

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 1**



**ANDROID BASIC WITH KOTLIN**

**Oleh:**

**Siti Ratna Dwinta Sari**

**NIM. 2310817120002**

**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 : Siti Ratna Dwinta Sari  
NIM : 2310817120002

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.....	11
C.    Pembahasan .....	14
D.    Tautan Git .....	15

## **DAFTAR GAMBAR**

Gambar 1. Tampilan Awal Aplikasi.....	11
Gambar 2. Tampilan Dadu Setelah di Roll, Jika Dadu Berbeda .....	12
Gambar 3. Tampilan Dadu Setelah di Roll, Jika Dadu Sama.....	13

## DAFTAR TABEL

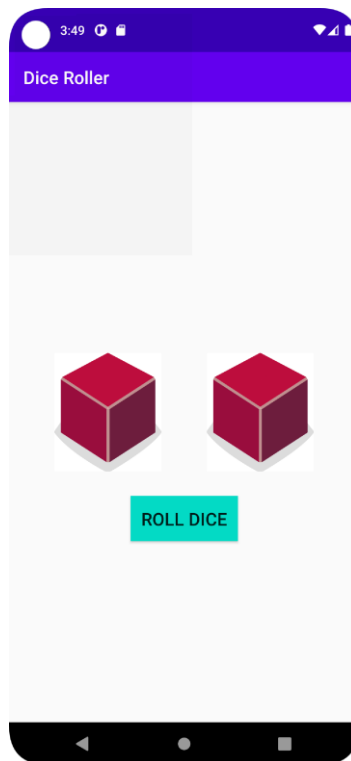
Tabel 1. Source Code Jawaban Soal 1.....	8
--	---

## 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.



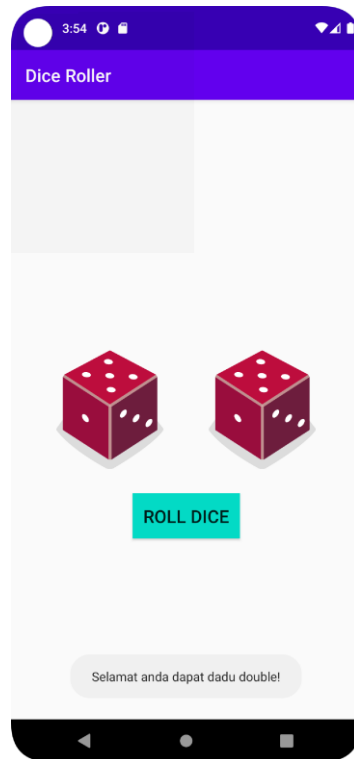
**Gambar 1 Tampilan Awal Aplikasi**

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.
4. Upload aplikasi yang telah anda buat kedalam repository github ke dalam **folder Module 1 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/u/0/uc?id=147HT2lIH5qin3z5ta7H9y2N\\_5OMW81Ll&export=download](https://drive.google.com/u/0/uc?id=147HT2lIH5qin3z5ta7H9y2N_5OMW81Ll&export=download)



**Gambar 3 Tampilan Roll Dadu Double**

## A. Source Code

### 1. MainActivity.kt

*Tabel 1. Source Code Jawaban Soal 1*

1	package com.example.diceroller
2	
3	
4	import android.widget.Toast
5	import androidx.compose.ui.platform.LocalContext
6	import android.os.Bundle
7	import androidx.activity.ComponentActivity
8	import androidx.activity.compose.setContent
9	import androidx.activity.enableEdgeToEdge
10	import androidx.compose.foundation.Image
11	import androidx.compose.foundation.layout.*
12	import androidx.compose.material3.*
13	import androidx.compose.runtime.*
14	import androidx.compose.ui.Alignment
15	import androidx.compose.ui.Modifier
16	import androidx.compose.ui.res.painterResource
17	import androidx.compose.ui.unit.dp
18	import androidx.compose.ui.unit.sp
19	import androidx.compose.ui.tooling.preview.Preview
20	import com.example.diceroller.ui.theme.DiceRollerTheme



```

21 import kotlin.random.Random
22 class MainActivity : ComponentActivity() {
23     override fun onCreate(savedInstanceState: Bundle?) {
24         super.onCreate(savedInstanceState)
25         enableEdgeToEdge()
26         setContent {
27             DiceRollerTheme {
28                 Scaffold(modifier = Modifier.fillMaxSize()) {
29                     innerPadding ->
30                         DiceGameScreen(
31                             modifier
32                             Modifier.padding(innerPadding)
33                         )
34                     }
35                 }
36             }
37         }
38     @Composable
39     fun DiceGameScreen(modifier: Modifier = Modifier) {
40         var dice1 by remember { mutableStateOf(0) }
41         var dice2 by remember { mutableStateOf(0) }
42         val context = LocalContext.current
43         var resultText by remember { mutableStateOf("") }
44
45         val getDiceImage = { value: Int ->
46             when (value) {
47                 1 -> R.drawable.dice_1
48                 2 -> R.drawable.dice_2
49                 3 -> R.drawable.dice_3
50                 4 -> R.drawable.dice_4
51                 5 -> R.drawable.dice_5
52                 6 -> R.drawable.dice_6
53                 else -> R.drawable.empty_dice
54             }
55         }
56
57         Column(
58             modifier = modifier
59                 .fillMaxSize()
60                 .padding(24.dp),
61             horizontalAlignment = Alignment.CenterHorizontally,
62             verticalArrangement = Arrangement.Center
63         ) {
64             Row(horizontalArrangement
65                 Arrangement.spacedBy(16.dp)) {
66                 Image(
67                     painter
68                     = painterResource(id
69                     = getDiceImage(dice1)),
69                     contentDescription = "Dadu 1",

```

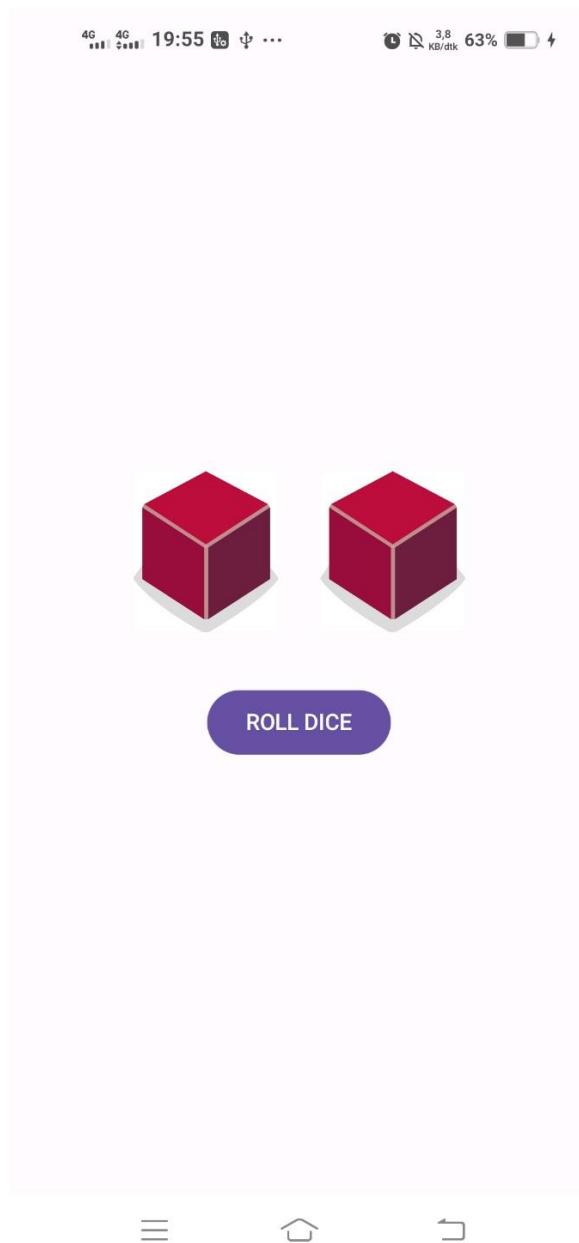
```

69         modifier = Modifier.size(100.dp)
70     )
71     Image(
72         painter = painterResource(id =
73         getDiceImage(dice2)),
74         contentDescription = "Dadu 2",
75         modifier = Modifier.size(100.dp)
76     )
77
78     Spacer(modifier = Modifier.height(32.dp))
79
80     Button(onClick = {
81         dice1 = Random.nextInt(1, 6)
82         dice2 = Random.nextInt(1, 6)
83         if (dice1 == dice2) {
84             Toast.makeText(context, "Selamat anda
mendapatkan dadu double!", Toast.LENGTH_SHORT).show()
85         } else {
86             Toast.makeText(context, "Anda belum
beruntung!", Toast.LENGTH_SHORT).show()
87         }
88     }) {
89         Text("ROLL DICE")
90     }
91
92     Spacer(modifier = Modifier.height(16.dp))
93
94     if (resultText.isNotEmpty()) {
95         Surface(
96             modifier = Modifier
97                 .fillMaxWidth()
98                 .padding(top = 16.dp),
99             shape = MaterialTheme.shapes.medium,
100             tonalElevation = 2.dp,
101             color =
102             MaterialTheme.colorScheme.surfaceVariant
103         ) {
104             Box(modifier = Modifier.padding(12.dp)) {
105                 Text(
106                     text = resultText,
107                     fontSize = 16.sp,
108                     modifier =
109                     Modifier.align(Alignment.Center)
110                 )
111             }
112         }
113     }
114
115     @Preview(showBackground = true)

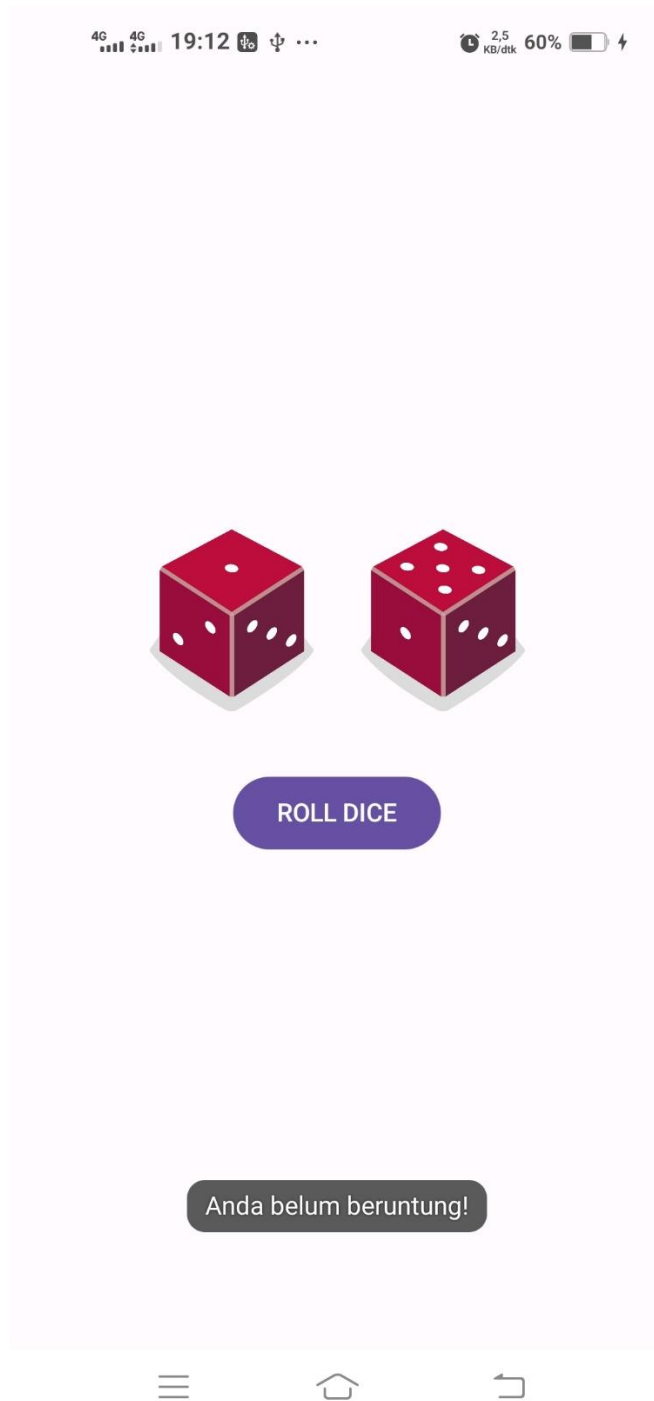
```

```
116 @Composable
117 fun DiceGamePreview() {
118     DiceRollerTheme {
119         DiceGameScreen()
120     }
121 }
```

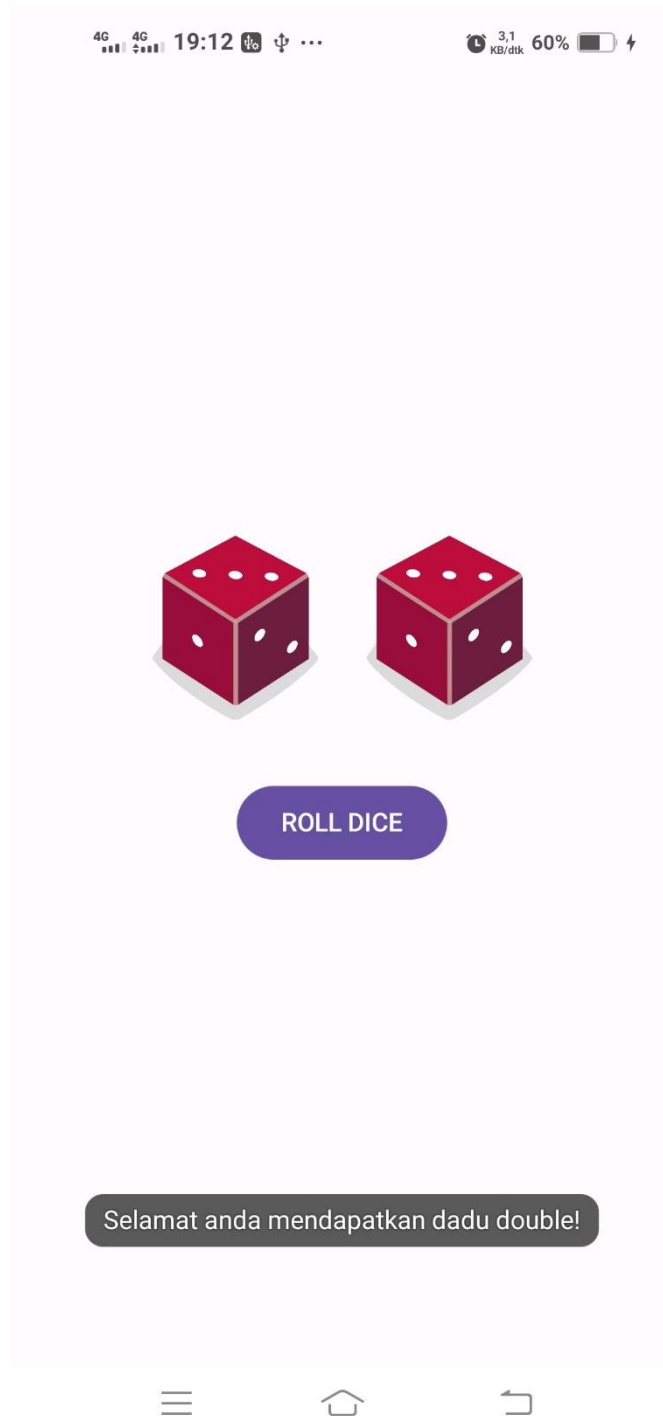
## B. Output Program



*Gambar 1. Tampilan Awal Aplikasi*



*Gambar 2. Tampilan Dadu Setelah di Roll, Jika Dadu Berbeda*



*Gambar 3. Tampilan Dadu Setelah di Roll, Jika Dadu Sama*

## C. Pembahasan

### 1. MainActivity.kt:

Pada praktikum ini dibuat sebuah aplikasi Android sederhana bernama Dice Roller menggunakan Jetpack Compose. Aplikasi ini berfungsi untuk melakukan simulasi lempar dua buah dadu secara acak. Proses utama aplikasi berada di dalam kelas `MainActivity`, di mana di dalam metode `onCreate`, dipanggil fungsi `setContent` untuk menampilkan UI dengan tema `DiceRollerTheme`. UI utama dibuat dalam fungsi `DiceGameScreen`.

Di dalam `DiceGameScreen`, terdapat dua buah variabel `dice1` dan `dice2` yang masing-masing menyimpan angka hasil lemparan dadu. Variabel tersebut dibuat menggunakan `remember` dan `mutableStateOf` sehingga perubahan nilainya akan langsung memicu perubahan tampilan pada layar. Selain itu, terdapat juga variabel `resultText` untuk menyimpan hasil teks yang ingin ditampilkan (meskipun pada implementasi saat ini belum digunakan secara optimal).

Untuk menampilkan gambar dadu sesuai angka yang dihasilkan, dibuat sebuah fungsi `getDiceImage`. Fungsi ini menerima angka dari 1 hingga 6 dan mengembalikan resource gambar dadu yang sesuai. Jika nilai dadu belum diatur (default 0), maka akan ditampilkan gambar dadu kosong.

Tata letak antarmuka dibangun menggunakan `Column`, sehingga elemen UI tersusun secara vertikal. Gambar dua buah dadu ditempatkan dalam `Row` agar tampil secara horizontal dan diberi jarak antar gambar. Sebuah tombol dengan teks "ROLL DICE" diletakkan di bawah gambar. Ketika tombol ditekan, aplikasi akan mengacak nilai `dice1` dan `dice2` dengan `Random.nextInt(1, 6)`, kemudian membandingkan kedua nilai tersebut. Jika hasilnya sama, maka akan ditampilkan pesan Toast "Selamat anda mendapatkan dadu double!". Jika berbeda, akan ditampilkan Toast "Anda belum beruntung!".

Selain itu, disiapkan juga sebuah `Surface` yang seharusnya berfungsi untuk menampilkan teks hasil di bawah tombol. Namun karena `resultText` tidak diubah dalam aksi tombol, bagian ini belum berfungsi secara penuh. Akhirnya, dibuat fungsi `DiceGamePreview` untuk mempermudah melihat pratinjau tampilan aplikasimelalui fitur preview di Android Studio tanpa harus menjalankan aplikasi di emulator atau perangkat fisik.

#### **D. Tautan Git**

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/annacoded/Pemrograman-Mobile>