LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1

ANDROID BASIC WITH KOTLIN

Oleh:

Akmallullail Sya'ban NIM. 2310817310010

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 Prakitkum ini dikerjakan oleh:

Nama Praktikan : Akmallullail Sya'ban
NIM : 2310817310010

Menyetujui, Asisten Praktikum	Mengetahui, Dosen Penanggung Jawab Praktikum
Muhammad Raka Azwar NIM. 2210817210012	Andreyan Rizky Baskara, S.Kom., M.Kom. NIP. 199307032019030101 1

DAFTAR ISI

LEMBAR PENGESAHAN 2
DAFTAR ISI 3
DAFTAR GAMBAR 4
DAFTAR TABEL 5
SOAL 1 6
A. Source Code 8
B. Output Program 15
C. Pembahasan 18
D. Tautan Git 20

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. Tampilan Awal Dadu Aplikasi XML 15
Gambar 5. Tampilan Dadu Double Aplikasi XML 15
Gambar 6. Tampilan Dadu Saat di Roll Aplikasi XML 16
Gambar 7. Tampilan Awal Dadu Aplikasi Jetpack Compose 16
Gambar 8. Tampilan Dadu Saat di Roll Aplikasi Jetpack Compose 17
Gambar 9. Tampilan Dadu Double Aplikasi Jetpack Compose 17

DAFTAR TABEL

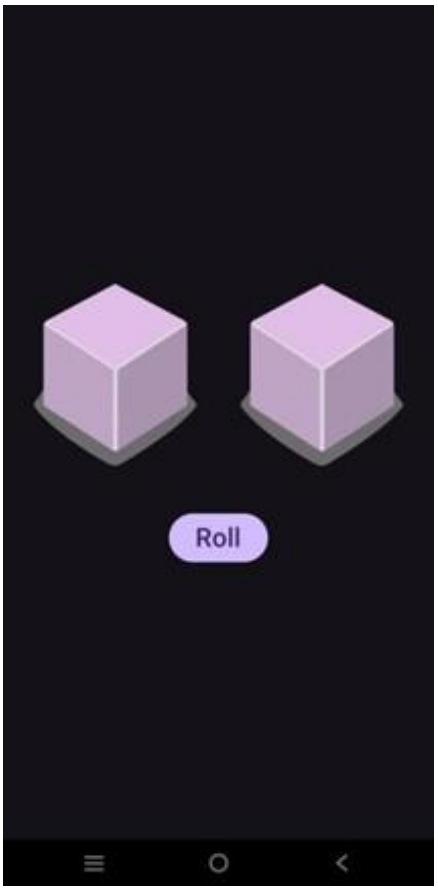
Tabel 1. Source Code Jawaban Soal 1 XML 10
Tabel 2. Source Code Jawaban Soal 1 XML 11
Tabel 3. Source Code Jawaban Soal 1 Jetpack Compose 13
Tabel 4. Source Code Jawaban Soal 1 Jetpack Compose 14

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.

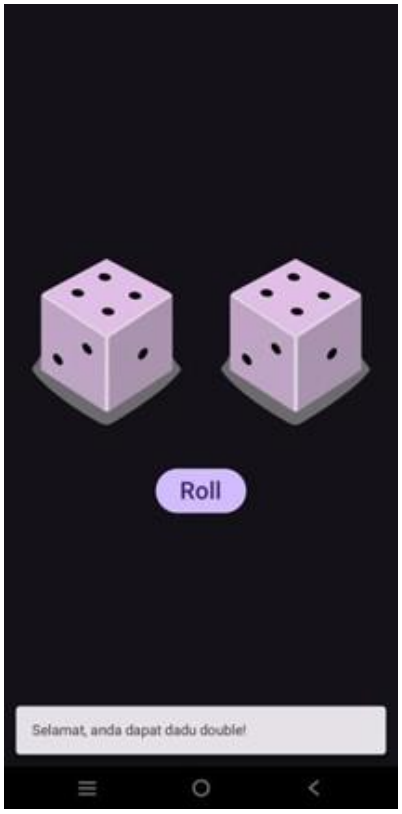


Gambar 1. Tampilan Awal Aplikasi

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

A. Source Code

XML:

MainActivity.kt

```
1  package com.example.dicerollerxml
2
3  import android.os.Bundle
4  import android.widget.Button
5  import androidx.activity.enableEdgeToEdge
6  import androidx.appcompat.app.AppCompatActivity
7  import
   com.example.dicerollerxml.databinding.ActivityMainBinding
8  import com.google.android.material.snackbar.Snackbar
9
10 class MainActivity : AppCompatActivity() {
11     private lateinit var binding: ActivityMainBinding
```

```
12
13      override fun onCreate(savedInstanceState: Bundle?) {
14          super.onCreate(savedInstanceState)
15          enableEdgeToEdge()
16          binding = ActivityMainBinding.inflate(layoutInflater)
17          setContentView(binding.root)
18
19          binding.button.setOnClickListener {
20              rollDice()
21          }
22      }
23
24      private fun rollDice()
25      {
26          val dice = Dice(6)
27          val sides = dice.diceRoll()
28          val sides2 = dice.diceRoll()
29          val button = findViewById<Button>(R.id.button)
30          val dR1 = when(sides)
31          {
32              1 -> R.drawable.dice_1
33              2 -> R.drawable.dice_2
34              3 -> R.drawable.dice_3
35              4 -> R.drawable.dice_4
36              5 -> R.drawable.dice_5
37              else -> R.drawable.dice_6
38          }
39
40          val dR2 = when (sides2)
41          {
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              else -> R.drawable.dice_6
48          }
49          binding.imageView1.setImageResource(dR1)
50          binding.imageView2.setImageResource(dR2)
51
52          val massage = if (sides == sides2) {
53              "Selamat, anda mendapatkan dadu double"
54          } else {
55              "Anda belum beruntung"
56          }
57          Snackbar.make(button, massage, Snackbar.LENGTH_LONG)
58              .show()
59      }
60  }
61  class Dice(private val numSides : Int)
62  {
63      fun diceRoll() : Int
```

```
64      {
65          return                              (1..numSides).random()
66      }
67  }
```

Tabel 1. Source Code Jawaban Soal 1 XML

activity_main.xml

```
1   <?xml             version="1.0"             encoding="utf-8"?>
2   <androidx.constraintlayout.widget.ConstraintLayout
3   xmlns:android="http://schemas.android.com/apk/res/android"
4       xmlns:app="http://schemas.android.com/apk/res-auto"
5       xmlns:tools="http://schemas.android.com/tools"
6       android:id="@+id/main"
7       android:layout_width="match_parent"
8       android:layout_height="match_parent"
9       tools:context=".MainActivity">
10
11      <ImageView
12          android:id="@+id/imageView1"
13          android:layout_width="186dp"
14          android:layout_height="231dp"
15          android:layout_marginTop="195dp"
16          android:layout_marginEnd="16dp"
17          app:layout_constraintBottom_toTopOf="@+id/button"
18          app:layout_constraintEnd_toEndOf="parent"
19          app:layout_constraintHorizontal_bias="1.0"
20          app:layout_constraintStart_toEndOf="@+id/imageView2"
21          app:layout_constraintTop_toTopOf="parent"
22          app:layout_constraintVertical_bias="1.0"
23          app:srcCompat="@drawable/dice_0"                    />
24
25      <Button
26          android:id="@+id/button"
27          android:layout_width="wrap_content"
28          android:layout_height="wrap_content"
29          android:layout_marginStart="160dp"
30          android:layout_marginEnd="161dp"
31          android:layout_marginBottom="256dp"
32          android:text="Roll"
33          app:layout_constraintBottom_toBottomOf="parent"
34          app:layout_constraintEnd_toEndOf="parent"
35          app:layout_constraintStart_toStartOf="parent"      />
36
37      <ImageView
38          android:id="@+id/imageView2"
39          android:layout_width="195dp"
40          android:layout_height="231dp"
41          android:layout_marginStart="14dp"
42          android:layout_marginTop="195dp"
43          app:layout_constraintBottom_toTopOf="@+id/button"
44          app:layout_constraintEnd_toStartOf="@+id/imageView1"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0"
        app:srcCompat="@drawable/dice_0" />

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

Tabel 2. Source Code Jawaban Soal 1 XML

Compose:

MainActivity.kt

```
package com.example.diceroller

import android.os.Bundle
import android.view.Gravity
import android.widget.Toast
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.annotation.ContentView
import androidx.annotation.GravityInt
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.layout.width
import androidx.compose.foundation.layout.wrapContentSize
import androidx.compose.material.icons.Icons
import androidx.compose.material3.Button
import
androidx.compose.material3.ExtendedFloatingActionButton
import androidx.compose.material3.Icon
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.mutableStateOf
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.tooling.preview.Preview
import com.example.diceroller.ui.theme.DiceRollerTheme
import androidx.compose.ui.unit.dp
import androidx.compose.runtime.getValue
import androidx.compose.runtime.setValue
import androidx.compose.runtime.remember
```

```
import androidx.compose.ui.focus.focusModifier
import androidx.compose.ui.graphics.painter.Painter
import androidx.compose.ui.platform.LocalContext
import kotlinx.coroutines.launch

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            DiceRollerTheme {
                DiceRollerApp()
            }
        }
    }
}

@Composable
fun DiceWithButtonAndImage(modifier: Modifier = Modifier) {
    var result by remember { mutableStateOf(0) }
    var result2 by remember { mutableStateOf(0) }
    val context = LocalContext.current

    val imageResource = when (result) {
        0 -> R.drawable.dice_0
        1 -> R.drawable.dice_1
        2 -> R.drawable.dice_2
        3 -> R.drawable.dice_3
        4 -> R.drawable.dice_4
        5 -> R.drawable.dice_5
        else -> R.drawable.dice_6
    }

    val imageResource2 = when (result2) {
        0 -> R.drawable.dice_0
        1 -> R.drawable.dice_1
        2 -> R.drawable.dice_2
        3 -> R.drawable.dice_3
        4 -> R.drawable.dice_4
        5 -> R.drawable.dice_5
        else -> R.drawable.dice_6
    }
        Column(
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = modifier
    ) {
        Row(
            horizontalArrangement = Arrangement.Center,
            modifier = Modifier.fillMaxWidth()
        ) {
            Image(
                painter = painterResource(imageResource),
```

```
92                  contentDescription    =    result.toString(),
93                  modifier      =      Modifier.height(200.dp)
94              )
95              Spacer(modifier    =    Modifier.height(100.dp))
96              Image(
97                  painter  =  painterResource(imageResource2),
98                  contentDescription   =   result2.toString(),
99                  modifier      =      Modifier.height(200.dp)
100             )
101         }
102
103         Spacer(modifier      =      Modifier.height(16.dp))
104
105         Button(onClick                  =                  {
106             result             =             (1..6).random()
107             result2             =             (1..6).random()
108             val   resultText   =   if   (result   ==   result2)
109                 "Selamat,  anda  mendapatkan  dadu  double"
110             else
111                 "Anda            belum            beruntung"
112             val  toast  =  Toast.makeText(context,resultText,
    Toast.LENGTH_SHORT)
113             toast.setGravity(Gravity.BOTTOM                or
    Gravity.CENTER_HORIZONTAL,              0,              150)
114
115             toast.show()
116
117         })                                                  {
118             Text(stringResource(R.string.Roll))
119             }
120         }
121     }
122
123 @Preview(showBackground                =                true)
124 @Composable
125 fun                     DiceRollerApp()                     {
126     DiceWithButtonAndImage(
127         modifier                =                Modifier
128             .fillMaxSize()
129             .wrapContentSize(Alignment.Center)
130     )
131 }
```

Tabel 3. Source Code Jawaban Soal 1 Jetpack Compose

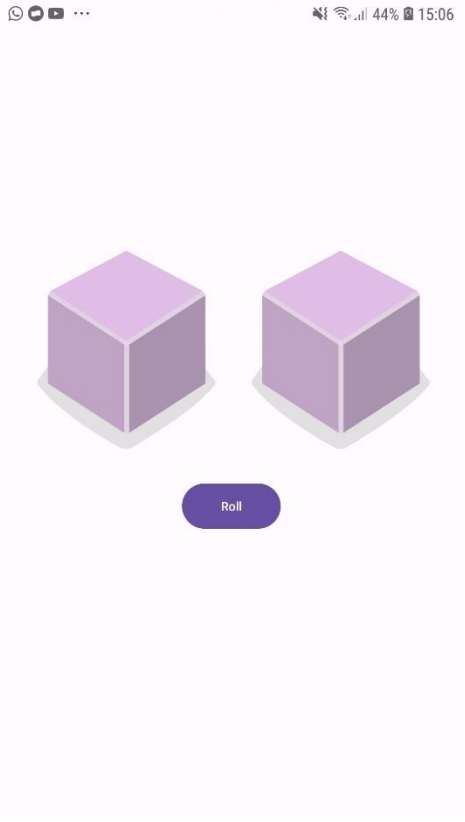
AndroidManifest.xml

```
1   <?xml            version="1.0"            encoding="utf-8"?>
2   <manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
3       xmlns:tools="http://schemas.android.com/tools"          >
4
5       <application
```

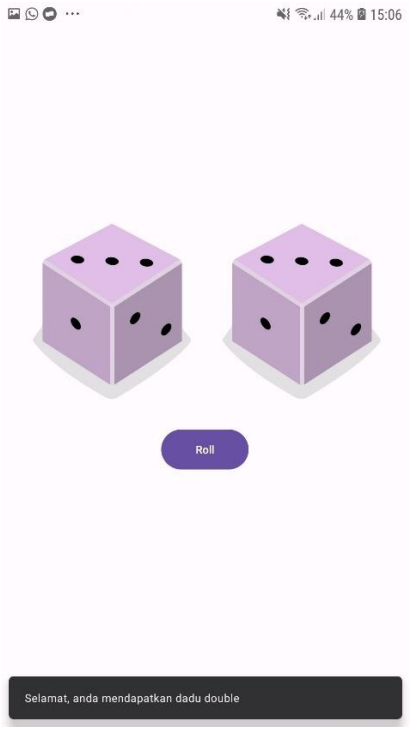
```
6           android:allowBackup="true"

7   android:dataExtractionRules="@xml/data_extraction_rules"
8           android:fullBackupContent="@xml/backup_rules"
9           android:icon="@mipmap/ic_launcher"
10          android:label="@string/Roll"
11          android:roundIcon="@mipmap/ic_launcher_round"
12          android:supportsRtl="true"
13          android:theme="@style/Theme.DiceRoller"
14          tools:targetApi="31"                                    >
15          <activity
16              android:name=".MainActivity"
17              android:exported="true"
18              android:label="@string/Roll"
19              android:theme="@style/Theme.DiceRoller"             >
20              <intent-filter>
21                  <action
    android:name="android.intent.action.MAIN"                      />
22
23                  <category
    android:name="android.intent.category.LAUNCHER"                />
24              </intent-filter>
25          </activity>
26      </application>
27
28  </manifest>
```

Tabel 4. Source Code Jawaban Soal 1 Jetpack Compose

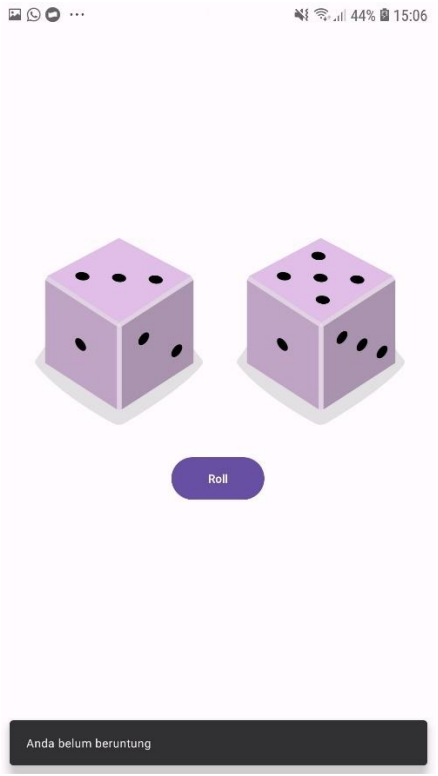
B. Output Program



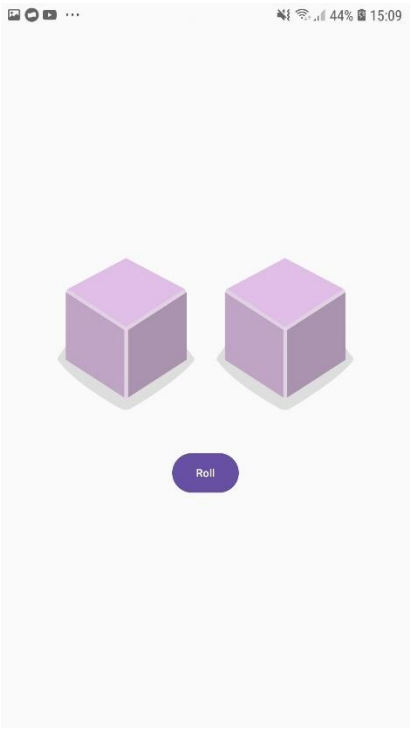
Gambar 4. Tampilan Awal Dadu Aplikasi XML



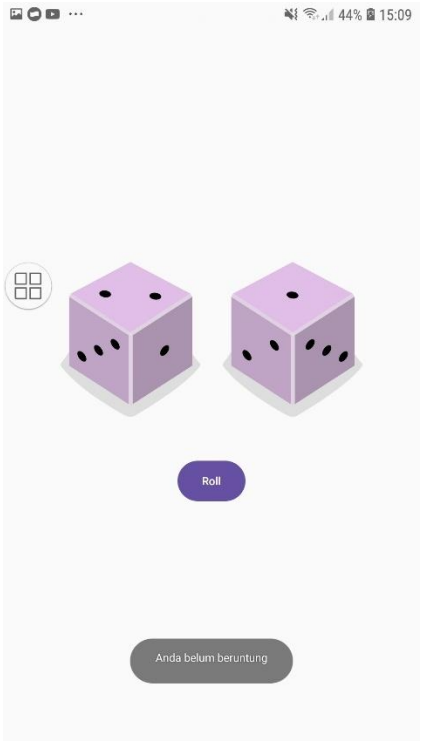
Gambar 5. Tampilan Dadu Double Aplikasi XML



Gambar 6. Tampilan Dadu Saat di Roll Aplikasi XML



Gambar 7. Tampilan Awal Dadu Aplikasi Jetpack Compose



Gambar 8. Tampilan Dadu Saat di Roll Aplikasi Jetpack Compose

Gambar 9. Tampilan Dadu Double Aplikasi Jetpack Compose

C. Pembahasan

XML:

MainActivity.kt:

Pada baris [11], private lateinit var binding: ActivityMainBinding memiliki fungsi untuk mengakses komponen dari activity_main.xml tanpa findViewById

Pada baris [13 – 22], adalah blok kode yang menghubungkan layout dengan binding, dan root dari binding nya adalah root dari XML. Binding button berfungsi ketika tombol di klik, maka akan memanggil fungsi rollDice()

Pada baris [26 – 28], terdapat 3 variable yang pertama mendeklarasikan objek dice dengan 6 sisi, dan membuat objek yang berfungsi untuk meng-roll dua dadu dan menyimpan hasilnya di sides dan sides2

Pada baris [29], memanggil objek tombol dengan id button yang terhubung di activity_main.xml dengan id button.

Pada baris [30 – 48], fungsi when untuk menentukan gambar hasil roll, dan R.drawable.dice adalah sebagai id dari gambar yang ada pada drawable. Hal ini dilakukan berulang kali pada dua sisi dadu.

Pada baris [49 – 50], menampilkan hasil gambar dadu masing masing dengan imageView dan dari dadu yang sudah di kondisikan pada when secara acak dari 1 – 6

Pada baris [52 – 58], pada blok ini adalah komponen snackbar yang berfungsi untuk menampilkan pop up kecil di bawah layar yang menampilkan hasil dadu ketika sama atau pun tidak dengan pesan yang sudah ditentukan.

Pada baris [61 – 67], pada blok ini adalah class dadu yang berisi numSides sebagai jumlah sisi dadu dengan fungsi diceRoll yang akan menghasilkan angka acak dari satu sampai enam sisi dadu.

activity_main.xml:

Pada baris [10 – 22], adalah image view layout dadu kanan dengan nama ImageView1 dengan tinggi, lebar, jarak, margin atas, bawah yang sudah ditentukan.

Pada baris [24 – 34], adalah button yang dipanggil pada MainActivity.kt yang akan diposisikan pada tengah tengah layar di bagian bawah dengan wrap_content yang lebar dan tinggi mengikuti ukuran teks tombol. Pada bagian ini juga mengatur margin kiri dan kanan untuk tetap berada di tengah

Pada baris [36 – 47], adalah image view layout dadu kiri dengan nama ImageView2 dengan tinggi, lebar, jarak, margin atas, bawah yang sudah ditentukan. hampir mirip dengan ImageView1.

Jetpack Compose:

MainActivity.kt:

Pada baris [45 – 55], terdapat beberapa fungsi seperti enableEdgeToEdge untuk aplikasi menggunakan seluruh layar, setContent untuk UI pada Jetpack Compose dan DiceRollerTheme menerapkan tema aplikasi untuk di terapkan di DiceRollerApp

Pada baris [58 – 60], adalah baris kode yang menampilkan fungsi utama untuk dua dadu dan tombol untuk roll dadu, dengan menyimpan nilai dadu secara dinamis, remember menjaga nilai, dan mutableStateOf untuk nilai berubah saat tombol ditekan.

Pada baris [61], ini akan digunakan sebagai Toast untuk mengambil context

Pada baris [63 – 81], adalah baris kode yang menyesuaikan angka dadu dengan gambar di drawable dengan fungsi when yang akan disimpan di variable result dengan nama ImageResource. Begitu juga pada ImageResource2 melakukan hal yang sama untuk dadu yang kedua.

Pada baris [82 – 85], adalah column untuk menampilkan hasil secara vertikal dan rata tengah.

Pada baris [86 – 89], adalah row untuk menampilkan hasil secara horizontal bersamping-sampingan.

Pada baris [90 – 101], adalah fungsi Image yang berisi beberapa objek seperti painter yang menampilkan gambar dari drawable resource, contentDescription sebagai aksesibilitas berdasarkan hasil roll dadu, dan mengatur tinggi gambar.

Pada baris [105 – 117], adalah membuat sebuah tombol dengan parameter onClick yang berfungsi ketika di tekan. Selain itu ada fungsi result untuk mengrandom nilai dari 1 sampai 6 untuk dadu dan disimpan pada variable result juga. Ada juga variable yang menyimpan hasil result jika dadu sama ataupun tidak sama, jika dadu sama menampilkan pesan “anda mendapatkan dadu double” dan jika tidak “anda belum beruntung”. Terdapat juga variable untuk menampilkan pesan dengan komponen toast untuk membuat pesan ketika tombol ditekan dan mengeluarkan hasilnya dengan show

Pada baris [118], merupakan isi dari tombolnya dengan nama yang akan muncul yaitu Roll

Pada baris [123 – 131], pada blok ini berfungsi sebagai preview dan tampilan aplikasi utama diceRollerApp

AndroidManifest.xml:

Pada baris [5 – 14], ini merupakan beberapa hal penting yang dimana terdapat backup, icon, nama aplikasi yang muncul, dan tema dari aplikasi dengan target API 31

Pada baris [15 – 19], ini merujukkan kemana file aplikasi yaitu MainActivity dan nama aplikasinya juga ditentukan yaitu Roll.

Pada baris [20 – 24], mengartikan bahwa MainActivity adalah apa yang utama dari aplikasi dengan menunjukkan main sebagai aktivitas pertama yang dibuka dan launcher agar bisa dimunculkan di device seperti HP.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
https://github.com/Akmlsybn/Praktikum-Mobile