

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 1**



**ANDROID BASIC WITH KOTLIN**

**Oleh:**

**Naufal Elyzar NIM. 231017210019**

**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 : Naufal Elyzar  
NIM : 2310817210019

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 .....	2
DAFTAR ISI .....	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL .....	5
SOAL 1 .....	6
A.    Source Code .....	8
B.    Output Program.....	12
C.    Pembahasan .....	13
D.    Tautan Git .....	15

## **DAFTAR GAMBAR**

Gambar 1 Screenshot Hasil Jawaban Soal 1 XML .....	12
Gambar 2 Screenshot Hasil Jawaban Soal 1 Jetpack Compose .....	13

## DAFTAR TABEL

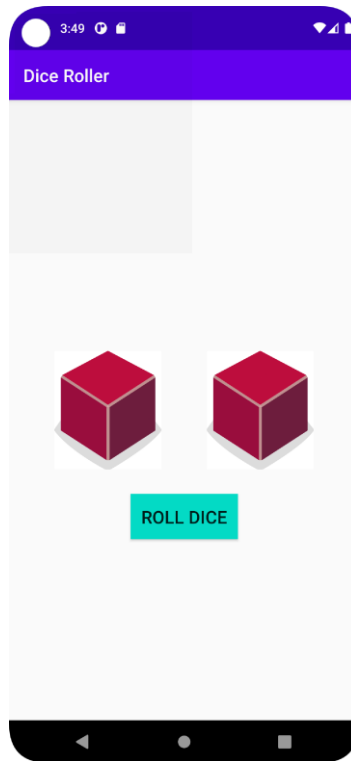
Tabel 1. Source Code Jawaban Soal 1 .....	<b>Error! Bookmark not defined.</b>
Tabel 2. Source Code Jawaban Soal 1 .....	10

## 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.



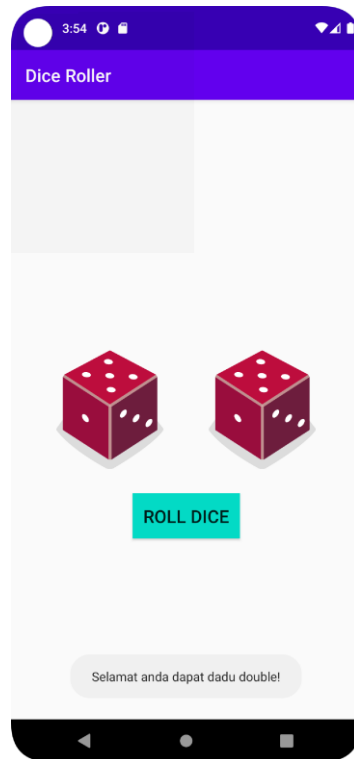
**Gambar 1 Tampilan Awal Aplikasi**

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 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/u/0/uc?id=147HT2IIH5qin3z5ta7H9y2N\\_5OMW81LI&export=download](https://drive.google.com/u/0/uc?id=147HT2IIH5qin3z5ta7H9y2N_5OMW81LI&export=download)



**Gambar 3 Tampilan Roll Dadu Double**

## **A. Source Code**

### **MainActivity.kt**

```
1 package com.example.praktiikum1
2
3 import android.os.Bundle
4 import androidx.activity.ComponentActivity
5 import androidx.activity.compose.setContent
6 import androidx.compose.foundation.Image
7 import androidx.compose.foundation.background
8 import androidx.compose.foundation.layout.*
9 import androidx.compose.material3.*
10 import androidx.compose.runtime.*
11 import androidx.compose.ui.Alignment
12 import androidx.compose.ui.Modifier
13 import androidx.compose.ui.graphics.Color
14 import androidx.compose.ui.res.painterResource
15 import androidx.compose.ui.res.stringResource
16 import androidx.compose.ui.tooling.preview.Preview
17 import androidx.compose.ui.unit.dp
18 import androidx.compose.ui.unit.sp
19 import com.example.praktiikum1.ui.theme.Praktiikum1Theme
20 import kotlinx.coroutines.launch
21
22
```



```

23 class MainActivity : ComponentActivity() {
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContent {
27             Praktikum1Theme {
28                 Surface(
29                     modifier = Modifier.fillMaxSize(),
30                     color                                     =
MaterialTheme.colorScheme.background
31                 ) {
32                     DiceRollerApp()
33                 }
34             }
35         }
36     }
37 }
38
39 @Preview
40 @Composable
41 fun DiceRollerApp() {
42     DiceWithButtonAndImage(modifier = Modifier
43         .fillMaxSize()
44         .background(Color.Black)
45         .wrapContentSize(Alignment.Center)
46     )
47 }
48
49 @Composable
50 fun DiceWithButtonAndImage(modifier: Modifier = Modifier) {
51     var result1 by remember { mutableStateOf( 0) }
52     var result2 by remember { mutableStateOf( 0) }
53     val snackbarHostState = remember { SnackbarHostState() }
54     val coroutineScope = rememberCoroutineScope()
55     val imageResource1 = when(result1) {
56         0 -> R.drawable.dice_0
57         1 -> R.drawable.dice_1
58         2 -> R.drawable.dice_2
59         3 -> R.drawable.dice_3
60         4 -> R.drawable.dice_4
61         5 -> R.drawable.dice_5
62         else -> R.drawable.dice_6
63     }
64
65     val imageResource2 = when(result2) {
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	
75	Scaffold( 76       modifier = modifier, 77       snackbarHost = { 78           SnackbarHost(hostState = snackbarHostState) { data -> 7                Snackbar(snackbarData = data, containerColor = 9       Color.LightGray, contentColor = Color.Black) 80           } 81       } 82    ) { padding -> 83       Column( 84          modifier = modifier, 85          horizontalAlignment = Alignment.CenterHorizontally 86        ) { 87          Spacer(modifier = Modifier.height(150.dp)) 88          Row(modifier = Modifier.padding(16.dp)) { 89            Image( 90              painter = painterResource(imageResource1), 91              contentDescription = result1.toString() 92            ) 93            Spacer(modifier = Modifier.weight(1f)) 94            Image( 95              painter = painterResource(imageResource2), 96              contentDescription = result2.toString() 97            ) 98            } 99          } 100        Spacer(modifier = Modifier.height(10.dp)) 101        Button(onClick = { 102              result1 = (1..6).random() 103              result2 = (1..6).random() 104              if (result1 == result2) { 105                coroutineScope.launch { 106                  snackbarHostState.showSnackbar("Selamat,    anda    dapat    dadu 107                  double!") 108                } 109              } 110            }) { 111              Text(stringResource(R.string.roll), fontSize = 112              24.sp) 113            } 114        } 115      } 116    }

*Tabel 1 Source Code Jawaban Soal 1*

**activity\_main.xml**

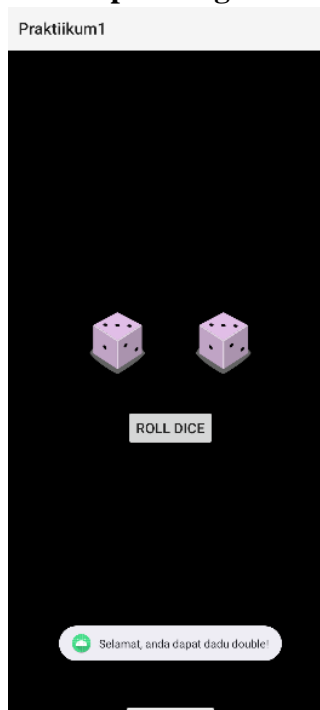
```

1 package com.example.praktikum1xml
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 lateinit var diceImage1: ImageView
11     private lateinit var diceImage2: ImageView
12     private lateinit var rollButton: Button
13
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         setContentView(R.layout.activity_main)
17
18         diceImage1 = findViewById(R.id.dice_image_1)
19         diceImage2 = findViewById(R.id.dice_image_2)
20         rollButton = findViewById(R.id.roll_button)
21         diceImage1.setImageResource(R.drawable.dice_0)
22         diceImage2.setImageResource(R.drawable.dice_0)
23
24
25         rollButton.setOnClickListener {
26             val result1 = (1..6).random()
27             val result2 = (1..6).random()
28             val imageRes1 = getDiceImage(result1)
29             val imageRes2 = getDiceImage(result2)
30
31             diceImage1.setImageResource(imageRes1)
32             diceImage2.setImageResource(imageRes2)
33
34             if (result1 == result2) {
35                 Toast.makeText(this, "Selamat, anda dapat dadu
double!", Toast.LENGTH_SHORT).show()
36             }
37         }
38     }
39
40     private fun getDiceImage(result: Int): Int {
41         return when (result) {
42             0 -> R.drawable.dice_0
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     }
51 }

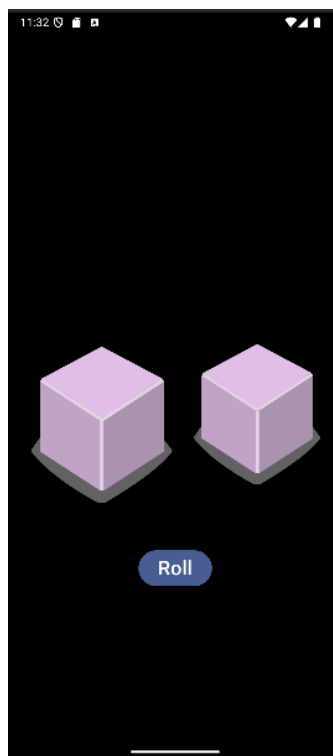
```

Tabel 2. Source Code Jawaban Soal 1

## B. Output Program



Gambar 1 Screenshot Hasil Jawaban Soal 1 XML



### **C. Pembahasan**

#### **MainActivity.kt:**

Pada line 1, dideklarasikan package bernama com.example.praktiikum1 yg merupakan identitas unik dari aplikasi ini.

Pada line 3, diimport Bundle yg digunakan untuk menyimpan dan mengelola data antar aktivitas

Pada line 4, diimport untuk Activity yg digunakan pada Jetpack Compose.

Pada line 5, diimport digunakan untuk menentukan tampilan UI komposisi.

Pada line 6, diimport Image dari Jetpack Compose yg memungkinkan menampilkan gambar di UI.

Pada line 7, diimport untuk memberi latar warna pada elemen UI.

Pada line 8, diimport layout dari Jetpack Compose, yg mencakup beberapa fungsi untuk mengatur tata letak seperti Column, Row, dan Spacer.

Pada line 9, diimport material3.\* yg mencakup komponen Material Design 3 seperti Button, Text, Surface, dll.

Pada line 10, diimport untuk pengelolaan state dan penandaan fungsi komposabel.

Pada line 11, diimport untuk mengatur penyusunan elemen dalam layout.

Pada line 12, diimport untuk memodifikasi tampilan seperti ukuran, padding, dan warna.

Pada line 13, diimport untuk menggunakan warna.

Pada line 14, diimport untuk memuat gambar dari resource drawable berdasarkan ID.

Pada line 15, diimport untuk memuat string dari resource (seperti teks tombol).

Pada line 16, diimport untuk melihat tampilan UI di Android Studio Preview.

Pada line 17, diimport untuk menentukan ukuran dimensi dalam density-independent pixels.

Pada line 18, diimport untuk menentukan ukuran font dalam scale-independent pixels.

Pada line 19, diimport Praktikum1Theme, tema khusus aplikasi yg mendefinisikan skema warna dan bentuk komponen.

Pada line 20, diimport launch yg digunakan untuk menjalankan tugas asinkron seperti menampilkan snackbar.

Pada line 23, dideklarasikan kelas MainActivity yg merupakan turunan dari ComponentActivity, aktivitas utama aplikasi.

Pada line 24, di-override fungsi onCreate, yg dipanggil saat Activity pertama kali dibuat.

Pada line 25, dipanggil fungsi super.onCreate(savedInstanceState) untuk memanggil versi onCreate dari superclass.

Pada line 26, digunakan fungsi setContent untuk mendefinisikan UI aplikasi menggunakan Jetpack Compose.

Pada line 27-31, digunakan tema Praktikum1Theme dan Surface sebagai container utama, lalu pada baris [32] dipanggil fungsi DiceRollerApp() yg akan membentuk UI.

Pada line 41, fungsi DiceRollerApp dideklarasikan sebagai fungsi @Composable dan juga @Preview.

Pada line 42-45, dipanggil fungsi DiceWithButtonAndImage dengan modifier yang mengatur ukuran layar penuh, latar belakang hitam, dan konten di tengah layar.

Pada line 50, fungsi DiceWithButtonAndImage dideklarasikan sebagai fungsi @Composable dengan parameter modifier.

Pada line 51, dideklarasikan variabel result1 yang menyimpan hasil lemparan dadu pertama, menggunakan remember dan mutableStateOf.

Pada line 52, dideklarasikan variabel result2 untuk dadu kedua, sama seperti result1.

Pada line 53, dibuat snackbarHostState untuk mengatur state dari snackbar.

Pada line 54, dibuat coroutineScope untuk menjalankan coroutine seperti snackbar secara asinkron.

Pada line 55-63, dideklarasikan imageResource1 yang akan menentukan gambar dadu pertama berdasarkan nilai result1.

Pada line 65-73, dideklarasikan imageResource2 yang akan menentukan gambar dadu kedua berdasarkan nilai result2.

Pada line 75, digunakan komponen Scaffold sebagai struktur UI utama yang memiliki snackbarHost.

Pada line 76-81, didefinisikan bagaimana tampilan snackbar ditampilkan, termasuk warna latar dan teks.

Pada line 84, komponen Column digunakan untuk menyusun elemen secara vertikal dan terpusat secara horizontal.

Pada line 88, ditambahkan Spacer untuk memberi jarak dari atas layar sejauh 150dp.

Pada line 90, dibuat Row untuk menampilkan 2 gambar dadu dalam satu baris.

Pada line 91-94, dadu pertama ditampilkan sebagai gambar dengan Image, menggunakan painterResource(imageResource1) dan contentDescription sesuai angka dadu.

Pada line 96-99, dadu kedua ditampilkan sebagai gambar.

Pada line 101, Spacer memberi jarak ke bawah sejauh 10dp.

Pada line 103, didefinisikan Button yang saat diklik akan melempar kedua dadu.

Pada line 104-105, hasil dadu result1 dan result2 akan diisi dengan angka acak dari 1 sampai 6.

Pada line 106-108, jika kedua hasil sama (dadu double), maka akan ditampilkan snackbar dengan teks "Selamat, anda dapat dadu double!" secara asinkron menggunakan coroutine.

Terakhir, pada line 112, teks tombol diambil dari string resource roll, dengan ukuran font 24sp.

### **activity\_main.xml:**

Pada line 1, terdapat versi XML dan encoding yang digunakan

Pada line 3, diimport Bundle.

Pada line 4, diimport Button.

Pada line 5, diimport ImageView.

Pada line 6, diimport Toast.

Pada line 7, diimport AppCompatActivity

Pada line 9, dideklarasikan kelas MainActivity yang merupakan turunan dari AppCompatActivity, dan menjadi Activity utama yang dijalankan.

Pada line 10, dideklarasikan properti diceImage1 sebagai ImageView yang nantinya akan mereferensikan gambar dadu pertama.

Pada line 11, dideklarasikan properti diceImage2 sebagai ImageView untuk gambar dadu kedua.

Pada line 12, dideklarasikan properti rollButton sebagai Button yang akan digunakan untuk melempar dadu saat diklik.

Pada line 14, fungsi onCreate di-override, yang akan dipanggil ketika Activity pertama kali dimulai.

Pada line 15, dipanggil super.onCreate(savedInstanceState) untuk memastikan siklus Activity berjalan dengan benar.

Pada line 16, layout utama Activity diatur menggunakan setContentView yang memuat activity\_main.xml.

Pada line 18, diceImage1 dihubungkan dengan ImageView dari XML yang memiliki ID dice\_image\_1.

Pada line 19, diceImage2 dihubungkan dengan ImageView dari XML yang memiliki ID dice\_image\_2.

Pada line 20, rollButton dihubungkan dengan Button dari XML yang memiliki ID roll\_button.

Pada line 21, gambar awal untuk diceImage1 diatur menjadi dice\_0, gambar dadu kosong.

Pada line 22, gambar awal untuk diceImage2 juga diatur menjadi dice\_0.

Pada line 25, ditambahkan listener setOnClickListener pada rollButton yang akan dijalankan ketika tombol ditekan.

Pada line 26, dideklarasikan variabel result1 yang merupakan angka acak dari 1 sampai 6 untuk dadu pertama.

Pada line 27, dideklarasikan variabel result2 sebagai angka acak dari 1 sampai 6 untuk dadu kedua.

Pada line 28, dipanggil fungsi getDiceImage(result1) untuk mendapatkan resource gambar sesuai angka result1.

Pada line 29, dipanggil fungsi getDiceImage(result2) untuk gambar dadu kedua sesuai angka result2.

Pada line 31, gambar diceImage1 diperbarui menggunakan imageRes1.

Pada line 32, gambar diceImage2 diperbarui menggunakan imageRes2.

Pada line 34-35, terdapat pengecekan jika hasil kedua dadu sama maka akan ditampilkan pesan "Selamat, anda dapat dadu double!".

Pada line 40, dideklarasikan fungsi getDiceImage sebagai fungsi yang menerima result berupa angka hasil dadu.

Pada line 41, fungsi when digunakan untuk mencocokkan angka hasil result dan mengembalikan resource gambar yang sesuai.

Pada line 40, jika hasil 0, gambar dice\_0 akan dikembalikan.

Pada line 42-47, jika hasil antara 1 hingga 5, maka akan dikembalikan gambar dice\_1 hingga dice\_5.

Pada line 48, untuk nilai di luar yang disebutkan (default), maka akan dikembalikan gambar dice\_6.

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

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/NaufalElyzar11/PraktikumMobile>