LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1

ANDROID BASIC WITH COMPOSE

Oleh:

Adrian Bintang Saputera NIM. 2310817110006

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2025

LEMBAR PENGESAHAN

LAPORAN PRAKTIKUM PEMROGRAMAN I
MODUL 1

Laporan Praktikum Pemrograman Mobile Modul 1: Android Basic with Compose ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Prakitkum ini dikerjakan oleh:

Nama Praktikan : Adrian Bintang Saputera
NIM : 2310817110006

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

DAFTAR GAMBAR

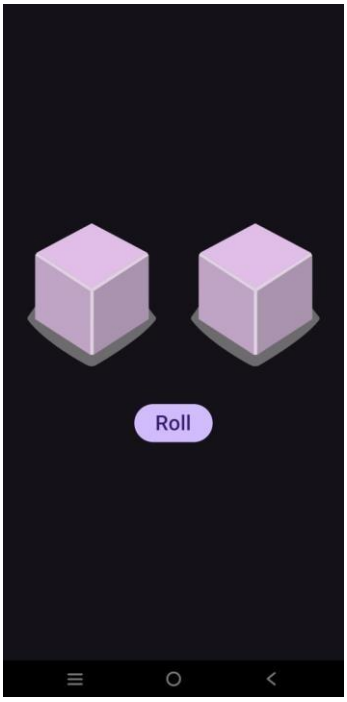
DAFTAR TABEL

SOAL 1

Soal Praktikum:

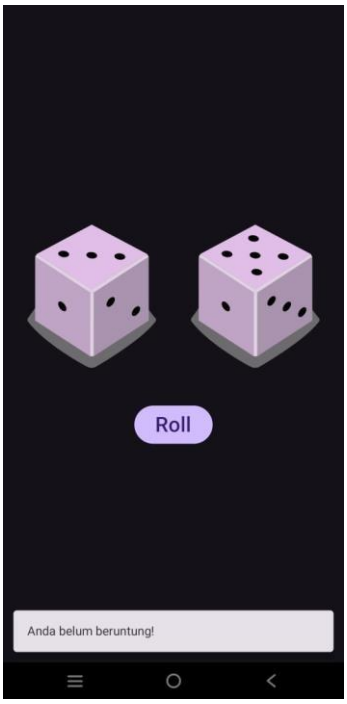
Buatlah sebuah aplikasi yang dapat menampilkan 2 buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



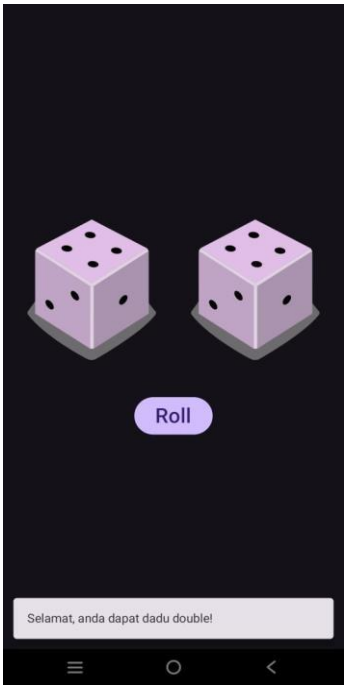
Gambar 1. Tampilan Awal Aplikasi

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:

MainActivity.kt

Table 1. Source Code Jawaban Soal 1 XML

```
1  package com.example.dicerollxml
2
3  import android.os.Bundle
4  import android.widget.Button
5  import android.widget.ImageView
6  import android.widget.Toast
7  import androidx.appcompat.app.AppCompatActivity
8
9  class MainActivity : AppCompatActivity() {
10     private var lastDiceRoll: Int? = null
11     override fun onCreate(savedInstanceState:
   Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14
15         val rollButton: Button =
   findViewById(R.id.button)
16         rollButton.setOnClickListener { rollDice();
   rollDice2() }
17
18         val diceImage: ImageView =
   findViewById(R.id.imageView)
19         val diceImage2: ImageView =
   findViewById(R.id.imageView2)
20         diceImage.setImageResource(R.drawable.dice_0)

21 diceImage2.setImageResource(R.drawable.dice_0)
22     }
23
```

```
24      private fun rollDice() {
25          val dice = Dice(6)
26          val diceRoll = dice.roll()
27          val diceImage: ImageView =
    findViewById(R.id.imageView)
28
29          val drawableResource = when (diceRoll) {
30              1 -> R.drawable.dice_1
31              2 -> R.drawable.dice_2
32              3 -> R.drawable.dice_3
33              4 -> R.drawable.dice_4
34              5 -> R.drawable.dice_5
35              else -> R.drawable.dice_6
36          }
37
38          diceImage.setImageResource(drawableResource)
39
40          diceImage.contentDescription =
    diceRoll.toString()
41
42          lastDiceRoll = diceRoll
43      }
44
45      private fun rollDice2() {
46          val dice = Dice(6)
47          val diceRoll = dice.roll()
48          val diceImage: ImageView =
    findViewById(R.id.imageView2)
49
50          val drawableResource = when (diceRoll) {
51              1 -> R.drawable.dice_1
52              2 -> R.drawable.dice_2
53              3 -> R.drawable.dice_3
54              4 -> R.drawable.dice_4
55              5 -> R.drawable.dice_5
56              else -> R.drawable.dice_6
57          }
58
59          diceImage.setImageResource(drawableResource)
60
61          diceImage.contentDescription =
    diceRoll.toString()
62
63          if (diceRoll == lastDiceRoll) {
64              Toast.makeText(this, "Selamat, anda dapat
```

```
    dadu double!", Toast.LENGTH_SHORT).show()
65          }
66          else{
67              Toast.makeText(this, "Anda belum
    beruntung!", Toast.LENGTH_SHORT).show()
68          }
69      }
70
71      class Dice(private val numSides: Int) {
72
73          fun roll(): Int {
74              return (1..numSides).random()
75          }
76      }
77  }
```

activity_main.xml

Table 2. Source Code Jawaban Soal 1 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      tools:layout_editor_absoluteX="-1dp"
11      tools:layout_editor_absoluteY="62dp">
12
13      <Button
14          android:id="@+id/button"
15          android:layout_width="wrap_content"
16          android:layout_height="wrap_content"
17          android:layout_marginBottom="308dp"
18          android:text="@string/roll"
19          android:textSize="20sp"
20          app:layout_constraintBottom_toBottomOf="parent"
21          app:layout_constraintEnd_toEndOf="parent"
22          app:layout_constraintHorizontal_bias="0.498"
23          app:layout_constraintStart_toStartOf="parent" />
24
25      <ImageView
```

```
        android:id="@+id/imageView"
        android:layout_width="160dp"
        android:layout_height="200dp"
        android:layout_marginEnd="200dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.803"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.429"
        app:srcCompat="@drawable/dice_1" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="160dp"
        android:layout_height="200dp"
        android:layout_marginStart="200dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.176"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.429"
        app:srcCompat="@drawable/dice_2" />

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

Jetpack Compose:

MainActivity.kt

Table 3. Source Code Jawaban Soal 1 Jetpack Compose

```
package com.example.dicerolljetcom

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.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
```

```
14  import androidx.compose.foundation.layout.fillMaxSize
15  import androidx.compose.foundation.layout.fillMaxWidth
16  import androidx.compose.foundation.layout.height
17  import androidx.compose.foundation.layout.padding
18  import androidx.compose.foundation.layout.width
19  import androidx.compose.material3.Button
20  import androidx.compose.material3.Text
21  import androidx.compose.runtime.Composable
22  import androidx.compose.ui.Alignment
23  import androidx.compose.ui.Modifier
24  import androidx.compose.ui.res.painterResource
25  import androidx.compose.ui.res.stringResource
26  import androidx.compose.ui.tooling.preview.Preview
27  import
    com.example.dicerolljetcom.ui.theme.DiceRollJetComTheme
28  import androidx.compose.ui.unit.dp
29  import androidx.compose.material3.ButtonDefaults
30  import androidx.compose.ui.graphics.Color
31  import androidx.compose.ui.unit.sp
32  import androidx.compose.runtime.getValue
33  import androidx.compose.runtime.mutableIntStateOf
34  import androidx.compose.runtime.setValue
35  import androidx.compose.runtime.remember
36  import androidx.compose.ui.platform.LocalContext
37
38  class MainActivity : ComponentActivity() {
39      override fun onCreate(savedInstanceState: Bundle?)
    {
40          super.onCreate(savedInstanceState)
41          enableEdgeToEdge()
42          setContent {
43              DiceRollJetComTheme {
44                  DiceRollerApp()
45              }
46          }
47      }
48  }
49
50  @Composable
51  fun DiceWithButtonAndImage(modifier: Modifier =
    Modifier) {
52      var result by remember { mutableIntStateOf(0) }
53      var result2 by remember { mutableIntStateOf(0) }
54      val context = LocalContext.current
55
```

```
56      val imageResource = when (result) {
57          0 -> R.drawable.dice_0
58          1 -> R.drawable.dice_1
59          2 -> R.drawable.dice_2
60          3 -> R.drawable.dice_3
61          4 -> R.drawable.dice_4
62          5 -> R.drawable.dice_5
63          else -> R.drawable.dice_6
64      }
65
66      val imageResource2 = when (result2) {
67          0 -> R.drawable.dice_0
68          1 -> R.drawable.dice_1
69          2 -> R.drawable.dice_2
70          3 -> R.drawable.dice_3
71          4 -> R.drawable.dice_4
72          5 -> R.drawable.dice_5
73          else -> R.drawable.dice_6
74      }
75      Column(
76          horizontalAlignment =
    Alignment.CenterHorizontally,
77          modifier = modifier
78      ) {
79          Row(
80              horizontalArrangement = Arrangement.Center,
81              modifier = Modifier.fillMaxWidth()
82          ) {
83              Image(
84                  painter =
    painterResource(imageResource),
85                  contentDescription = result.toString(),
86                  modifier = Modifier.height(200.dp)
87              )
88              Spacer(modifier = Modifier.width(10.dp))
89              Image(
90                  painter =
    painterResource(imageResource2),
91                  contentDescription =
    result2.toString(),
92                  modifier = Modifier.height(200.dp)
93              )
94          }
95
96          Spacer(modifier = Modifier.height(5.dp))
```

```
97
98          Button(onClick = {
99              result = (1..6).random()
100             result2 = (1..6).random()
101             val resultText = if (result == result2)
102                 "Selamat, anda dapat dadu double!"
103             else
104                 "Anda belum beruntung!"
105             val toast =
    Toast.makeText(context,resultText, Toast.LENGTH_SHORT)
106             toast.setGravity(Gravity.BOTTOM or
    Gravity.CENTER_HORIZONTAL, 0, 150)
107
108             toast.show()
109
110         },
111             colors = ButtonDefaults.buttonColors(
112                 containerColor = Color(0xFF6750A4),
113                 contentColor = Color.White
114             )
115         ) {
116             Text(
117                 text = stringResource(R.string.roll),
118                 fontSize = 25.sp
119             )
120         }
121     }
122 }
123
124 @Preview(showBackground = true)
125 @Composable
126 fun DiceRollerApp() {
127     DiceWithButtonAndImage(
128         modifier = Modifier
129             .fillMaxSize()
130             .padding(top = 300.dp)
131     )
132 }
```

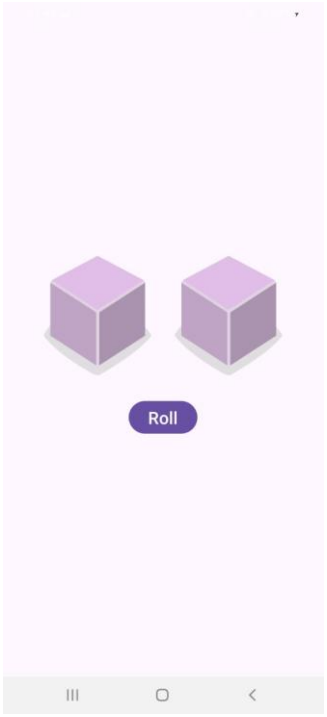
AndroidManifest.xml

Table 4. Source Code Jawaban Soal 1 Jetpack Compose

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

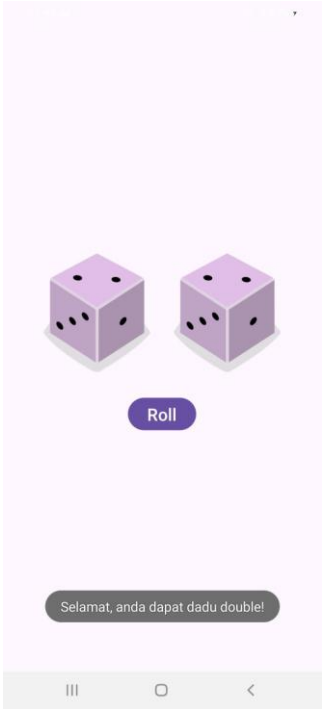
```
3       xmlns:tools="http://schemas.android.com/tools">
4
5       <application
6           android:allowBackup="true"
7   android:dataExtractionRules="@xml/data_extraction_rules"
8           android:fullBackupContent="@xml/backup_rules"
9           android:icon="@mipmap/ic_launcher"
10          android:label="@string/app_name"
11          android:roundIcon="@mipmap/ic_launcher_round"
12          android:supportsRtl="true"
13          android:theme="@style/Theme.DiceRollJetCom"
14          tools:targetApi="31">
15          <activity
16              android:name=".MainActivity"
17              android:exported="true"
18              android:label="@string/app_name"
19              android:theme="@style/Theme.DiceRollJetCom">
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>
```

B. Output Program

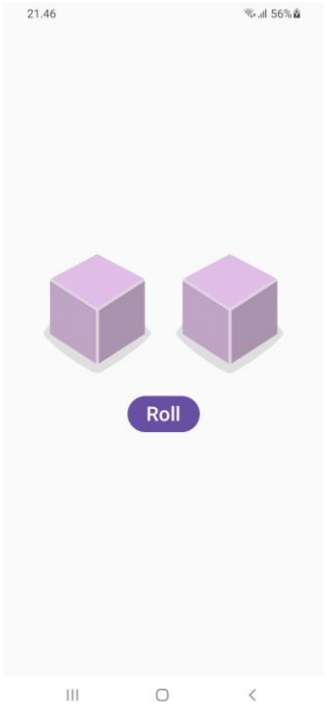


Gambar 4. Tampilan Awal Dadu Aplikasi XML

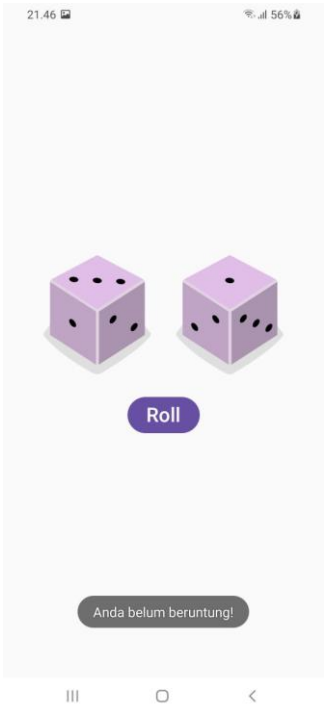
Gambar 5. Tampilan Dadu Saat Di Roll Aplikasi XML



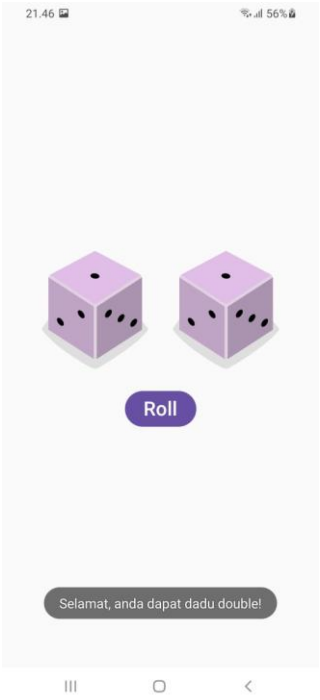
Gambar 6. Tampilan Dadu Double Aplikasi XML



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:

Blok 1: Package dan Import (line 1, 3–7)

Pada line 1, dideklarasikan package com.example.dicerollxml sebagai namespace dari aplikasi.

Pada line 3–7, dilakukan import terhadap class penting dari Android seperti Bundle, Button, ImageView, Toast, dan AppCompatActivity, yang digunakan untuk membangun UI dan logika aplikasi berbasis Activity.

Blok 2: Deklarasi Class dan Variabel (line 9–10)

Pada line 9, dideklarasikan kelas MainActivity sebagai turunan dari AppCompatActivity, yaitu entry point utama aplikasi.

Pada line 10, dibuat properti lastDiceRoll bertipe Int?, digunakan untuk menyimpan hasil lemparan dadu pertama.

Blok 3: Fungsi onCreate() (line 11–13, 15–16, 18–22)

Pada line 11, fungsi onCreate() dioverride sebagai titik awal ketika activity dijalankan.

Pada line 12, super.onCreate(savedInstanceState) dipanggil untuk mewarisi logika dasar dari AppCompatActivity.

Pada line 13, layout activity_main.xml dipasang sebagai tampilan utama.

Pada line 15, Button dengan ID button diakses dari layout dan disimpan ke dalam variabel rollButton.

Pada line 16, listener dipasang ke tombol tersebut untuk memanggil fungsi rollDice() dan rollDice2() saat tombol ditekan.

Pada line 18, ImageView pertama dengan ID imageView diambil dan disimpan ke variabel diceImage.

Pada line 19, ImageView kedua dengan ID imageView2 diambil dan disimpan ke variabel diceImage2.

Pada line 20, gambar awal dadu pertama diatur ke dice_0.

Pada line 21, gambar awal dadu kedua juga diatur ke dice_0.
Pada line 22, onCreate() ditutup.

Blok 4: Fungsi rollDice() (line 24–27, 29–36, 38, 40, 42–43)
Pada line 24, fungsi rollDice() dideklarasikan untuk menangani logika lemparan dadu pertama.
Pada line 25, objek Dice dibuat dengan 6 sisi.
Pada line 26, hasil lemparan dadu disimpan ke variabel diceRoll.
Pada line 27, ImageView pertama diakses kembali dari layout.
Pada line 29–36, digunakan struktur when untuk menentukan gambar berdasarkan nilai diceRoll.
Pada line 38, gambar pada ImageView diperbarui sesuai hasil lemparan.
Pada line 40, deskripsi konten gambar diatur ke nilai diceRoll untuk keperluan aksesibilitas.
Pada line 42, nilai lemparan disimpan ke lastDiceRoll.
Pada line 43, fungsi rollDice() ditutup.

Blok 5: Fungsi rollDice2() (line 45–48, 50–57, 59, 61, 63–69)
Pada line 45, fungsi rollDice2() dideklarasikan untuk menangani logika lemparan dadu kedua.
Pada line 46, objek Dice dibuat dengan 6 sisi.
Pada line 47, hasil lemparan disimpan ke variabel diceRoll.
Pada line 48, ImageView kedua diakses dari layout.
Pada line 50–57, digunakan struktur when untuk menentukan gambar berdasarkan nilai diceRoll.
Pada line 59, gambar pada ImageView kedua diperbarui.
Pada line 61, deskripsi konten gambar diatur berdasarkan diceRoll.
Pada line 63, dilakukan pengecekan apakah diceRoll sama dengan lastDiceRoll.
Pada line 64, jika sama, maka Toast dengan pesan keberuntungan ditampilkan.
Pada line 65, blok if ditutup.

Pada line 66, blok else dibuka jika hasil dadu tidak sama.

Pada line 67, Toast dengan pesan belum beruntung ditampilkan.

Pada line 68, blok else ditutup.

Pada line 69, fungsi rollDice2() ditutup.

Blok 6: Kelas Dice (line 71, 73–76)

Pada line 71, dideklarasikan inner class Dice dengan properti numSides yang menentukan jumlah sisi dadu.

Pada line 73, fungsi roll() dideklarasikan untuk menghasilkan angka acak.

Pada line 74, angka acak antara 1 hingga jumlah sisi dikembalikan menggunakan (1..numSides).random().

Pada line 75, fungsi roll() ditutup.

Pada line 76, kelas Dice ditutup.

Blok 7: Penutup Kelas MainActivity (line 77)

Pada line 77, kelas MainActivity ditutup secara keseluruhan.

Activity_main.xml:

Blok 1: Deklarasi XML dan Root Layout (line 1–11)

Pada line 1, dideklarasikan deklarasi XML standar <?xml version="1.0" encoding="utf-8"?>.

Pada line 2, elemen root ConstraintLayout digunakan dari AndroidX untuk mengatur posisi komponen UI secara fleksibel berdasarkan constraint.

Pada line 3–5, namespace didefinisikan untuk android, app, dan tools, yang diperlukan untuk atribut-atribut dalam layout.

Pada line 6, layout diberi ID @+id/main untuk keperluan referensi dari kode Kotlin.

Pada line 7–8, layout diatur agar memenuhi seluruh lebar dan tinggi layar menggunakan match_parent.

Pada line 9, konteks layout ditetapkan untuk MainActivity agar layout ini dapat dikenali sebagai tampilan milik aktivitas tersebut.

Pada line 10–11, atribut tools:layout_editor_absoluteX dan tools:layout_editor_absoluteY digunakan hanya di layout editor untuk menentukan posisi awal tampilan.

Blok 2: Tombol (Button) (line 13–23)

Pada line 13, elemen <Button> dibuat dengan ID @+id/button.

Pada line 14–15, ukuran tombol disetel ke wrap_content agar menyesuaikan ukuran konten.

Pada line 16, diberi margin bawah sebesar 308dp untuk posisi tombol lebih ke atas.

Pada line 17, teks tombol diambil dari resource string @string/roll.

Pada line 18, ukuran teks diatur ke 20sp.

Pada line 19–23, constraint digunakan untuk menempatkan tombol di tengah bawah layar: menyematkan ke bawah (Bottom_toBottomOf="parent"), ke sisi kanan dan kiri (End_toEndOf, Start_toStartOf), dan menggunakan Horizontal_bias="0.498" untuk menjaga posisi tengah.

Blok 3: Gambar Dadu Pertama (ImageView) (line 25–36)

Pada line 25, elemen <ImageView> pertama dibuat dengan ID @+id/imageView.

Pada line 26–27, ukuran gambar diatur menjadi 160dp lebar dan 200dp tinggi.

Pada line 28, margin end ditetapkan sebesar 200dp agar gambar terdorong ke kiri dari sisi kanan layar.

Pada line 29–31, constraint digunakan agar gambar diletakkan di bagian tengah vertikal layar dan tetap di dalam parent (bottom dan end).

Pada line 32, Horizontal_bias="0.803" digunakan agar gambar berada di sisi kanan layar.

Pada line 33–34, gambar disematkan juga ke atas dan ke sisi kiri layar menggunakan constraint Top_toTopOf dan Start_toStartOf.

Pada line 35, Vertical_bias="0.429" digunakan untuk mengatur posisi vertikalnya agar berada sedikit di atas tengah.

Pada line 36, gambar awal yang ditampilkan adalah @drawable/dice_1.

Blok 4: Gambar Dadu Kedua (ImageView) (line 38–49)

Pada line 38, elemen <ImageView> kedua dibuat dengan ID @+id/imageView2.

Pada line 39–40, ukuran gambar disamakan dengan gambar pertama: 160dp × 200dp.

Pada line 41, margin start ditetapkan sebesar 200dp agar terdorong ke kanan dari sisi kiri layar.

Pada line 42–44, constraint untuk bottom dan end disetel ke parent, memastikan gambar tetap berada dalam batas layar.

Pada line 45, Horizontal_bias="0.176" digunakan agar posisi gambar berada di sisi kiri layar.

Pada line 46–47, constraint ke atas dan ke sisi kiri disetel untuk keselarasan vertikal dan horizontal.

Pada line 48, Vertical_bias="0.429" diatur agar posisinya sejajar vertikal dengan gambar pertama.

Pada line 49, gambar awal yang ditampilkan adalah @drawable/dice_2.

Blok 5: Penutup Layout (line 51)

Pada line 51, elemen penutup </androidx.constraintlayout.widget.ConstraintLayout> digunakan untuk menutup layout utama aplikasi.

Jetpack Compose:

MainActivity.kt:

Blok 1: Package dan Import (line 1–36)

Pada line 1, dideklarasikan package com.example.dicerolljetcom sebagai namespace aplikasi.

Pada line 3–36, dilakukan import terhadap berbagai komponen dari Android dan Jetpack Compose, seperti Bundle, Toast, ComponentActivity, setContent, enableEdgeToEdge, Image, Column, Row, Modifier, Button, Text, painterResource, stringResource, Color, dp, sp, remember, dan LocalContext. Semua import ini digunakan untuk membangun UI dengan Compose dan mengatur interaktivitas serta tampilan aplikasi.

Blok 2: Kelas MainActivity dan onCreate() (line 38–47)

Pada line 38, dideklarasikan kelas MainActivity yang mewarisi dari ComponentActivity.

Pada line 39–40, fungsi onCreate() dioverride untuk menangani proses saat activity pertama kali dibuka.

Pada line 41, fungsi enableEdgeToEdge() dipanggil agar tampilan aplikasi dapat menggunakan area layar secara penuh.

Pada line 42, setContent digunakan untuk mulai menyusun UI dengan pendekatan deklaratif dari Jetpack Compose.

Pada line 43, tema aplikasi DiceRollJetComTheme diterapkan pada konten UI.

Pada line 44, fungsi DiceRollerApp() dipanggil untuk memuat tampilan utama aplikasi.

Pada line 45–46, blok penutup untuk theme dan setContent.

Pada line 47, menutup fungsi onCreate() dan kelas MainActivity.

Blok 3: Fungsi DiceWithButtonAndImage - Variabel dan Gambar (line 48–94)

Pada line 48, fungsi DiceWithButtonAndImage() dideklarasikan sebagai @Composable, dengan parameter opsional modifier.

Pada line 50–51, dua variabel result dan result2 disiapkan menggunakan remember dan mutableIntStateOf, untuk menyimpan nilai dadu.

Pada line 52, variabel context digunakan untuk menampilkan Toast.

Pada line 54–63, blok when digunakan untuk memilih gambar dadu pertama berdasarkan nilai result. Jika hasilnya 0, gambar dice_0 ditampilkan, dan seterusnya hingga dice_6.

Pada line 66–75, blok when kedua serupa digunakan untuk memilih gambar dadu kedua berdasarkan result2.

Pada line 76–78, elemen Column digunakan untuk menata komponen secara vertikal dan memusatkan secara horizontal.

Pada line 79–81, elemen Row digunakan untuk menata dua ImageView secara horizontal dan memenuhi lebar penuh.

Pada line 82–85, Image pertama ditampilkan dengan gambar dadu pertama dan tinggi 200.dp.

Pada line 86, Spacer ditambahkan dengan lebar 10.dp sebagai jarak antar gambar.

Pada line 87–90, Image kedua ditampilkan dengan gambar dadu kedua.

Pada line 91–92, Row dan Column ditutup.

Blok 4: Spacer dan Tombol (line 96–122)

Pada line 96, Spacer vertikal ditambahkan dengan tinggi 5.dp.

Pada line 98, Button dideklarasikan dan diberi aksi onClick.

Pada line 99–100, result dan result2 diisi ulang dengan angka acak dari 1 hingga 6.

Pada line 101–104, dilakukan pengecekan apakah kedua hasil dadu sama. Jika iya, teks "Selamat…" akan digunakan, jika tidak, teks "Anda belum beruntung!".

Pada line 105, objek Toast dibuat dengan teks hasil tersebut dan durasi pendek.

Pada line 106, posisi Toast diatur agar muncul di bawah tengah layar.

Pada line 107, Toast ditampilkan ke layar.

Pada line 108, penutup blok onClick.

Pada line 110–112, warna tombol diatur menggunakan ButtonDefaults.buttonColors dengan latar ungu dan teks putih.

Pada line 113, isi tombol didefinisikan sebagai teks dari string resource roll.

Pada line 114, ukuran teks diatur menjadi 25.sp.

Pada line 115–116, penutup Text dan Button.

Blok 5: Preview dan Komposisi Utama (line 124–131)

Pada line 124, anotasi @Preview(showBackground = true) digunakan agar komponen dapat ditampilkan di preview editor Android Studio.

Pada line 125, fungsi DiceRollerApp() dideklarasikan sebagai @Composable.

Pada line 126, fungsi DiceWithButtonAndImage() dipanggil sebagai komponen utama aplikasi.

Pada line 127, modifier digunakan untuk mengisi seluruh layar (fillMaxSize).

Pada line 128, padding atas sebesar 300.dp diterapkan agar elemen turun ke bawah layar.

Pada line 129–132, penutup modifier, fungsi DiceRollerApp(), dan akhir file.

AndroidManifest.xml:

Blok 1: Deklarasi XML dan Tag Root (line 1–3)

Pada line 1, ditentukan bahwa file ini adalah file XML dengan encoding UTF-8.

Pada line 2, elemen <manifest> dideklarasikan sebagai root dari file ini, dan diberikan namespace Android.

Pada line 3, ditambahkan namespace tools untuk keperluan spesifik Android Studio (seperti tools:targetApi).

Blok 2: Tag <application> dan Atributnya (line 5–15)

Pada line 5, tag <application> digunakan untuk mendeklarasikan konfigurasi utama dari aplikasi.

Pada line 6, android:allowBackup="true" memperbolehkan sistem mencadangkan data aplikasi.

Pada line 7, android:dataExtractionRules menunjuk ke file XML yang berisi aturan ekstraksi data pengguna.

Pada line 8, android:fullBackupContent menunjuk ke file XML lain untuk aturan pencadangan penuh.

Pada line 9, android:icon menentukan ikon aplikasi dari resource @mipmap/ic_launcher.

Pada line 10, android:label memberikan label (nama) aplikasi yang ditampilkan.

Pada line 11, android:roundIcon memberikan versi ikon bundar aplikasi.

Pada line 12, android:supportsRtl="true" menyatakan bahwa aplikasi mendukung layout kanan ke kiri.

Pada line 13, android:theme menetapkan tema utama aplikasi menggunakan @style/Theme.DiceRollJetCom.

Pada line 14, atribut tools:targetApi="31" memberi tahu Android Studio bahwa target API layout preview adalah 31.

Pada line 15, dibuka tag <activity> untuk mendeklarasikan activity utama aplikasi.

Blok 3: Tag <activity> dan Intent Filter (line 16–26)

Pada line 16, activity utama MainActivity dideklarasikan, dengan atribut android:name=".MainActivity".

Pada line 17, android:exported="true" menyatakan bahwa activity ini bisa diakses dari luar (dibutuhkan sejak Android 12).

Pada line 18, android:label="@string/app_name" menyetel label activity.

Pada line 19, android:theme="@style/Theme.DiceRollJetCom" menyetel tema activity agar sesuai tema aplikasi.

Pada line 20, dibuka tag <intent-filter> untuk menentukan bagaimana activity ini akan dipanggil.

Pada line 21, action android.intent.action.MAIN menyatakan bahwa ini adalah entry point aplikasi.

Pada line 23, category android.intent.category.LAUNCHER menunjukkan bahwa activity ini ditampilkan di launcher.

Pada line 24, intent filter ditutup.

Pada line 25, tag <activity> ditutup.

Pada line 26, tag <application> ditutup.

Blok 4: Penutup Manifest (line 28)

Pada line 28, tag <manifest> ditutup, mengakhiri deklarasi seluruh konfigurasi aplikasi.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
https://github.com/MadeByBintang/PraktikumMobile