LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1

ANDROID BASIC WITH KOTLIN

Oleh:

Muhammad Raihan NIM. 2310817110008

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 Raihan
NIM : 2310817110008

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

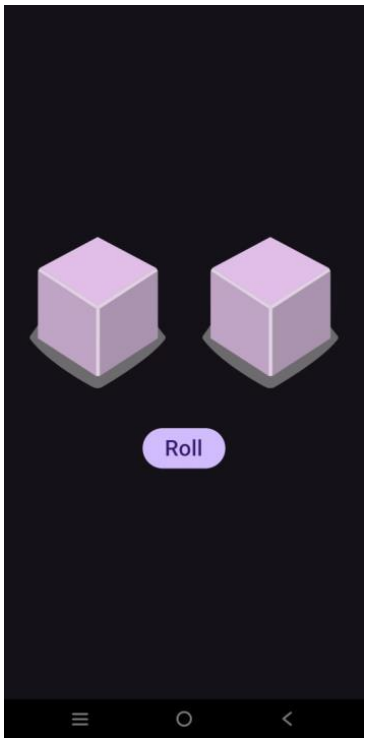
DAFTAR TABEL

SOAL 1

Soal Praktikum:

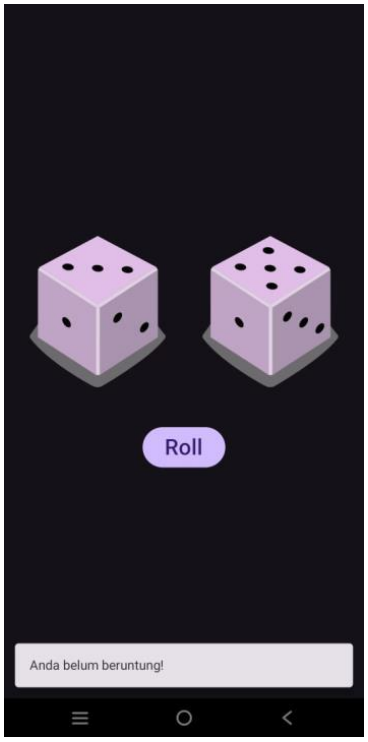
Buatlah sebuah aplikasi yang dapat menampilkan 2 buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



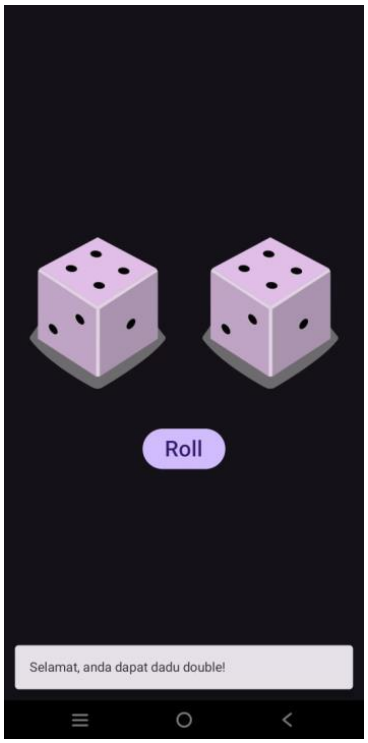
Gambar 1. Tampilan Awal Aplikasi

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

1) Versi Jetpack Compose

MainActivity.kt

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

```
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.Image
8   import androidx.compose.foundation.layout.Arrangement
9   import androidx.compose.foundation.layout.Column
10  import androidx.compose.foundation.layout.Row
11  import androidx.compose.foundation.layout.Spacer
12  import androidx.compose.foundation.layout.fillMaxSize
13  import androidx.compose.foundation.layout.height
14  import androidx.compose.foundation.layout.padding
15  import androidx.compose.foundation.layout.size
16  import
    androidx.compose.foundation.layout.wrapContentSize
17  import androidx.compose.material3.Button
18  import androidx.compose.material3.MaterialTheme
19  import androidx.compose.material3.Scaffold
20  import androidx.compose.material3.SnackbarDuration
21  import androidx.compose.material3.SnackbarHost
22  import androidx.compose.material3.SnackbarHostState
23  import androidx.compose.material3.Surface
24  import androidx.compose.material3.Text
25  import androidx.compose.runtime.Composable
26  import androidx.compose.runtime.getValue
27  import androidx.compose.runtime.mutableStateOf
28  import androidx.compose.runtime.remember
29  import
    androidx.compose.runtime.rememberCoroutineScope
30  import androidx.compose.runtime.setValue
```

```
31 import androidx.compose.ui.Alignment
32 import androidx.compose.ui.Modifier
33 import androidx.compose.ui.res.painterResource
34 import androidx.compose.ui.res.stringResource
35 import androidx.compose.ui.tooling.preview.Preview
36 import androidx.compose.ui.unit.dp
37 import androidx.compose.ui.unit.sp
38 import
   com.example.dicecompose.ui.theme.DiceComposeTheme
39 import kotlinx.coroutines.launch
40
41 class MainActivity : ComponentActivity() {
42     override     fun     onCreate(savedInstanceState:
   Bundle?) {
43         super.onCreate(savedInstanceState)
44         enableEdgeToEdge()
45         setContent {
46             DiceComposeTheme {
47                 Surface (
48                     modifier                          =
   Modifier.fillMaxSize(),
49                     color                             =
   MaterialTheme.colorScheme.background
50                 ) {
51                     DiceRollerApp()
52                 }
53             }
54         }
55     }
56 }
57
58 @Preview(showBackground = true)
59 @Composable
60 fun DiceRollerApp() {
61     DiceWithButtonAndImage(modifier = Modifier
62         .fillMaxSize()
63         .wrapContentSize(Alignment.Center)
64     )
65 }
66
67 @Composable
68 fun   DiceWithButtonAndImage(modifier:   Modifier   =
   Modifier) {
69     var result1 by remember { mutableStateOf(0) }
70     var result2 by remember { mutableStateOf(0) }
71
```

```
72      val snackbarHostState = remember { SnackbarHostState() }
73      val coroutineScope = rememberCoroutineScope()
74
75      val imageResource1 = getDiceImageResource(result1)
76      val imageResource2 = getDiceImageResource(result2)
77
78      Scaffold (
79          snackbarHost = { SnackbarHost(snackbarHostState) },
80          content = {padding ->
81              Column(
82                  modifier = modifier
83                      .padding(padding),
84                  verticalArrangement = Arrangement.Center,
85                  horizontalAlignment = Alignment.CenterHorizontally
86              ) {
87                  Row {
88                      Image(
89                          modifier = Modifier.size(200.dp),
90                          painter = painterResource(imageResource1),
91                          contentDescription = result1.toString()
92                      )
93                      Image(
94                          modifier = Modifier.size(200.dp),
95                          painter = painterResource(imageResource2),
96                          contentDescription = result2.toString()
97                      )
98                  }
99                  Spacer(modifier = Modifier.height(16.dp))
100                 Button(onClick = {
101                     result1 = (1..6).random()
102                     result2 = (1..6).random()
103
104                     coroutineScope.launch {
```

```
105
    snackbarHostState.showSnackbar(
106                             message                 =
    getSnackbarText(result1, result2),
107                             duration                =
    SnackbarDuration.Short
108                         )
109                     }
110                 }) {

111 Text(stringResource(R.string.roll), fontSize = 20.sp)
112                 }
113             }
114         }
115     )
116 }
117
118 fun getDiceImageResource (result: Int): Int {
119     return when (result) {
120         1 -> R.drawable.dice_1
121         2 -> R.drawable.dice_2
122         3 -> R.drawable.dice_3
123         4 -> R.drawable.dice_4
124         5 -> R.drawable.dice_5
125         6 -> R.drawable.dice_6
126         else -> R.drawable.dice_0
127     }
128 }
129
130 fun getSnackbarText (result1: Int, result2: Int):
    String {
131     return if(result1 == result2) "Selamat, Anda
    mendapatkan angka double!"
132     else "Anda belum beruntung!"
133 }
```

2) Versi XML

MainActivity.kt

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

```
1   package com.example.dicexml
2
3   import android.os.Bundle
4   import android.widget.Button
5   import android.widget.ImageView
6   import androidx.activity.enableEdgeToEdge
```

```
7  import androidx.appcompat.app.AppCompatActivity
8  import
   androidx.constraintlayout.widget.ConstraintLayout
9  import com.google.android.material.snackbar.Snackbar
10
11 class MainActivity : AppCompatActivity() {
12     override     fun     onCreate(savedInstanceState:
   Bundle?) {
13         super.onCreate(savedInstanceState)
14         enableEdgeToEdge()
15         setContentView(R.layout.activity_main)
16         val     mainLayout:     ConstraintLayout     =
   findViewById(R.id.main)
17         val         dice1:         ImageView         =
   findViewById(R.id.dice1)
18         val         dice2:         ImageView         =
   findViewById(R.id.dice2)
19         val         rollBtn:         Button         =
   findViewById(R.id.rollBtn)
20
21         rollBtn.setOnClickListener {
22             val result1 = (1..6).random()
23             val result2 = (1..6).random()
24
25             val           imageResource1           =
   getDiceImageResource(result1)
26             val           imageResource2           =
   getDiceImageResource(result2)
27
28             dice1.setImageResource(imageResource1)
29             dice2.setImageResource(imageResource2)
30
31             val            snackbarText            =
   getSnackbarText(result1, result2)
32             Snackbar.make(mainLayout,   snackbarText,
   Snackbar.LENGTH_SHORT).show()
33         }
34     }
35 }
36
37 fun getDiceImageResource (result: Int): Int {
38     return when (result) {
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
```

```
        6 -> R.drawable.dice_6
        else -> R.drawable.dice_0
    }
}

fun getSnackbarText (result1: Int, result2: Int): String {
    return if(result1 == result2) "Selamat, Anda mendapatkan angka double!"
    else "Anda belum beruntung!"
}
```

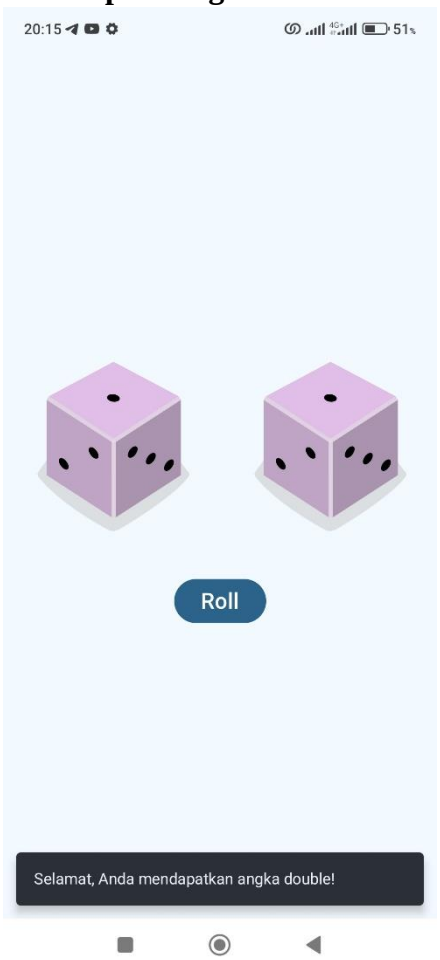
activity_main.xml

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

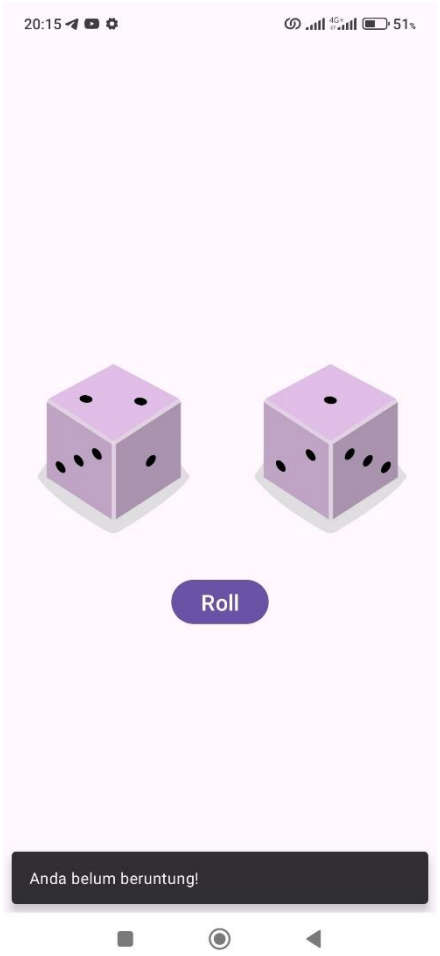
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintBottom_toBottomOf="parent"
            android:orientation="horizontal"
            android:gravity="center"
            android:layout_marginBottom="15dp">

            <ImageView
                android:id="@+id/dice1"
                android:layout_width="wrap_content"
```

```
                android:layout_height="200dp"
                android:src="@drawable/dice_0"

android:contentDescription="@string/dice1"/>
            <ImageView
                android:id="@+id/dice2"
                android:layout_width="wrap_content"
                android:layout_height="200dp"
                android:src="@drawable/dice_0"

android:contentDescription="@string/dice2"/>
        </LinearLayout>

        <Button
            android:id="@+id/rollBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/roll"
            android:textSize="20sp" />
    </LinearLayout>



</androidx.constraintlayout.widget.ConstraintLayout>
```

B. Output Program



Gambar 4. Screenshot Hasil Jawaban Soal 1 Versi Jetpack Compose



Gambar 5. Screenshot Hasil Jawaban Soal 1 Versi XML

C. Pembahasan

1) Versi Jetpack Compose

MainActivity.kt

Pada baris [1], terdapat nama package file Kotlin yang dideklarasikan

Pada baris [3] sampai [39], terdapat beberapa komponen yang diimpor untuk kebutuhan pembuatan aplikasi.

Pada baris [41], terdapat class MainActivity yang dibuat, dimana class tersebut mewarisi dari ComponentActivity() yang merupakan activity yang mendukung Jetpack Compose.

Pada baris [42], terdapat fungsi onCreate() yang di-override, dimana fungsi tersebut akan dipanggil saat activity dimulai.

Pada baris [43], terdapat onCreate() yang dipanggil dari superclass-nya.

Pada baris [44], terdapat fungsi enableEdgeToEdge() yang dipanggil, dimana fungsi ini berguna untuk membuat tampilan layar aplikasi bisa menggunakan seluruh area layar.

Pada baris [45], terdapat fungsi setContent yang merupakan fungsi khusus dari Jetpack Compose, dimana fungsi ini berfungsi untuk menetapkan isi UI yang akan ditampilkan.

Pada baris [46], terdapat fungsi DiceComposeTheme yang berfungsi untuk menerapkan tema aplikasi seperti warna, tipografi, dan bentuk komponen pada seluruh UI.

Pada baris [47], terdapat sebuah container dengan nama Surface yang digunakan sebagai latar belakang untuk mengisi layar penuh dengan modifier = Modifier.fillMaxSize() dan mengatur warna latar belakang dengan color = MaterialTheme.colorScheme.background.

Pada baris [51], terdapat fungsi DiceRollerApp() yang dijalankan di dalam Surface, dimana fungsi ini merupakan fungsi yang akan menampilkan UI utama aplikasi.

Pada baris [60], terdapat fungsi DiceRollerApp() yang dibuat dan diberi anotasi @Preview(showBackground = true) untuk menampilkan preview UI aplikasi dengan tampilan latar belakangnya, serta anotasi @Composable yang menyatakan bahwa fungsi ini merupakan fungsi yang akan menampilkan UI di Compose.

Pada baris [61], terdapat fungsi DiceWithButtonAndImage() yang dijalankan di dalam DiceRollerApp(). Disini fungsi ini diberi argumen modifier = Modifier.fillMaxSize() yang dimana berfungsi untuk mengisi seluruh ukuran layar dan wrapContentSize(Alignment.Center) yang berfungsi untuk membuat isi di dalamnya ditaruh di tengah layar. Fungsi DiceWithButtonAndImage() berfungsi untuk menampilkan gambar dadu dan tombol, serta mengatur logikanya saat ditampilkan di layar.

Pada baris [68], terdapat fungsi DiceWithButtonAndImage() yang dibuat, dimana fungsi ini berfungsi untuk membentuk tampilan dan logika dari UI utama. Fungsi ini akan menampilkan dua buah dadu, sebuah tombol untuk mengacak dadu, dan sebuah snackbar saat tombol ditekan. Fungsi ini juga memiliki parameter modifier yang bertipe data Modifier dengan nilai default Modifier kosong. Fungsi ini juga menggunakan anotasi @Composable yang menyatakan bahwa fungsi ini merupakan fungsi yang akan menampilkan UI di Compose.

Pada baris [69] dan [70], terdapat variabel result1 dan result2 yang berfungsi untuk menyimpan nilai dadu dari 1-6. Adapun statement remember pada kedua variabel tersebut berfungsi agar Compose tetap mengingat variabel meskipun terjadi perubahan pada UI. Statement mutableStateOf(0) mendefinisikan nilai awalnya adalah 0.

Pada baris [72], terdapat variabel snackbarHostState yang akan digunakan untuk mengatur dan menampilkan Snackbar.

Pada baris [73], terdapat variabel coroutineScope yang akan diperlukan untuk menjalankan fungsi showSnackbar(), dikarenakan fungsi ini merupakan fungsi suspend yang dimana harus ada coroutineScope untuk menjalankannya.

Pada baris [75] dan [76], terdapat variabel imageResource1 dan imageResource2 yang berfungsi untuk menyimpan resource gambar berdasarkan angka dadu dari result1 dan result2. Variabel imageResource1 akan menyimpan gambar berdasarkan hasil dari fungsi getDiceImageResource() dengan argumen result1 dan variabel imageResource2 akan menyimpan gambar berdasarkan hasil dari fungsi getDiceImageResource() dengan argumen result2.

Pada baris [78], terdapat Scaffold yang merupakan layout yang akan memberikan area untuk Snackbar dan konten utama. Kode di dalamnya dimulai dari baris [78] sampai baris [115].

Pada baris [79], terdapat snackbarHost yang didefinisikan berdasarkan snackbarHostState yang telah dibuat sebelumnya.

Pada baris [80], terdapat content yang merupakan tempat untuk konten utama yang akan berisi gambar dadu dan button.

Pada baris [81], terdapat Column yang akan menumpuk item secara vertikal. Item yang akan ditumpuk secara vertikal antara lain adalah Row, Spacer, dan Button. Column ini juga diberi argumen agar menggunakan padding dari content untuk menghindari overlap, membuat posisinya rata tengah secara vertikal dengan verticalArrangement = Arrangement.Center, dan membuat posisinya rata tengah secara horizontal dengan horizontalAlignment = Alignment.CenterHorizontally.

Pada baris [87], terdapat Row yang akan menumpuk item secara horizontal. Row disini berfungsi untuk menampilkan 2 buah item Image dadu secara horizontal.

Pada baris [88], terdapat Image yang resourcenya mengambil dari variabel imageResource1, ini akan menampilkan dadu pertama yang berada di sebelah kiri. Image ini diatur size-nya sebesar 200 dp.

Pada baris [93], terdapat Image yang resourcenya mengambil dari variabel imageResource2, ini akan menampilkan dadu kedua yang berada di sebelah kanan. Image ini diatur size-nya sebesar 200 dp.

Pada baris [99], terdapat Spacer yang berfungsi untuk memberi jarak antara Row dan Button setinggi 16 dp.

Pada baris [100], terdapat Button yang dibuat untuk mengacak dadu. Saat di klik, variabel result1 dan result2 akan diberi nilai acak dalam rentang angka 1-6. Lalu terdapat coroutineScope.launch untuk menjalankan fungsi suspend showSnackbar dari snackbarHostState. Snackbar akan menampilkan pesan yang didapat dari fungsi

getSnackbarText yang diisi dengan argumen result1 dan result2 secara berurutan. Durasi snackbar saat tampil adalah Short.

Pada baris [111], terdapat Text dimana ini berfungsi untuk memberikan teks di dalam Button. Isi teks tersebut diambil dari resource file string.xml dengan nama “roll” yang isinya adalah Roll. Teks ini ukuran font-nya diatur menjadi sebesar 20 sp.

Pada baris [118], terdapat fungsi getDiceImageResource() yang dimana telah digunakan sebelumnya untuk mengambil resource gambar dadu berdasarkan result yang didapatkan saat menekan Button. Fungsi ini akan mengambil parameter result dan akan mengembalikan resource gambar dadu berdasarkan nilai dari result tersebut. Jika result bernilai 1, maka akan mengembalikan resource dice_1. Jika result bernilai 2, maka akan mengembalikan resource dice_2. Begitu seterusnya sampai result bernilai 6, maka akan mengembalikan resource dice_6. Jika result bernilai selain dari angka 1-6, maka akan mengembalikan resource dice_0 yang dimana itu merupakan dadu kosong.

Pada baris [130], terdapat fungsi getSnackbarText() yang telah digunakan sebelumnya pada fungsi showSnackbar yang berfungsi untuk mengembalikan string yang akan ditampilkan sebagai pesan pada Snackbar. Fungsi ini memiliki parameter result1 dan result2 yang dimana jika nilai dari result1 dan result2 sama, maka akan mengembalikan string “Selamat, Anda mendapatkan angka double!”. Sedangkan jika tidak sama, maka akan mengembalikan string “Anda belum beruntung!”.

2) Versi XML

MainActivity.kt:

Pada baris [1], terdapat nama package file Kotlin yang dideklarasikan

Pada baris [3] sampai [9], terdapat beberapa komponen yang diimpor untuk kebutuhan pembuatan aplikasi.

Pada baris [11], terdapat class MainActivity yang dibuat, dimana class tersebut mewarisi dari AppCompactActivity() yang merupakan class bawaan Android yang menyediakan fitur activity modern dan digunakan pada project yang menggunakan XML.

Pada baris [12], terdapat fungsi onCreate() yang di-override, dimana fungsi tersebut akan dipanggil saat activity dimulai.

Pada baris [13], terdapat onCreate() yang dipanggil dari superclass-nya.

Pada baris [14], terdapat fungsi enableEdgeToEdge() yang dipanggil, dimana fungsi ini berguna untuk membuat tampilan layar aplikasi bisa menggunakan seluruh area layar.

Pada baris [15], terdapat fungsi setContentView yang berfungsi untuk mendeklarasikan layout pada activity_main.xml untuk ditampilkan ke layar.

Pada baris [16], terdapat variabel mainLayout yang dideklarasikan untuk menyimpan layout dari elemen ConstraintLayout dengan id ‘main’ yang berasal dari file xml.

Pada baris [17], terdapat variabel dice1 yang dideklarasikan untuk menyimpan resource gambar dari elemen ImageView dengan id ‘dice1’ yang berasal dari file xml.

Pada baris [18], terdapat variabel dice2 yang dideklarasikan untuk menyimpan resource gambar dari elemen ImageView dengan id ‘dice2’ yang berasal dari file xml.

Pada baris [19], terdapat variabel rollBtn yang dideklarasikan untuk menyimpan elemen Button dengan id ‘rollBtn’ yang berasal dari file xml.

Pada baris [21], terdapat variabel rollBtn yang diberi fungsi setOnClickListener untuk mengatur logika yang dilakukan ketika Button tersebut ditekan. Kode untuk logikanya terdapat pada baris [22] sampai [32].

Pada baris [22] dan [23], terdapat variabel result1 dan result2 yang dideklarasikan, dimana saat Button diklik maka akan mengisikan kedua variabel ini dengan angka acak dalam rentang 1-6.

Pada baris [25] dan [26], terdapat variabel imageResource1 dan imageResource2 yang berfungsi untuk menyimpan resource gambar kedua dadu. Variabel imageResource1 diisikan dengan nilai yang didapat dari fungsi getDiceImageResource yang diisi dengan argumen result1. Sedangkan variabel imageResource2 diisikan dengan nilai yang didapat dari fungsi getDiceImageResource yang diisi dengan argumen result2.

Pada baris [28] dan [29], terdapat variabel dice1 dan dice2 dengan masing-masing memanggil fungsi setImageResource(). Variabel dice1 memanggil fungsi setImageResource() dengan argumen imageResource1, yang artinya gambar pada dice1 akan digantikan dengan gambar yang ada pada imageResource1. Sedangkan variabel dice2 memanggil fungsi setImageResource() dengan argumen imageResource2, yang artinya gambar pada dice1 akan digantikan dengan gambar yang ada pada imageResource2.

Pada baris [31], terdapat variabel snackbarText yang berfungsi untuk menyimpan isi teks atau pesan yang akan ditampilkan pada Snackbar. Isi dari variabel ini adalah hasil return dari fungsi getSnackbarText dengan argumen result1 dan result2.

Pada baris [32], terdapat class Snackbar yang dipanggil dengan fungsi make. Di dalam fungsi make terdapat argumen mainLayout yang berfungsi sebagai context, snackbarText untuk teks atau pesan yang akan ditampilkan pada snackbar, dan Snackbar.LENGTH_SHORT untuk durasi Snackbar-nya. Selanjutnya langsung dipanggil fungsi show() untuk langsung menampilkan Snackbar pada layar.

Pada baris [37], terdapat fungsi getDiceImageResource() yang dimana telah digunakan sebelumnya untuk mengambil resource gambar dadu berdasarkan result yang didapatkan saat menekan Button. Fungsi ini akan mengambil parameter result dan akan mengembalikan resource gambar dadu berdasarkan nilai dari result tersebut. Jika result bernilai 1, maka akan mengembalikan resource dice_1. Jika result bernilai 2, maka akan mengembalikan resource dice_2. Begitu seterusnya sampai result bernilai 6, maka akan mengembalikan resource dice_6. Jika result bernilai selain dari angka 1-6, maka akan mengembalikan resource dice_0 yang dimana itu merupakan dadu kosong.

Pada baris [49], terdapat fungsi getSnackbarText() yang telah digunakan sebelumnya saat membuat Snackbar yang berfungsi untuk mengembalikan string yang akan ditampilkan sebagai pesan pada Snackbar. Fungsi ini memiliki parameter result1 dan result2 yang dimana jika nilai dari result1 dan result2 sama, maka akan mengembalikan string "Selamat, Anda mendapatkan angka double!". Sedangkan jika tidak sama, maka akan mengembalikan string "Anda belum beruntung!".

activity_main.xml:

Pada baris [1], terdapat kode yang menjelaskan terkait versi XML dan encoding yang digunakan. Disini versinya adalah 1.0 dan encoding-nya adalah UTF-8.

Pada baris [2], terdapat ConstraintLayout yang menjadi dasar latar belakang tampilan aplikasi. Layout ini memiliki id 'main'. Selain itu juga memiliki beberapa atribut seperti layout_width="match_parent" agar lebarnya menjadi seluas layar, layout_height="match_parent" agar tingginya menjadi seluas layar, dan tools:context=".MainActivity" yang berarti akan digunakan sebagai preview di Android Studio.

Pada baris [10], terdapat LinearLayout bagian luar yang berfungsi untuk menampung 2 elemen di dalamnya, yaitu LinearLayout bagian dalam dan sebuah Button. Layout ini diatur lebar dan tingginyanya agar seluas layar. Lalu, layout ini juga diatur agar susunan orientasi elemen di dalamnya menjadi menumpuk secara vertikal. Selanjutnya, layout ini diatur dengan atribut gravity="center" agar elemen-elemen di dalamnya menjadi rata tengah.

Pada baris [16], terdapat LinearLayout bagian dalam yang berfungsi untuk menampung 2 buah gambar dadu. Layout ini diatur lebarnya agar seluas layar dan tingginya menyesuaikan ukuran isi elemen di dalamnya. Posisi layout ini diatur sedemikian rupa dengan ConstraintTop dan ConstraintBottom agar posisinya menjadi rata tengah. Susunan orientasi elemen di dalam layout ini diatur menjadi horizontal. Lalu, layout ini diatur dengan atribut gravity="center" agar elemen dua dalamnya menjadi rata tengah. Selanjutnya, layout ini diberi marginBottom sebesar 15 dp untuk memberi jarak kepada Button.

Pada baris [25], terdapat ImageView dengan id ‘dice1’ yang mengambil sumber gambarnya dari dice_0 pada folder drawable. Gambar dice_0 yang berupa dadu kosong diambil sebagai gambar default sebelum Button ditekan. Gambar ini diberi tinggi sebesar 200 dp dengan lebarnya menyesuaikan. ContentDescription pada gambar ini diambil dari string dengan nama ‘dice1’ pada file string.xml.

Pada baris [31], terdapat ImageView dengan id ‘dice2’ yang mengambil sumber gambarnya dari dice_0 pada folder drawable. Gambar dice_0 yang berupa dadu kosong diambil sebagai gambar default sebelum Button ditekan. Gambar ini diberi tinggi sebesar 200 dp dengan lebarnya menyesuaikan. ContentDescription pada gambar ini diambil dari string dengan nama ‘dice2’ pada file string.xml.

Pada baris [39], terdapat Button dengan id ‘rollBtn’ yang berisi teks dari string dengan nama ‘roll’ dari file string.xml. Lebar dan tinggi Button ini menyesuaikan ukuran isi teks di dalamnya. Ukuran teks di dalam Button ini adalah sebesar 20 sp.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
Versi Jetpack Compose:
https://github.com/raihan2030/Praktikum-Pemrograman-Mobile/tree/main/Praktikum-1/Compose
Versi XML:
https://github.com/raihan2030/Praktikum-Pemrograman-Mobile/tree/main/Praktikum-1/XML