LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1

ANDROID BASIC WITH KOTLIN

Oleh:

Nazmi Hakim NIM. 2310817210012

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MARET 2025

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 Prakitkum ini dikerjakan oleh:

Nama Praktikan : Nazmi Hakim
NIM : 2310817210012

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

DAFTAR GAMBAR

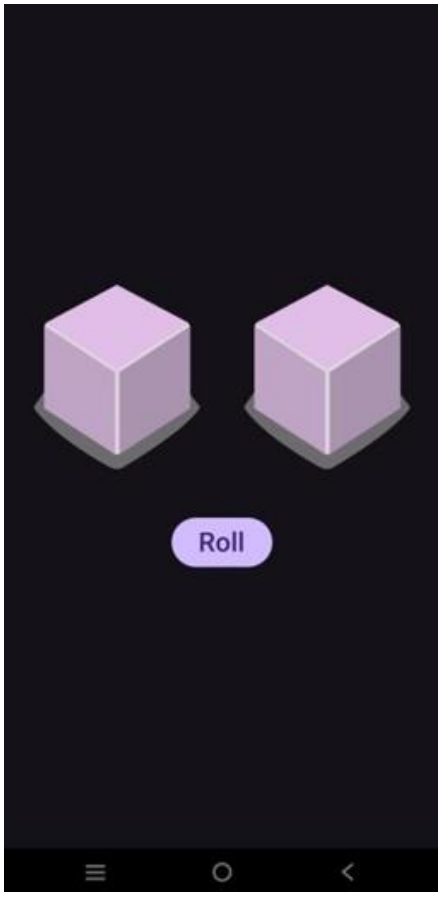
DAFTAR TABEL

SOAL 1

Soal Praktikum:

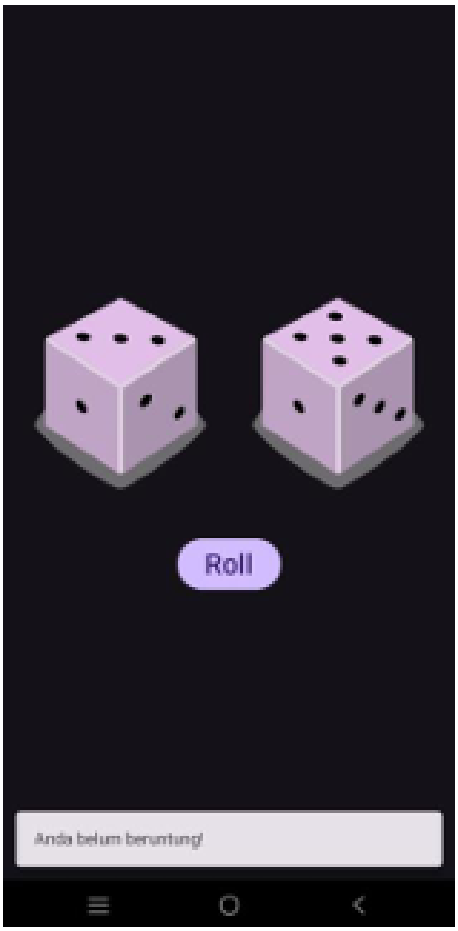
Buatlah sebuah aplikasi yang dapat menampilkan 2 buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol "Roll". Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



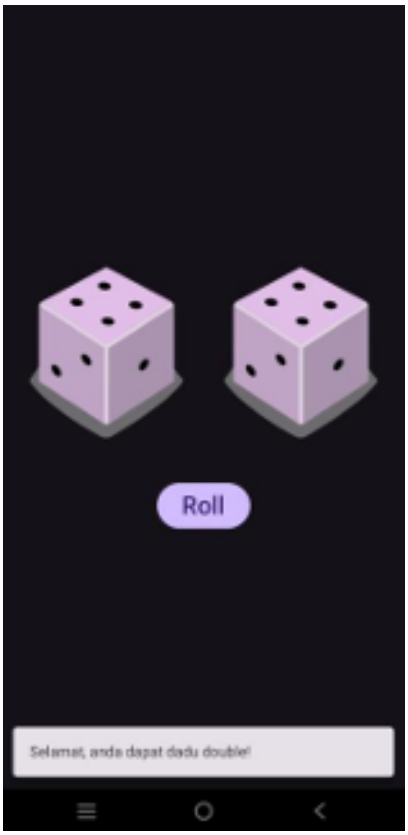
Gambar 1. Tampilan Awal Aplikasi

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

XML :

activity_main.xml :

Tabel 1. Source Code Jawaban Soal 1 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"
```

```
    android:gravity="center"
    android:padding="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center"
        android:layout_marginBottom="16dp">

        <ImageView
            android:id="@+id/imageView1"
            android:layout_width="0dp"
            android:layout_height="200dp"
            android:layout_weight="1"
            android:contentDescription="Dice 1"
            android:src="@drawable/dice_0"
            android:scaleType="centerCrop"/>

        <ImageView
            android:id="@+id/imageView2"
            android:layout_width="0dp"
            android:layout_height="200dp"
            android:layout_weight="1"
            android:contentDescription="Dice 2"
            android:src="@drawable/dice_0"
            android:scaleType="centerCrop"/>
    </LinearLayout>

    <Button
        android:id="@+id/rollButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:backgroundTint="@android:color/holo_purple"
        android:text="Roll" />
</LinearLayout>
```

MainActivity.kt :

Tabel 2. Source Code Jawaban Soal 1 XML

```
package com.example.dicerollerxml

import android.os.Bundle
import android.widget.Button
import android.widget.ImageView
import androidx.appcompat.app.AppCompatActivity
import com.google.android.material.snackbar.Snackbar

class MainActivity : AppCompatActivity() {
```

```
10
11      lateinit var imageView1: ImageView
12      lateinit var imageView2: ImageView
13      lateinit var rollButton: Button
14
15      override fun onCreate(savedInstanceState: Bundle?) {
16          super.onCreate(savedInstanceState)
17          setContentView(R.layout.activity_main)
18
19          imageView1 = findViewById(R.id.imageView1)
20          imageView2 = findViewById(R.id.imageView2)
21          rollButton = findViewById(R.id.rollButton)
22
23          rollButton.setOnClickListener {
24              val hasil = (1..6).random()
25              val hasil2 = (1..6).random()
26
27              setImage(imageView1, hasil)
28              setImage(imageView2, hasil2)
29
30              val message = if (hasil == hasil2) "Selamat, anda
31  dapat dadu double!" else "Anda belum beruntung!"
32              Snackbar.make(findViewById(android.R.id.content),
33  message, Snackbar.LENGTH_SHORT).show()
34          }
35      }
36
37      private fun setImage(imageView: ImageView, hasil: Int) {
38          val imageResId = when (hasil) {
39              1 -> R.drawable.dice_1
40              2 -> R.drawable.dice_2
41              3 -> R.drawable.dice_3
42              4 -> R.drawable.dice_4
43              5 -> R.drawable.dice_5
44              else -> R.drawable.dice_6
45          }
46          imageView.setImageResource(imageResId)
      }
  }
```

Jetpack Compose :

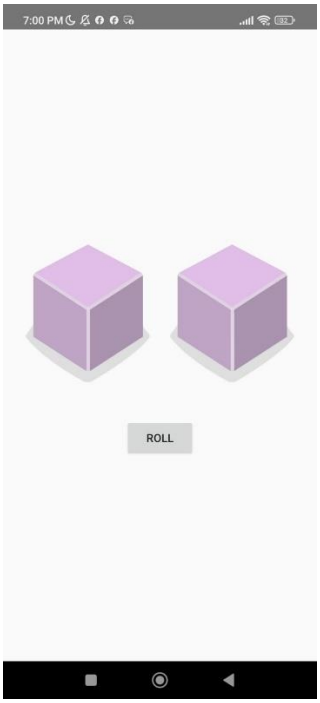
MainActivity.kt :

Tabel 3. Source Code Jawaban Soal 1 Jetpack Compose

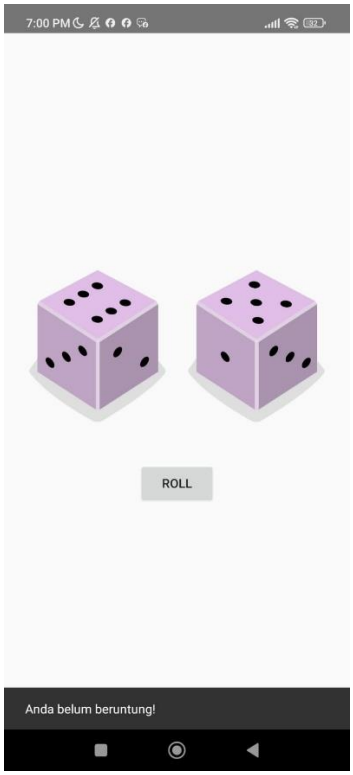
```
1  package com.example.dicerollercompose
2
3  import android.os.Bundle
4  import androidx.activity.ComponentActivity
5  import androidx.activity.compose.setContent
6  import androidx.compose.foundation.Image
7  import androidx.compose.foundation.layout.*
8  import androidx.compose.material3.*
9  import androidx.compose.runtime.*
10 import androidx.compose.ui.Alignment
11 import androidx.compose.ui.Modifier
12 import androidx.compose.ui.res.painterResource
13 import androidx.compose.ui.tooling.preview.Preview
14 import androidx.compose.ui.unit.dp
15 import
16 com.example.dicerollercompose.ui.theme.DiceRollerComposeTheme
17 import kotlinx.coroutines.launch
18
19 class MainActivity : ComponentActivity() {
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22         setContent {
23             DiceRollerComposeTheme {
24                 DiceRollerApp()
25             }
26         }
27     }
28 }
29
30 @Preview(showBackground = true)
31 @Composable
32 fun DiceRollerApp() {
33     DiceWithButtonAndImage()
34 }
35
36 @Composable
37 fun DiceWithButtonAndImage() {
38     var hasil by remember { mutableStateOf(0) }
39     var hasil2 by remember { mutableStateOf(0) }
40     val snackbarHostState = remember { SnackbarHostState() }
41     val coroutineScope = rememberCoroutineScope()
42
43     fun getDiceImage(hasil: Int) = when (hasil) {
44         0 -> R.drawable.dice_0
45         1 -> R.drawable.dice_1
46         2 -> R.drawable.dice_2
```

```
47          3 -> R.drawable.dice_3
48          4 -> R.drawable.dice_4
49          5 -> R.drawable.dice_5
50          else -> R.drawable.dice_6
51      }
52
53      val gambarDadu = getDiceImage(hasil)
54      val gambarDadu2 = getDiceImage(hasil2)
55
56      Box(
57          modifier = Modifier.fillMaxSize(),
58          contentAlignment = Alignment.Center
59      ) {
60          Column(
61              horizontalAlignment = Alignment.CenterHorizontally
62          ) {
63              Row(
64                  horizontalArrangement = Arrangement.Center,
65                  verticalAlignment = Alignment.CenterVertically
66              ) {
67                  Image(painter   =   painterResource(gambarDadu),
68  contentDescription = null, modifier = Modifier.size(200.dp))
69                  Image(painter  =  painterResource(gambarDadu2),
70  contentDescription = null, modifier = Modifier.size(200.dp))
71              }
72              Spacer(modifier = Modifier.height(16.dp))
73              Button(onClick = {
74                  hasil = (1..6).random()
75                  hasil2 = (1..6).random()
76                  val message = if (hasil == hasil2) "Selamat, anda
77  dapat dadu double!" else "Anda belum beruntung!"
78
79                  coroutineScope.launch {
80
81  snackbarHostState.currentSnackbarData?.dismiss()
82                      snackbarHostState.showSnackbar(message,
83  duration = SnackbarDuration.Short)
84                  }
85              }) {
86                  Text(text = "Roll")
87              }
88          }
89
90          SnackbarHost(
91              hostState = snackbarHostState,
92              modifier = Modifier
                    .fillMaxWidth()
                    .align(Alignment.BottomCenter)
                    .padding(bottom = 16.dp)
          )
      }
  }
```

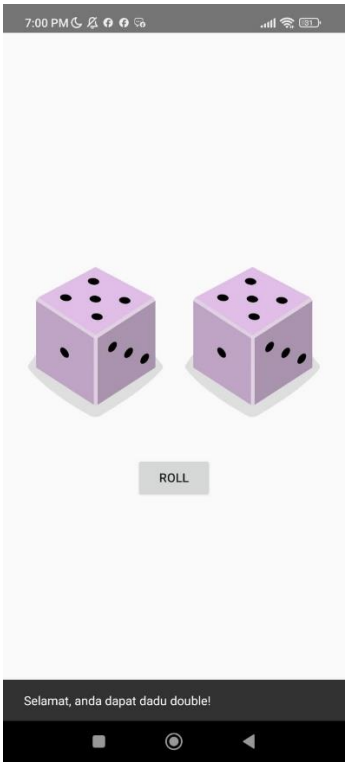
B. Output Program



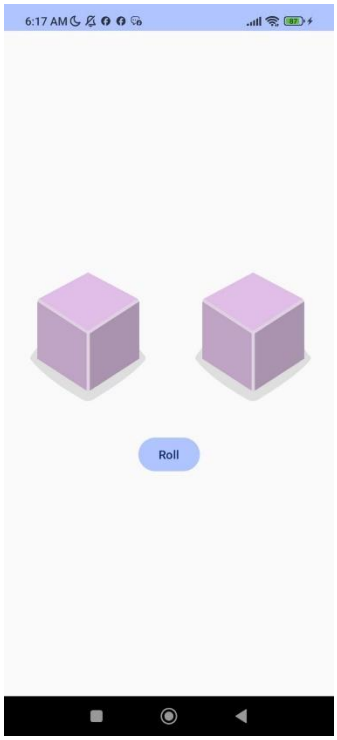
Gambar 4. Screenshot Hasil Jawaban Soal 1 XML



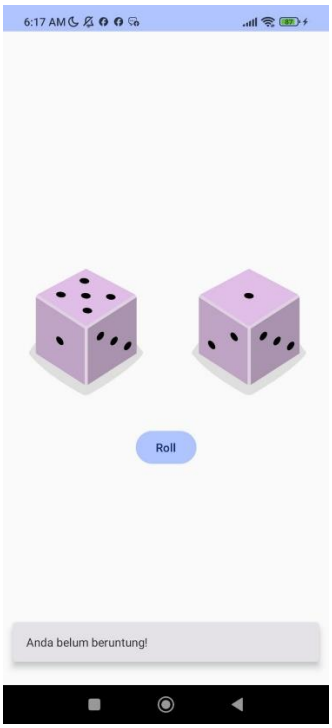
Gambar 5. Screenshot Hasil Jawaban Soal 1 XML



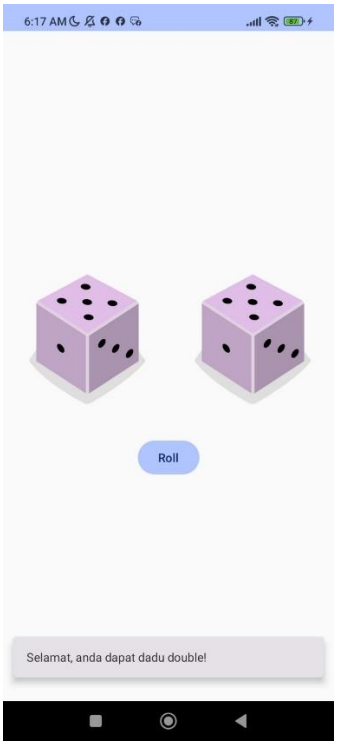
Gambar 6. Screenshot Hasil Jawaban Soal 1 XML



Gambar 7. Screenshot Hasil Jawaban Soal 1 Jetpack Compose



Gambar 8. Screenshot Hasil Jawaban Soal 1 Jetpack Compose



Gambar 9. Screenshot Hasil Jawaban Soal 1 Jetpack Compose

C. Pembahasan

XML :

activity_main.xml :

Line 1, deklarasi standar XML yang menandai awal dokumen.
Line 2, elemen root layout: LinearLayout, menyusun elemen secara vertikal karena android:orientation="vertical" (line 5). xmlns:android adalah deklarasi namespace Android agar atribut seperti android:layout_width dikenali.
Line 3 dan 4, Layout akan mengisi seluruh lebar dan tinggi layar menggunakan ”match_parent”
Line 5, semua elemen anak (children) akan ditata dari atas ke bawah.
Line 6, seluruh isi LinearLayout akan di-tengah-kan secara horizontal dan vertikal.
Line 7, memberi jarak 16dp dari sisi layout terhadap konten (padding di semua sisi).
Line 9 sampai 14, Ini adalah layout horizontal di dalam layout utama, untuk menampung 2 gambar dadu berdampingan. layout_marginBottom="16dp" memberikan jarak antara baris gambar dan tombol di bawahnya.
Line 16 sampai 23, @+id/imageView1 adalah ID unik yang bisa diakses dari Kotlin. layout_width="0dp" + layout_weight="1" → Membagi lebar sama rata dengan ImageView lain. src="@drawable/dice_0" → Gambar awal adalah dadu kosong (nilai 0). scaleType="centerCrop" → Gambar dipotong supaya penuh tanpa merusak rasio.
Line 25 sampai 32, gambar dadu kedua, sama seperti line 16 sampai 23.
Line 35 sampai 41, menampilkan tombol yang ukurannya mengikuti teks atau konten didalamnya, dengan warna latar belakang ungu dan teks ”Roll”

MainActivity.kt :

Line 1, adalah nama package sesuai dengan struktur project
Line 3 sampai 7, Mengimpor komponen View (ImageView, Button) dan Snackbar dari Material Design.
Line 9, adalah aktivitas utama aplikasi, mewarisi AppCompatActivity.
Line 11 sampai 13, Variabel View yang akan dihubungkan ke XML nanti dengan findViewById.
Line 15 sampai 16, Fungsi yang dijalankan saat aplikasi pertama kali dibuka.
Line 17, Menghubungkan tampilan layout XML ke kode Kotlin (menampilkan activity_main.xml).
Line 19 sampai 21, Menghubungkan variabel di Kotlin dengan komponen yang ada di XML menggunakan ID masing-masing.
Line 23 sampai 32, menambahkan action ke tombol atau button, yaitu jika di-klik akan menghasilkan angka acak dari 1 sampai 6 dan menyimpannya untuk variabel hasil dan hasil2 secara terpisah. Lalu memanggil fungsi setImage untuk mengganti gambar berdasarkan angka, lalu jika nilai dari hasil dan hasil2 sama maka akan mengisi variabel message dengan teks "Selamat, anda dapat dadu double!", Jika tidak maka akan diisi dengan "Anda belum beruntung!", lalu menampilkan notifikasi snackbar dengan pesan yang ada di variabel message dengan waktu yang singkat

Line 35 sampai 44, adalah fungsi untuk mengatur gambar dadu sesuai dengan variabel hasil di fungsi SetImage. Jika hasil adalah 1 maka akan menapilkan gambar dadu 1, jika 2 maka akan memperlihankan gambar dadu 2, dan seterusnya.

Jetpack Compose :

MainActivity.kt:

Line 1, package com.example.dicerollercompose menentukan namespace aplikasi yang berguna untuk membedakan aplikasi android diceroller dari aplikasi lainnya.
Line 3 sampai 16, berisi import import yang berfungsi agar bisa memakai kembali fungsi, komponen, kelas, dan fitur dari library lain ke dalam kode seperti :

1. android.os.Bundle :
   Digunakan untuk menyimpan dan mengelola data yang dikirimkan antar aktivitas dalam aplikasi Android. Dalam kode ini, digunakan di dalam onCreate untuk menangani instance state aplikasi.
2. androidx.activity.ComponentActivity :
   Merupakan kelas dasar untuk aktivitas yang menggunakan Jetpack Compose. MainActivity mewarisi ComponentActivity agar bisa menjalankan UI berbasis Compose.
3. androidx.activity.compose.setContent :
   Digunakan untuk menetapkan UI aplikasi menggunakan Jetpack Compose. Dalam kode ini, setContent memanggil DiceRollerComposeTheme { DiceRollerApp() } untuk menampilkan UI utama.
4. androidx.compose.foundation.Image :
   Komponen Jetpack Compose yang digunakan untuk menampilkan gambar. Dalam kode ini, digunakan untuk menampilkan dadu dengan gambar yang sesuai.
5. androidx.compose.foundation.layout.* :
   Mengimpor berbagai komponen tata letak seperti Box, Column, Row, Spacer, dan Modifier. Digunakan untuk menyusun elemen-elemen UI seperti posisi dadu, tombol, dan snackbar.
6. androidx.compose.material3.* :
   Mengimpor komponen dari Material Design 3, termasuk Button, Text, SnackbarHost, dan SnackbarHostState. Digunakan untuk membuat tombol "Roll" dan menampilkan snackbar sebagai notifikasi.
7. *androidx.compose.runtime .* :
   Mengimpor fungsi-fungsi yang mendukung state management seperti remember dan mutableStateOf. Digunakan untuk menyimpan dan memperbarui nilai dadu saat tombol ditekan.
8. androidx.compose.ui.Alignment :
   Digunakan untuk mengatur perataan elemen dalam tata letak, misalnya Alignment.Center dalam Box untuk memusatkan dadu.
9. androidx.compose.ui.Modifier:
   Digunakan untuk mengubah atau menyesuaikan tampilan dan perilaku elemen UI. Dalam kode ini, digunakan untuk mengatur ukuran gambar, margin, padding, dan posisi elemen.
10. androidx.compose.ui.res.painterResource :

Digunakan untuk memuat gambar dari drawable berdasarkan ID sumber daya. Dalam kode ini, digunakan untuk menampilkan gambar dadu berdasarkan nilai yang dihasilkan.

11. androidx.compose.ui.tooling.preview.Preview :
    Digunakan untuk menampilkan pratinjau UI dalam Android Studio. Dalam kode ini, digunakan di DiceRollerApp() agar tampilan aplikasi bisa dilihat sebelum dijalankan.
12. androidx.compose.ui.unit.dp :
    Digunakan untuk mengatur ukuran elemen dalam satuan density-independent pixels (dp). Contohnya, Modifier.size(200.dp) digunakan untuk menentukan ukuran gambar dadu.
13. com.example.dicerollercompose.ui.theme.DiceRollerComposeTheme :
    Mengimpor tema khusus yang telah dibuat dalam proyek untuk konsistensi tampilan aplikasi.
14. kotlinx.coroutines.launch
    Digunakan untuk menjalankan operasi asinkron dalam coroutine. Dalam kode ini, digunakan untuk menampilkan snackbar secara non-blocking ketika tombol ditekan.

Line 18, Mendeklarasikan kelas MainActivity, mewarisi ComponentActivity.
Line 19, Override onCreate, dipanggil saat Activity dibuat.
Line 20, Memanggil super.onCreate() dari parent class.
Line 21 sampai 27, Mengatur tampilan UI menggunakan Jetpack Compose dengan tema dan fungsi utama DiceRollerApp().
Line 29, @Preview(showBackground = true) Menampilkan tampilan UI dalam mode preview di Android Studio.
Line 30, Mendeklarasikan DiceRollerApp sebagai fungsi compose, menandakan bahwa fungsi ini adalah UI yang dapat dikomposisi ulang (Jetpack Compose).
Line 31, DiceRollerApp(), Fungsi utama yang memanggil DiceWithButtonAndImage() untuk menampilkan dadu dan tombol roll.
Line 32 sampai 33, Memanggil UI utama dari aplikasi, DiceWithButtonAndImage().
Line 35, Mendeklarasikan DiceWithButtonAndImage sebagai fungsi compose, menandakan bahwa fungsi ini adalah UI yang dapat dikomposisi ulang (Jetpack Compose).
Line 36, Fungsi utama yang menyusun antarmuka dan logika aplikasi.
Line 37 sampai 40, hasil & hasil2 adalah variabel yang menyimpan angka dadu yang akan ditampilkan, mutableStateOf(0) artinya nilai awalnya adalah 0. remember memastikan nilai state tetap bertahan selama UI tidak direkomposisi ulang. snackbarHostState menyimpan state untuk snackbar yang menampilkan pesan notifikasi. coroutineScope digunakan untuk menjalankan proses asynchronous dalam coroutine.
Line 42 getDiceImage(hasil: Int) adalah fungsi untuk mengembalikan gambar dadu berdasarkan nilai hasil.
Line 43 sampai 49, Struktur kontrol seperti switch-case (when). Jika hasil adalah 0, akan menampilkan dice_0, jika 1 akan dice_1, dan seterusnya.
Line 52 dan 53, gambarDadu & gambarDadu2 menyimpan gambar dadu yang sesuai dengan nilai hasil dan hasil2.
Line 55 sampai 58, membuat layout Box dengan ukuran layar penuh dan isi di tengah layar.
Line 59 sampai 61, menyusun elemen UI secara vertikal dan sejajar tengah secara horizontal.
Line 62 sampai 65, menyusun agar dua dadu ditempatkan secara horizontal di tengah layar.
Line 66 dan 67, menampilkan dua gambar dadu dengan ukuran 200dp.

Line 69, menambahkan jarak vertikal 16dp sebelum tombol.
Line 70 sampai 73, menampilkan sebuah tombol atau button yang apabila di tekan atau klik maka akan mengacak variabel hasil dan hasil2 dari angka 1 sampai angka 6 menggunakan fungsi random. Apabila nilai Dari variabel hasil dan hasil2 berbeda maka akan menampilkan pesan ”Anda belum beruntung!”, sebaliknya ”Selamat, anda dapat dadu double!”
Line 75, coroutineScope.launch menjalankan operasi snackbar dalam coroutine.
Line 76, snackbarHostState.currentSnackbarData?.dismiss() berguna jika ada snackbar yang sedang ditampilkan, tutup terlebih dahulu.
Line 77, snackbarHostState.showSnackbar(message, duration = SnackbarDuration.Short) → Menampilkan snackbar dengan pesan yang sesuai (dari fungsi message), dan dengan durasi yang singkat menggunakan duration = SnackbarDuration.Short
Line 79 sampai 81, menampilkan teks ”Roll”
Line 84, SnackbarHost menampilkan snackbar di layar.
Line 85, hostState = snackbarHostState digunakan untuk menghubungkan SnackbarHost dengan state pengatur snackbar agar dapat menampilkan pesan notifikasi.
Line 86, modifier = Modifier berfungsi untuk mengatur ukuran, posisi, dan tata letak visual dari elemen UI
Line 87, Modifier.fillMaxWidth() memastikan snackbar mengisi lebar layar
Line 88, align(Alignment.BottomCenter) memposisikan snackbar di bawah layar.
Line 89, padding(bottom = 16.dp) → memberi jarak 16 dp dari bawah.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
https://github.com/NazmiHakim/Pemrograman-Mobile/tree/main/Modul1