

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Bi'ahlil Haq Aulia Akbar Awaludin NIM. 2010817110011

**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 : Bi'ahlil Haq Aulia Akbar Awaludin
NIM : 2010817110011

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Raka Azwar
NIM. 2210817210012

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

DAFTAR ISI

LEMBAR PENGESAHAN.....	1
DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
DAFTAR TABEL.....	4
SOAL 1.....	5
A. Source Code.....	8
Compose.....	8
XML.....	10
B. Output Program.....	14
C. Pembahasan.....	15
Compose.....	15
XML.....	15
D. Tautan Git.....	16

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1 Compose	14
Gambar 2. Screenshot Hasil Jawaban Soal 1 XML	14

DAFTAR TABEL

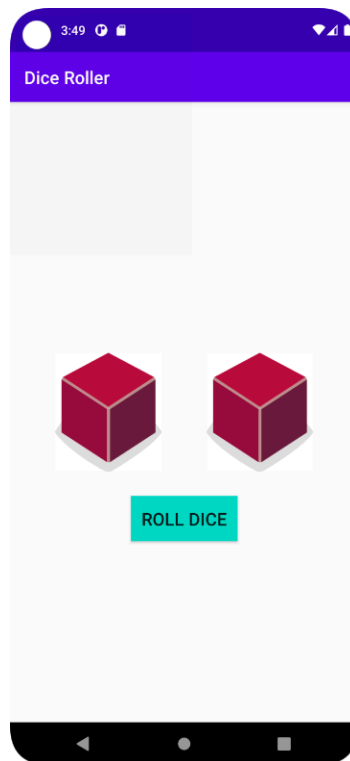
Tabel 1. Source Code Jawaban Soal 1	8
Tabel 2. Source Code Jawaban Soal 1	10
Tabel 3. Source Code Jawaban Soal 1	12

SOAL 1

Soal Praktikum:

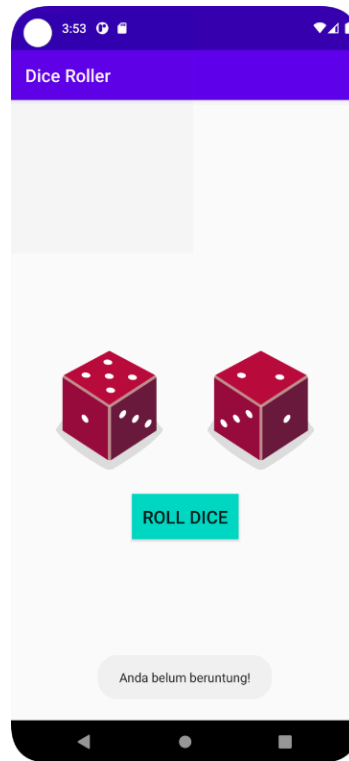
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



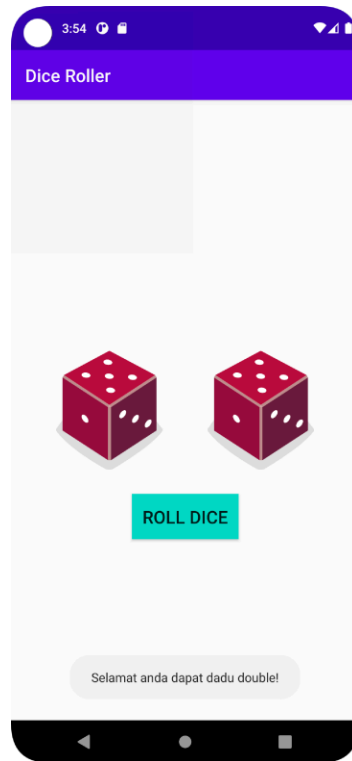
Gambar 1 Tampilan Awal Aplikasi

2. Setelah user menekan tombol “Roll” maka masing-masing dadu akan memperlihatkan sisi dadunya dengan angka antara 1 s/d 6. Apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2, maka aplikasi akan menampilkan pesan “Anda belum beruntung!” seperti yang 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.
Untuk gambar dadu dapat didownload pada link berikut:
https://drive.google.com/file/d/14V3qXGdFnuoYN4AGd_9SgFh8kw8X9ySm/view?usp=sharing

A. Source Code

Compose MainActivity.kt

```
1 package com.example.diceroller
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.fillMaxSize
14 import androidx.compose.foundation.layout.fillMaxWidth
15 import androidx.compose.foundation.layout.height
16 import androidx.compose.foundation.layout.padding
17 import androidx.compose.foundation.layout.wrapContentSize
18 import androidx.compose.material3.BottomAppBar
19 import androidx.compose.material3.Button
20 import androidx.compose.material3.MaterialTheme
21 import androidx.compose.material3.Scaffold
22 import androidx.compose.material3.Surface
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.graphics.Color
33 import androidx.compose.ui.res.painterResource
34 import androidx.compose.ui.res.stringResource
35 import androidx.compose.ui.tooling.preview.Preview
36 import androidx.compose.ui.unit.dp
37 import com.example.diceroller.ui.theme.DiceRollerTheme
38
39 class MainActivity : ComponentActivity() {
40     override fun onCreate(savedInstanceState: Bundle?) {
41         super.onCreate(savedInstanceState)
42         enableEdgeToEdge()
43         setContent {
44             DiceRollerTheme {
```

```

45         Surface(
46             modifier = Modifier.fillMaxSize(),
47         ) {
48             DiceRollerApp()
49         }
50     }
51 }
52 }
53 }
54
55 @Preview
56 @Composable
57 fun DiceRollerApp() {
58     DiceWithButtonAndImage(modifier = Modifier
59         .fillMaxSize()
60         .wrapContentSize(Alignment.Center)
61         .background(MaterialTheme.colorScheme.background)
62     )
63 }
64
65 @Composable
66 private fun DiceWithButtonAndImage(modifier: Modifier =
67     Modifier) {
68     var dice1 by remember { mutableIntStateOf(0) }
69     var dice2 by remember { mutableIntStateOf(0) }
70     val imageResource1 = when (dice1) {
71         0 -> R.drawable.dice_0
72         1 -> R.drawable.dice_1
73         2 -> R.drawable.dice_2
74         3 -> R.drawable.dice_3
75         4 -> R.drawable.dice_4
76         5 -> R.drawable.dice_5
77         else -> R.drawable.dice_6
78     }
79     val imageResource2 = when (dice2) {
80         0 -> R.drawable.dice_0
81         1 -> R.drawable.dice_1
82         2 -> R.drawable.dice_2
83         3 -> R.drawable.dice_3
84         4 -> R.drawable.dice_4
85         5 -> R.drawable.dice_5
86         else -> R.drawable.dice_6
87     }
88     Column(
89         modifier = Modifier,
90         horizontalAlignment = Alignment.CenterHorizontally
91     ) {
92         Spacer(modifier = Modifier.weight(1f))
93         Row {
94             Image(painter = painterResource(imageResource1),
95                 contentDescription = dice1.toString())

```

94	Image(painter = painterResource(imageResource2),
95	contentDescription = dice2.toString())
96	}
97	Spacer(modifier = Modifier.height(22.dp))
98	Button(onClick = {
99	dice1 = (1..6).random()
100	dice2 = (1..6).random()
101	}) { Text(stringResource(R.string.roll)) }
102	Spacer(modifier = Modifier.weight(1f))
103	if (dice1 != 0 dice2 != 0)
104	Text(
105	stringResource(
106	if (dice1 == dice2) R.string.twins else
107	R.string.zonk
108),
109	color =
110	MaterialTheme.colorScheme.background,
111	style =
112	MaterialTheme.typography.labelMedium,
113	modifier = Modifier
114	.fillMaxWidth()
115	.background(MaterialTheme.colorScheme.primary)
116	.padding(8.dp)
117)
	}
	}

Tabel 1. Source Code Jawaban Soal 1

XML

MainActivity.kt

1	package com.example.dicerollerxml
2	
3	import android.os.Bundle
4	import android.widget.Button
5	import android.widget.ImageView
6	import android.widget.TextView
7	import android.widget.Toast
8	import androidx.activity.enableEdgeToEdge
9	import androidx.appcompat.app.AppCompatActivity
10	import androidx.core.view.ViewCompat
11	import androidx.core.view.WindowInsetsCompat
12	
13	class MainActivity : AppCompatActivity() {
14	override fun onCreate(savedInstanceState: Bundle?) {
15	super.onCreate(savedInstanceState)
16	enableEdgeToEdge()
17	setContentView(R.layout.activity_main)
18	val diceImage1: ImageView = findViewById(R.id.dice1)
19	val diceImage2: ImageView = findViewById(R.id.dice2)

```

20         val rollButton: Button =
21     findViewById(R.id.rollButton)
22         val bottomText: TextView =
23     findViewById(R.id.messageText)
24
25         rollButton.setOnClickListener {
26             val dice1 = (1..6).random()
27             val dice2 = (1..6).random()
28             val drawable1 = when (dice1) {
29                 1 -> R.drawable.dice_1
30                 2 -> R.drawable.dice_2
31                 3 -> R.drawable.dice_3
32                 4 -> R.drawable.dice_4
33                 5 -> R.drawable.dice_5
34                 else -> R.drawable.dice_6
35             }
36             val drawable2 = when (dice2) {
37                 1 -> R.drawable.dice_1
38                 2 -> R.drawable.dice_2
39                 3 -> R.drawable.dice_3
40                 4 -> R.drawable.dice_4
41                 5 -> R.drawable.dice_5
42                 else -> R.drawable.dice_6
43             }
44             diceImage1.setImageResource(drawable1)
45             diceImage2.setImageResource(drawable2)
46
47             bottomText.apply {
48                 text = getString(
49                     if (dice1 == dice2) R.string.twins else
50 R.string.zonk
51                 )
52
53             setBackgroundColor(getColor(R.color.purple_500))
54                 setTextColor(getColor(R.color.white))
55                 setPadding(16, 16, 16, 16)
56                 visibility = TextView.VISIBLE
57             }
58         }
59
60     ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.
61 main)) { v, insets ->
62
63         val systemBars =
64     insets.getInsets(WindowInsetsCompat.Type.systemBars())
65         v.setPadding(systemBars.left, systemBars.top,
66     systemBars.right, systemBars.bottom)
67         insets
68     }
69 }
70 }

```

Tabel 2. Source Code Jawaban Soal 1

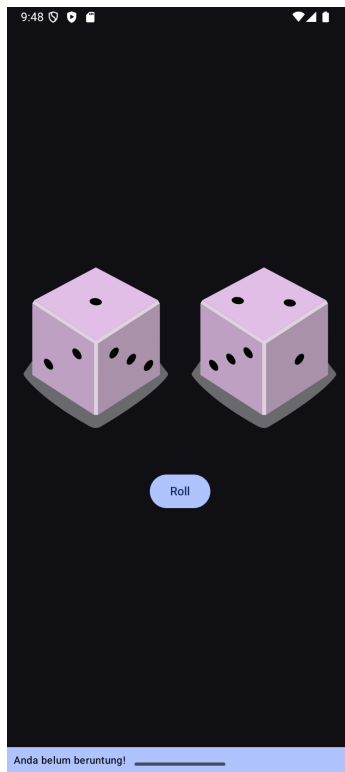
activity_main.xml

```
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/main"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:contentDescription="@string/dice2"
10     tools:context=".MainActivity"
11     tools:layout_editor_absoluteX="10dp"
12     tools:layout_editor_absoluteY="32dp">
13
14
15     <ImageView
16         android:id="@+id/dice1"
17         android:layout_width="100dp"
18         android:layout_height="100dp"
19
20         android:layout_marginTop="260dp"
21         android:contentDescription="@string/dice1"
22         android:src="@drawable/dice_0"
23
24         app:layout_constraintEnd_toStartOf="@+id/dice2"
25         app:layout_constraintStart_toStartOf="parent"
26         app:layout_constraintHorizontal_chainStyle="packed"
27
28         app:layout_constraintTop_toTopOf="parent"
29
30     app:layout_constraintBottom_toTopOf="@+id/rollButton"
31         app:layout_constraintVertical_bias="0.0"
32     />
33
34     <ImageView
35         android:id="@+id/dice2"
36         android:layout_width="100dp"
37         android:layout_height="100dp"
38
39         android:contentDescription="@string/dice2"
40
41         android:src="@drawable/dice_0"
42         app:layout_constraintBottom_toBottomOf="@+id/dice1"
43
44     app:layout_constraintBottom_toTopOf="@+id/rollButton"
45         app:layout_constraintEnd_toEndOf="parent"
46         app:layout_constraintHorizontal_chainStyle="packed"
```

47	
48	app:layout_constraintStart_toEndOf="@+id/dice1"
49	app:layout_constraintTop_toTopOf="@+id/dice1"
50	app:layout_constraintVertical_bias="0.0" />
51	
52	<Button
53	android:id="@+id/rollButton"
54	android:layout_width="wrap_content"
55	android:layout_height="wrap_content"
56	android:layout_marginTop="40dp"
57	android:backgroundTint="#D0BFFF"
58	android:text="@string/roller"
59	android:textColor="#000000"
60	
61	app:layout_constraintEnd_toEndOf="parent"
62	app:layout_constraintHorizontal_bias="0.498"
63	app:layout_constraintStart_toStartOf="parent"
64	app:layout_constraintTop_toBottomOf="@id/dice1"
65	
66	/>
67	
68	<TextView
69	android:id="@+id/messageText"
70	android:layout_width="0dp"
71	android:layout_height="wrap_content"
72	android:layout_margin="10dp"
73	android:layout_marginBottom="8dp"
74	android:background="#FFFFFF"
75	android:padding="12dp"
76	android:text="@string/zonk"
77	android:textColor="#000000"
78	android:visibility="invisible"
79	app:layout_constraintBottom_toBottomOf="parent"
80	app:layout_constraintEnd_toEndOf="parent"
	app:layout_constraintHorizontal_bias="1.0"
	app:layout_constraintStart_toStartOf="parent" />
	</androidx.constraintlayout.widget.ConstraintLayout>

Tabel 3. Source Code Jawaban Soal 1

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1 Compose



Gambar 2. Screenshot Hasil Jawaban Soal 1 XML

C. Pembahasan

Compose

MainActivity.kt:

Pada line 44, dideklarasikan theme default yang dibuat oleh compose secara otomatis, didalamnya akan dipanggil fungsi `dicerollerapp` yang membungkus keseluruhan ui

Pada line 57, dideklarasikan `preview` sekaligus fungsi `dicerollerapp` yang membungkus keseluruhan ui

Pada line 65, dideklarasikan fungsi `DiceWithButtonAndImage` untuk ui image dice dan buttonnya

Pada line 68 - 86, terdapat logic untuk mengatur source dari image painter sesuai dengan state nya, disini juga dideklarasikan variabel untuk menyimpan state `dice1` dan `dice2`

Pada line 67, diimport `Column layout` milik compose untuk mengatur image dice dan button

Pada line 92, diimport `Row layout` milik compose untuk mengatur image dice 1 dan dice 2 sehingga bisa bersebelahan

Pada line 103, terdapat logic untuk menampilkan `Text` milik compose jika state dari dice 1 dan dice 2 berubah

Pada line 105, terdapat logic untuk mengubah `stringresource Text` milik compose jika value dari dice 1 dan dice 2 sama

XML

MainActivity.kt:

Pada line 7, diimport `AppCompatActivity` untuk base class activity yang mendukung versi terdahulu Android

Pada line 17, diimport fungsi `setContentView` untuk menjadikan `activity_main` sebagai layout tujuan

Pada line 18 - 23, dideklarasikan variable yang merujuk pada setiap komponen ui yang ada pada layout

Pada line 25, terdapat fungsi yang memuat logic apabila button di klik

Pada line 26 - 45, dideklarasikan variable untuk menyimpan nilai dadu dan logic untuk mengubah tampilan image dadu sesuai dengan nilai dari dadu

Pada line 47 - 58, dipanggil scope fungsi pada `bottomText` untuk memberikan logic dimana text tidak akan muncul jika button tidak diklik, dan text akan berubah jika dadu 1 dan 2 memiliki nilai sama atau tidak

activity_main.xml:

Pada line 15 - 31, dideklarasikan komponen `imageview` untuk dadu 1, dengan `constraint` `endto startof` ke dadu 2 dan juga `constraintHorizontal_chainStyle` `packed`

Pada line 33 - 48, dideklarasikan komponen imageview untuk dadu 2, dengan constraint Bottom_toBottomOf ke dice1, Bottom_toTopOf ke rollButton, Horizontal_chainStyle packed

Pada line 50 - 64, dideklarasikan komponen button untuk rollButton, dengan constraint Top_toBottomOf ke dice1

Pada line 66 - 80, dideklarasikan komponen textview untuk messageText, dengan visibility invisible, dan constraint Bottom_toBottomOf ke parent, End_toEndOf ke parent, Start_toStartOf ke parent

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/biahlil/Pemrograman-Mobile.git>