

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASICS WITH COMPOSE

Oleh:

Muhammad Azrianzan NIM. 2310817210003

**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 Basics with Compose ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Muhammad Azrianzan
NIM : 2310817210003

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..... | 1 |
| DAFTAR ISI | 2 |
| DAFTAR GAMBAR..... | 3 |
| DAFTAR TABEL | 4 |
| SOAL 1 | 5 |
| A. Source Code | 7 |
| B. Output Program..... | 14 |
| C. Pembahasan | 19 |
| D. Tautan Git | 22 |

DAFTAR GAMBAR

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

DAFTAR TABEL

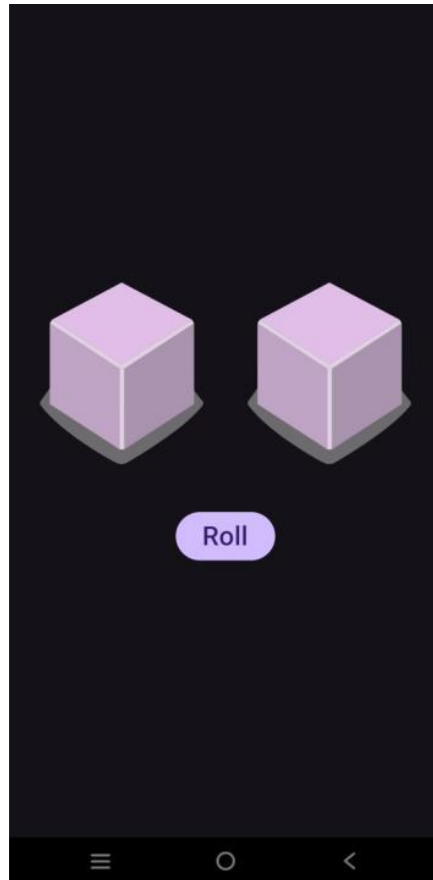
| | |
|--|----|
| Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt untuk Compose..... | 10 |
| Tabel 2. Source Code Jawaban Soal 1 MainActivity.kt untuk XML..... | 12 |
| Tabel 3. Source Code Jawaban Soal 1 activity_main.xml untuk XML | 14 |

SOAL 1

Soal Praktikum:

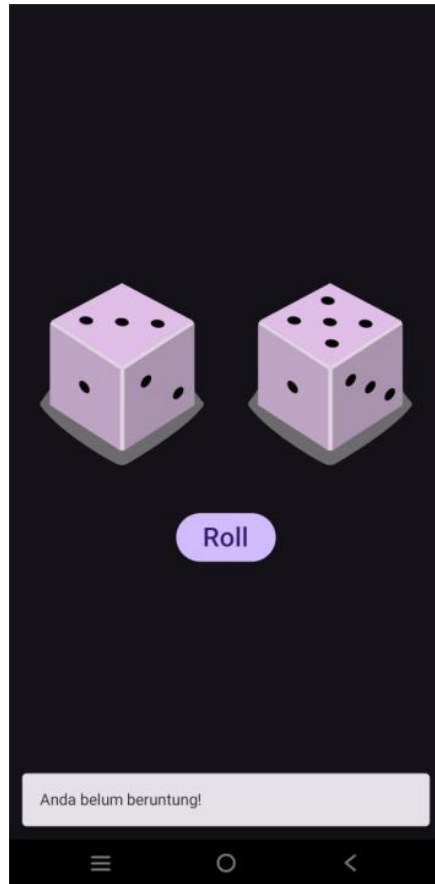
Buatlah sebuah aplikasi yang dapat menampilkan 2 buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



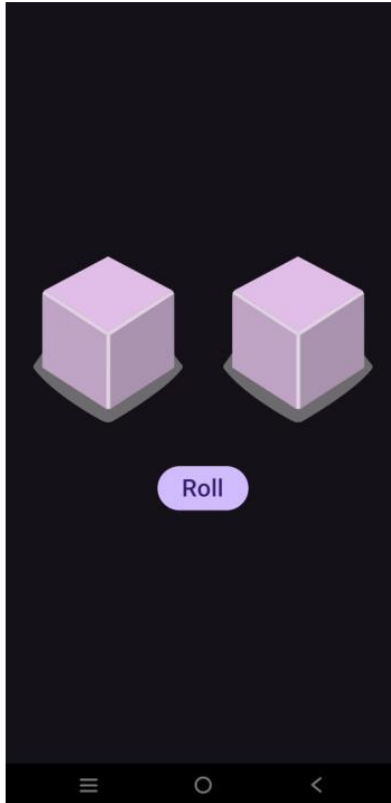
Gambar 1. Tampilan Awal Aplikasi

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

A. Source Code

Compose:

MainActivity.kt


```

1 package com.example.modull
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.foundation.layout.wrapContentSize
9 import androidx.compose.material3.Text
10 import androidx.compose.runtime.Composable
11 import androidx.compose.ui.Alignment
12 import androidx.compose.ui.Modifier
13 import androidx.compose.ui.tooling.preview.Preview
14 import com.example.modull.ui.theme.ModullTheme
15 import androidx.compose.foundation.layout.Column
16 import androidx.compose.material3.Button
17 import androidx.compose.ui.res.stringResource
18 import androidx.compose.foundation.Image
19 import androidx.compose.foundation.layout.Row
20 import androidx.compose.foundation.layout.Spacer
21 import androidx.compose.ui.res.painterResource
22 import androidx.compose.foundation.layout.height
23 import androidx.compose.foundation.layout.padding
24 import androidx.compose.material3.Scaffold
25 import androidx.compose.material3.SnackbarDuration
26 import androidx.compose.material3.SnackbarHost
27 import androidx.compose.material3.SnackbarHostState
28 import androidx.compose.ui.unit.dp
29 import androidx.compose.runtime.mutableStateOf
30 import androidx.compose.runtime.remember
31 import androidx.compose.runtime.getValue
32 import androidx.compose.runtime.rememberCoroutineScope
33 import androidx.compose.runtime.setValue
34 import kotlinx.coroutines.launch
35
36
37 class MainActivity : ComponentActivity() {
38     override fun onCreate(savedInstanceState: Bundle?) {
39         super.onCreate(savedInstanceState)
40         enableEdgeToEdge()
41         setContent {
42             ModullTheme {
43                 DiceRollerApp()
44             }
45         }
46     }
47 }

```

```

48
49 @Preview
50 @Composable
51 fun DiceRollerApp() {
52     DiceWithButtonAndImage(modifier = Modifier
53         .fillMaxSize()
54         .wrapContentSize(Alignment.Center))
55 }
56
57 @Composable
58 fun DiceWithButtonAndImage(modifier: Modifier = Modifier)
59 {
60     var result1 by remember { mutableStateOf(0) }
61     var result2 by remember { mutableStateOf(0) }
62     var message by remember { mutableStateOf("") }
63
64     val scope = rememberCoroutineScope()
65     val snackbarHostState = remember
66     {SnackbarHostState()}
67
68     fun getDiceImage(result: Int): Int {
69         return when (result) {
70             1 -> R.drawable.dice_1
71             2 -> R.drawable.dice_2
72             3 -> R.drawable.dice_3
73             4 -> R.drawable.dice_4
74             5 -> R.drawable.dice_5
75             6 -> R.drawable.dice_6
76             else -> R.drawable.dice_0
77         }
78     }
79
80     Scaffold (
81         snackbarHost = {SnackbarHost(hostState =
82         snackbarHostState)}
83     ) { paddingValues ->
84         Column (
85             modifier = modifier.padding(paddingValues),
86             horizontalAlignment =
87             Alignment.CenterHorizontally
88         ) {
89             Row {
90                 Image(
91                     painter =
92                     painterResource(getDiceImage(result1)),
93                     contentDescription =
94                     result1.toString()
95                 )

```

| | |
|-----|---|
| 89 | Image(|
| 90 | painter = |
| | painterResource(getDiceImage(result2)), |
| 91 | contentDescription = |
| | result2.toString() |
| 92 |) |
| 93 | } |
| 94 | Spacer(modifier = Modifier.height(16.dp)) |
| 95 | Button(onClick = { |
| 96 | result1 = (1..6).random() |
| 97 | result2 = (1..6).random() |
| 98 | scope.launch { |
| 99 | message = if (result1 == result2) |
| | "Selamat, anda dapat dadu double!" else "Anda belum |
| | beruntung!" |
| 100 | snackbarHostState.showSnackbar(|
| 101 | message, |
| 102 | duration = SnackbarDuration.Short |
| 103 |) |
| 104 | } |
| 105 | }) { |
| 106 | Text(stringResource(R.string.roll)) |
| 107 | } |
| 108 | } |
| 109 | } |
| 110 | } |

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

XML:

MainActivity.kt

```

1 package com.example.modul1_xml
2
3 import android.os.Bundle
4 import com.google.android.material.button.MaterialButton
5 import com.google.android.material.snackbar.Snackbar
6 import androidx.appcompat.app.AppCompatActivity
7 import android.widget.ImageView
8 import kotlin.random.Random
9
10
11 class MainActivity : AppCompatActivity() {
12     private lateinit var diceImage1: ImageView
13     private lateinit var diceImage2: ImageView
14
15     private val diceImages = listOf(
16         R.drawable.dice_1,
17         R.drawable.dice_2,
18         R.drawable.dice_3,
19         R.drawable.dice_4,
20         R.drawable.dice_5,
21         R.drawable.dice_6
22     )
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContentView(R.layout.activity_main)
27
28         diceImage1 = findViewById(R.id.dice_image_1)
29         diceImage2 = findViewById(R.id.dice_image_2)
30
31         val rollButton: MaterialButton =
32             findViewById(R.id.roll_button)
33         val coordinatorLayout:
34             androidx.coordinatorlayout.widget.CoordinatorLayout =
35                 findViewById(R.id.coordinator_layout)
36
37         rollButton.setOnClickListener {
38             val result1 = Random.nextInt(6)
39             val result2 = Random.nextInt(6)
40
41             diceImage1.setImageResource(diceImages[result1])
42             diceImage2.setImageResource(diceImages[result2])
43
44             val message = if (result1 == result2) {
45                 "Selamat, anda dapat dadu double!"
46             }
47         }
48     }
49 }

```

| | |
|----|---|
| 44 | } else { |
| 45 | "Anda belum beruntung!" |
| 46 | } |
| 47 | |
| 48 | Snackbar.make(coordinatorLayout, message, |
| | Snackbar.LENGTH_SHORT).show() |
| 49 | } |
| 50 | } |
| 51 | } |

Tabel 2. Source Code Jawaban Soal 1 MainActivity.kt untuk XML

activity_main.xml

```

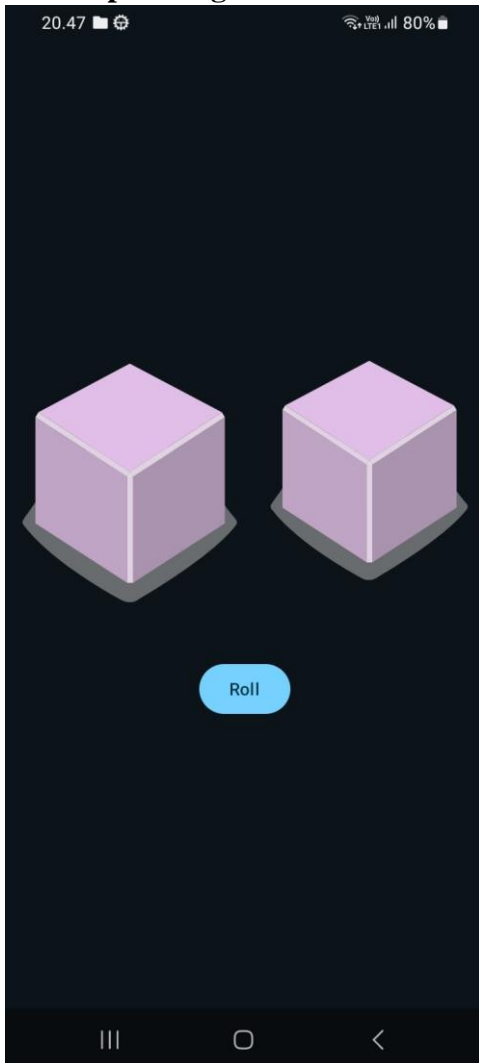
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.coordinatorlayout.widget.CoordinatorLayout
3  xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/coordinator_layout"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <LinearLayout
11         android:id="@+id/main_content"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"
14         android:gravity="center"
15         android:orientation="vertical"
16         android:padding="16dp">
17
18         <LinearLayout
19             android:layout_width="wrap_content"
20             android:layout_height="wrap_content"
21             android:orientation="horizontal"
22             android:gravity="center"
23             android:layout_marginBottom="24dp">
24
25             <ImageView
26                 android:id="@+id/dice_image_1"
27                 android:layout_width="200dp"
28                 android:layout_height="200dp"
29                 android:src="@drawable/dice_0"
30                 android:layout_margin="8dp" />
31
32             <ImageView
33                 android:id="@+id/dice_image_2"
34                 android:layout_width="200dp"
35                 android:layout_height="200dp"
36                 android:src="@drawable/dice_0"
37                 android:layout_margin="8dp" />
38         </LinearLayout>
39
40         <com.google.android.material.button.MaterialButton
41             android:id="@+id/roll_button"
42             android:layout_width="wrap_content"
43             android:layout_height="wrap_content"
44             android:text="@string/roll" />
45
46     </LinearLayout>

```

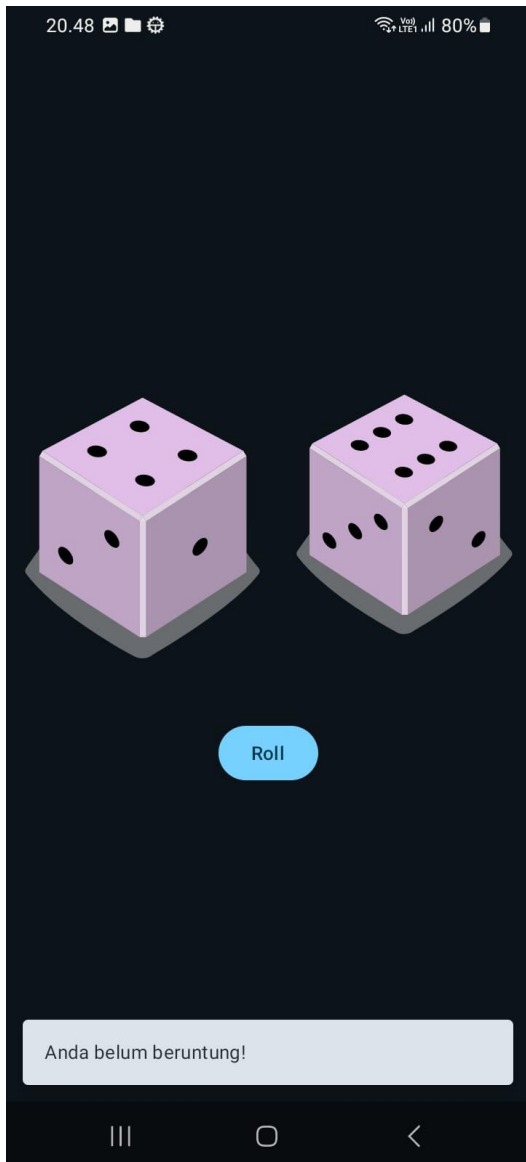
| | |
|----|--|
| 47 | </androidx.coordinatorlayout.widget.CoordinatorLayout> |
|----|--|

Tabel 3. Source Code Jawaban Soal 1 activity_main.xml untuk XML

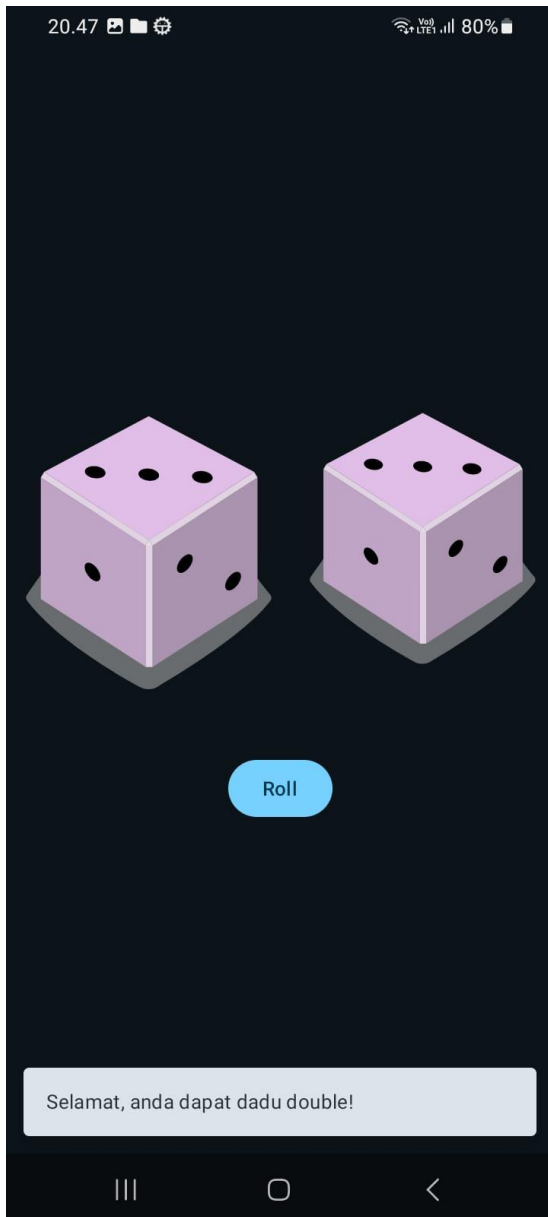
B. Output Program



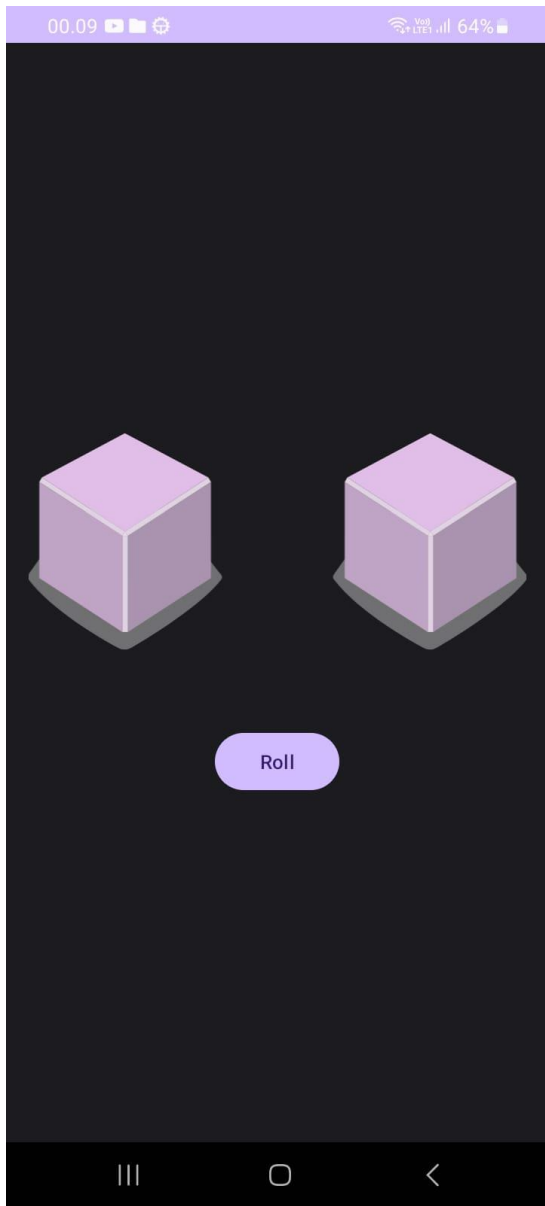
Gambar 4. Screenshot Hasil Jawaban Soal 1 Tampilan Awal Aplikasi dengan Compose



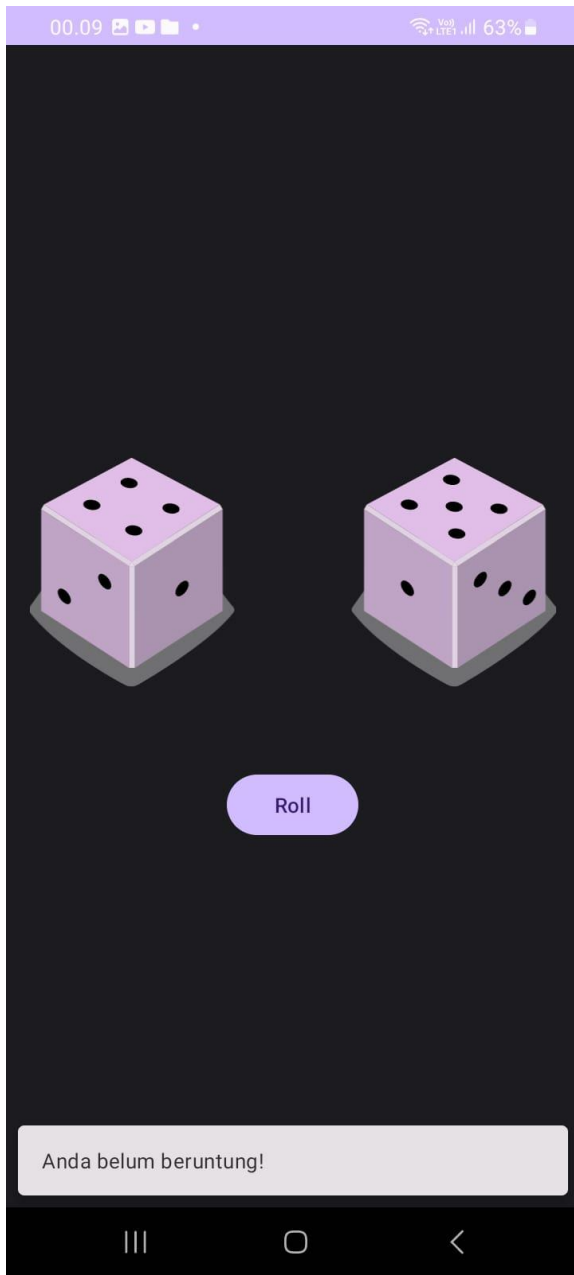
Gambar 5. Screenshot Hasil Jawaban Soal 1 Tampilan Dadu Setelah Di-Roll Compose



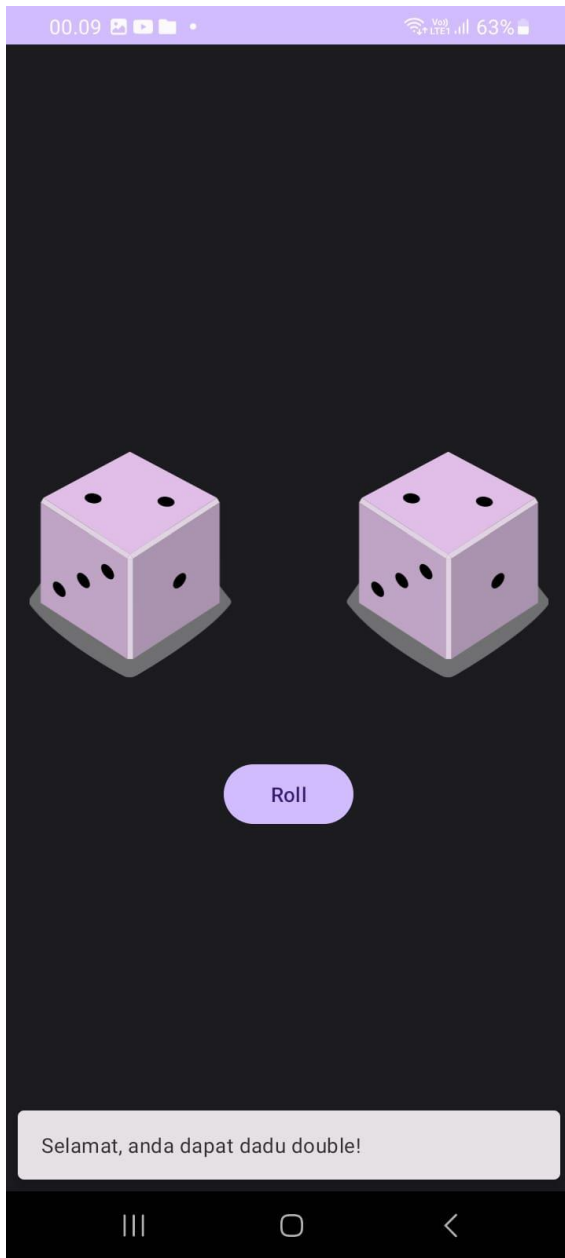
Gambar 6. Screenshot Hasil Jawaban Soal 1 Tampilan Saat Roll Dadu Double Compose



Gambar 7. Screenshot Hasil Jawaban Soal 1 Tampilan Awal Aplikasi dengan XML



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



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

C. Pembahasan

Compose:

MainActivity.kt:

Pada baris [1] didefinisikan paket aplikasi sebagai "com.example.modul1". Pada baris [3] hingga [6] diimport kelas-kelas untuk mengatur aktivitas Android dan Jetpack Compose. Pada baris [7] dan [8] diimport modifier untuk mengatur ukuran layout. Pada baris [9] diimport komponen `Text` dari Material Design 3 untuk menampilkan teks. Pada baris [10] diimport

anotasi untuk menandai fungsi sebagai Composable function. Pada baris [11] dan [12] diimpor kelas untuk alignment dan modifier dalam UI. Pada baris [13] diimpor anotasi untuk preview composable di Android Studio. Pada baris [14] diimpor tema aplikasi yang telah didefinisikan yaitu `Modul1Theme`. Pada baris [15] diimpor komponen layout `Column` untuk pengaturan vertikal. Pada baris [16] diimpor komponen `Button` untuk interaksi. Pada baris [17] diimpor fungsi untuk mengakses string dari resource. Pada baris [18] diimpor komponen `Image` untuk menampilkan gambar. Pada baris [19] diimpor komponen layout `Row` untuk pengaturan horizontal. Pada baris [20] diimpor komponen `Spacer` untuk memberikan ruang kosong. Pada baris [21] diimpor fungsi untuk mengakses gambar dari resource. Pada baris [22] dan [23] diimpor modifier untuk mengatur tinggi dan padding. Pada baris [24] hingga [27] diimpor komponen untuk `Scaffold` dan `Snackbar`. Pada baris [28] diimpor unit `dp` (density-independent pixels) untuk dimensi. Pada baris [29] hingga [33] diimpor fungsi untuk state management di Compose. Pada baris [34] diimpor fungsi `coroutine` untuk operasi asinkron.

Pada baris [37] didefinisikan kelas utama yang mewarisi `ComponentActivity`. Pada baris [38] dan [39] dipanggil override fungsi `onCreate` saat aktivitas dibuat untuk memanggil implementasi parent. Pada baris [40] diaktifkan tampilan edge-to-edge untuk UI yang lebih immersive. Pada baris [42] hingga [45] ditetapkan konten UI menggunakan Compose, dengan tema `Modul1Theme` dan composable `DiceRollerApp`. Pada baris [46] dan [47] ada tutup kurung kurawal dari fungsi `onCreate` dan kelas utama. Pada baris [49] hingga [51] didefinisikan fungsi composable `DiceRollerApp` yang mana fungsi tersebut juga ditandai dengan anotasi `Preview`. Pada baris [52] hingga [55] yang merupakan isi dari fungsi composable `DiceRollerApp` akan memanggil composable `DiceWithButtonAndImage` dengan modifier untuk mengisi seluruh layar dan memusatkan konten.

Pada baris [57] dan [58] didefinisikan composable utama aplikasi dengan parameter modifier default. Pada baris [59] dan [60] dibuat state untuk menyimpan nilai hasil dadu 1 yaitu `result1` dan dadu 2 yaitu `result2`. Pada baris [61] dibuat state bernama `message` untuk menyimpan pesan yang akan ditampilkan. Pada baris [63] dan [64] disimpan `coroutine scope` untuk operasi asinkron pada variabel `scope` dan dibuat state bernama `snackbarHostState` untuk mengontrol `Snackbar`. Pada baris [66] hingga [76] dibuat fungsi `getDiceImage` untuk mendapatkan resource gambar dadu 0 hingga 6 sesuai nilai parameter `hasil`. Pada baris [77] hingga [79] digunakan `Scaffold` sebagai layout dasar dengan konfigurasi `SnackbarHost`. Pada baris [80] hingga [83] dibuat layout `Column` dengan padding dan alignment horizontal center. Pada baris [84] dibuat layout `Row` untuk menempatkan dua gambar dadu secara horizontal. Pada baris [85] hingga [88] ditampilkan gambar dadu pertama sesuai dengan nilai `result1` menggunakan `Image`. Pada baris [89] hingga [93] ditampilkan gambar dadu kedua sesuai dengan nilai `result2` menggunakan `Image`. Pada baris [94] dibuat ruang vertikal sepanjang `16dp` menggunakan `Spacer`. Pada baris [95] hingga [97] dibuat tombol menggunakan `Button` dengan fungsi `onClick` yang mengacak nilai dadu 1 (`result1`) dan 2 (`result2`) antara 1-6. Pada baris [98] dan [99] digunakan `coroutine` untuk mengatur pesan berdasarkan kondisi apakah kedua dadu sama, jika kedua dadu sama maka pesan atau state `message` diatur isinya menjadi "Selamat, anda

dapat dadu double!" dan jika tidak maka isinya diatur menjadi "Anda belum beruntung!". Pada baris [99] hingga [105] ditampilkan snackbar sesuai dengan pesan yang telah diatur tadi selama durasi singkat. Pada baris [106] hingga [110] ditampilkan teks pada tombol dari resource string "roll" dan ada kurung tutup kurawal untuk menutup blok Column, Scaffold, dan fungsi DiceWithButtonAndImage.

XML:

MainActivity.kt:

Pada baris [1] didefinisikan paket aplikasi sebagai "com.example.modul1_xml". Pada baris [3] diimpor kelas Bundle untuk menyimpan dan mengirim data antar komponen Android. Pada baris [4] diimpor kelas MaterialButton dari library Material Design untuk tombol. Pada baris [5] diimpor kelas Snackbar dari library Material Design untuk menampilkan pesan singkat. Pada baris [6] diimpor kelas dasar AppCompatActivity yang menyediakan kompatibilitas Material Design pada versi Android lama. Pada baris [7] dan [8] diimpor kelas ImageView untuk menampilkan gambar dadu dan kelas Random dari Kotlin untuk menghasilkan angka acak. Pada baris [11] didefinisikan kelas utama MainActivity yang mewarisi AppCompatActivity. Pada baris [12] dan [13] dideklarasikan dua variabel ImageView bernama diceImage1 dan diceImage2 dengan penundaan inisialisasi (akan diinisialisasi nanti). Pada baris [15] hingga [22] dibuat daftar resource gambar dadu yang berisi gambar dadu 1 hingga 6 yang akan digunakan untuk menampilkan hasil lemparan. Pada baris [24] dan [25] dipanggil override fungsi onCreate saat aktivitas dibuat yang memanggil impelementasi parent. Pada baris [26] digunakan setContentView untuk menetapkan layout XML yang akan digunakan untuk aktivitas ini yaitu activity_main. Pada baris [28] dan [29] diinisialisasi variabel ImageView diceImage1 dan diceImage2 tadi dengan mencari elemen dari layout berdasarkan ID-nya. Pada baris [31] dicari tombol untuk melempar dadu dari layout berdasarkan ID-nya yang hasilnya akan disimpan ke variabel rollButton. Pada baris [32] dan [33] dicari juga layout berdasarkan ID-nya yang hasilnya disimpan ke variabel CoordinatorLayout untuk tempat menampilkan Snackbar. Pada baris [35] ditetapkan listener untuk menangani ketika tombol roll ditekan. Ketika tombol roll tersebut ditekan, dijalankan blok kode di dalamnya yaitu pada baris [36] dan [37] dihasilkan dua angka acak antara 0-5 untuk hasil lemparan dadu yang hasilnya disimpan ke variabel result1 dan result2 kemudian di baris [39] dan [40] ditetapkan gambar dadu pada ImageView diceImage1 dan diceImage2 sesuai dengan hasil acak yang didapatkan. Lalu berlanjut pada baris [42] hingga [46] yang menentukan isi dari variabel message sebagai pesan yang akan ditampilkan berdasarkan hasil lemparan dadu. Jika nilai dadu 1 (result1) dan dadu 2 (result2) sama maka isi variabel message adalah "Selamat, anda dapat dadu double!", namun jika nilainya berbeda, maka isi variabel message adalah "Anda belum beruntung!". Serta pada baris [48] dibuat dan ditampilkan snackbar dengan pesan yang telah ditentukan tadi selama durasi singkat. Pada baris [49] hingga [51] ada tutup kurung kurawal untuk menutup blok setOnClickListener, fungsi onCreate, dan kelas MainActivity.

activity_main.xml:

Pada baris [1] dideklarasikan XML dengan versi 1.0 dan encoding UTF-8. Pada baris [2] dan [3] ada elemen root menggunakan CoordinatorLayout dari AndroidX dengan namespace android dan tools. Pada baris [4] ditetapkan ID untuk CoordinatorLayout agar bisa direferensikan di kode Kotlin. Pada baris [5] dan [6] diatur lebar dan tinggi layout agar mengisi seluruh layar parent. Pada baris [7] ditentukan konteks untuk tools Android Studio, menunjukkan bahwa layout ini digunakan oleh MainActivity. Pada baris [9] dan [10] dibuat LinearLayout dengan ID main_content sebagai kontainer utama. Pada baris [11] dan [12] diatur lebar dan tinggi LinearLayout agar mengisi seluruh layar parent. Pada baris [13] menetapkan posisi konten LinearLayout di tengah secara horizontal dan vertikal. Pada baris [14] diatur orientasi LinearLayout menjadi vertikal (disusun dari atas ke bawah). Pada baris [15] ditambahkan padding 16dp pada semua sisi LinearLayout. Pada baris [17] hingga [19] dibuat LinearLayout bersarang (nested LinearLayout) dengan ukuran yang menyesuaikan kontennya. Pada baris [20] diatur orientasi LinearLayout bersarang menjadi horizontal (disusun dari kiri ke kanan). Pada baris [21] ditempatkan konten LinearLayout bersarang di tengah secara horizontal dan vertikal. Pada baris [22] ditambahkan margin bawah sebesar 24dp untuk LinearLayout bersarang. Pada baris [24] dan [25] dibuat ImageView pertama dengan ID dice_image_1 untuk gambar dadu pertama. Pada baris [26] dan [27] diatur lebar dan tinggi ImageView pertama menjadi 200dp x 200dp. Pada baris [28] dan [29] ditetapkan gambar default untuk ImageView pertama menggunakan resource dice_0 (gambar dadu 0) dan ditambahkan margin 8dp pada semua sisi ImageView pertama. Pada baris [31] dan [32] dibuat ImageView kedua dengan ID dice_image_2 untuk gambar dadu kedua. Pada baris [33] dan [34] diatur lebar dan tinggi ImageView kedua menjadi 200dp x 200dp. Pada baris [35] dan [36] ditetapkan gambar default untuk ImageView kedua menggunakan resource dice_0 (gambar dadu 0) juga dan ditambahkan margin 8dp pada semua sisi ImageView kedua. Pada baris [37] ada tag penutup LinearLayout bersarang yang berisi kedua ImageView.

Pada baris [39] dan [40] dibuat tombol Material Design dengan ID roll_button. Pada baris [41] dan [42] diatur lebar dan tinggi tombol agar menyesuaikan kontennya. Pada baris [43] ditetapkan teks tombol menggunakan string resource dengan nama "roll". Pada baris [45] ada tag penutup LinearLayout utama. Pada baris [47] ada tag penutup CoordinatorLayout root.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/Azrianzan/Praktikum_Pemrograman_Mobile/tree/main/Modul%201