

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASICS WITH COMPOSE

Oleh:

Avantio Fierza Patria NIM. 2310817310001

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MARET 2025**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN Mobile
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 : Avantio Fierza Patria
NIM : 2310817310001

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..... | i |
| DAFTAR ISI | ii |
| DAFTAR GAMBAR..... | iii |
| DAFTAR TABEL | iv |
| SOAL 1 | 1 |
| A. Source Code menggunakan Jetpack Compose | 3 |
| B. Source Code menggunakan XML | 7 |
| C. Output Program | 12 |
| D. Pembahasan | 13 |
| D. Tautan GIT | 16 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1. 1 Screenshot Hasil Jawaban Soal 1 | 12 |
|---|----|

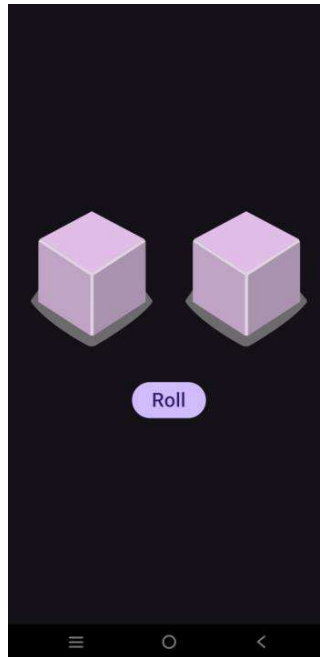
DAFTAR TABEL

SOAL 1

Soal Praktikum:

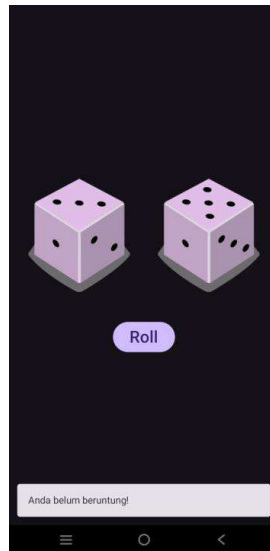
Buatlah sebuah aplikasi yang dapat menampilkan 2 buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



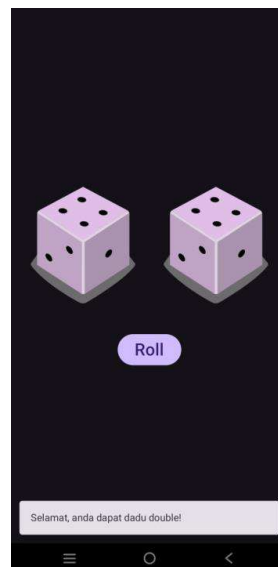
Gambar 1 Tampilan Awal Aplikasi

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 menggunakan Jetpack Compose MainActivity.kt

```
1 package com.example.diceroller
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.*
8 import androidx.compose.material3.*
9 import androidx.compose.runtime.*
10 import androidx.compose.ui.Alignment
11 import androidx.compose.ui.Modifier
12 import androidx.compose.ui.graphics.Color
13 import androidx.compose.ui.res.painterResource
14 import androidx.compose.ui.unit.dp
15 import androidx.compose.ui.unit.sp
16 import kotlin.random.Random
17
18 class MainActivity : ComponentActivity() {
19     override fun onCreate(savedInstanceState: Bundle?) {
20         super.onCreate(savedInstanceState)
21         setContent {
22             DiceRollerApp()
23         }
24     }
25 }
```



```

24     }
25 }
26
27 @Composable
28 fun DiceRollerApp() {
29     MaterialTheme {
30         Surface(
31             modifier = Modifier.fillMaxSize(),
32             color = Color(0xFFFFF5E1) // Warna krem
33         ) {
34             DiceContent()
35         }
36     }
37 }
38
39 @Composable
40 fun DiceContent() {
41     var dice1Value by remember { mutableStateOf(1) }
42     var dice2Value by remember { mutableStateOf(1) }
43     var message by remember { mutableStateOf("Tekan tombol
44     untuk mulai!") }
45
46     Column(
47         modifier = Modifier
48             .fillMaxSize()
49             .padding(16.dp),
50         horizontalAlignment = Alignment.CenterHorizontally,
51         verticalArrangement = Arrangement.Center
52     ) {
53         Text(text = "Roll the Dice!", fontSize = 24.sp, color
54         = Color.Black)

```

```

55         Spacer(modifier = Modifier.height(16.dp))
56
57         Row(
58             horizontalArrangement = Arrangement.Center
59         ) {
60             DiceImage(dice1Value)
61             Spacer(modifier = Modifier.width(16.dp))
62             DiceImage(dice2Value)
63         }
64
65         Spacer(modifier = Modifier.height(16.dp))
66
67         Text(text = message, fontSize = 18.sp, color =
68         Color.DarkGray)
69
70         Spacer(modifier = Modifier.height(24.dp))
71
72         Button(
73             onClick = {
74                 dice1Value = Random.nextInt(1, 7)
75                 dice2Value = Random.nextInt(1, 7)
76
77                 message = if (dice1Value == dice2Value) {
78                     "🎉 Selamat, anda dapat dadu double! 🎉"
79                 } else {
80                     "Anda belum beruntung! 😞"
81                 }
82             }
83         ) {
84             Text(text = "Roll", fontSize = 18.sp)
85         }

```

```

86     }
87 }
88
89 @Composable
90 fun DiceImage(diceValue: Int) {
91     Image(
92         painter = painterResource(id =
93         getDiceImageResource(diceValue)),
94         contentDescription = "Dice $diceValue",
95         modifier = Modifier.size(100.dp)
96     )
97 }
98
99 fun getDiceImageResource(value: Int): Int {
100     return when (value) {
101         1 -> R.drawable.dice_1
102         2 -> R.drawable.dice_2
103         3 -> R.drawable.dice_3
104         4 -> R.drawable.dice_4
105         5 -> R.drawable.dice_5
106         6 -> R.drawable.dice_6
107         else -> R.drawable.dice_1
108     }
109 }
110
111

```

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

B. Source Code menggunakan XML

MainActivity.kt

```
1 package com.example.diceroller2
2
3 import android.os.Bundle
4 import android.widget.Button
5 import android.widget.ImageView
6 import android.widget.TextView
7 import androidx.appcompat.app.AppCompatActivity
8 import kotlin.random.Random
9
10 class MainActivity : AppCompatActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14
15         // Ambil referensi dari UI
16         val dice1Image: ImageView =
17         findViewById(R.id.dice1Image)
18         val dice2Image: ImageView =
19         findViewById(R.id.dice2Image)
20         val btnRoll: Button = findViewById(R.id.btnRoll)
21         val tvMessage: TextView = findViewById(R.id.tvMessage)
```

```

21
22         btnRoll.setOnClickListener {
23             val dice1Value = Random.nextInt(1, 7)
24             val dice2Value = Random.nextInt(1, 7)
25
26             // Update gambar dadu
27
28             dice1Image.setImageResource(getDiceImage(dice1Value))
29             dice2Image.setImageResource(getDiceImage(dice2Value))
30
31             // Update pesan berdasarkan hasil
32             tvMessage.text = if (dice1Value == dice2Value) {
33                 "🎉 Selamat, anda dapat dadu double! 🎉"
34             } else {
35                 "😞 Anda belum beruntung!"
36             }
37         }
38     }
39
40     private fun getDiceImage(value: Int): Int {
41         return when (value) {
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             6 -> R.drawable.dice_6
48             else -> R.drawable.dice_1
49         }
50     }
51 }

```

| | |
|----|--|
| 52 | |
| 53 | |
| 54 | |

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

activity_main.xml

| | |
|----|--|
| 1 | <?xml version="1.0" encoding="utf-8"?> |
| 2 | <LinearLayout |
| 3 | xmlns:android="http://schemas.android.com/apk/res/android" |
| 4 | android:layout_width="match_parent" |
| 5 | android:layout_height="match_parent" |
| 6 | android:orientation="vertical" |
| 7 | android:background="@color/cream" |
| 8 | android:gravity="center" |
| 9 | android:padding="16dp"> |
| 10 | |
| 11 | <!-- Judul --> |
| 12 | <TextView |
| 13 | android:id="@+id/tvTitle" |
| 14 | android:layout_width="wrap_content" |
| 15 | android:layout_height="wrap_content" |
| 16 | android:text="Roll the Dice!" |
| 17 | android:textSize="24sp" |
| 18 | android:textStyle="bold" |
| 19 | android:textColor="@color/black" |
| 20 | android:layout_marginBottom="16dp"/> |
| 21 | |
| 22 | <!-- Dadu --> |
| 23 | <LinearLayout |
| 24 | android:layout_width="wrap_content" |
| 25 | android:layout_height="wrap_content" |
| 26 | android:orientation="horizontal" |

```

27         android:gravity="center">
28
29         <ImageView
30             android:id="@+id/dice1Image"
31             android:layout_width="100dp"
32             android:layout_height="100dp"
33             android:src="@drawable/dice_1"/>
34
35         <Space
36             android:layout_width="16dp"
37             android:layout_height="wrap_content"/>
38
39         <ImageView
40             android:id="@+id/dice2Image"
41             android:layout_width="100dp"
42             android:layout_height="100dp"
43             android:src="@drawable/dice_1"/>
44     </LinearLayout>
45
46     <!-- Pesan -->
47     <TextView
48         android:id="@+id/tvMessage"
49         android:layout_width="wrap_content"
50         android:layout_height="wrap_content"
51         android:text="Tekan tombol untuk mulai!"
52         android:textSize="18sp"
53         android:textColor="@color/black"
54         android:layout_marginTop="16dp"/>
55
56     <!-- Tombol -->
57     <Button

```

| | |
|----|---|
| 58 | <code>android:id="@+id/btnRoll"</code> |
| 59 | <code>android:layout_width="wrap_content"</code> |
| 60 | <code>android:layout_height="wrap_content"</code> |
| 61 | <code>android:text="Roll"</code> |
| 62 | <code>android:textSize="18sp"</code> |
| 63 | <code>android:layout_marginTop="24dp"/></code> |
| | <code></LinearLayout></code> |

Tabel 1. 3 Source Code Jawaban Soal 1 activity_maint .XML

C. Output Program



Gambar 1. 1 Screenshot Hasil Jawaban Soal 1

D. Pembahasan

MainActivity.kt Jetpack Compose:

1. Import Library

- **ComponentActivity** → Kelas dasar untuk aktivitas yang mendukung **Jetpack Compose**.
- **setContent {}** → Fungsi untuk mengganti tampilan menggunakan **Compose**.
- **Image** → Digunakan untuk menampilkan gambar dadu.
- **Surface** → Digunakan untuk membungkus elemen UI dengan warna latar belakang.
- **MaterialTheme** → Menerapkan tema Material3.
- **Random** → Menghasilkan angka acak (untuk dadu).

2. MainActivity

- **MainActivity** → Kelas utama yang menjalankan aplikasi.
- **onCreate** → Metode yang dipanggil saat aplikasi pertama kali dijalankan.
- **setContent { DiceRollerApp() }** → Menetapkan tampilan utama aplikasi dengan Composable DiceRollerApp().

3. Fungsi DiceRollerApp()

- **@Composable** → Menandakan fungsi ini sebagai bagian dari UI Jetpack Compose.
- **MaterialTheme** → Menerapkan tema Material3 ke seluruh aplikasi.
- **Surface** → Membuat latar belakang dengan warna krem (#FFF5E1).
- **DiceContent()** → Memanggil fungsi yang menampilkan konten utama aplikasi.

4. Fungsi DiceContent()

- **var dice1Value by remember { mutableStateOf(1) }**
 - Menyimpan nilai dadu pertama, menggunakan remember agar tetap tersimpan saat UI diperbarui.
- **var dice2Value by remember { mutableStateOf(1) }**
 - Menyimpan nilai dadu kedua.
- **var message by remember { mutableStateOf("Tekan tombol untuk mulai!") }**
 - Menyimpan pesan hasil roll dadu.

5. Struktur Layout dengan Column

- **Column {}** → Menyusun elemen UI secara vertikal.
- **modifier = Modifier.fillMaxSize().padding(16.dp)**
 - Mengisi layar penuh dan memberikan padding **16dp**
- **horizontalAlignment = Alignment.CenterHorizontally**
 - Menyusun elemen ke **tengah horizontal**.
- **verticalArrangement = Arrangement.Center**
 - Menyusun elemen ke **tengah vertikal**.

6. Menampilkan Teks dan Gambar Dadu

- **Text("Roll the Dice!")** → Menampilkan teks judul.
- **Spacer** → Memberikan jarak antar elemen.
- **Row {}** → Menyusun elemen UI secara **horizontal** untuk kedua dadu.
- **DiceImage(dice1Value)** → Memanggil fungsi untuk menampilkan gambar dadu pertama.
- **DiceImage(dice2Value)** → Memanggil fungsi untuk menampilkan gambar dadu kedua.

7. Menampilkan Pesan Hasil Roll

- Menampilkan pesan hasil roll dadu (message).
- Warna abu-abu tua (**Color.DarkGray**) agar lebih terlihat di latar krem.

8. Tombol "Roll" untuk Mengacak Dadu

- **Button(onClick = { ... })** → Menjalankan aksi saat tombol ditekan.
- **Random.nextInt(1, 7)** → Menghasilkan angka acak **1-6** untuk dadu.
- Cek apakah dadu double:
 - Jika **sama** → tampilkan 🎉 Selamat, anda dapat dadu double! 🎉
 - Jika **berbeda** → tampilkan "Anda belum beruntung! 😞"

9. Fungsi DiceImage()

- Menampilkan gambar dadu berdasarkan nilai **diceValue**.
- Ukuran gambar dadu = **100.dp**.

10. Fungsi getDiceImageResource

- Mengembalikan gambar dadu yang sesuai dengan angka yang muncul.

MainActivity.kt XML:

1. Import Library

- **AppCompatActivity** → Kelas dasar untuk aktivitas berbasis XML dengan fitur kompatibilitas lama.
- **ImageView** → Untuk menampilkan gambar dadu.
- **Button** → Untuk tombol "Roll".
- **TextView** → Untuk menampilkan pesan hasil roll dadu.
- **Random** → Untuk menghasilkan angka acak (1-6).

2. MainActivity

- **MainActivity** → Kelas utama yang menjalankan aplikasi.
- **onCreate** → Metode yang dipanggil saat aplikasi pertama kali dijalankan.
- **setContentView(R.layout.activity_main)** → Menetapkan tampilan utama dari file XML **activity_main.xml**.

3. Menghubungkan Elemen UI dari XML

- **Menghubungkan elemen-elemen UI** dengan **findViewById()** agar bisa digunakan dalam kode.
- **dice1Image** → Gambar dadu pertama.
- **dice2Image** → Gambar dadu kedua.
- **btnRoll** → Tombol untuk meroll dadu.
- **tvMessage** → Teks yang menampilkan pesan setelah roll.

4. Menetapkan Aksi pada Tombol "Roll"

- **btnRoll.setOnClickListener {}** → Menjalankan aksi saat tombol Roll ditekan.
- **Random.nextInt(1, 7)** → Menghasilkan angka acak antara 1-6 untuk setiap dadu.
setImageResource(getDiceImage(dice1Value)) → Mengubah gambar dadu sesuai hasil.
- Cek apakah dadu double:
- **Jika nilai dadu sama** → Tampilkan "🎉 Selamat, anda dapat dadu double!"
- **Jika nilai berbeda** → Tampilkan "😞 Anda belum beruntung!"

5. Fungsi getDiceImage ()

- Mengembalikan gambar dadu yang sesuai dengan angka yang muncul.
- Misalnya, jika angka 3 keluar, akan mengembalikan **R.drawable.dice_3** (gambar dadu 3).

6. activity_main.xml:

- **LinearLayout** → Tata letak vertikal untuk menyusun elemen dari atas ke bawah.
- **android:background="#FFF5E1"** → Mengatur warna latar belakang menjadi krem.
- **TextView (tvMessage)** → Menampilkan pesan hasil roll dadu.
- **ImageView (dice1Image & dice2Image)** → Menampilkan gambar dadu.

- **Button (btnRoll)** → Tombol untuk roll dadu.

E. Tautan GIT

<https://github.com/gr1ff0m/Pemrograman-Mobile-Praktikum>