

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Rifky Putra Mahardika NIM. 2310817210023

**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 Kotlin ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Rifky Putra Mahardika
NIM : 2310817210023

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar
NIM. 2210817210026

Muti`a Maulida S.Kom M.T.I
NIP. 198810272019032013

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	22

DAFTAR GAMBAR

Gambar 1. Contoh Tampilan Awal Aplikasi	6
Gambar 2. Contoh Tampilan Dadu Setelah Di Roll	7
Gambar 3. Contoh Tampilan Roll Dadu Double	8
Gambar 4. Screenshot Hasil Jawaban Soal 1 Tampilan Awal	15
Gambar 5. Screenshot Hasil Jawaban Soal 1 Tidak Double	16
Gambar 6. Screenshot Hasil Jawaban Soal 1 Double	17
Gambar 7. Screenshot Hasil Jawaban Soal 1 Mode Landscape	18

DAFTAR TABEL

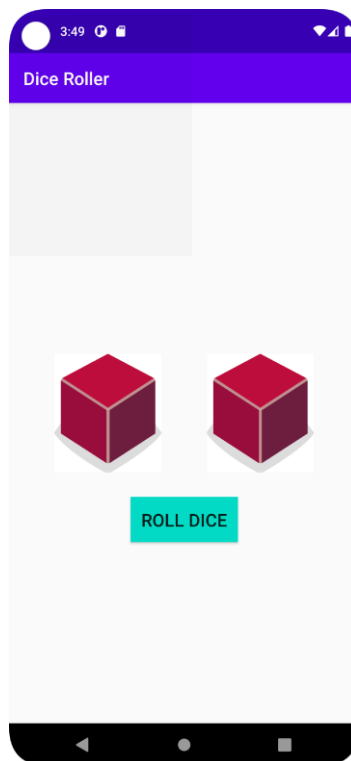
Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt	12
Tabel 2. Source Code Jawaban Soal 1 activity_main.xml	14

SOAL 1

Soal Praktikum:

Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. 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.



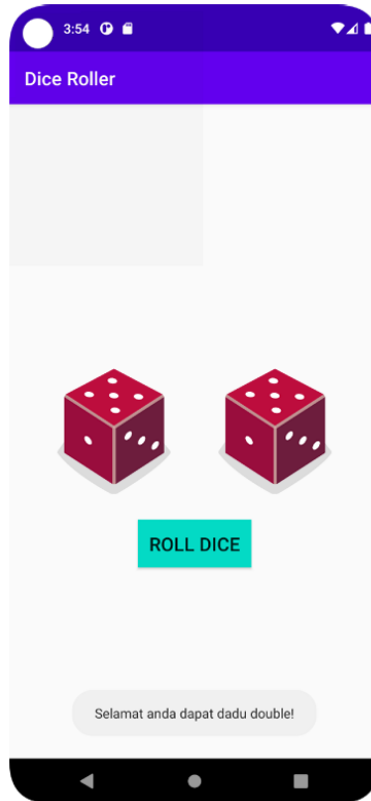
Gambar 1. Contoh Tampilan Awal Aplikasi

2. Setelah user menekan tombol “Roll Dice” maka masing-masing dadu akan memunculkan sisi dadu masing-masing dengan angka antara 1 s/d 6. Apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2 maka akan menampilkan pesan “Anda belum beruntung!” seperti dapat dilihat pada Gambar 2.



Gambar 2. Contoh 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 dapat dilihat pada Gambar 3.



Gambar 3. Contoh Tampilan Roll Dadu Double

4. Upload aplikasi yang telah anda buat kedalam repository github ke dalam **folder Module 2 dalam bentuk project**. Jangan lupa untuk melakukan **Clean Project** sebelum mengupload pekerjaan anda pada repo.
5. Untuk gambar dadu dapat didownload pada link berikut:
https://drive.google.com/file/d/14V3qXGdFnuaYN4AGd_9SgFh8kw8X9ySm/view

A. Source Code

1. MainActivity.kt

1	<code>package com.example.prakmodul1</code>
2	
3	<code>import android.os.Bundle</code>
4	<code>import android.widget.Toast</code>


```

5 import androidx.activity.ComponentActivity
6 import androidx.activity.compose.setContent
7 import androidx.compose.foundation.Image
8 import androidx.compose.foundation.layout.*
9 import androidx.compose.foundation.shape.RoundedCornerShape
10 import androidx.compose.material3.*
11 import androidx.compose.runtime.*
12 import androidx.compose.runtime.saveable.rememberSaveable
13 import androidx.compose.ui.Alignment
14 import androidx.compose.ui.Modifier
15 import androidx.compose.ui.graphics.Color
16 import androidx.compose.ui.platform.LocalContext
17 import androidx.compose.ui.res.painterResource
18 import androidx.compose.ui.unit.dp
19 import com.example.prakmodul1.ui.theme.Prakmodul1Theme
20
21 class MainActivity : ComponentActivity() {
22     companion object {
23         private const val pesandouble = "Selamat anda dapat
24         dadu double!"
25         private const val kurangberuntung = "Anda belum
26         beruntung!"
27     }
28
29     override fun onCreate(savedInstanceState: Bundle?) {
30         super.onCreate(savedInstanceState)
31         setContent {
32             Prakmodul1Theme {
33                 DiceRollerApp()
34             }
35         }
36
37         @OptIn(ExperimentalMaterial3Api::class)

```

```

37     @Composable
38     fun DiceRollerApp() {
39         var dice1 by rememberSaveable { mutableStateOf(0) }
40         var dice2 by rememberSaveable { mutableStateOf(0) }
41
42         val context = LocalContext.current
43
44         Scaffold(
45             topBar = {
46                 TopAppBar(
47                     title = {
48                         Text(
49                             text = "Dice Roller",
50                             style =
MaterialTheme.typography.headlineMedium
51                         )
52                     },
53                     colors =
TopAppBarDefaults.topAppBarColors(
54                         containerColor = Color(0xFF6200EE),
55                         titleContentColor = Color.White
56                     )
57                 )
58             }
59         ) { innerPadding ->
60             Column(
61                 modifier = Modifier
62                     .fillMaxSize()
63                     .padding(innerPadding)
64                     .padding(16.dp),
65                 horizontalAlignment =
Alignment.CenterHorizontally,
66                 verticalArrangement = Arrangement.Center
67             ) {

```

```

68         Row(
69             horizontalArrangement =
Arrangement.spacedBy(32.dp),
70             verticalAlignment =
Alignment.CenterVertically
71         ) {
72             DiceImage(diceValue = dice1)
73             DiceImage(diceValue = dice2)
74         }
75
76         Spacer(modifier = Modifier.height(48.dp))
77
78         Button(
79             onClick = {
80                 dice1 = (1..6).random()
81                 dice2 = (1..6).random()
82
83                 val message = if (dice1 == dice2)
pesandouble else kurangberuntung
84                 Toast.makeText(context, message,
Toast.LENGTH_SHORT).apply {
85 view?.setBackgroundResource(android.R.color.transparent)
86                 }.show()
87             },
88             shape = RoundedCornerShape(0.dp),
89             colors = ButtonDefaults.buttonColors(
90                 containerColor = Color(0xFF00BCD4),
91                 contentColor = Color.White
92             ),
93             modifier = Modifier
94                 .width(200.dp)
95                 .height(50.dp)
96         ) {
97             Text(

```

98	text = "ROLL DICE",
99	style =
	MaterialTheme.typography.labelLarge
100)
101	}
102	}
103	}
104	}
105	
106	@Composable
107	fun DiceImage(diceValue: Int) {
108	val imageRes = when (diceValue) {
109	0 -> R.drawable.dice_0
110	1 -> R.drawable.dice_1
111	2 -> R.drawable.dice_2
112	3 -> R.drawable.dice_3
113	4 -> R.drawable.dice_4
114	5 -> R.drawable.dice_5
115	else -> R.drawable.dice_6
116	}
117	
118	Image(
119	painter = painterResource(id = imageRes),
120	contentDescription = "Dice showing \$diceValue",
121	modifier = Modifier.size(100.dp)
122)
123	}
124	}

Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt

2. activity_main.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<LinearLayout

```

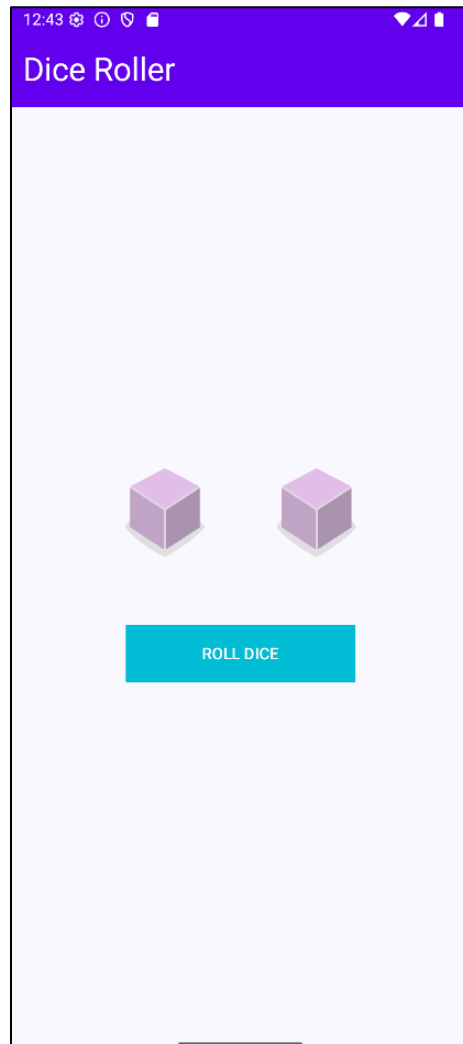
3  xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
4  android:layout_width="match_parent"
5  android:layout_height="match_parent"
6  android:orientation="vertical"
7  android:gravity="center"
8  android:padding="16dp"
9  tools:context=".MainActivity">
10
11  <LinearLayout
12      android:layout_width="wrap_content"
13      android:layout_height="wrap_content"
14      android:orientation="horizontal"
15      android:layout_marginBottom="32dp">
16
17      <ImageView
18          android:id="@+id/dice1"
19          android:layout_width="100dp"
20          android:layout_height="100dp"
21          android:src="@drawable/dice_0"
22          android:layout_marginEnd="16dp"/>
23
24      <ImageView
25          android:id="@+id/dice2"
26          android:layout_width="100dp"
27          android:layout_height="100dp"
28          android:src="@drawable/dice_0"/>
29  </LinearLayout>
30
31  <TextView
32      android:id="@+id/resultText"
33      android:layout_width="wrap_content"
34      android:layout_height="wrap_content"
35      android:layout_marginBottom="16dp"

```

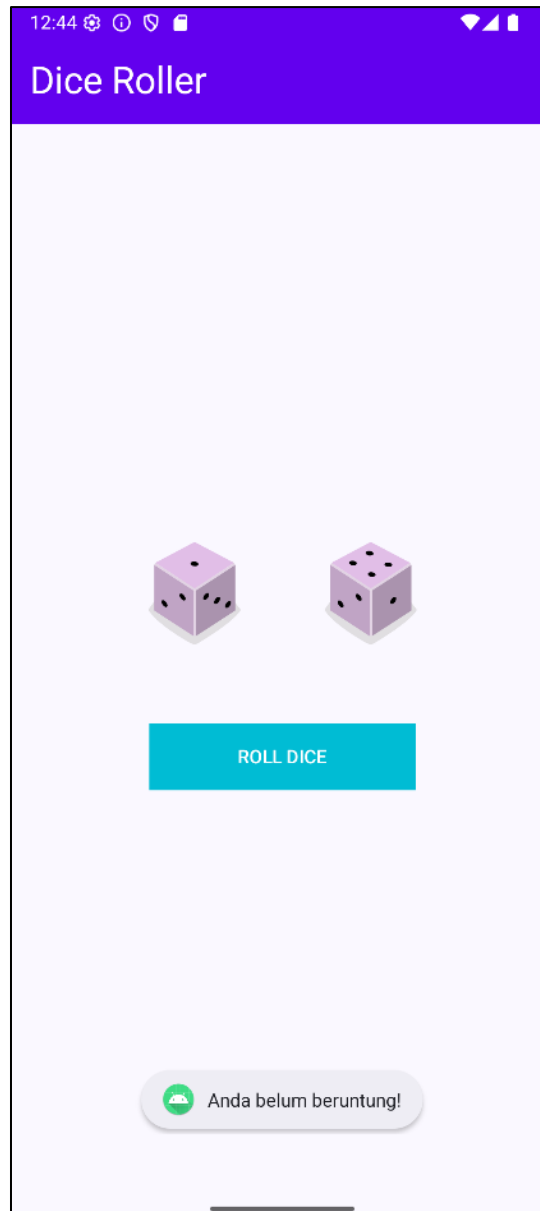
36	android:textSize="18sp"
37	android:textStyle="bold"/>
38	
39	<Button
40	android:id="@+id/Button"
41	android:layout_width="wrap_content"
42	android:layout_height="wrap_content"
43	android:text="Roll Dice"
44	android:textAllCaps="false"
45	android:paddingHorizontal="32dp"
46	android:paddingVertical="8dp"/>
47	
48	</LinearLayout>
49	

Tabel 2. Source Code Jawaban Soal 1 activity_main.xml

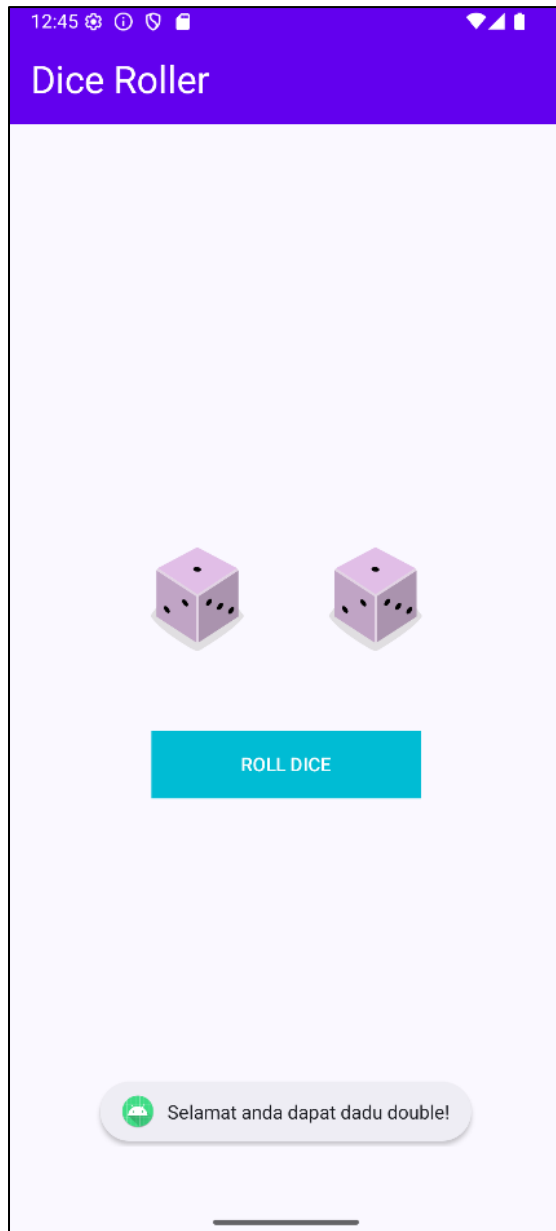
B. Output Program



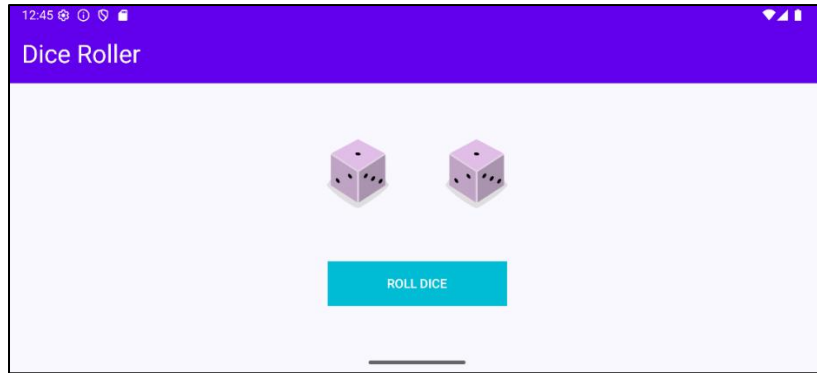
Gambar 4. Screenshot Hasil Jawaban Soal 1 Tampilan Awal



Gambar 5. Screenshot Hasil Jawaban Soal 1 Tidak Double



Gambar 6. Screenshot Hasil Jawaban Soal 1 Double



Gambar 7. Screenshot Hasil Jawaban Soal 1 Mode Landscape

C. Pembahasan

1. MainActivity.kt:

- Pada baris [1], dideklarasikan nama package file Kotlin yang dikelompokkan file ini ke dalam `com.example.prakmodul1`.
- Pada baris [3] hingga [19], `import` adalah perintah yang digunakan untuk mengimpor kelas, fungsi, atau objek dari package lain tanpa harus menyebutkan path lengkapnya.
- Pada baris [21], `class MainActivity : AppCompatActivity()` ini digunakan sebagai titik awal aplikasi yang akan mengatur tampilan aplikasi.
- Pada baris [22], `companion object` digunakan untuk menyimpan anggota (variabel/fungsi) yang bersifat statis.
- Pada baris [23] dan [24], `private const val pesandouble = "Selamat anda dapat dadu double!"` nantinya digunakan untuk menyimpan teks yang akan ditampilkan ketika hasil kedua dadu nilainya sama atau double. `private const val kurangberuntung = "Anda belum beruntung!"` ini nantinya untuk menyimpan teks yang akan ditampilkan ketika kedua dadu tidak sama.
- Pada baris [27], `override fun onCreate(savedInstanceState: Bundle?)` berfungsi untuk menerima fungsi `onCreate` dari `ComponentActivity`.

- Pada baris [28], `super.onCreate(savedInstanceState)` berfungsi untuk memastikan fungsi bawaan Android tetap dijalankan dengan benar, seperti pelacakan lifecycle dan manajemen sistem.
- Pada baris [29] hingga [31], `setContent{...}` digunakan untuk menetapkan tampilan UI dari aplikasi, dan diterapkan tampilan dari `Prakmodul1Theme`, dan `DiceRollerApp()` yang berisi UI pada dadu.
- Pada baris [36], `@OptIn(ExperimentalMaterial3Api::class)` merupakan penggunaan fitur yang masih bersifat eksperimental di Jetpack Compose Material 3.
- Pada baris [38] hingga [42], `fun DiceRollerApp()` akan menampilkan antarmuka aplikasi dadu ini, lalu `var dice1 by rememberSaveable { mutableStateOf(0)` untuk memastikan nilainya tetap bertahan walau terjadi perubahan konfigurasi (seperti rotasi layar). `val context = LocalContext.current` digunakan untuk mengambil konteks saat ini dari lingkungan Compose.
- Pada baris [44], `Scaffold` adalah komponen layout utama di Jetpack Compose yang menyediakan struktur standar UI aplikasi.
- Pada baris [45] hingga [50], `topBar` Menentukan konten yang akan ditampilkan di bagian atas layar dan ditetapkan judul Dice Roller pada bagian `topBar` tersebut.
- Pada baris [53] hingga [55], `colors = TopAppBarDefaults.topAppBarColors(...)` ini digunakan untuk mengatur warna dari app bar tadi.
- Pada baris [59], `innerPadding` adalah penutup dari `Scaffold` untuk menghindari tumpang tindih dengan sistem UI.
- Pada baris [60] hingga [66], `Column` adalah layout vertikal di Compose dan Semua elemen di dalamnya akan ditumpuk dari atas ke bawah. `Modifier` digunakan untuk mengatur ukuran dan tampilan komponen.
- Pada baris [68] hingga [70], `Row` digunakan untuk mengatur layout horizontal pada Compose dan diatur ukuran dan tampilan komponennya pada baris ini.

- Pada baris [72] dan [73], `DiceImage` digunakan untuk menampilkan gambar dan angka dari kedua dadu tersebut.
- Pada baris [78], `Button` merupakan komponen tombol dari Jetpack Compose dan dapat diedit sendiri.
- Pada baris [79] hingga [81], `onClick` merupakan proses yang dijalankan pada saat tombol diklik dan dadu akan dirandom angkanya.
- Pada baris [83], `val message = if (dice1 == dice2) pesandouble else kurangberuntung` digunakan untuk menampilkan pesan dadu double maupun tidak double.
- Pada baris [84], `Toast.makeText(...)` digunakan untuk menampilkan pesan singkat di layar berdasarkan kondisi.
- Pada baris [88] hingga [91], `shape = RoundedCornerShape(0.dp)` digunakan untuk mengatur tombol menjadi bentuk persegi, dan selanjutnya diatur warna pada tombol.
- Pada baris [93], `modifier = Modifier ...` digunakan untuk mengatur ukuran tombol.
- Pada baris [97] hingga [99], `Text` digunakan untuk menampilkan **teks** di dalam UI aplikasi.
- Pada baris [106], `@Composable` merupakan anotasi dari Compose, yang dimana ini sebagai penanda bahwa `DiceImage` adalah fungsi yang bisa digunakan dalam UI deklaratif Jetpack Compose.
- Pada baris [107], `fun DiceImage(diceValue: Int)` ini nantinya akan menerima satu nilai dadu dari 0 hingga 6 dalam bentuk integer dan menentukan gambar mana yang akan tampil nanti.
- Pada baris [109] hingga [115], akan ditampilkan gambar dadu sesuai dengan nilai dadu yang sudah diacak sebelumnya.
- Pada baris [118] hingga [121], `Image` merupakan komponen yang nantinya akan menampilkan gambar pada output. `painter = painterResource(id = imageRes)` akan mengambil gambar dari resource drawable berdasarkan `imageRes`. `contentDescription =`

"Dice showing \$diceValue" sebagai deskripsi gambar bagi pengguna.

2. activity_main.xml

- Pada baris [1], `<?xml version="1.0" encoding="utf-8"?>` ini digunakan sebagai deklarasi XML dengan menggunakan XML versi 1.0 dengan encoding UTF-8.
- Pada baris [2] hingga [9], `LinearLayout` ini digunakan sebagai container utama dengan orientasi vertikal (komponen ditata dari atas ke bawah). `xmlns:android` merupakan namespace standar untuk atribut Android. Untuk selanjutnya diatur lebar layar, tinggi layar, layout akan disejajarkan di Tengah, dan lain sebagainya.
- Pada baris [11] hingga [15], `LinearLayout` ini digunakan sebagai container untuk dua dadu.
- Pada baris [17] hingga [22], `ImageView` ini nantinya digunakan untuk menampilkan gambar dadu, yang dimana pada bagian ini untuk mengatur tampilan pada dadu pertama. Pada bagian ini diatur sumber gambar dadu, lebar, dan juga tinggi dadu.
- Pada baris [24] hingga [28], `ImageView` ini nantinya digunakan untuk menampilkan gambar dadu, yang dimana pada bagian ini untuk mengatur tampilan pada dadu kedua. Pada bagian ini diatur sumber gambar dadu, lebar, dan juga tinggi dadu.
- Pada baris [31] hingga [37], `TextView` ini nantinya akan digunakan untuk menampilkan hasil dari roll dadu dan diatur lebar, tinggi, ketebalan, dan ukuran teks dari hasil roll dadu ini.
- Pada baris [39] hingga [46], `Button` merupakan tombol yang nantinya akan digunakan untuk melakukan roll dadu pada saat tombol ditekan dan diatur pula teks pada tombol ini yaitu "Roll Dice".
- Pada baris [48], `</LinearLayout>` digunakan untuk menutup `LinearLayout` utama dari file XML ini.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.