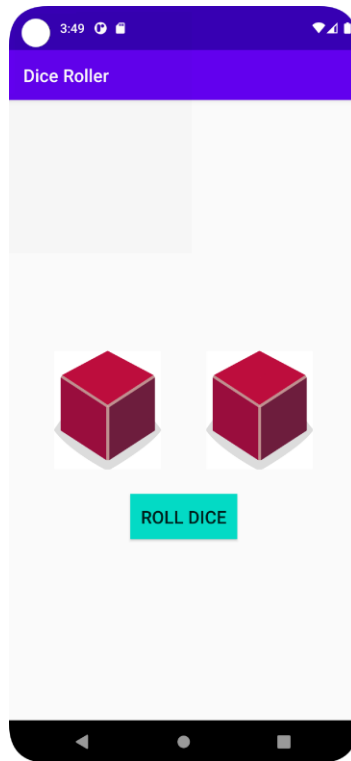
Muhammad Firda Ryanifar NIM. 2010817120007

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MARET 2022

SOAL

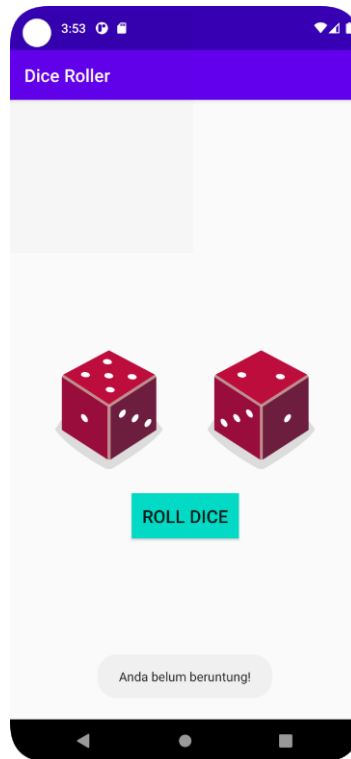
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



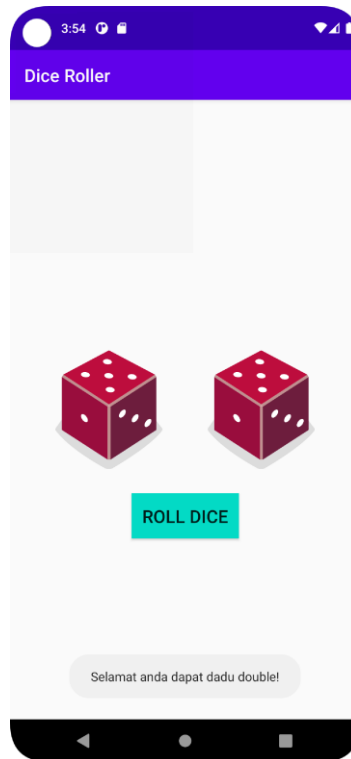
Gambar 5 Tampilan Awal Aplikasi (Modul 2)

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 6 Tampilan Dadu Setelah Di Roll (Modul 2)

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



Gambar 7 Tampilan Roll Dadu Double (Modul 2)

A. Source Code

Table 5 Source Code activity_main.xml (Modul 2)

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 6 Source Code MainActivity.kt (Modul 2)

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	override fun onCreate(savedInstanceState: Bundle?) {
11	super.onCreate(savedInstanceState)
12	setContentView(R.layout.activity_main)
13	
14	

```

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

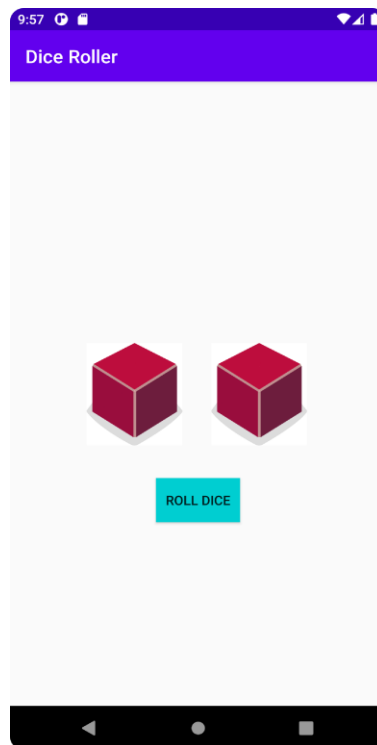
```

```

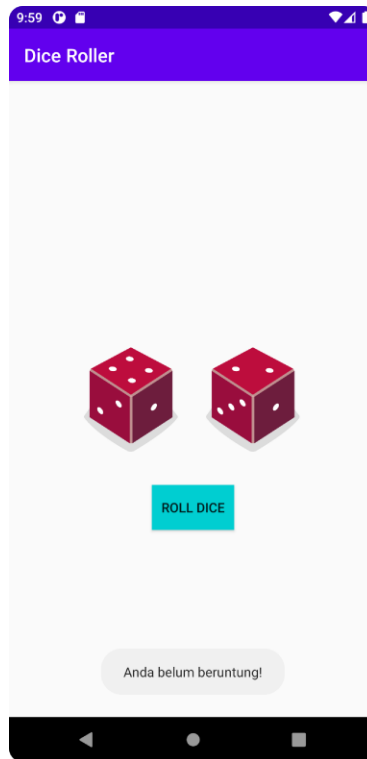
62         diceImage1.contentDescription =
63         diceRoll1.toString()
64         diceImage2.contentDescription =
65         diceRoll2.toString()
66
67         if (diceRoll1 == diceRoll2) {
68             Toast.makeText(applicationContext, "Selamat
69 anda dapat dadu double!", Toast.LENGTH_LONG).show()
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 }

```

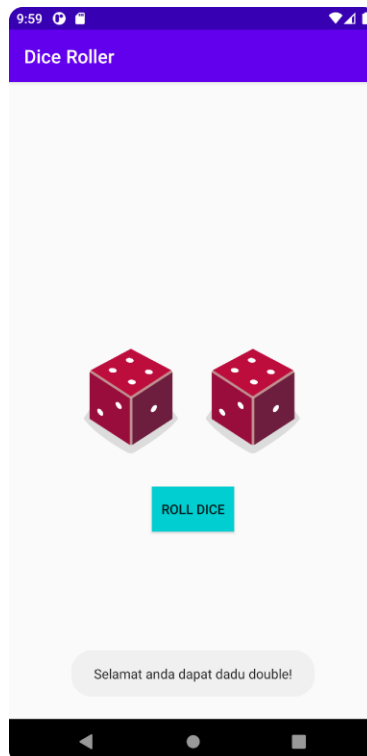
B. Output Program



Gambar 8 Screenshot Hasil Tampilan Awal Aplikasi (Modul 2)



Gambar 9 Screenshot Hasil Tampilan Setelah di Roll (Modul 2)



Gambar 10 Screenshot Hasil Tampilan Roll Dadu Double (Modul 2)

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 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>