

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 1**



**ANDROID BASIC WITH KOTLIN**

**Oleh:**

**Muhammad Bukhari Fitri NIM. 2310817210015**

**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 Bukhari Fitri  
NIM : 2310817210015

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
A. Source Code.....	8
B. Output Program .....	14
C. Pembahasan .....	19
D. Tautan Git .....	25

## DAFTAR GAMBAR

Gambar 1. Tampilan Awal Aplikasi.....	6
Gambar 2. Tampilan Dadu Setelah Di-Roll .....	7
Gambar 3. Tampilan Roll Dadu Double.....	8
Gambar 4. Screenshot Hasil Jawaban Soal 1 Tampilan Awal Aplikasi XML.....	14
Gambar 5. Screenshot Hasil Jawaban Soal 1 Tampilan Dadu Setelah Di-Roll XML.....	15
Gambar 6. Screenshot Hasil Jawaban Soal 1 Tampilan Roll Dadu Double XML.....	16
Gambar 7. Screenshot Hasil Jawaban Soal 1 Tampilan Awal Aplikasi Compose.....	17
Gambar 8. Screenshot Hasil Jawaban Soal 1 Tampilan Dadu Setelah Di-Roll Compose ...	18
Gambar 9. Screenshot Hasil Jawaban Soal 1 Tampilan Roll Dadu Double Compose.....	19

## DAFTAR TABEL

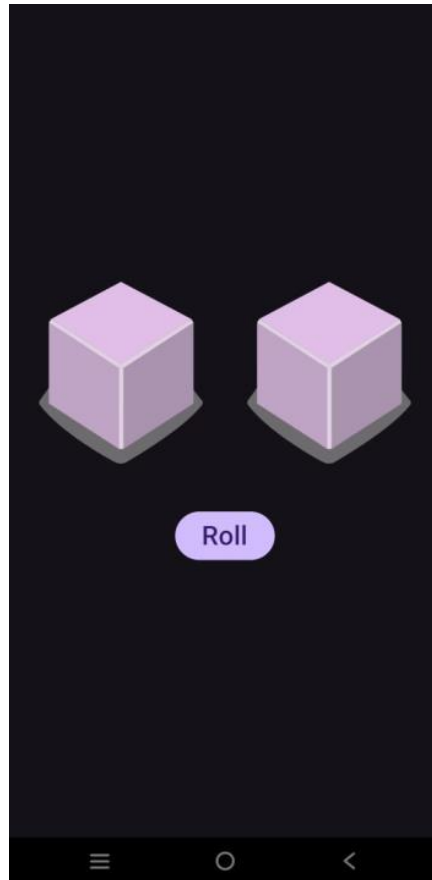
Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt XML .....	9
Tabel 2. Source Code Jawaban Soal 1 DiceRollerApp.kt XML .....	10
Tabel 3. Source Code Jawaban Soal 1 activity_main.xml XML .....	11
Tabel 4. Source Code Jawaban Soal 1 MainActivity.kt Compose .....	11
Tabel 5. Source Code Jawaban Soal 1 DiceRollerApp.kt Compose .....	12
Tabel 6. Source Code Jawaban Soal 1 DiceWithButtonAndImage.kt Compose .....	14

## SOAL 1

### Soal Praktikum:

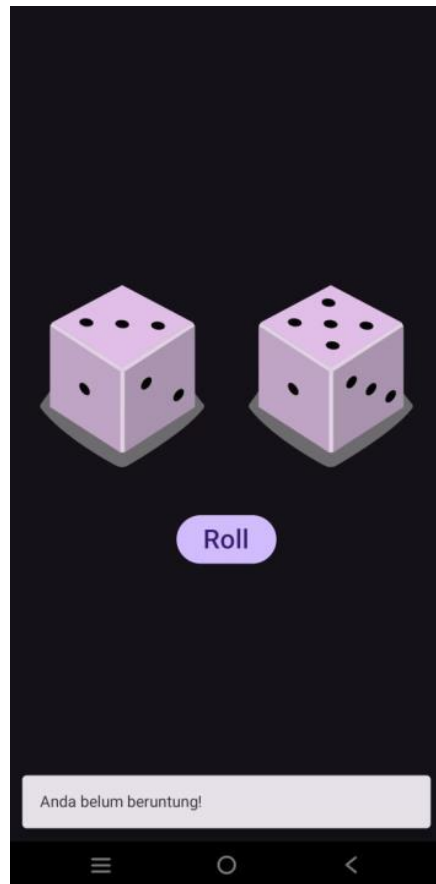
Buatlah sebuah aplikasi yang dapat menampilkan 2 buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



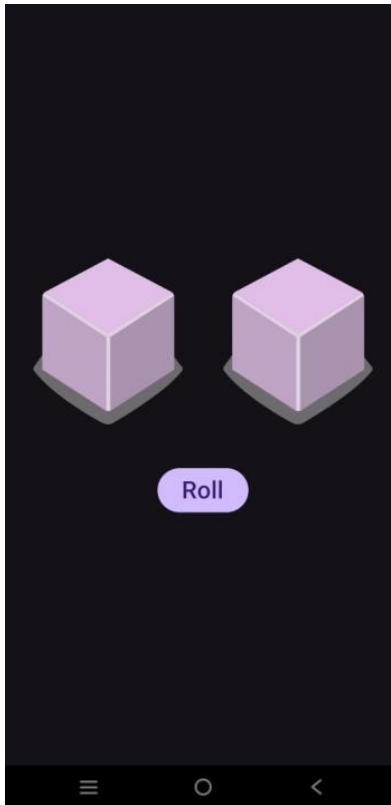
*Gambar 1. Tampilan Awal Aplikasi*

2. Setelah user menekan tombol “Roll” maka masing-masing dadu akan memperlihatkan sisi dadunya 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 yang 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](https://drive.google.com/file/d/14V3qXGdFnuoYN4AGd_9SgFh8kw8X9ySm/view?usp=sharing)

## A. Source Code

### XML:

### MainActivity.kt

```
1 package com.example.dicexml
2
3 import android.os.Bundle
4 import androidx.activity.enableEdgeToEdge
5 import androidx.appcompat.app.AppCompatActivity
6 import com.example.dicexml.databinding.ActivityMainBinding
7
8
9 class MainActivity : AppCompatActivity() {
```



```

10     private lateinit var binding: ActivityMainBinding
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         enableEdgeToEdge()
14
15         binding = ActivityMainBinding.inflate(layoutInflater)
16         setContentView(binding.root)
17
18         binding.button.setOnClickListener {
19             diceRollerApp(binding)
20         }
21     }
22 }

```

Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt XML

### DiceRollerApp.kt

```

1 package com.example.dicexml
2
3 import com.example.dicexml.databinding.ActivityMainBinding
4 import com.google.android.material.snackbar.Snackbar
5
6 fun diceRollerApp(binding: ActivityMainBinding) {
7     val result1 = randomDice()
8     val result2 = randomDice()
9
10    val imageResource1 = getDiceImage(result1)
11    val imageResource2 = getDiceImage(result2)
12
13    binding.imageView.setImageResource(imageResource1)
14    binding.imageView2.setImageResource(imageResource2)
15
16    if (result1 == result2) {
17        Snackbar.make(binding.button, "Selamat, anda dapat dadu
double", Snackbar.LENGTH_LONG).show()
18    }
19    else if (result1 != result2) {
20        Snackbar.make(binding.button, "Anda belum beruntung!",
Snackbar.LENGTH_LONG).show()
21    }
22 }
23
24 fun randomDice() : Int {
25     return (1..6).random()
26 }
27
28 fun getDiceImage(result: Int) : Int {
29     return when (result) {
30         1 -> R.drawable.dice_1
31         2 -> R.drawable.dice_2
32         3 -> R.drawable.dice_3
33         4 -> R.drawable.dice_4

```

34	5 -> R.drawable.dice_5
35	else -> R.drawable.dice_6
36	}
37	}

Tabel 2. Source Code Jawaban Soal 1 DiceRollerApp.kt XML

### 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	android:background="#2B2A2A"
9	tools:context=".MainActivity">
10	
11	<Button
12	android:id="@+id/button"
13	android:layout_width="wrap_content"
14	android:layout_height="wrap_content"
15	android:layout_marginStart="162dp"
16	android:layout_marginTop="450dp"
17	android:layout_marginEnd="162dp"
18	android:layout_marginBottom="234dp"
19	android:backgroundTint="#D460ED"
20	android:text="@string/roll"
21	app:layout_constraintBottom_toBottomOf="parent"
22	app:layout_constraintEnd_toEndOf="parent"
23	app:layout_constraintStart_toStartOf="parent"
24	app:layout_constraintTop_toTopOf="parent" />
25	
26	
27	<ImageView
28	android:id="@+id/imageView"
29	android:layout_width="wrap_content"
30	android:layout_height="wrap_content"
31	android:layout_marginTop="171dp"
32	android:layout_marginEnd="4dp"
33	android:layout_marginBottom="29dp"
34	android:contentDescription="@string/image_description"
35	app:layout_constraintBottom_toTopOf="@+id/button"
36	app:layout_constraintEnd_toEndOf="parent"
37	app:layout_constraintStart_toEndOf="@+id/imageView2"
38	app:layout_constraintTop_toTopOf="parent"
39	app:srcCompat="@drawable/dice_0" />
40	
41	<ImageView
42	android:id="@+id/imageView2"
43	android:layout_width="wrap_content"

44	android:layout_height="wrap_content"
45	android:layout_marginTop="171dp"
46	android:layout_marginBottom="29dp"
47	android:contentDescription="@string/image_description"
48	app:layout_constraintBottom_toTopOf="@+id/button"
49	app:layout_constraintEnd_toStartOf="@+id/imageView"
50	app:layout_constraintStart_toStartOf="parent"
51	app:layout_constraintTop_toTopOf="parent"
52	app:srcCompat="@drawable/dice_0" />
53	</androidx.constraintlayout.widget.ConstraintLayout>

Tabel 3. Source Code Jawaban Soal 1 activity\_main.xml XML

## Compose:

### MainActivity.kt

1	package com.example.dicecompose
2	
3	import android.os.Bundle
4	import androidx.activity.ComponentActivity
5	import androidx.activity.compose.setContent
6	import androidx.activity.enableEdgeToEdge
7	import androidx.compose.foundation.layout.fillMaxSize
8	import androidx.compose.material3.Scaffold
9	import androidx.compose.ui.Modifier
10	import com.example.dicecompose.component.DiceRollerApp
11	import com.example.dicecompose.ui.theme.DiceComposeTheme
12	
13	class MainActivity : ComponentActivity() {
14	override fun onCreate(savedInstanceState: Bundle?) {
15	super.onCreate(savedInstanceState)
16	enableEdgeToEdge()
17	setContent {
18	DiceComposeTheme {
19	Scaffold(modifier = Modifier.fillMaxSize()) {
20	{ innerPadding ->
21	DiceRollerApp()
22	}
23	}
24	}
25	}
26	}

Tabel 4. Source Code Jawaban Soal 1 MainActivity.kt Compose

### DiceRollerApp.kt

1	package com.example.dicecompose.component
2	
3	import androidx.compose.foundation.layout.*
4	import androidx.compose.runtime.Composable
5	import androidx.compose.ui.Alignment
6	import androidx.compose.ui.Modifier

```

7 import androidx.compose.ui.tooling.preview.Preview
8
9 @Composable
10 fun DiceRollerApp() {
11     DiceWithButtonAndImage(
12         modifier = Modifier
13             .fillMaxSize()
14             .wrapContentSize(Alignment.Center)
15     )
16 }
17
18 @Preview(showBackground = true)
19 @Composable
20 fun DiceRollerAppPreview() {
21     DiceRollerApp()
22 }

```

*Tabel 5. Source Code Jawaban Soal 1 DiceRollerApp.kt Compose*

### **DiceWithButtonAndImage.kt**

```

1 package com.example.dicecompose.component
2
3 import androidx.compose.foundation.Image
4 import androidx.compose.foundation.layout.Column
5 import androidx.compose.foundation.layout.Row
6 import androidx.compose.foundation.layout.Spacer
7 import androidx.compose.foundation.layout.height
8 import androidx.compose.material3.Button
9 import androidx.compose.material3.Scaffold
10 import androidx.compose.material3.SnackbarDuration
11 import androidx.compose.material3.SnackbarHost
12 import androidx.compose.material3.SnackbarHostState
13 import androidx.compose.material3.Text
14 import androidx.compose.runtime.Composable
15 import androidx.compose.runtime.getValue
16 import androidx.compose.runtime.mutableStateOf
17 import androidx.compose.runtime.remember
18 import androidx.compose.runtime.rememberCoroutineScope
19 import androidx.compose.runtime.setValue
20 import androidx.compose.ui.Alignment
21 import androidx.compose.ui.Modifier
22 import androidx.compose.ui.res.painterResource
23 import androidx.compose.ui.res.stringResource
24 import androidx.compose.ui.unit.dp
25 import com.example.dicecompose.R
26
27 import kotlinx.coroutines.launch
28
29 @Composable
30 fun DiceWithButtonAndImage(modifier: Modifier = Modifier) {
31     var result1 by remember { mutableStateOf(0) }
32     var result2 by remember { mutableStateOf(0) }

```

```

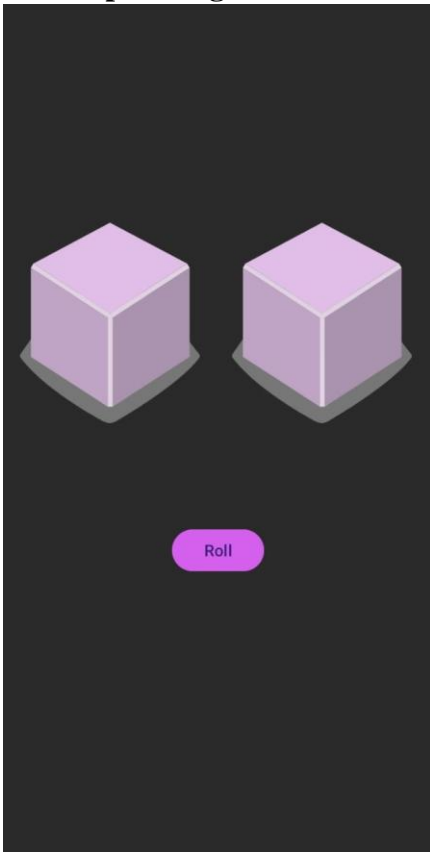
33     val imageResource1 = getDiceImage(result1)
34     val imageResource2 = getDiceImage(result2)
35
36     val scope = rememberCoroutineScope()
37     val snackbarHostState = remember { SnackbarHostState() }
38
39     Scaffold(
40         snackbarHost = { SnackbarHost(hostState =
snackbarHostState) }
41     ) {
42         paddingValues ->
43         Column (
44             modifier = modifier,
45             horizontalAlignment = Alignment.CenterHorizontally
46         ) {
47             Row {
48                 Image(
49                     painter = painterResource(imageResource1),
50                     contentDescription = result1.toString()
51                 )
52                 Image(
53                     painter = painterResource(imageResource2),
54                     contentDescription = result2.toString()
55                 )
56             }
57
58             Spacer(modifier = Modifier.height(12.dp))
59             Button(onClick = {
60                 result1 = (1..6).random()
61                 result2 = (1..6).random()
62
63                 scope.launch {
64                     snackbarHostState.currentSnackbarData?.dismiss()
65
66                     val message = if (result1 == result2) {
67                         "Selamat, anda dapat dadu double!"
68                     } else {
69                         "Anda belum beruntung!"
70                     }
71                     snackbarHostState.showSnackbar(
72                         message,
73                         duration = SnackbarDuration.Short
74                     )
75                 }
76             }) {
77                 Text(stringResource(R.string.roll))
78             }
79         }
80     }
81
82     fun getDiceImage(result: Int) : Int {
83         return when (result) {

```

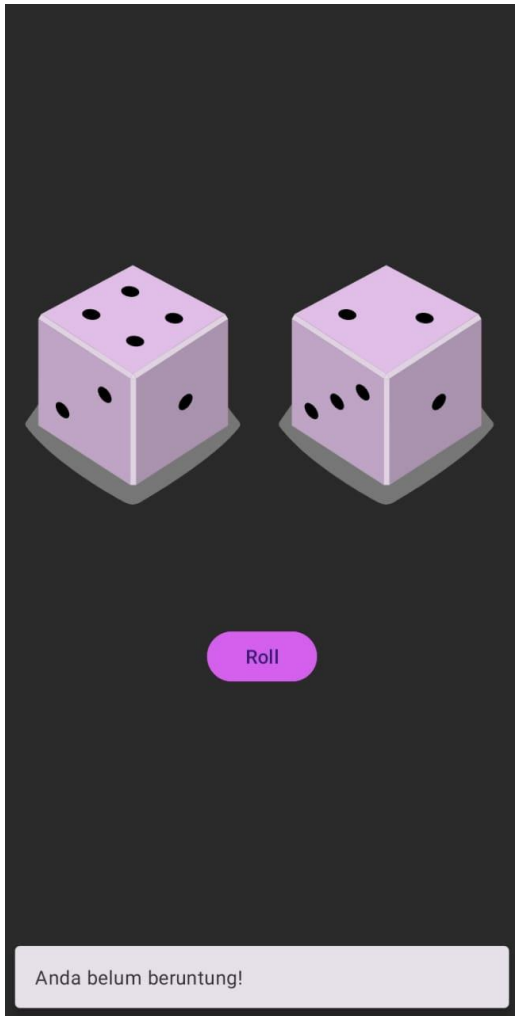
```
84         0 -> R.drawable.dice_0
85         1 -> R.drawable.dice_1
86         2 -> R.drawable.dice_2
87         3 -> R.drawable.dice_3
88         4 -> R.drawable.dice_4
89         5 -> R.drawable.dice_5
90         else -> R.drawable.dice_6
91     }
92 }
```

*Tabel 6. Source Code Jawaban Soal 1 DiceWithButtonAndImage.kt Compose*

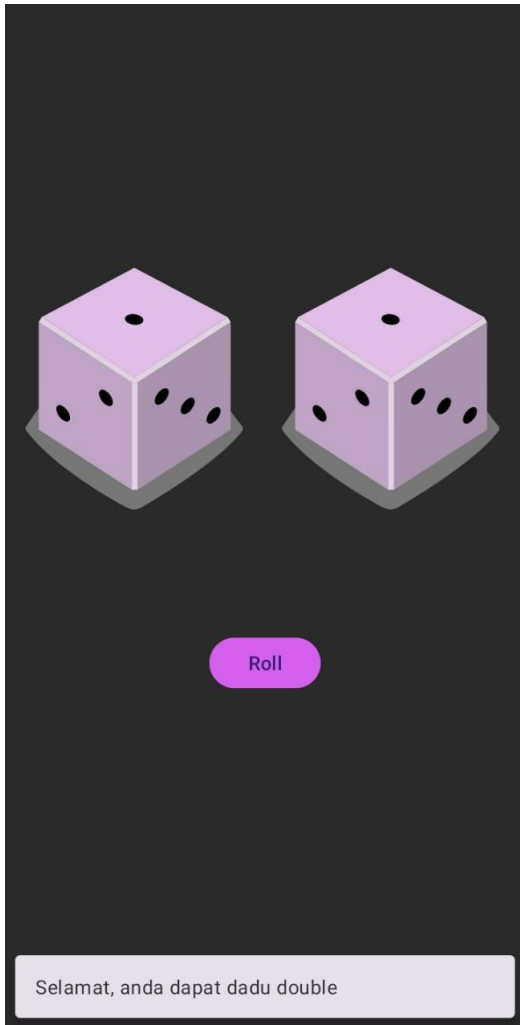
## **B. Output Program**



*Gambar 4. Screenshot Hasil Jawaban Soal 1 Tampilan Awal Aplikasi XML*

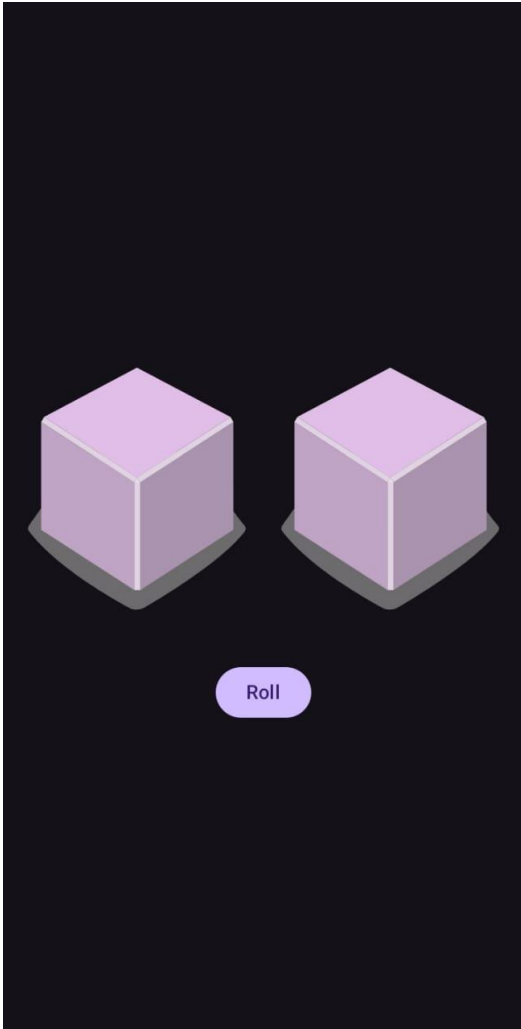


*Gambar 5. Screenshot Hasil Jawaban Soal 1 Tampilan Dadu Setelah Di-Roll XML*

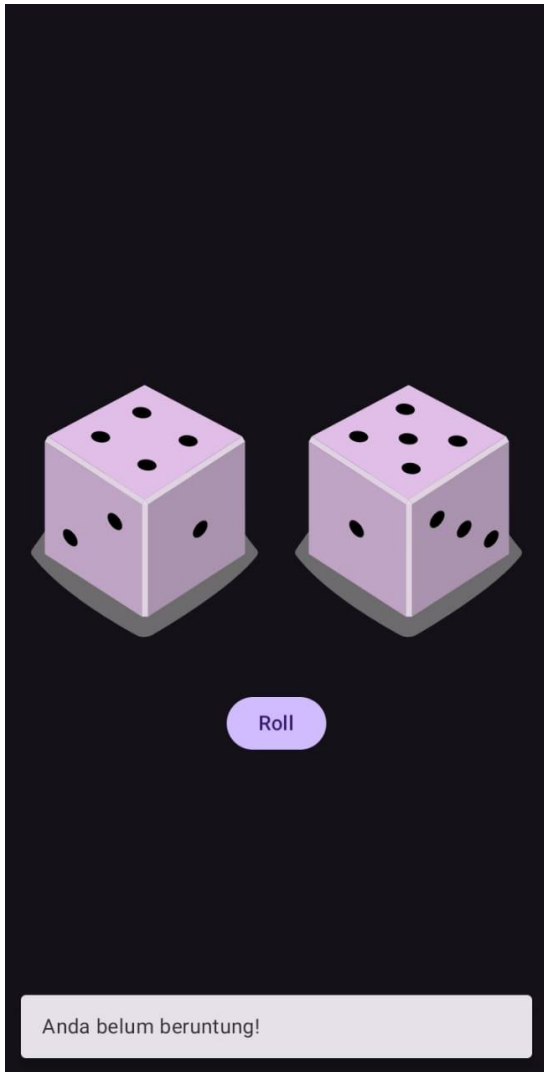


*Gambar 6. Screenshot Hasil Jawaban Soal 1 Tampilan Roll Dadu Double XML*

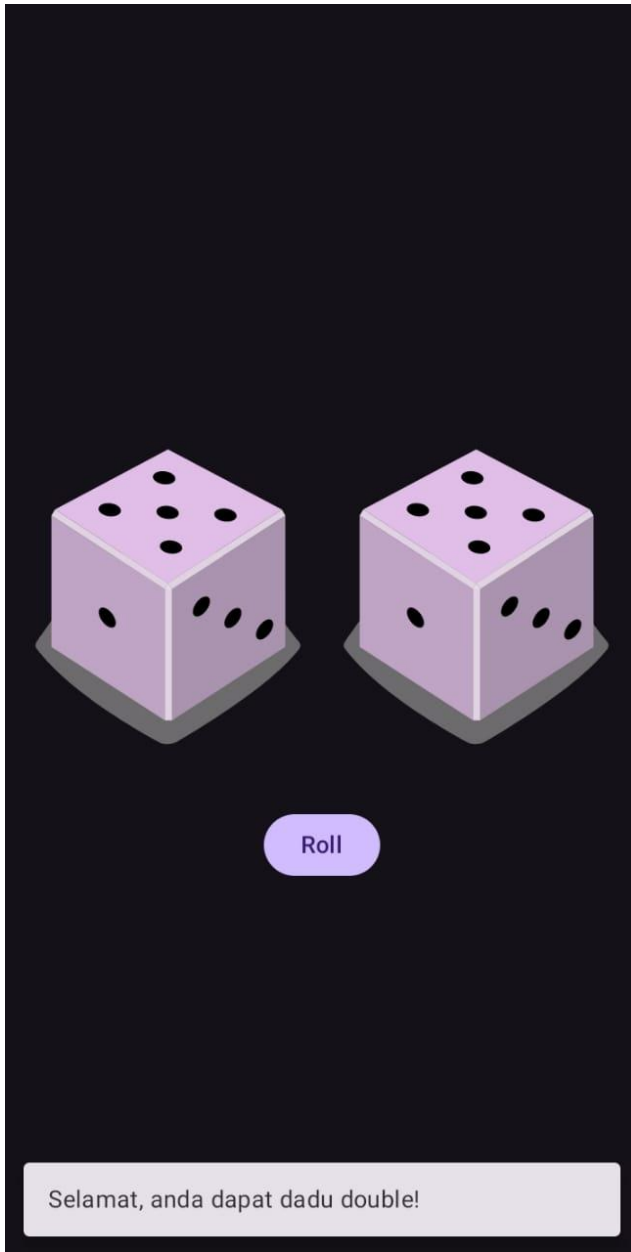




*Gambar 7. Screenshot Hasil Jawaban Soal 1 Tampilan Awal Aplikasi Compose*



*Gambar 8. Screenshot Hasil Jawaban Soal 1 Tampilan Dadu Setelah Di-Roll Compose*



*Gambar 9. Screenshot Hasil Jawaban Soal 1 Tampilan Roll Dadu Double Compose*

### **C. Pembahasan**

#### **XML:**

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

Pada line 1, dideklarasikan nama package file Kotlin

Pada line 3, diimport Bundle untuk mengkomunikasikan data antar aktivitas

Pada line 4, diimport enableEdgeToEdge untuk menampilkan konten dengan layar penuh

Pada line 5, diimport AppCompatActivity untuk base class activity yang mendukung versi terdahulu Android

Pada line 6, diimport ActivityMainBinding untuk menghubungkan layout XML dengan kode kotlin

Pada line 9, dideklarasikan kelas bernama MainActivity yang extend dari AppCompatActivity

Pada line 10, terdapat variabel binding bertipe ActivityMainBinding dengan deklarasi lateinit untuk diinisialisasi nanti

Pada line 11, terdapat override pada fungsi onCreate yang dipanggil saat activity dimulai

Pada line 12, akan dipanggil superclass dari onCreate

Pada line 13, akan dipanggil enableEdgeToEdge() agar tampilan memenuhi hingga tepi layar

Pada line 15, inialisasi binding untuk menghubungkan layout XML ke Kotlin menggunakan View Binding

Pada line 16, setContentView(binding.root) untuk menampilkan tampilan XML ke layar

Pada line 18, menetapkan aksi saat tombol dengan id button ditekan

Pada line 19, memanggil function diceRollerApp(binding)

### **DiceRollerApp.kt:**

Pada line 1, dideklarasikan nama package file Kotlin

Pada line 3, diimport ActivityMainBinding untuk menghubungkan layout XML dengan kode Kotlin

Pada line 3, diimport Snackbar untuk menampilkan notifikasi

Pada line 6, dideklarasikan fungsi diceRollerApp dengan parameter binding yang akan menjalankan logika dari aplikasi

Pada line 7, dideklarasikan variable result1 yang berisi hasil acak dari fungsi randomDice()

Pada line 8, dideklarasikan variable result2 yang berisi hasil acak dari fungsi randomDice()

Pada line 10, dideklarasikan variable imageResource1 yang akan mendapatkan gambar dadu sesuai dengan nilai result1 melalui fungsi getDiceImage()

Pada line 11, dideklarasikan variable imageResource2 yang akan mendapatkan gambar dadu sesuai dengan nilai result1 melalui fungsi getDiceImage()

Pada line 13, gambar dadu pertama akan ditampilkan di komponen ImageView dengan id imageView menggunakan fungsi setImageResource()

Pada line 14, gambar dadu kedua akan ditampilkan di komponen ImageView dengan id imageView2 menggunakan fungsi setImageResource()

Pada line 15, inialisasi binding untuk menghubungkan layout XML ke Kotlin menggunakan View Binding

Pada line 16, terdapat pengecekan kondisi if jika nilai result1 dan result2 sama (dadu double)

Pada line 17, jika kondisi if terpenuhi, maka akan ditampilkan Snackbar dengan pesan "Selamat, anda dapat dadu double" selama LENGTH\_LONG

Pada line 19, terdapat pengecekan kondisi else if jika result1 dan result2 tidak sama

Pada line 20, jika kondisi tersebut terpenuhi, maka akan ditampilkan Snackbar dengan pesan "Anda belum beruntung!" selama LENGTH\_LONG

Pada line 24, dideklarasikan fungsi randomDice() yang akan mengembalikan nilai Int acak dari 1 sampai 6 menggunakan (1..6).random()

Pada line 28, dideklarasikan fungsi getDiceImage() dengan parameter result bertipe Int yang akan mengembalikan resource gambar dadu yang sesuai

Pada line 29-35, digunakan struktur when untuk menentukan resource gambar dadu berdasarkan nilai result

Pada line 30, jika nilai 1, maka dikembalikan R.drawable.dice\_1  
Pada line 31, jika nilai 2, maka dikembalikan R.drawable.dice\_2  
Pada line 32, jika nilai 3, maka dikembalikan R.drawable.dice\_3  
Pada line 33, jika nilai 4, maka dikembalikan R.drawable.dice\_4  
Pada line 34, jika nilai 5, maka dikembalikan R.drawable.dice\_5  
Pada line 35, jika tidak ada yang cocok, maka akan dikembalikan gambar default R.drawable.dice\_6

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

Pada line 1, terdapat versi XML dan encoding yang digunakan  
Pada line 2, dideklarasikan root dari file layout, yakni ConstraintLayout dan namespace XML untuk atribut Android  
Pada line 3, dideklarasikan namespace untuk atribut kustom  
Pada line 4, dideklarasikan namespace untuk atribut tools dalam aplikasi  
Pada line 5, terdapat atribut android:id dengan nilai @+id/main yang menjadi id dari ConstraintLayout  
Pada line 6, layout\_width diatur match\_parent, artinya lebar layout mengikuti lebar layar perangkat  
Pada line 7, layout\_height diatur match\_parent, artinya tinggi layout mengikuti tinggi layar perangkat  
Pada line 8, background layout diatur dengan warna #2B2A2A, yaitu abu gelap  
Pada line 10, terdapat tag deklarasi element Button  
Pada line 9, tools:context mengaitkan layout ini dengan kelas MainActivity.kt  
Pada line 11, terdapat deklarasi elemen Button  
Pada line 12, dideklarasikan id dari Button, yaitu @+id/button  
Pada line 13, layout\_width diatur wrap\_content, menyesuaikan dengan lebar isi tombol  
Pada line 14, layout\_height diatur wrap\_content, menyesuaikan dengan tinggi isi tombol  
Pada line 15, layout\_marginStart diatur sebesar 162dp untuk memberi jarak dari sisi kiri  
Pada line 16, layout\_marginTop diatur sebesar 450dp sebagai jarak dari atas layar  
Pada line 17, layout\_marginEnd diatur sebesar 162dp untuk memberi jarak dari sisi kanan  
Pada line 18, layout\_marginBottom diatur sebesar 234dp sebagai jarak dari bawah  
Pada line 19, backgroundTint diatur dengan warna ungu #D460ED sebagai warna latar tombol  
Pada line 20, android:text diatur menggunakan string resource @string/roll (misalnya: "Roll")  
Pada line 21, app:layout\_constraintBottom\_toBottomOf="parent" artinya bagian bawah tombol dirapatkan ke bawah parent  
Pada line 22, app:layout\_constraintEnd\_toEndOf="parent" artinya bagian kanan tombol dirapatkan ke kanan parent  
Pada line 23, app:layout\_constraintStart\_toStartOf="parent" artinya bagian kiri tombol dirapatkan ke kiri parent  
Pada line 24, app:layout\_constraintTop\_toTopOf="parent" artinya bagian atas tombol dirapatkan ke atas parent  
Pada line 27, terdapat deklarasi elemen ImageView pertama  
Pada line 28, id dari ImageView adalah @+id/imageView  
Pada line 29, lebar ImageView diatur wrap\_content

Pada line 30, tinggi ImageView diatur wrap\_content  
 Pada line 31, layout\_marginTop diatur sebesar 171dp sebagai jarak dari atas  
 Pada line 32, layout\_marginEnd diatur sebesar 4dp sebagai jarak dari sebelah kanan  
 Pada line 33, layout\_marginBottom diatur sebesar 29dp sebagai jarak dari bawah  
 Pada line 34, contentDescription menggunakan string resource @string/image\_description sebagai deskripsi gambar untuk aksesibilitas  
 Pada line 35, app:layout\_constraintBottom\_toTopOf="@+id/button" artinya bagian bawah ImageView ditempel ke atas tombol  
 Pada line 36, app:layout\_constraintEnd\_toEndOf="parent" artinya sisi kanan ImageView ditempel ke sisi kanan parent  
 Pada line 37, app:layout\_constraintStart\_toEndOf="@+id/imageView2" artinya sisi kiri ImageView ditempel ke sisi kanan imageView2  
 Pada line 38, app:layout\_constraintTop\_toTopOf="parent" artinya bagian atas ImageView ditempel ke atas parent  
 Pada line 39, gambar awal ImageView ditampilkan menggunakan @drawable/dice\_0, gambar dadu kosong  
 Pada line 41, terdapat deklarasi elemen ImageView kedua  
 Pada line 42, id dari ImageView adalah @+id/imageView2  
 Pada line 43, lebar ImageView diatur wrap\_content  
 Pada line 44, tinggi ImageView diatur wrap\_content  
 Pada line 45, layout\_marginTop diatur sebesar 171dp  
 Pada line 46, layout\_marginBottom diatur sebesar 29dp  
 Pada line 47, contentDescription menggunakan string resource @string/image\_description  
 Pada line 48, app:layout\_constraintBottom\_toTopOf="@+id/button" artinya bagian bawah ImageView kedua ditempel ke atas tombol  
 Pada line 49, app:layout\_constraintEnd\_toStartOf="@+id/imageView" artinya sisi kanan ImageView2 ditempel ke kiri ImageView  
 Pada line 50, app:layout\_constraintStart\_toStartOf="parent" artinya sisi kiri ImageView2 ditempel ke sisi kiri parent  
 Pada line 51, app:layout\_constraintTop\_toTopOf="parent" artinya bagian atas ImageView2 ditempel ke atas parent  
 Pada line 52, app:srcCompat="@drawable/dice\_0" digunakan untuk menampilkan gambar awal dadu kosong  
 Pada line 53, merupakan penutup tag dari elemen root ConstraintLayout

### **Compose:**

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

Pada line 1, dideklarasikan nama package file Kotlin  
 Pada line 3, diimport Bundle untuk mengkomunikasikan data antar aktivitas  
 Pada line 4, diimport ComponentActivity sebagai base activity khusus untuk Jetpack Compose  
 Pada line 5, diimport setContent, yang digunakan untuk menetapkan tampilan berbasis Compose  
 Pada line 6, diimport enableEdgeToEdge agar tampilan aplikasi bisa memenuhi seluruh layar termasuk area status bar dan navigation bar

Pada line 7, diimport `fillMaxSize` yang merupakan modifier Compose agar elemen memenuhi ukuran layar penuh

Pada line 8, diimport `Scaffold`, yang digunakan sebagai struktur dasar layout di Jetpack Compose

Pada line 9, diimport Modifier dari Jetpack Compose untuk mengatur tampilan layout

Pada line 10, diimport `DiceRollerApp` dari package component, yaitu fungsi utama yang menampilkan UI aplikasi dadu

Pada line 11, diimport `DiceComposeTheme` dari package `ui.theme` untuk mengatur tema tampilan aplikasi

Pada line 13, deklarasi kelas `MainActivity` sebagai tempat logika utama aplikasi dijalankan

Pada line 14, terdapat override function yaitu `onCreate` untuk dipanggil saat aktivitas pertama kali dibuat

Pada line 15, memanggil function `super.onCreate` untuk memanggil konstruktor dari parent

Pada line 16, memanggil function `enableEdgeToEdge` agar tampilan aplikasi bisa memenuhi seluruh layar

Pada line 17, memanggil function `setContent` untuk menetapkan tampilan UI dengan Jetpack Compose

Pada line 18, memanggil `DiceComposeTheme`, tema kustom yang membungkus tampilan aplikasi

Pada line 19, memanggil `Scaffold` untuk layout utama aplikasi

Pada line 20, terdapat parameter `innerPadding` untuk mengatur padding dalam elemen (belum digunakan)

Pada line 21, memanggil function `DiceRollerApp()` untuk menampilkan komponen UI utama dari aplikasi

### **DiceRollerApp.kt:**

Pada line 1, dideklarasikan nama package file Kotlin

Pada line 3, diimport `layout.*` yang berisi fungsi-fungsi pengaturan layout

Pada line 4, diimport `Composable` untuk fungsi UI berbasis compose

Pada line 5, diimport `Alignment` untuk mengatur kesejajaran pada layout

Pada line 7, diimport `Preview` untuk menampilkan pratinjau UI

Pada line 9, dideklarasikan anotasi `Composable` untuk memberitahu function sebagai elemen UI compose

Pada line 10, dideklarasikan function `DiceRollerApp()` untuk memanggil function `DiceWithButtonAndImage()`

Pada line 11-14, dideklarasikan function yang terdapat parameter modifier dengan properti `fillMaxSize()` untuk memenuhi seluruhlayar dan `wrapContentSize(Alignment.Center)` untuk memusatkan isi konten jadi di tengah

Pada line 18, dideklarasikan anotasi `Preview(showBackground = true)` untuk menampilkan pratinjau UI dengan tampilan background

Pada line 19, dideklarasikan anotasi `Composable` untuk memberitahu function sebagai elemen UI compose

Pada line 21-21, dideklarasikan function `DiceRollerAppPreview` yang digunakan untuk memanggil function `DiceRollerApp()` untuk menampilkan preview dari function `DiceRollerApp`

## **DiceWithButtonAndImage.kt:**

Pada line 1, dideklarasikan nama package file Kotlin

Pada line 3, diimport Image untuk menampilkan gambar pada UI

Pada line 4, diimport Column untuk membuat kolom pada layout UI

Pada line 5, diimport Row untuk membuat Row pada layout UI

Pada line 6, diimport Spacer untuk membuat ruang kosong di antara 2 komponen

Pada line 7, diimport height untuk mengatur tinggi layout

Pada line 8, diimport Button untuk menyediakan komponen Button

Pada line 9, diimport Scaffold untuk struktur dasar layout di Jetpack Compose

Pada line 10, diimport SnackbarDuration untuk mengatur durasi notifikasi Snackbar

Pada line 11-12, diimport SnackbarHost untuk menampilkan komponen Snackbar

Pada line 13, diimport SnackbarHostState

Pada line 13, diimport Text untuk menampilkan teks pada UI

Pada line 14, diimport Composable untuk fungsi UI berbasis compose

Pada line 15, diimport getValue untuk mengakses nilai dari properti

Pada line 16, diimport mutableStateOf untuk membuat state count yang dimulai dari 0

Pada line 17, diimport remember untuk menyimpan nilai counter

Pada line 18, diimport rememberCoroutineScope untuk membuat CoroutineScope

Pada line 19, diimport setValue untuk mengubah nilai dari mutableStateOf dengan by

Pada line 20, diimport Alignment untuk mengatur kesejajaran pada layout

Pada line 21, diimport Modifier untuk mengatur tampilan layout

Pada line 22, diimport painterResource untuk mengambil/meload gambar dari sumber folder

Pada line 23, diimport stringResource untuk mengambil/meload string dari sumber folder

Pada line 24, diimport dp untuk mengatur ukuran dalam satuan density-independent pixels (dp)

Pada line 25, diimport R untuk menyediakan resources (sumber)

Pada line 27, diimport launch untuk menjalankan kode secara asinkronus dalam Coroutine Scope

Pada line 29, dideklarasikan anotasi Composable untuk memberitahu function sebagai elemen UI compose

Pada line 30, dideklarasikan function DiceWithButtonAndImage(modifier: Modifier = Modifier) untuk untuk membuat UI button dan dadu. Function ini memiliki parameter Modifier dengan nilai defaultnya modifier untuk modifikasi layout

Pada line 31-32, dideklarasikan variable (mutable) state result1 dan result2 untuk menyimpan hasil roll dua dadu yang nilai awalnya 0

Pada line 33-34, dideklarasikan variable (immutable) imageResource1 dan imageResource2 yang memanggil function getDiceImage(result) untuk menampilkan dadu apa yang muncul sesuai dengan nilai result

Pada line 36, dideklarasikan variable immutable scope yang memanggil function rememberCoroutineScope() untuk menampilkan snackbar secara asynchronous

Pada line 37, dideklarasikan variable immutable snackbarHostState untuk menyimpan state dan konten snackbar

Pada line 39-40, memanggil function Scaffold untuk layout utama aplikasi dan tempat untuk menampilkan snackbar melalui snackbarHost

Pada line 42, terdapat parameter innerPadding untuk mengatur padding dalam elemen (belum digunakan)



Pada line 43-45, menggunakan Column untuk membuat kolom pada layout dengan parameter modifier untuk modifikasi dan `horizontalAlignment` untuk mengatur isi konten ke Tengah secara horizontal.

Pada line 47-54, menggunakan Row untuk membuat baris pada layout yang didalamnya memanggil komponen `compose Image` untuk menampilkan gambar menggunakan function `painterResource()` dan deskripsi konten menggunakan function `toString()`

Pada line 58, memanggil `Spacer` ruang kosong dengan parameter modifier untuk memodifikasi tinggi pada layout sebesar 12dp

Pada line 59, memanggil `Button` dengan parameter `onClick` untuk melakukan aksi saat diklik

Pada line 60-61, mendeklarasikan variable `result1` dan `result2` yang menghasilkan nilai antara 1 sampai 6 secara acak menggunakan function `random()`

Pada line 63, menggunakan `scope.launch` untuk menampilkan `snackbar` secara asinkron

Pada line 64, jika notifikasi `snackbar` sebelumnya masih tampil, function `dismiss()` digunakan untuk menyembunyikan notifikasi sebelumnya

Pada line 66-69, mendeklarasikan variable `message immutable` untuk menampilkan pesan "Selamat, anda dapat dadu double!" jika nilai dari variable `result1` dan `result2` sama, jika selain itu (nilai tidak sama) maka menampilkan pesan "Anda belum beruntung!"

Pada line 71, memanggil function `showSnackbar()` untuk menampilkan `Snackbar` dengan parameter argument variable `message` dan deklarasi variable `duration` untuk mengatur durasi `Snackbar` secara singkat

Pada line 77, menggunakan komponen `Text` untuk menampilkan teks "Roll" yang diambil dari file `strings.xml`

Pada line 82-90, mendeklarasikan function `getDiceImage()` dengan parameter `result` bertipe data `Int` dengan return type `Int` untuk mengembalikan resource ID gambar berdasarkan nilai `result`. Jika `result` adalah 0, akan mengembalikan gambar awal `dice_0` (saat belum roll). Nilai 1-6 akan mengembalikan gambar `dice_1` sampai `dice_6` sesuai hasil roll. Jika selain itu, akan mengembalikan gambar `dice_6` sebagai default.

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

Berikut adalah tautan untuk source code yang telah dibuat.

[Bukhrii/Praktikum Mobile](#)