

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Muhammad Raka Azwar NIM. 2210817210012

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 : Raymond Hariyono
NIM : 2310817210007

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Salsabila Syifa
NIM. 2010817320004

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 01 011

DAFTAR ISI

LEMBAR PENGESAHAN	3
DAFTAR ISI	4
DAFTAR GAMBAR.....	5
DAFTAR TABEL	6
SOAL 1	7
A. Source Code.....	9
Source Code XML.....	9
Source Code Compose.....	Error! Bookmark not defined.
B. Output Program	19
C. Pembahasan	20
Tautan Git.....	24

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	11
---	----

DAFTAR TABEL

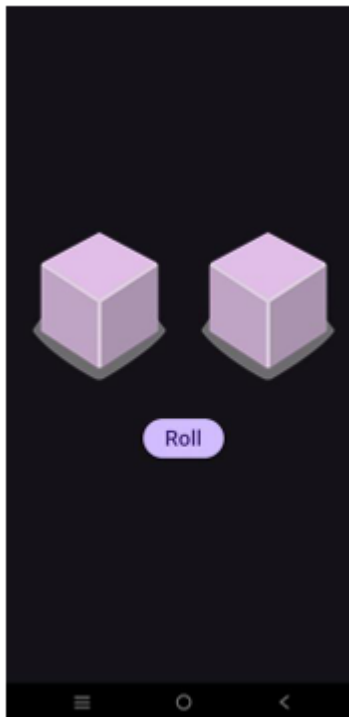
Tabel 1. Source Code Jawaban Soal 1	Error! Bookmark not defined.
Tabel 2. Source Code Jawaban Soal 1	10

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.



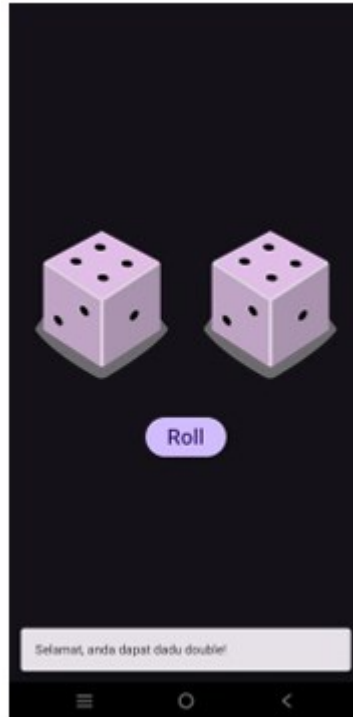
Gambar 1 Tampilan Awal Aplikasi

2. Setelah user menekan tombol “Roll 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 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 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

Source Code XML

MainActivity.kt

1	package com.example.modul1
2	
3	import android.os.Bundle
4	import android.widget.Button

```
5 import android.widget.ImageView
6 import android.widget.TextView
7 import androidx.activity.ComponentActivity
8
9
10 class diceRollerXML : ComponentActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14
15         val buttonRoll =
16 findViewById<Button>(R.id.buttonRoll)
17         val diceImageLeft =
18 findViewById<ImageView>(R.id.DiceLeft)
19         val diceImageRight =
20 findViewById<ImageView>(R.id.DiceRight)
21         val resultMessage =
22 findViewById<TextView>(R.id.resultMessage)
23
24
25         diceImageLeft.setImageResource(R.drawable.dice_0)
26         diceImageRight.setImageResource(R.drawable.dice_0)
27
28         buttonRoll.setOnClickListener {
29             val randomNumberLeft = (1..6).random()
30             val randomNumberRight = (1..6).random()
31
32             val drawableResourceLeft = when
33 (randomNumberLeft) {
34                 1 -> R.drawable.dice_1
35                 2 -> R.drawable.dice_2
36                 3 -> R.drawable.dice_3
37                 4 -> R.drawable.dice_4
38                 5 -> R.drawable.dice_5
```

39	else -> R.drawable.dice_6
40	}
41	val drawableResourceRight = when
42	(randomNumberRight) {
43	1 -> R.drawable.dice_1
44	2 -> R.drawable.dice_2
45	3 -> R.drawable.dice_3
46	4 -> R.drawable.dice_4
47	5 -> R.drawable.dice_5
48	else -> R.drawable.dice_6
49	}
50	diceImageRight
51	.setImageResource(drawableResourceRight)
52	
53	diceImageLeft.setImageResource(drawableResourceLeft)
54	
55	if (randomNumberLeft == randomNumberRight) {
56	resultMessage.text = "Selamat, anda mendapat
57	dadu double!"
58	
59	resultMessage.setBackgroundResource(R.drawable.rounded_borde
60	r_win_textview)
61	} else {
62	resultMessage.text = "Anda Belum beruntung"
63	
64	resultMessage.setBackgroundResource(R.drawable.rounded_borde
65	r_lose_textview)
66	
67	}
68	}
69	}
70	}

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/constraintLayout"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      tools:context=".diceRollerXML">
10
11
12      <Button
13          android:id="@+id/buttonRoll"
14          android:layout_width="wrap_content"
15          android:layout_height="wrap_content"
16          android:text="@string/roll"
17          android:background="@drawable/rounded_border_button"
18          app:layout_constraintBottom_toBottomOf="parent"
19          app:layout_constraintEnd_toEndOf="parent"
20          app:layout_constraintHorizontal_bias="0.498"
21          app:layout_constraintStart_toStartOf="parent"
22          app:layout_constraintTop_toTopOf="parent"
23          app:layout_constraintVertical_bias="0.639" />
24
25      <ImageView
26          android:id="@+id/DiceLeft"
27          android:layout_width="200dp"
28          android:layout_height="200dp"
29          app:layout_constraintBottom_toBottomOf="parent"
30          app:layout_constraintEnd_toEndOf="parent"
31          app:layout_constraintHorizontal_bias="1.0"
32          app:layout_constraintStart_toStartOf="parent"
33          app:layout_constraintTop_toTopOf="parent"
```

34	app:layout_constraintVertical_bias="0.393"
35	app:srcCompat="@drawable/dice_0" />
36	
37	<ImageView
38	android:id="@+id/DiceRight"
39	android:layout_width="200dp"
40	android:layout_height="200dp"
41	app:layout_constraintBottom_toBottomOf="parent"
42	app:layout_constraintEnd_toEndOf="parent"
43	app:layout_constraintHorizontal_bias="0.075"
44	app:layout_constraintStart_toStartOf="parent"
45	app:layout_constraintTop_toTopOf="parent"
46	app:layout_constraintVertical_bias="0.393"
47	app:srcCompat="@drawable/dice_0"
48	tools:ignore="ImageContrastCheck" />
49	
50	<TextView
51	android:id="@+id/resultMessage"
52	android:layout_width="0dp"
53	android:layout_height="40dp"
54	android:gravity="center"
55	android:textSize="15sp"
56	android:textColor="@android:color/black"
57	app:layout_constraintBottom_toBottomOf="parent"
58	app:layout_constraintEnd_toEndOf="parent"
59	app:layout_constraintStart_toStartOf="parent" />
60	</androidx.constraintlayout.widget.ConstraintLayout>

Tabel 2. Source Code Jawaban Soal 1 XML

Source Code Compose

MainActivity.kt

```
1 Package com.example.diceRollerCompose
2
3 import android.os.Bundle
4 import androidx.activity.ComponentActivity
5 import androidx.activity.compose.setContent
6 import androidx.activity.enableEdgeToEdge
7 import androidx.compose.foundation.Image
8 import androidx.compose.foundation.background
9 import androidx.compose.foundation.layout.Arrangement
10 import androidx.compose.foundation.layout.Column
11 import androidx.compose.foundation.layout.Row
12 import androidx.compose.foundation.layout.Spacer
13 import androidx.compose.foundation.layout.WindowInsets
14 import androidx.compose.foundation.layout.asPaddingValues
15 import androidx.compose.foundation.layout.fillMaxSize
16 import androidx.compose.foundation.layout.fillMaxWidth
17 import androidx.compose.foundation.layout.height
18 import androidx.compose.foundation.layout.padding
19 import androidx.compose.foundation.layout.safeDrawing
20 import androidx.compose.foundation.layout.size
21 import androidx.compose.foundation.shape.RoundedCornerShape
22 import androidx.compose.material3.Button
23 import androidx.compose.material3.Text
24 import androidx.compose.runtime.Composable
25 import androidx.compose.runtime.getValue
26 import androidx.compose.runtime.mutableIntStateOf
27 import androidx.compose.runtime.mutableStateOf
28 import androidx.compose.runtime.remember
29 import androidx.compose.runtime.setValue
30 import androidx.compose.ui.Alignment
31 import androidx.compose.ui.Modifier
32 import androidx.compose.ui.draw.clip
```

```
33 import androidx.compose.ui.graphics.Color
34 import androidx.compose.ui.res.painterResource
35 import androidx.compose.ui.res.stringResource
36 import androidx.compose.ui.tooling.preview.Preview
37 import androidx.compose.ui.unit.dp
38 import
39 com.example.diceRollerCompose.ui.theme.diceRollerComposeTheme
40
41 class MainActivity : ComponentActivity() {
42     override fun onCreate(savedInstanceState: Bundle?) {
43         super.onCreate(savedInstanceState)
44         enableEdgeToEdge()
45         setContent {
46             diceRollerComposeTheme {
47                 DiceRollerApp()
48             }
49         }
50     }
51 }
52
53 @Composable
54 fun DiceRollerApp() {
55     DiceWithButtonImageAndMessage()
56 }
57
58
59 @Composable
60 fun DiceWithButtonImageAndMessage() {
61     var resultLeft by remember { mutableIntStateOf(0) }
62     var resultRight by remember { mutableIntStateOf(0) }
63     var messageResult by remember { mutableStateOf("") }
64
65     val imageResourceLeft = when (resultLeft) {
66         0 -> R.drawable.dice_0
```

```

67         1 -> R.drawable.dice_1
68         2 -> R.drawable.dice_2
69         3 -> R.drawable.dice_3
70         4 -> R.drawable.dice_4
71         5 -> R.drawable.dice_5
72         else -> R.drawable.dice_6
73     }
74     val imageResourceRight = when (resultRight) {
75         0 -> R.drawable.dice_0
76         1 -> R.drawable.dice_1
77         2 -> R.drawable.dice_2
78         3 -> R.drawable.dice_3
79         4 -> R.drawable.dice_4
80         5 -> R.drawable.dice_5
81         else -> R.drawable.dice_6
82     }
83
84     Column(
85         modifier = Modifier
86             .fillMaxSize()
87
88         .padding(WindowInsets.safeDrawing.asPaddingValues())
89             .padding(16.dp),
90         horizontalAlignment = Alignment.CenterHorizontally,
91         verticalArrangement = Arrangement.Center
92     ) {
93         Spacer(modifier = Modifier.weight(1f))
94         Row(
95             modifier = Modifier.fillMaxWidth(),
96             horizontalArrangement = Arrangement.Center
97         ) {
98             Image(
99                 painter = painterResource(imageResourceLeft),
100                contentDescription = "Dice $resultLeft",

```



```

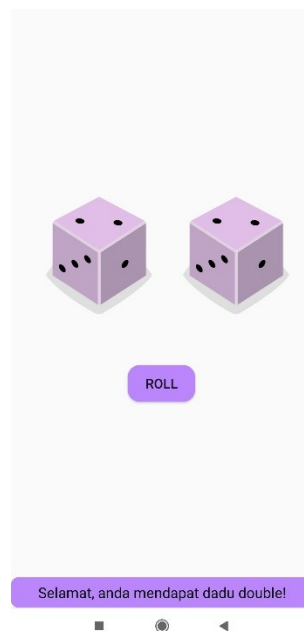
101         modifier = Modifier.size(200.dp)
102     )
103     Image(
104         painter                                     =
105     painterResource(imageResourceRight),
106         contentDescription = "Dice $resultRight",
107         modifier = Modifier.size(200.dp)
108     )
109 }
110
111     Spacer(modifier = Modifier.height(16.dp))
112
113     Button(
114         onClick = {
115             resultLeft = (1..6).random()
116             resultRight = (1..6).random()
117
118             messageResult = if (resultLeft == resultRight)
119 {
120                 "Selamat, Anda dapat dadu double"
121             } else {
122                 "Anda belum beruntung, coba lagi!"
123             }
124         }
125     ) {
126         Text(stringResource(R.string.roll))
127     }
128
129     Spacer(modifier = Modifier.weight(1f))
130
131     if (messageResult.isNotEmpty()) {
132         Text(
133             text = messageResult,
134             color = Color.Black,

```

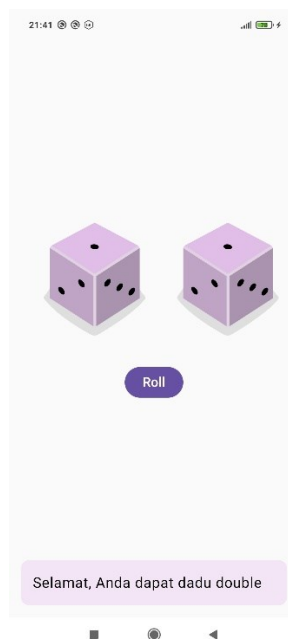
135	modifier = Modifier
136	.fillMaxWidth()
137	.clip(RoundedCornerShape(12.dp))
138	.background(Color(0xFFF3E5F5))
139	.padding(16.dp)
140)
141	} else {
142	Spacer(modifier = Modifier.height(56.dp))
143	}
144	}
145	}
146	
147	@Preview(showBackground = true)
148	@Composable
149	fun PreviewDiceRollerApp() {
150	diceRollerComposeTheme {
151	DiceRollerApp()
152	}
153	}

Tabel 3. Source Code Jawaban Soal 1 Compose

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1 XML



Gambar 2. Screenshot Hasil Jawaban Soal 1 Compose

C. Pembahasan

Pembahasan XML

MainActivity.kt:

Line 1: mendeklarasikan file merupakan file XML dengan encoding UTF-8

Line 2 – 6 : membuka ConstraintLayout sebagai layout utama. Yang memiliki tiga namespace yang dideklarasikan android: untuk atribut dasar Android. app: untuk atribut khusus AndroidX dan ConstraintLayout. tools: untuk preview di Android Studio (tidak dibawa ke runtime).

Line 7: Memberikan id pada layout, yaitu @+id/constraintLayout.

Line 9: Tinggi layout juga mengisi seluruh tinggi layar (match_parent).

Line 10: Menyetel context preview ke kelas diceRollerXML,

Line 13-14: Membuat komponen Button dengan ID @+id/buttonRoll agar bisa dipanggil ke mainActivity.

Line 15 – 16: Ukuran tombol mengikuti isi teks (wrap_content).

Line 17: Teks tombol diambil dari file string resource @string/roll.

Line 18: Memberikan latar belakang kustom dari file drawable rounded_border_button.

Line 19-24: Menentukan constraint untuk mengatur posisi tombol berada ditengah dan memiliki jarak.

Line 27 membuat ImageView dengan ID @+id/DiceLeft

Line 28-29 mengatur ukuran gambar dadu dengan panjang dan lebar 200dp x 200 dp

Line 30-35 mengatur posisis ImageView ditengah agak keatas

Line 36: Gambar awal dadu diambil dari @drawable/dice_0.

Line 39: membuat ImageView dengan ID @+id/DiceRight.

Line 40 – 41: mengatur ukuran gambar dadu dengan panjang dan lebar 200dp x 200 dp

Line 42 – 47: Posisi dadu kanan diletakkan di tengah atas

Line 48: Gambar awal dadu diambil dari @drawable/dice_0.

Line 49: tools:ignore digunakan agar Android Studio mengabaikan peringatan kontras pada preview gambar.

Line 52: Membuat TextView dengan ID `@+id/resultMessage` untuk menampilkan pesan hasil roll.

Line 52: Lebar 0dp berarti akan mengikuti constraint start dan end (match constraint).

Line 54: Menentukan tinggi TextView 40dp

Line 55: teks di align ditengah dari background (gravity="center").

Line 56: Menentukan ukuran teks menjadi 15sp.

Line 57: Menggunakan warna teks hitam dari sistem android

Line 58 – 60: TextView ditempatkan di tengah bawah layar dengan constraint start dan end ke parent.

activity_main.xml:

Line 1: merupakan package file Kotlin yaitu `com.example.modul1`.

Line 3 – 6: Mengimpor class yang dibutuhkan: Bundle: digunakan saat activity dibuat. Button, ImageView, TextView: komponen UI. AppCompatActivity: kelas dasar dari activity yang digunakan.

Line 8: mendeklarasikan class `diceRollerXML` yang merupakan turunan dari AppCompatActivity

Line 9 – 11: Override fungsi `onCreate`, fungsi ini dipanggil saat activity pertama kali dibuat. Baris 10 memanggil fungsi super untuk menjalankan proses bawaan Android. Baris 11 mengatur layout yang digunakan oleh activity ini, yaitu `activity_main.xml`.

Line 13 – 16: Mendeklarasikan variabel dan menghubungkan dengan elemen UI di XML menggunakan `findViewById()`: `buttonRoll`: tombol untuk melempar dadu. `diceImageLeft` dan `diceImageRight`: gambar dadu kiri dan kanan. `resultMessage`: teks hasil lemparan dadu.

Line 19 – 20: Mengatur gambar awal kedua buah dadu default dengan dadu tanpa angka (`dice_0`).

Line 22: Membuat listener saat tombol `buttonRoll` diklik menggunakan `buttonRoll.setOnClickListener`

Line 23 – 24: Menghasilkan aksi klik pada tombol memakai random()

Line 26 – 33: Pengkondisian when yang mana akan menampilkan angka dadu sebelah kiri sesuai angka acak dan mengambil UI dari drawable

Line 34-41: Pengkondisian when yang mana akan menampilkan angka dadu sebelah kiri sesuai angka acak dan mengambil UI dari drawable

Line 41 – 42 menampilkan gambar dadu kiri dan kanan berdasarkan angka acak yang mengambil imageresource dari drawable

Line 45 – 47: jika kedua angka sama maka akan menampilkan pesan dadu double dan mengatur backgrounds sesuai dengan di drawable yang sudah diatur untuk pesan menang

Line 48 – 51: jika kedua angka sama maka akan menampilkan pesan dadu double dan mengatur backgrounds sesuai dengan di drawable yang sudah diatur untuk pesan kalah

Line 52 – 54 block penutup setOnClickListener, onCreate, dan class.

Pembahasan Compose

MainActivity.kt

Line 1: menunjukkan bahwa file Kotlin berada pada package com.example.diceRollerCompose.

Line 4 – 31: merupakan import statement untuk fungsi dan class dari library android dan jetpack compose

Line 5: Bundle: digunakan untuk menyimpan dan memulihkan data dalam lifecycle Android.

Line 6: ComponentActivity: superclass untuk activity berbasis Compose.

Line 7: setContent: menetapkan konten UI menggunakan Jetpack Compose.

Line 8: enableEdgeToEdge: mengaktifkan tampilan edge-to-edge agar UI menyesuaikan dengan layar penuh (menyentuh notifikasi bar dan navigation bar).

Line 9 – 14: Layout dasar Compose seperti Column, Row, Spacer, Modifier, padding, fillMaxSize, dll.

Line 15: RoundedCornerShape: digunakan untuk membuat sudut membulat pada elemen UI.

Line 16: Button: tombol interaktif dari Material3.

Line 17: Text: untuk menampilkan teks pada layar.

Line 18 – 22: State management: remember, mutableStateOf, mutableIntStateOf

Line 23 – 24: painterResource dan stringResource: untuk memuat gambar dan string dari resource XML.

Line 25: Preview: memungkinkan melihat tampilan UI langsung di Android Studio.

Line 26: dp: satuan ukuran yang digunakan Compose.

Line 27: Mengimpor tema khusus dari folder ui.theme.

Line 32 – 42: Merupakan activity utama

Line 32: Meng-extend ComponentActivity, yang kompatibel dengan Jetpack Compose.

Line 33: Lifecycle method yang dipanggil saat aktivitas dimulai.

Line 34: Memanggil onCreate dari superclass.

Line 35: Mengatur tampilan agar menggunakan seluruh layar.

Line 36- 40: digunakan untuk mengisi UI dengan compose, diceRollerComposeTheme menerapkan styling (warna, font, ukuran), DiceRollerApp() adalah fungsi utama untuk memanggil tampilan aplikasi.

Line 41 – 42: menutup fungsi onCreate() dan MainActivity.

Line 44 – 46: fungsi yang bersifat composable, yang digunakan untuk menghasilkan UI

Line 49 – 114 : Fungsi tampilan utama

Line 49 – 50: Fungsi utama yang menampilkan UI lengkap, dengan 2 gambar dadu, tombol dan pesan.

Line 51 – 53 : resultLeft & resultRight: angka dadu kiri & kanan. messageResult: string pesan saat user menekan tombol. remember: menyimpan nilai selama komposisi ulang.

Line 55 – 63 menentukan gambar dadu kiri berdasarkan angka resultLeft

Line 64 – 72 : menentukan gambar dadu kanan berdasarkan angka resultRight

Line 74 – 113: merupakan kode untuk layout

Line 74 – 81: membuat column atau layout vertical untuk semua layout, mengisi ukuran penuh dengan fillMaxSize, memberi padding agar tidak memenuhi status bar, dan padding 16dp dari tepi

Line 82: Spacer untuk memberi jarak diatas dadu

Line 83- 97: membuat row untuk menampilkan gambar dadu secara horizontal dan berada ditengah

Line 87 – 96: menampilkan gambar dadu dari imagesource dan mengatur ukuran ke 200 dp

Line 98-111: Mengacak dua angka dari 1–6. Menampilkan pesan berdasarkan hasil:

jika angka sama (double): menampilkan pesan selamat anda dapat dadu double. Jika beda: user mendapat pesan anda belum beruntung, coba lagi. `stringResource(R.string.roll)`: Mengambil teks tombol dari file strings.xml.

Line 113 – 126: Jika `messageResult` tidak kosong, tampilkan pesan ke layar. Jika kosong (belum klik tombol), beri jarak bawah agar layout tetap rapi.

Baris 130 – 136: Fungsi ini digunakan oleh Android Studio untuk melihat tampilan UI secara langsung di editor. Tidak digunakan saat aplikasi dijalankan di device.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/raymondhariyono/mobile-praktikum>