

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Muhammad Erza Raihan NIM. 2310817210027

**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 : Muhammad Erza Raihan
NIM : 2310817210027

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.....	2
DAFTAR ISI	3
DAFTAR GAMBAR	4
DAFTAR TABEL.....	5
SOAL 1	6
A. Source Code.....	9
B. Output Program.....	14
C. Pembahasan.....	17
D. Tautan Git	20

DAFTAR GAMBAR

Gambar 1. Tampilan Awal Aplikasi	6
Gambar 2. Tampilan Dadu Setelah Di-Roll	7
Gambar 3. Tampilan Roll Dadu Double	8
Gambar 4. Screenshot Hasil Jawaban Soal 1 Compose	15
Gambar 5. Screenshot Hasil Jawaban Soal 1 Compose	15
Gambar 6. Screenshot Hasil Jawaban Soal 1 XML	16
Gambar 7. Screenshot Hasil Jawaban Soal 1 XML	17

DAFTAR TABEL

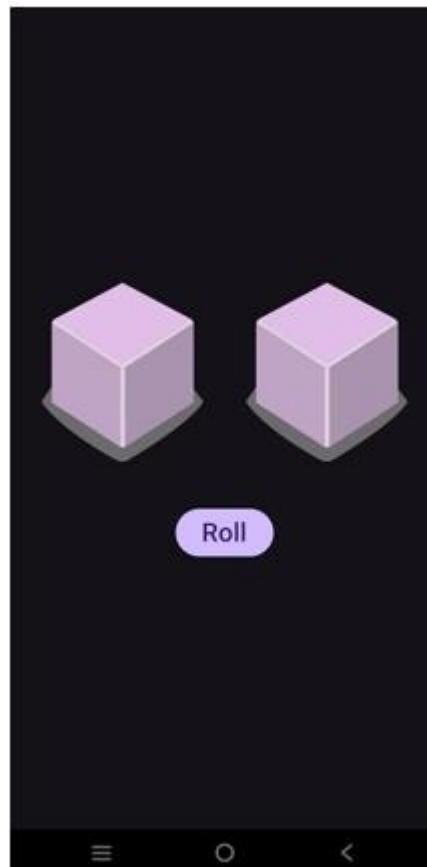
Tabel 1. Source Code Mainactivity XML Soal 1	10
Tabel 2. Source Code MainActivity Compose Soal 1	13
Tabel 3. Source Code activity_main XML Soal 1	14

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”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

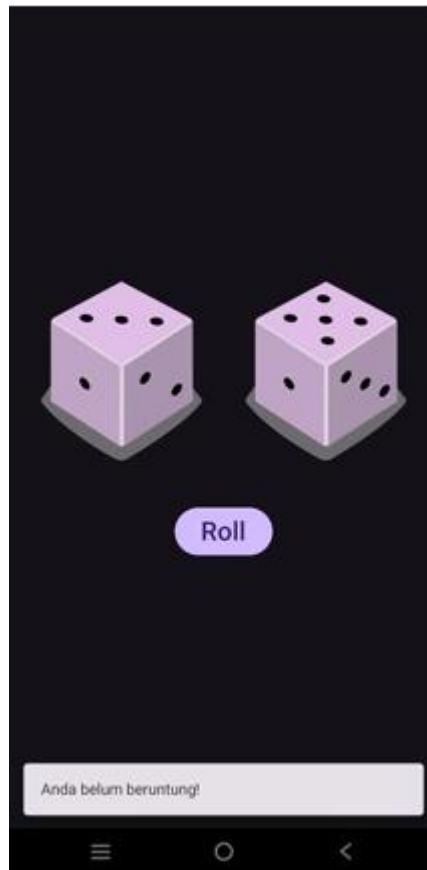
1. Tampilan awal aplikasi setelah dijalankan akan menampilkan 2 buah dadu kosong seperti dapat dilihat pada Gambar 1.



Gambar 1. Tampilan Awal Aplikasi

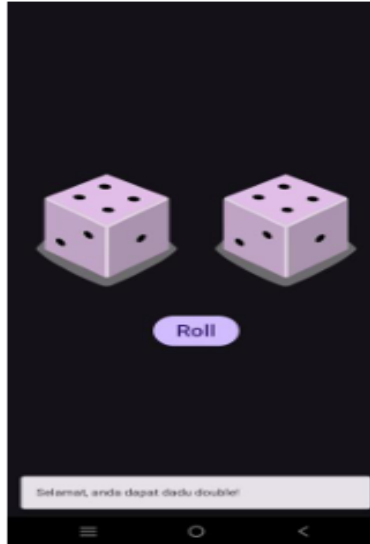
2. Setelah user menekan tombol “Roll” maka masing-masing dadu akan memperlihatkan sisi dadu masing-masing dengan angka antara 1 s/d 6. Apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2 maka aplikasi

akan menampilkan pesan “Anda belum beruntung!” seperti 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 yang 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%20%20

A. Source Code

MainActivity.kt XML

```
1 package com.example.modul1xml
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 class MainActivity : AppCompatActivity() {
9
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         val diceImage01: ImageView =
15         findViewById(R.id.imageView)
16         diceImage01.setImageResource(R.drawable.dice_1)
17
18         val diceImage02: ImageView =
19         findViewById(R.id.imageView2)
20         diceImage02.setImageResource(R.drawable.dice_1)
21
22         val rollButton: Button = findViewById(R.id.button)
23
24         rollButton.setOnClickListener { rollDice() }
25     }
26
27     private fun rollDice() {
28         val dice1 = Dice(6)
29         val diceRoll1 = dice1.roll()
30
31         val diceImage1: ImageView =
32         findViewById(R.id.imageView)
33         when (diceRoll1) {
34             1 ->
35             diceImage1.setImageResource(R.drawable.dice_1)
36             2 ->
37             diceImage1.setImageResource(R.drawable.dice_2)
38             3 ->
39             diceImage1.setImageResource(R.drawable.dice_3)
40             4 ->
41             diceImage1.setImageResource(R.drawable.dice_4)
42             5 ->
43             diceImage1.setImageResource(R.drawable.dice_5)
44         }
45     }
46 }
```

48	6	->
49	diceImage1.setImageResource(R.drawable.dice_6)	
50	}	
51		
52	val dice2 = Dice(6)	
53	val diceRoll2 = dice2.roll()	
54		
55		
56	val diceImage2: ImageView	=
57	findViewById(R.id.imageView2)	
58	when (diceRoll2) {	
59	1	->
60	diceImage2.setImageResource(R.drawable.dice_1)	
61	2	->
62	diceImage2.setImageResource(R.drawable.dice_2)	
63	3	->
64	diceImage2.setImageResource(R.drawable.dice_3)	
65	4	->
66	diceImage2.setImageResource(R.drawable.dice_4)	
67	5	->
68	diceImage2.setImageResource(R.drawable.dice_5)	
69	6	->
70	diceImage2.setImageResource(R.drawable.dice_6)	
71	}	
72		
73		
74	if (diceRoll1 == diceRoll2){	
75	val toast = Toast.makeText(this, "Selamat anda	
76	dapat dadu double!", Toast.LENGTH_SHORT)	
77	toast.show()	
78	} else {	
79	val toast = Toast.makeText(this, "Anda belum	
80	beruntung!", Toast.LENGTH_SHORT)	
81	toast.show() }	
82		
83		
84	}	
85	}	
86		
87	class Dice(private val numSides: Int) {	
88		
89	fun roll(): Int {	
90	return (1..numSides).random()	
91	}	
92	}	
93		
94		
95		

Tabel 1. Source Code Mainactivity XML Soal 1

MainActivity.kt Compose

```
1 package com.example.praktikum1
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.layout.Column
8 import androidx.compose.foundation.layout.fillMaxSize
9 import androidx.compose.foundation.layout.wrapContentSize
10 import androidx.compose.foundation.layout.height
11 import androidx.compose.foundation.layout.Spacer
12 import androidx.compose.foundation.layout.wrapContentSize
13 import androidx.compose.material3.Text
14 import androidx.compose.runtime.Composable
15 import androidx.compose.ui.Alignment
16 import androidx.compose.ui.Modifier
17 import androidx.compose.ui.res.painterResource
18 import androidx.compose.ui.tooling.preview.Preview
19 import com.example.praktikum1.ui.theme.PRAKTIKUM1Theme
20 import androidx.compose.foundation.layout.height
21 import androidx.compose.foundation.layout.Spacer
22 import androidx.compose.ui.unit.dp
23 import androidx.compose.material3.Button
24 import androidx.compose.ui.res.stringResource
25 import androidx.compose.runtime.getValue
26 import androidx.compose.runtime.remember
27 import androidx.compose.runtime.setValue
28 import androidx.compose.runtime.mutableStateOf
29
30 class MainActivity : ComponentActivity() {
31     override fun onCreate(savedInstanceState: Bundle?) {
32         super.onCreate(savedInstanceState)
33         setContent {
34             PRAKTIKUM1Theme {
35                 DiceRollerApp()
36             }
37         }
38     }
39 }
40
41 @Preview()
42 @Composable
43 fun DiceRollerApp() {
44     DiceWithButtonAndImage()
45 }
46
47
48
49
50 @Composable
```

```

51 fun DiceWithButtonAndImage(modifier: Modifier = Modifier
52     .fillMaxSize()
53     .wrapContentSize(Alignment.Center)) {
54
55     var result1 by remember { mutableStateOf(1) }
56     var result2 by remember { mutableStateOf(1) }
57     var message by remember { mutableStateOf("") }
58
59
60     val imageResource1 = when (result1) {
61         1 -> R.drawable.dice_1
62         2 -> R.drawable.dice_2
63         3 -> R.drawable.dice_3
64         4 -> R.drawable.dice_4
65         5 -> R.drawable.dice_5
66         else -> R.drawable.dice_6
67     }
68
69     val imageResource2 = when (result2) {
70         1 -> R.drawable.dice_1
71         2 -> R.drawable.dice_2
72         3 -> R.drawable.dice_3
73         4 -> R.drawable.dice_4
74         5 -> R.drawable.dice_5
75         else -> R.drawable.dice_6
76     }
77
78
79     Column (
80         modifier = modifier,
81         horizontalAlignment
82     Alignment.CenterHorizontally
83     ) {
84         androidx.compose.foundation.layout.Row {
85             Image(
86                 painter
87             painterResource(imageResource1),
88                 contentDescription = result1.toString()
89             )
90
91             Spacer(modifier = Modifier.height(8.dp))
92
93             Image(
94                 painter
95             painterResource(imageResource2),
96                 contentDescription = result2.toString()
97             )
98         }
99     }
100
101
102

```

```

103
104         Spacer(modifier = Modifier.height(16.dp))
105
106         val doubleMessage =
107 stringResource(R.string.double_message)
108         val badMessage =
109 stringResource(R.string.bad_message)
110
111         Button(onClick = {
112             result1 = (1..6).random()
113             result2 = (1..6).random()
114
115             message = if (result1 == result2) {
116                 doubleMessage
117             } else {
118                 badMessage
119             }
120         }) {
121             Text(stringResource(R.string.roll))
122         }
123
124         Spacer(modifier = Modifier.height(16.dp))
125
126         Text(text = message)
127
128
129
130
131     }}
132
133

```

Tabel 2. Source Code MainActivity Compose 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  "
5      xmlns:app="http://schemas.android.com/apk/res-auto"
6      xmlns:tools="http://schemas.android.com/tools"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      tools:context=".MainActivity">
10
11
12  <Button

```

```

13         android:id="@+id/button"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:layout_marginTop="52dp"
17         android:text="ROLL"
18         app:layout_constraintEnd_toEndOf="parent"
19         app:layout_constraintHorizontal_bias="0.498"
20         app:layout_constraintStart_toStartOf="parent"
21
22     app:layout_constraintTop_toBottomOf="@+id/imageView" />
23
24     <ImageView
25         android:id="@+id/imageView"
26         android:layout_width="150dp"
27         android:layout_height="150dp"
28         app:layout_constraintBottom_toBottomOf="parent"
29         app:layout_constraintEnd_toEndOf="parent"
30         app:layout_constraintHorizontal_bias="0.201"
31         app:layout_constraintStart_toStartOf="parent"
32         app:layout_constraintTop_toTopOf="parent"
33         app:layout_constraintVertical_bias="0.449"
34     />
35
36     <ImageView
37         android:id="@+id/imageView2"
38         android:layout_width="150dp"
39         android:layout_height="150dp"
40         app:layout_constraintBottom_toBottomOf="parent"
41         app:layout_constraintEnd_toEndOf="parent"
42         app:layout_constraintHorizontal_bias="0.837"
43         app:layout_constraintStart_toStartOf="parent"
44         app:layout_constraintTop_toTopOf="parent"
45         app:layout_constraintVertical_bias="0.449"
46     />
47
48 </androidx.constraintlayout.widget.ConstraintLayout>
49
50
51
52

```

Tabel 3. Source Code activity_main XML Soal 1

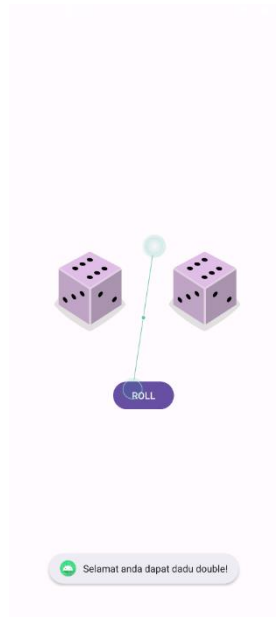
B. Output Program



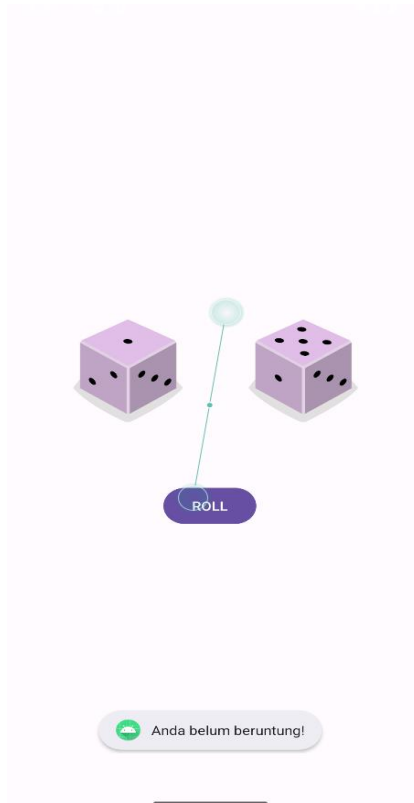
Gambar 4. Screenshot Hasil Jawaban Soal 1 Compose



Gambar 5. Screenshot Hasil Jawaban Soal 1 Compose



Gambar 6. Screenshot Hasil Jawaban Soal 1 XML



Gambar 7. Screenshot Hasil Jawaban Soal 1 XML

C. Pembahasan

MainActivity.kt XML:

Kode ini merupakan implementasi aplikasi Android sederhana yang menampilkan dua gambar dadu dan sebuah tombol untuk "mengacak" atau **roll** kedua dadu tersebut. Proyek ini menggunakan Kotlin dengan antarmuka XML pada `activity_main.xml`. Tujuannya adalah untuk memberikan visualisasi pelemparan dua buah dadu secara acak serta menampilkan notifikasi (Toast) ketika kedua dadu menunjukkan angka yang sama (double).

Pada bagian `onCreate`, komponen UI dari layout diambil dan diatur. Dua `ImageView` (`imageView` dan `imageView2`) digunakan untuk menampilkan gambar sisi dadu, dan satu `Button` (`button`) digunakan untuk memicu aksi roll dadu. Saat aplikasi dibuka, kedua gambar

dadu diinisialisasi terlebih dahulu dengan gambar dadu angka satu (dice_1). Kemudian, saat tombol ditekan, fungsi rollDice() akan dipanggil.

Fungsi rollDice() bertanggung jawab untuk mengacak angka pada masing-masing dadu menggunakan kelas Dice, yang menerima parameter jumlah sisi dadu (dalam hal ini 6). Setelah angka dadu diacak dengan metode .roll(), hasilnya digunakan dalam blok when untuk mengganti gambar pada masing-masing ImageView, menyesuaikan dengan angka yang didapat (misalnya jika hasilnya 3, maka akan ditampilkan gambar dice_3).

Terakhir, jika kedua dadu menunjukkan angka yang sama, maka aplikasi akan menampilkan pesan Toast yang menyatakan bahwa pengguna mendapatkan dadu double. Jika tidak sama, maka Toast akan memberitahukan bahwa pengguna belum beruntung. Ini memberikan elemen permainan sederhana dalam aplikasi.

activity_main.xml:

File XML adalah layout antarmuka pengguna (UI layout) untuk aplikasi Android yang menggunakan ConstraintLayout sebagai root element. ConstraintLayout memungkinkan kita menata elemen-elemen UI secara fleksibel dengan menetapkan posisi relatif antar elemen atau terhadap parent-nya, sehingga lebih efisien dan adaptif terhadap berbagai ukuran layar.

Dalam layout ini, terdapat **tiga komponen utama**: dua ImageView dan satu Button. Kedua ImageView digunakan untuk menampilkan gambar dadu, masing-masing memiliki ID imageView dan imageView2. Masing-masing dadu diberikan ukuran tetap sebesar 150dp baik lebar maupun tingginya. Posisi keduanya diatur agar sejajar secara vertikal di tengah layar, namun berbeda secara horizontal—yang satu lebih ke kiri (horizontal_bias="0.201"), yang satu lagi ke kanan (horizontal_bias="0.837"), sehingga terlihat seperti dua dadu berdampingan.

Di bawah kedua ImageView ini terdapat sebuah Button dengan ID button dan teks "**ROLL**". Tombol ini ditempatkan di bawah imageView (dadu kiri) dengan

`layout_constraintTop_toBottomOf="@+id/imageView"` dan diberi margin atas sebesar 52dp agar tidak terlalu menempel. Tombol juga diposisikan di tengah secara horizontal dengan `layout_constraintStart_toStartOf="parent"` dan `layout_constraintEnd_toEndOf="parent"` serta `horizontal_bias="0.498"` (hampir di tengah).

Penggunaan `tools:context=".MainActivity"` pada root `ConstraintLayout` adalah untuk memberi tahu Android Studio bahwa layout ini akan digunakan oleh `MainActivity`, sehingga saat desain dilihat di editor, bisa disimulasikan dengan konteks aktivitas yang tepat.

MainActivity.kt Compose:

Bagian utama dari aplikasi dimulai di dalam kelas `MainActivity`, yang mewarisi `ComponentActivity`. Di dalam metode `onCreate`, fungsi `setContent` digunakan untuk menampilkan UI berbasis Compose dengan tema `PRAKTIKUM1Theme`. Komponen utama aplikasi dipanggil melalui fungsi `DiceRollerApp`, yang akan memanggil `DiceWithButtonAndImage()`—sebuah composable function yang berisi keseluruhan tampilan dan logika utama aplikasi.

Fungsi `DiceWithButtonAndImage` menggunakan layout `Column` untuk menata elemen-elemen UI secara vertikal dan berada di tengah layar. Di dalamnya terdapat dua `Image` untuk menampilkan gambar hasil lemparan dadu, sebuah `Button` untuk melakukan aksi lempar, dan satu `Text` untuk menampilkan pesan hasil lemparan. Gambar dadu dipilih menggunakan struktur `when` berdasarkan nilai `result1` dan `result2`, yang diinisialisasi dengan nilai 1 dan dapat berubah setiap kali tombol ditekan, karena dibungkus dalam `remember` dan `mutableStateOf`. Ini memastikan bahwa UI akan merespons secara otomatis saat nilainya berubah.

Ketika pengguna menekan tombol `ROLL`, nilai `result1` dan `result2` akan diperbarui secara acak antara 1 hingga 6. Jika kedua nilai tersebut sama (`double`), maka pesan yang ditampilkan berasal dari `stringResource(R.string.double_message)`; jika tidak sama, maka pesan dari

bad_message akan ditampilkan. Penggunaan `stringResource()` membuat aplikasi ini lebih fleksibel dan mendukung internasionalisasi (misalnya, mudah untuk menerjemahkan ke bahasa lain).

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/MuhammadErzaRaihan>