

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 1**



**ANDROID BASIC WITH KOTLIN**

**Oleh:**

**Aiko Anatasha Wendiono**

**NIM. 2310817320013**

**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 : Aiko Anatasha Wendiono  
NIM : 2310817320013

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar  
NIM. 2210817210026

Muti`a Maulida S.Kom M.T.I  
NIP. 19881027 201903 20 13

## DAFTAR ISI

LEMBAR PENGESAHAN .....	2
DAFTAR ISI .....	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL .....	5
SOAL 1 .....	6
A. Source Code.....	8
B. Output Program .....	13
C. Pembahasan .....	13
D. Tautan Git .....	19

## **DAFTAR GAMBAR**

Gambar 1. Screenshot Hasil Jawaban Soal 1 .....	13
---	----

## DAFTAR TABEL

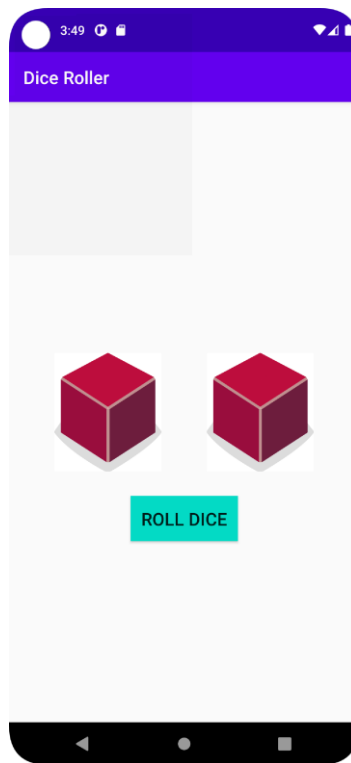
Tabel 1. Source Code.....	9
Tabel 2. Source Code.....	10
Tabel 3. Source Code.....	12

## SOAL 1

### Soal Praktikum:

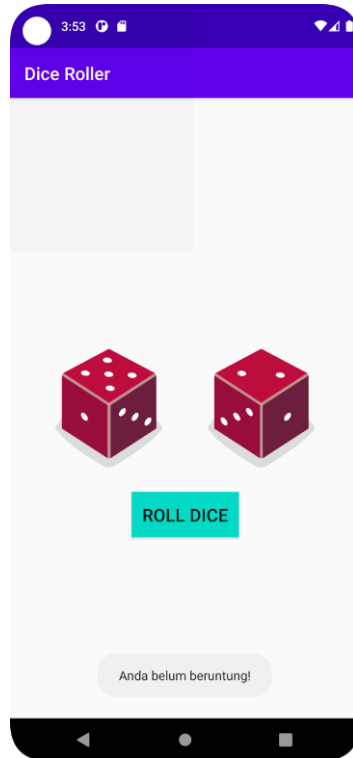
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



**Gambar 1 Tampilan Awal Aplikasi**

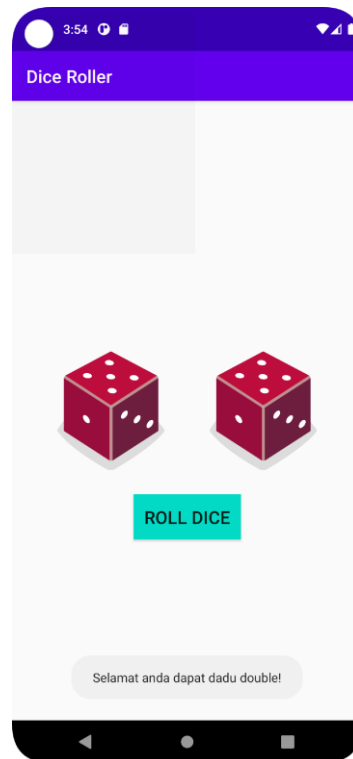
2. Setelah user menekan tombol “Roll Dice” maka masing-masing dadu akan memunculkan sisi dadu masing-masing dengan angka antara 1 s/d 6. Apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2 maka akan menampilkan pesan “Anda belum beruntung!” seperti dapat dilihat pada Gambar 2.



**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. Buatlah aplikasi tersebut menggunakan XML dan Jetpack Compose.
5. 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.
6. 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**

```

1 package com.example.diceroller
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 import kotlin.random.Random
9
10
11 class MainActivity : AppCompatActivity() {
12
13     private lateinit var diceImage1: ImageView
14     private lateinit var diceImage2: ImageView
15
16     override fun onCreate(savedInstanceState: Bundle?) {
17         super.onCreate(savedInstanceState)

```



18	setContentView(R.layout.activity_main)
19	
20	diceImage1 = findViewById(R.id.diceImage1)
21	diceImage2 = findViewById(R.id.diceImage2)
22	val
23	rollButton: Button =
24	findViewById(R.id.rollButton)
25	
26	rollButton.setOnClickListener {
27	rollDice()
28	}
29	}
30	
31	private fun rollDice() {
32	val dice1 = Random.nextInt(1, 7)
33	val dice2 = Random.nextInt(1, 7)
34	
35	diceImage1.setImageResource(getDiceImage(dice1))
36	diceImage2.setImageResource(getDiceImage(dice2))
37	
38	if (dice1 == dice2) {
39	Toast.makeText(this, "Selamat anda dapat dadu
40	double!", Toast.LENGTH_SHORT).show()
41	} else {
42	Toast.makeText(this, "Anda belum beruntung!",
43	Toast.LENGTH_SHORT).show()
44	}
45	}
46	
47	
48	private fun getDiceImage(number: Int): Int {
49	return when (number) {
50	1 -> R.drawable.dice_1
51	2 -> R.drawable.dice_2
52	3 -> R.drawable.dice_3
53	4 -> R.drawable.dice_4
54	5 -> R.drawable.dice_5
55	6 -> R.drawable.dice_6
56	else -> R.drawable.dice_0
57	}
58	}
59	}
60	}
61	}

Tabel 1. Source Code

## 2. activity\_main.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<LinearLayout
3	xmlns:android="http://schemas.android.com/apk/res/android"
4	android:layout_width="match_parent"

5	android:layout_height="match_parent"
6	android:gravity="center"
7	android:orientation="vertical"
8	android:padding="16dp">
9	
10	<LinearLayout
11	android:layout_width="wrap_content"
12	android:layout_height="wrap_content"
13	android:orientation="horizontal"
14	android:gravity="center"
15	android:layout_marginBottom="24dp">
16	
17	
18	<ImageView
19	android:id="@+id/diceImage1"
20	android:layout_width="100dp"
21	android:layout_height="100dp"
22	android:src="@drawable/dice_0" />
23	
24	<ImageView
25	android:id="@+id/diceImage2"
26	android:layout_width="100dp"
27	android:layout_height="100dp"
28	android:layout_marginStart="16dp"
29	android:src="@drawable/dice_0" />
30	
31	</LinearLayout>
32	
33	<Button
34	android:id="@+id/rollButton"
35	android:layout_width="wrap_content"
36	android:layout_height="wrap_content"
37	android:text="Roll Dice" />
38	
39	
40	</LinearLayout>
41	

Tabel 2. Source Code

### 3. Jetpack Compose

1	package com.example.myapplication
2	
3	import android.os.Bundle
4	import android.widget.Toast
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.material3.*
10	import androidx.compose.runtime.*
11	

```

12 import androidx.compose.ui.Alignment
13 import androidx.compose.ui.Modifier
14 import androidx.compose.ui.platform.LocalContext
15 import androidx.compose.ui.res.painterResource
16 import androidx.compose.ui.unit.dp
17 import kotlin.random.Random
18
19
20 class MainActivity : ComponentActivity() {
21     override fun onCreate(savedInstanceState: Bundle?) {
22         super.onCreate(savedInstanceState)
23         setContent {
24             Surface(
25                 modifier = Modifier.fillMaxSize(),
26                 color
27                 MaterialTheme.colorScheme.background
28             ) {
29                 DiceRoller()
30             }
31         }
32     }
33 }
34
35
36 @Composable
37 fun DiceRoller() {
38     var dice1 by remember { mutableStateOf(0) }
39     var dice2 by remember { mutableStateOf(0) }
40     val context = LocalContext.current
41
42     Column(
43         modifier = Modifier
44             .fillMaxSize()
45             .padding(16.dp),
46         verticalArrangement = Arrangement.Center,
47         horizontalAlignment
48         Alignment.CenterHorizontally
49     ) {
50         Row(
51             horizontalArrangement
52             Arrangement.spacedBy(16.dp),
53             verticalAlignment
54             Alignment.CenterVertically
55         ) {
56             DiceImage(dice1)
57             DiceImage(dice2)
58         }
59
60         Spacer(modifier = Modifier.height(24.dp))
61
62
63

```

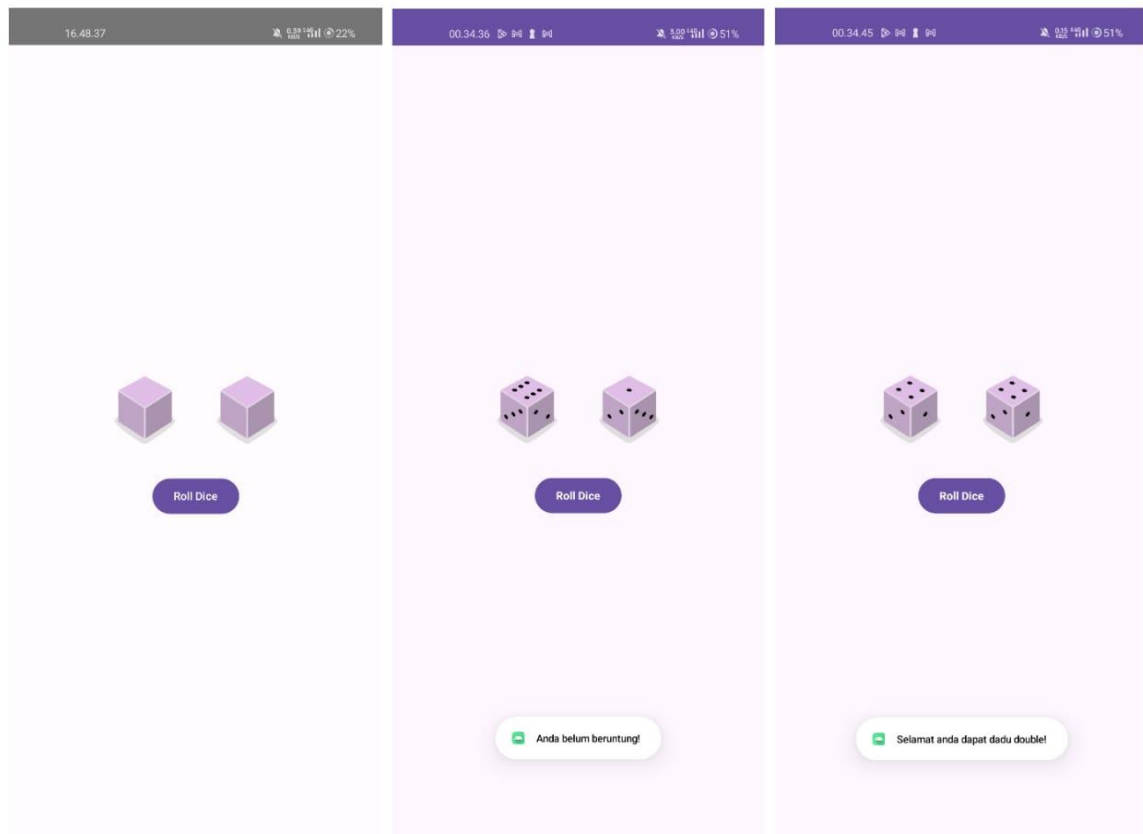
```

64
65         Button(onClick = {
66             dice1 = Random.nextInt(1, 7)
67             dice2 = Random.nextInt(1, 7)
68
69             if (dice1 == dice2) {
70                 Toast.makeText(context, "Selamat anda
71 dapat dadu double!", Toast.LENGTH_SHORT).show()
72             } else {
73                 Toast.makeText(context, "Anda belum
74 beruntung!", Toast.LENGTH_SHORT).show()
75             }
76         }) {
77             Text("Roll Dice")
78         }
79     }
80 }
81
82
83 @Composable
84 fun DiceImage(diceValue: Int) {
85     val imageResource = when (diceValue) {
86         1 -> R.drawable.dice_1
87         2 -> R.drawable.dice_2
88         3 -> R.drawable.dice_3
89         4 -> R.drawable.dice_4
90         5 -> R.drawable.dice_5
91         6 -> R.drawable.dice_6
92         else -> R.drawable.dice_0
93     }
94
95     Image(
96         painter = painterResource(id = imageResource),
97         contentDescription = "Gambar Dadu",
98         modifier = Modifier.size(100.dp)
99     )
100
101
102 }

```

Tabel 3. Source Code

## B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

## C. Pembahasan

### 1. MainActivity.kt:

**Pada line 1**, `package com.example.diceroller` menunjukkan bahwa file ini berada di dalam package bernama `com.example.diceroller`. Ini digunakan untuk mengatur struktur folder dan namespace aplikasi.

**Pada line 3-6**, dilakukan import terhadap beberapa class penting dari Android dan Kotlin yang akan digunakan dalam aplikasi:

- `android.os.Bundle` digunakan untuk menyimpan data instance activity.
- `android.widget.Button` dan `android.widget.ImageView` digunakan untuk tombol dan tampilan gambar.

- `android.widget.Toast` digunakan untuk menampilkan pesan pop-up.
- `androidx.appcompat.app.AppCompatActivity` adalah superclass untuk activity yang mendukung fitur-fitur kompatibel Android.
- `kotlin.random.Random` digunakan untuk menghasilkan angka acak (untuk dadu).

**Pada line 8**, class `MainActivity` dideklarasikan dan diwarisi dari `AppCompatActivity`, yang berarti ini adalah class utama dari activity pertama yang ditampilkan saat aplikasi berjalan.

**Pada line 10-11**, dideklarasikan dua variabel `diceImage1` dan `diceImage2` bertipe `ImageView`, yang nantinya akan menampilkan gambar dadu.

**Pada line 13-21**, terdapat method `onCreate()` yang dipanggil saat activity pertama kali dibuat.

- `super.onCreate(savedInstanceState)` memanggil implementasi `onCreate` dari superclass-nya.
- `setContentView(R.layout.activity_main)` menetapkan file layout yang digunakan, yaitu `activity_main.xml`.
- `findViewById(R.id.diceImage1)` dan `findViewById(R.id.diceImage2)` menghubungkan `ImageView` dari layout ke variabel di Kotlin.
- `findViewById(R.id.rollButton)` menghubungkan tombol `ROLL DICE` dari layout ke variabel `rollButton`.
- `rollButton.setOnClickListener` digunakan untuk memberikan aksi saat tombol ditekan, yaitu memanggil fungsi `rollDice()`.

**Pada line 23-32**, fungsi `rollDice()` berfungsi untuk mengacak nilai dadu dan menampilkan gambar serta pesan.

- `Random.nextInt(1, 7)` menghasilkan angka acak antara 1 hingga 6 untuk dua dadu.

- `diceImage1.setImageResource(...)` dan `diceImage2.setImageResource(...)` digunakan untuk mengubah gambar dadu berdasarkan angka yang dihasilkan.
- Kondisi `if (dice1 == dice2)` memeriksa apakah kedua dadu memiliki angka yang sama. Jika iya, tampilkan Toast dengan pesan "Selamat anda dapat dadu double!", jika tidak, tampilkan pesan "Anda belum beruntung!".

**Pada line 34-40**, fungsi `getDiceImage(number: Int): Int` digunakan untuk menentukan resource gambar berdasarkan angka dadu:

- Menggunakan `when` expression untuk mengembalikan `R.drawable.dice_1` hingga `R.drawable.dice_6` sesuai angka.
- Jika angka tidak termasuk dari 1-6, maka dikembalikan `R.drawable.dice_0` sebagai default.

## 2. `activity_main.xml`

**Pada line 1 dan 2**, `<LinearLayout>` adalah root dari layout ini, artinya seluruh elemen UI akan ditata di dalamnya secara linear (berurutan). Layout ini menggunakan beberapa atribut:

- `xmlns:android="http://schemas.android.com/apk/res/android"` adalah deklarasi *namespace* Android untuk mengenali atribut XML Android.
- `android:layout_width="match_parent"` dan `android:layout_height="match_parent"` artinya layout akan memenuhi seluruh lebar dan tinggi layar.
- `android:gravity="center"` artinya seluruh elemen UI akan ditempatkan di tengah layar.
- `android:orientation="vertical"` menyusun elemen secara vertikal dari atas ke bawah.
- `android:padding="16dp"` memberikan jarak padding di sekeliling layout sebesar 16dp.

**Pada line 4–15**, `LinearLayout` dalam ini digunakan untuk menampilkan dua `ImageView` secara horizontal.

- `android:layout_width="wrap_content"` dan `layout_height="wrap_content"` artinya lebar dan tinggi akan mengikuti isi kontennya.
- `android:orientation="horizontal"` menyusun elemen secara mendatar.
- `android:gravity="center"` memusatkan isinya.
- `android:layout_marginBottom="24dp"` memberikan jarak bawah dari layout ini ke elemen berikutnya.

**Pada line 6–10**, `ImageView` pertama untuk dadu kiri:

- `android:id="@+id/diceImage1"` adalah ID yang digunakan untuk mengakses dari Kotlin.
- `layout_width` dan `layout_height` diatur 100dp agar ukuran gambar tetap.
- `android:src="@drawable/dice_0"` digunakan sebagai gambar awal dadu.

**Pada line 12–16**, `ImageView` kedua untuk dadu kanan, sama seperti `diceImage1`, tapi memiliki `layout_marginStart="16dp"` agar ada jarak antara dua dadu.

**Pada line 18–21**, `Button` ini digunakan untuk mengacak dadu:

- `android:id="@+id/rollButton"` agar bisa dihubungkan dengan Kotlin code.
- `layout_width="wrap_content"` dan `layout_height="wrap_content"` agar tombol menyesuaikan isi.



- `android:text="Roll Dice"` adalah teks yang tampil di dalam tombol.

### 3. Jetpack Compose

**Pada line 1**, package `com.example.myapplication` menyatakan bahwa file ini berada di dalam package `com.example.myapplication`.

**Pada line 3–9**, Bagian import digunakan untuk:

- Mengakses elemen penting Compose seperti `Button`, `Image`, `Column`, dll.
- `Toast` untuk menampilkan pesan pop-up.
- `LocalContext` untuk mendapatkan `Context` saat ini.
- `Random` untuk menghasilkan angka acak dadu.

**Pada line 11**, `class MainActivity : ComponentActivity()` adalah activity utama yang mewarisi `ComponentActivity`, digunakan saat membuat UI dengan Jetpack Compose.

**Pada line 12–18**, `onCreate()` digunakan untuk membuat tampilan saat activity pertama kali dijalankan:

- `setContent { ... }` menandakan awal dari Compose UI.
- `Surface(...)` digunakan sebagai kontainer latar belakang.
- `DiceRoller()` adalah composable function yang berisi tampilan dan logika utama.

**Pada line 21–23:**

- `var dice1 by remember { mutableStateOf(0) }` dan `dice2` menyimpan nilai dadu, dan akan berubah secara dinamis jika ditekan tombol.
- `val context = LocalContext.current` digunakan untuk menampilkan `Toast` pada konteks sekarang.

**Pada line 25–32,** `Column` menyusun elemen UI secara vertikal dan berada di tengah layar.

- `modifier.fillMaxSize()` agar elemen mengisi seluruh layar.
- `padding(16.dp)` memberi jarak dari tepi layar.
- `verticalArrangement = Arrangement.Center` memusatkan secara vertikal.
- `horizontalAlignment = Alignment.CenterHorizontally` memusatkan secara horizontal.

**Pada line 33–37,** `Row` digunakan untuk menempatkan dua gambar dadu secara horizontal.

- `Arrangement.spacedBy(16.dp)` memberi jarak antar dadu.
- Memanggil fungsi `DiceImage(dice1)` dan `DiceImage(dice2)`.

**Pada line 39,** `Spacer(modifier = Modifier.height(24.dp))` memberi jarak vertikal antara gambar dadu dan tombol.

**Pada line 41–47,** `Button` digunakan untuk mengacak dadu:

- `Random.nextInt(1, 7)` menghasilkan angka 1–6 untuk kedua dadu.
- `if (dice1 == dice2)` akan menampilkan `Toast` jika dadu double.
- Jika tidak sama, akan muncul `Toast` “Anda belum beruntung!”.

**Pada line 51–59:**

- `val imageResource = when (diceValue) { ... }` menentukan gambar dadu berdasarkan angka yang dihasilkan (1–6).

- Jika nilai belum valid (sebelum tombol ditekan), akan ditampilkan gambar `dice_0`.

**Pada line 61–65**, `Image(...)` menampilkan gambar dadu:

- `painterResource(id = imageResource)` memuat gambar dari `resource`.
- `contentDescription = "Gambar Dadu"` untuk aksesibilitas.
- `modifier = Modifier.size(100.dp)` mengatur ukuran dadu 100x100 dp.

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

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/aikoanatashawendiono/Laporan-Praktikum-Pemrograman-Mobile>