

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 1**



**ANDROID BASIC WITH KOTLIN**

**Oleh:**

**Muhammad Erza Raihan NIM. 2310817210027**

**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 Erza Raihan  
NIM : 2310817210027

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Muhammad Raka Azwar  
NIM. 2210817210012

Andreyan Rizky Baskara, S.Kom., M.Kom.  
NIP. 19930703 201903 01 011

## DAFTAR ISI

LEMBAR PENGESAHAN .....	2
DAFTAR ISI .....	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL .....	5
SOAL 1 .....	6
9	
B.	15
C.	16
D.	<b>Error! Bookmark not defined.</b>

## **DAFTAR GAMBAR**

Gambar 1. Screenshot Hasil Jawaban Soal 1

11

## **DAFTAR TABEL**

Tabel 1. Source Code Jawaban Soal 1

Tabel 2. Source Code Jawaban Soal 1

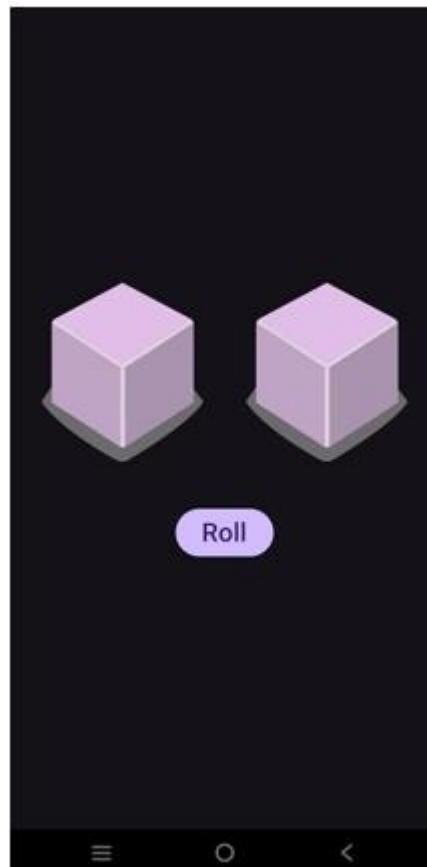
10

## 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”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

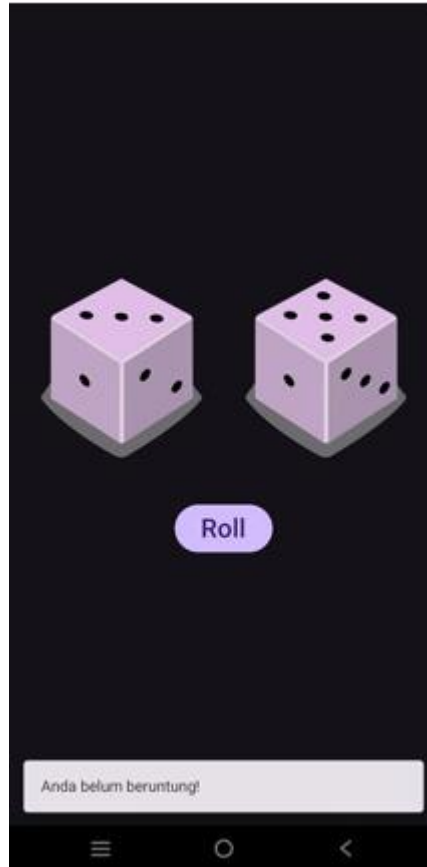
1. Tampilan awal aplikasi setelah dijalankan akan menampilkan 2 buah dadu kosong seperti dapat dilihat pada Gambar 1.



*Gambar 1. Tampilan Awal Aplikasi*

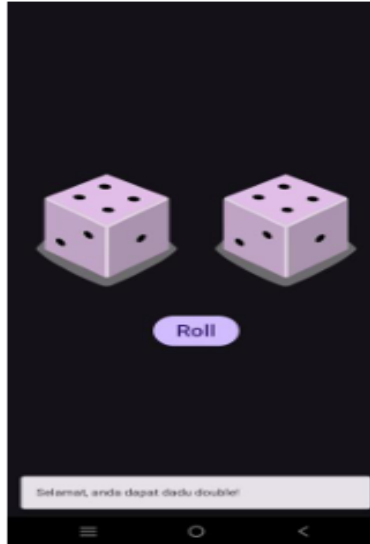
2. Setelah user menekan tombol “Roll” maka masing-masing dadu akan memperlihatkan sisi dadu masing-masing dengan angka antara 1 s/d 6. Apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2 maka aplikasi

akan menampilkan pesan “Anda belum beruntung!” seperti dapat dilihat pada Gambar 2.



*Gambar 2. Tampilan Dadu Setelah Di-Roll*

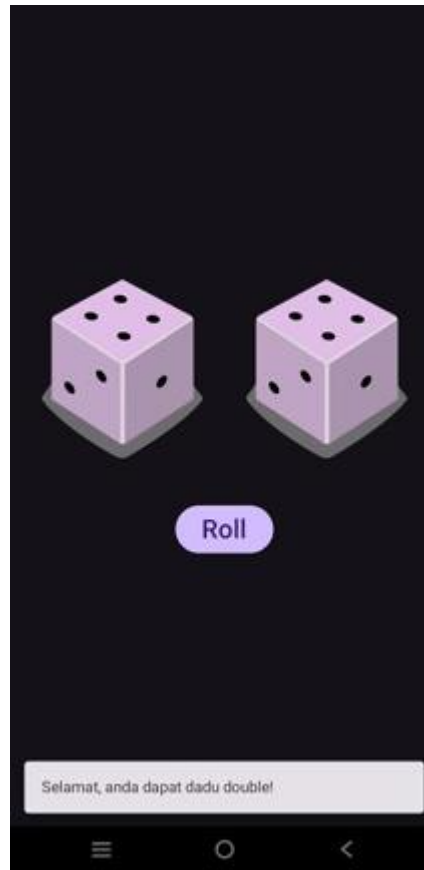
3. Apabila user mendapatkan nilai dadu yang sama antara Dadu 1 dan Dadu 2 atau nilai double, maka aplikasi akan menampilkan pesan “Selamat, anda dapat dadu double!” seperti yang dapat dilihat pada Gambar 3.



*Gambar 3. Tampilan Roll Dadu Double*

4. Buatlah aplikasi tersebut menggunakan XML dan Jetpack Compose.
5. Upload aplikasi yang telah anda buat ke dalam repository GitHub ke dalam folder Modul 1 dalam bentuk Project. Jangan lupa untuk melakukan Clean Project sebelum mengupload pekerjaan anda pada repository.
6. Untuk gambar dadu dapat didownload pada link berikut:  
[https://drive.google.com/file/d/14V3qXGdFnuoYN4AGd\\_9SgFh8kw8X9ySm/view?usp=sharing%20%20](https://drive.google.com/file/d/14V3qXGdFnuoYN4AGd_9SgFh8kw8X9ySm/view?usp=sharing%20%20)





## A. Source Code

### MainActivity.kt XML

```
1 package com.example.modul1xml
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 class MainActivity : AppCompatActivity() {
9
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         val diceImage01: ImageView =
15         findViewById(R.id.imageView)
16         diceImage01.setImageResource(R.drawable.dice_1)
17     }
```

```

18
19         val         diceImage02:         ImageView         =
20 findViewById(R.id.imageView2)
21         diceImage02.setImageResource(R.drawable.dice_1)
22
23         val rollButton: Button = findViewById(R.id.button)
24
25         rollButton.setOnClickListener { rollDice() }
26     }
27
28     private fun rollDice() {
29         val dice1 = Dice(6)
30         val diceRoll1 = dice1.roll()
31
32
33         val         diceImage1:         ImageView         =
34 findViewById(R.id.imageView)
35         when (diceRoll1) {
36             1
37             1
38 diceImage1.setImageResource(R.drawable.dice_1)
39             2
40 diceImage1.setImageResource(R.drawable.dice_2)
41             3
42 diceImage1.setImageResource(R.drawable.dice_3)
43             4
44 diceImage1.setImageResource(R.drawable.dice_4)
45             5
46 diceImage1.setImageResource(R.drawable.dice_5)
47             6
48 diceImage1.setImageResource(R.drawable.dice_6)
49         }
50
51         val dice2 = Dice(6)
52         val diceRoll2 = dice2.roll()
53
54
55         val         diceImage2:         ImageView         =
56 findViewById(R.id.imageView2)
57         when (diceRoll2) {
58             1
59 diceImage2.setImageResource(R.drawable.dice_1)
60             2
61 diceImage2.setImageResource(R.drawable.dice_2)
62             3
63 diceImage2.setImageResource(R.drawable.dice_3)
64             4
65 diceImage2.setImageResource(R.drawable.dice_4)
66             5
67 diceImage2.setImageResource(R.drawable.dice_5)
68
69

```

70	6	->
71	diceImage2.setImageResource(R.drawable.dice_6)	
72	}	
73		
74	if (diceRoll1 == diceRoll2){	
75	val toast = Toast.makeText(this, "Selamat anda	
76	dapat dadu double!", Toast.LENGTH_SHORT)	
77	toast.show()	
78	} else {	
79	val toast = Toast.makeText(this, "Anda belum	
80	beruntung!", Toast.LENGTH_SHORT)	
81	toast.show() }	
82		
83		
84	}	
85	}	
86		
87	class Dice(private val numSides: Int) {	
88		
89	fun roll(): Int {	
90	return (1..numSides).random()	
91		
92	}	
93	}	
94		
95		

Tabel 1. Source Code Mainactivity XML Soal 1

### MainActivity.kt Compose

1	package com.example.praktikum1
2	
3	import android.os.Bundle
4	import androidx.activity.ComponentActivity
5	import androidx.activity.compose.setContent
6	import androidx.compose.foundation.Image
7	import androidx.compose.foundation.layout.Column
8	import androidx.compose.foundation.layout.fillMaxSize
9	import androidx.compose.foundation.layout.wrapContentSize
10	import androidx.compose.material3.Text
11	import androidx.compose.runtime.Composable
12	import androidx.compose.ui.Alignment
13	import androidx.compose.ui.Modifier
14	import androidx.compose.ui.res.painterResource
15	import androidx.compose.ui.tooling.preview.Preview
16	import com.example.praktikum1.ui.theme.PRAKTIKUM1Theme
17	import androidx.compose.foundation.layout.height
18	import androidx.compose.foundation.layout.Spacer
19	

```

20 import androidx.compose.ui.unit.dp
21 import androidx.compose.material3.Button
22 import androidx.compose.ui.res.stringResource
23 import androidx.compose.runtime.getValue
24 import androidx.compose.runtime.remember
25 import androidx.compose.runtime.setValue
26 import androidx.compose.runtime.mutableStateOf
27
28
29 class MainActivity : ComponentActivity() {
30     override fun onCreate(savedInstanceState: Bundle?) {
31         super.onCreate(savedInstanceState)
32         setContent {
33             PRAKTIKUM1Theme {
34                 DiceRollerApp()
35             }
36         }
37     }
38 }
39
40
41 @Preview()
42 @Composable
43 fun DiceRollerApp() {
44     DiceWithButtonAndImage()
45 }
46
47
48
49 @Composable
50 fun DiceWithButtonAndImage(modifier: Modifier = Modifier
51     .fillMaxSize()
52     .wrapContentSize(Alignment.Center)) {
53
54     var result1 by remember { mutableStateOf(1) }
55     var result2 by remember { mutableStateOf(1) }
56     var message by remember { mutableStateOf("") }
57
58
59     val imageResource1 = when (result1) {
60         1 -> R.drawable.dice_1
61         2 -> R.drawable.dice_2
62         3 -> R.drawable.dice_3
63         4 -> R.drawable.dice_4
64         5 -> R.drawable.dice_5
65         else -> R.drawable.dice_6
66     }
67
68
69     val imageResource2 = when (result2) {
70         1 -> R.drawable.dice_1
71

```

```

72         2 -> R.drawable.dice_2
73         3 -> R.drawable.dice_3
74         4 -> R.drawable.dice_4
75         5 -> R.drawable.dice_5
76         else -> R.drawable.dice_6
77     }
78
79     Column (
80         modifier = modifier,
81         horizontalAlignment
82 Alignment.CenterHorizontally
83     ) {
84         androidx.compose.foundation.layout.Row {
85             Image(
86                 painter
87                 painterResource(imageResource1),
88                 contentDescription = result1.toString()
89             )
90
91             Spacer(modifier = Modifier.height(8.dp))
92
93             Image(
94                 painter
95                 painterResource(imageResource2),
96                 contentDescription = result2.toString()
97             )
98         }
99
100     }
101
102
103     Spacer(modifier = Modifier.height(16.dp))
104
105     val doubleMessage
106 stringResource(R.string.double_message)
107     val badMessage
108 stringResource(R.string.bad_message)
109
110     Button(onClick = {
111         result1 = (1..6).random()
112         result2 = (1..6).random()
113
114         message = if (result1 == result2) {
115             doubleMessage
116         } else {
117             badMessage
118         }
119     }) {
120         Text(stringResource(R.string.roll))
121     }
122
123

```

124	}
125	
126	Spacer(modifier = Modifier.height(16.dp))
127	
128	Text(text = message)
129	
130	
131	}}
132	
133	

*Tabel 2. Source Code MainActivity Compose Soal 1*

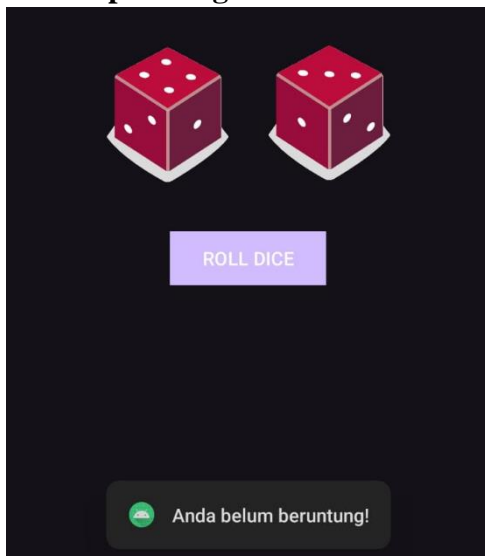
### **activity\_main.xml**

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
3	xmlns:android="http://schemas.android.com/apk/res/android
4	"
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	
12	<Button
13	android:id="@+id/button"
14	android:layout_width="wrap_content"
15	android:layout_height="wrap_content"
16	android:layout_marginTop="52dp"
17	android:text="ROLL"
18	app:layout_constraintEnd_toEndOf="parent"
19	app:layout_constraintHorizontal_bias="0.498"
20	app:layout_constraintStart_toStartOf="parent"
21	
22	
23	app:layout_constraintTop_toBottomOf="@+id/imageView" />
24	
25	<ImageView
26	android:id="@+id/imageView"
27	android:layout_width="150dp"
28	android:layout_height="150dp"
29	app:layout_constraintBottom_toBottomOf="parent"
30	app:layout_constraintEnd_toEndOf="parent"
31	app:layout_constraintHorizontal_bias="0.201"
32	app:layout_constraintStart_toStartOf="parent"
33	

34	app:layout_constraintTop_toTopOf="parent"
35	app:layout_constraintVertical_bias="0.449"
36	/>
37	
38	<ImageView
39	android:id="@+id/imageView2"
40	android:layout_width="150dp"
41	android:layout_height="150dp"
42	app:layout_constraintBottom_toBottomOf="parent"
43	app:layout_constraintEnd_toEndOf="parent"
44	app:layout_constraintHorizontal_bias="0.837"
45	app:layout_constraintStart_toStartOf="parent"
46	app:layout_constraintTop_toTopOf="parent"
47	app:layout_constraintVertical_bias="0.449"
48	/>
49	
50	
51	</androidx.constraintlayout.widget.ConstraintLayout>
52	

Tabel 3. Source Code activity\_main XML Soal 1

## B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

## C. Pembahasan

### MainActivity.kt XML:

Kode ini merupakan implementasi aplikasi Android sederhana yang menampilkan dua gambar dadu dan sebuah tombol untuk "mengacak" atau **roll** kedua dadu tersebut. Proyek ini menggunakan Kotlin dengan antarmuka XML pada `activity_main.xml`. Tujuannya adalah untuk memberikan visualisasi pelemparan dua buah dadu secara acak serta menampilkan notifikasi (Toast) ketika kedua dadu menunjukkan angka yang sama (double).

Pada bagian `onCreate`, komponen UI dari layout diambil dan diatur. Dua `ImageView` (`imageView` dan `imageView2`) digunakan untuk menampilkan gambar sisi dadu, dan satu `Button` (`button`) digunakan untuk memicu aksi roll dadu. Saat aplikasi dibuka, kedua gambar dadu diinisialisasi terlebih dahulu dengan gambar dadu angka satu (`dice_1`). Kemudian, saat tombol ditekan, fungsi `rollDice()` akan dipanggil.

Fungsi `rollDice()` bertanggung jawab untuk mengacak angka pada masing-masing dadu menggunakan kelas `Dice`, yang menerima parameter jumlah sisi dadu (dalam hal ini 6). Setelah angka dadu diacak dengan metode `.roll()`, hasilnya digunakan dalam blok `when` untuk mengganti gambar pada masing-masing `ImageView`, menyesuaikan dengan angka yang didapat (misalnya jika hasilnya 3, maka akan ditampilkan gambar `dice_3`).

Terakhir, jika kedua dadu menunjukkan angka yang sama, maka aplikasi akan menampilkan pesan Toast yang menyatakan bahwa pengguna mendapatkan dadu double. Jika tidak sama, maka Toast akan memberitahukan bahwa pengguna belum beruntung. Ini memberikan elemen permainan sederhana dalam aplikasi.



### **activity\_main.xml:**

File XML adalah layout antarmuka pengguna (UI layout) untuk aplikasi Android yang menggunakan ConstraintLayout sebagai root element. ConstraintLayout memungkinkan kita menata elemen-elemen UI secara fleksibel dengan menetapkan posisi relatif antar elemen atau terhadap parent-nya, sehingga lebih efisien dan adaptif terhadap berbagai ukuran layar.

Dalam layout ini, terdapat **tiga komponen utama**: dua ImageView dan satu Button. Kedua ImageView digunakan untuk menampilkan gambar dadu, masing-masing memiliki ID imageView dan imageView2. Masing-masing dadu diberikan ukuran tetap sebesar 150dp baik lebar maupun tingginya. Posisi keduanya diatur agar sejajar secara vertikal di tengah layar, namun berbeda secara horizontal—yang satu lebih ke kiri (`horizontal_bias="0.201"`), yang satu lagi ke kanan (`horizontal_bias="0.837"`), sehingga terlihat seperti dua dadu berdampingan.

Di bawah kedua ImageView ini terdapat sebuah Button dengan ID button dan teks **"ROLL"**. Tombol ini ditempatkan di bawah imageView (dadu kiri) dengan `layout_constraintTop_toBottomOf="@+id/imageView"` dan diberi margin atas sebesar 52dp agar tidak terlalu menempel. Tombol juga diposisikan di tengah secara horizontal dengan `layout_constraintStart_toStartOf="parent"` dan `layout_constraintEnd_toEndOf="parent"` serta `horizontal_bias="0.498"` (hampir di tengah).

Penggunaan `tools:context=".MainActivity"` pada root ConstraintLayout adalah untuk memberi tahu Android Studio bahwa layout ini akan digunakan oleh MainActivity, sehingga saat desain dilihat di editor, bisa disimulasikan dengan konteks aktivitas yang tepat.

### **MainActivity.kt Compose:**

Bagian utama dari aplikasi dimulai di dalam kelas MainActivity, yang mewarisi ComponentActivity. Di dalam metode onCreate, fungsi setContent digunakan untuk menampilkan UI berbasis Compose dengan tema PRAKTIKUM1Theme. Komponen utama aplikasi dipanggil melalui fungsi DiceRollerApp, yang akan memanggil DiceWithButtonAndImage()—sebuah composable function yang berisi keseluruhan tampilan dan logika utama aplikasi.

Fungsi DiceWithButtonAndImage menggunakan layout Column untuk menata elemen-elemen UI secara vertikal dan berada di tengah layar. Di dalamnya terdapat dua Image untuk menampilkan gambar hasil lemparan dadu, sebuah Button untuk melakukan aksi lempar, dan satu Text untuk menampilkan pesan hasil lemparan. Gambar dadu dipilih menggunakan struktur when berdasarkan nilai result1 dan result2, yang diinisialisasi dengan nilai 1 dan dapat berubah setiap kali tombol ditekan, karena dibungkus dalam remember dan mutableStateOf. Ini memastikan bahwa UI akan merespons secara otomatis saat nilainya berubah.

Ketika pengguna menekan tombol ROLL, nilai result1 dan result2 akan diperbarui secara acak antara 1 hingga 6. Jika kedua nilai tersebut sama (double), maka pesan yang ditampilkan berasal dari stringResource(R.string.double\_message); jika tidak sama, maka pesan dari bad\_message akan ditampilkan. Penggunaan stringResource() membuat aplikasi ini lebih fleksibel dan mendukung internasionalisasi (misalnya, mudah untuk menerjemahkan ke bahasa lain).

### **D. Tautan Git**

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/awan-ama/praktikummobile/tree/main/DiceRoller>