

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Damarjati Suryo Laksono

2310817210014

**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 : Damarjati Suryo Laksono
NIM : 2310817210014

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.....	9
B. Output Program	14
C. Pembahasan	15
D. Tautan Git.....	17

DAFTAR GAMBAR

Gambar 1 Screenshot Vertikal Aplikasi	14
Gambar 2 Screenshoot Horizontal Aplikasi	14

DAFTAR TABEL

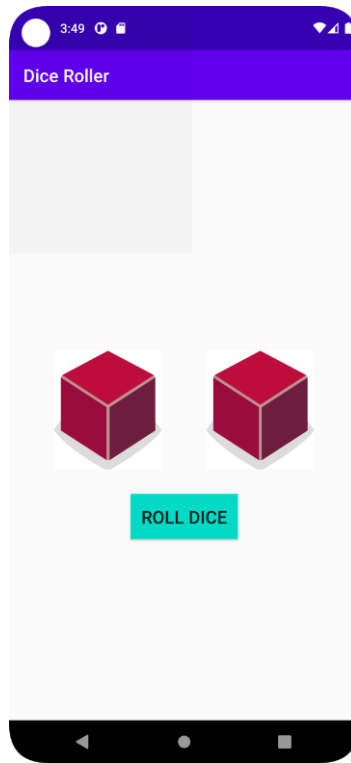
Tabel 1 Source Code MainActivity.kt.....	9
Tabel 2 Source Code DiceViewModel.kt.....	10
Tabel 3 Source Code activity_main.xml	11

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.



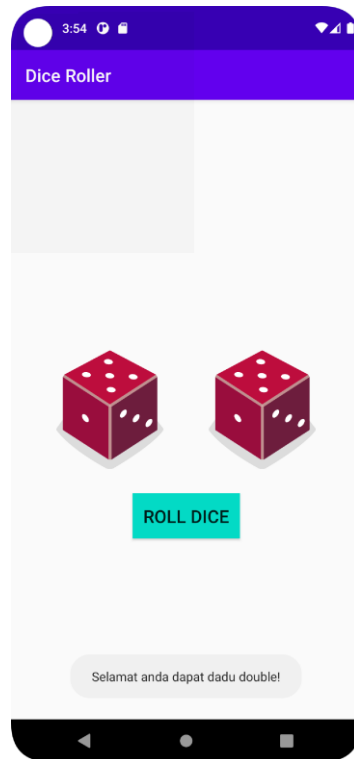
Gambar 1 Tampilan Awal Aplikasi

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



Gambar 3 Tampilan Roll Dadu Double

A. Source Code

1. MainActivity.kt

Tabel 1 Source Code MainActivity.kt

1	package com.example.diceroller
2	
3	import android.os.Bundle
4	import android.widget.Toast
5	import androidx.activity.viewModels
6	import androidx.appcompat.app.AppCompatActivity
7	import androidx.lifecycle.Observer
8	import com.example.diceroller.databinding.ActivityMainBinding
9	
10	class MainActivity : AppCompatActivity() {
11	
12	private lateinit var binding :ActivityMainBinding
13	private val viewModel : DiceViewModel by viewModels()
14	
15	override fun onCreate(savedInstanceState: Bundle?) {
16	super.onCreate(savedInstanceState)
17	binding = ActivityMainBinding.inflate(layoutInflater)
18	setContentView(binding.root)
19	
20	binding.button.setOnClickListener {
21	viewModel.rollDice()
22	}
23	viewModel.dice1.observe(this,Observer { value ->
24	binding.imageView.setImageResource(getDiceImage(value))
25	})
26	viewModel.dice2.observe(this,Observer { value ->
27	binding.imageView2.setImageResource(getDiceImage(value))
28	})

29	viewModel.isDouble.observe(this, Observer { isDouble ->
30	if (isDouble) {
31	Toast.makeText(this, "Selamat anda dapat dadu
	double", Toast.LENGTH_SHORT).show()
32	}else{
33	Toast.makeText(this, "Anda belum beruntung",
	Toast.LENGTH_SHORT).show()
34	}
35	})
36	}
37	
38	private fun getDiceImage(value: Int):Int{
39	return when(value) {
40	1 ->R.drawable.dice_1
41	2 ->R.drawable.dice_2
42	3 ->R.drawable.dice_3
43	4 ->R.drawable.dice_4
44	5 ->R.drawable.dice_5
45	else -> R.drawable.dice_6
46	}
47	}
48	}

2. DiceViewModel.kt

Tabel 2 Source Code DiceViewModel.kt

1	package com.example.diceroller
2	
3	import androidx.lifecycle.LiveData
4	import androidx.lifecycle.MutableLiveData
5	import androidx.lifecycle.ViewModel
6	import kotlin.random.Random
7	

8	class DiceViewModel : ViewModel() {
9	private val _dice1 = MutableLiveData(1)
10	val dice1 : LiveData<Int>
11	get() = _dice1
12	
13	private val _dice2 = MutableLiveData(1)
14	val dice2 : LiveData<Int>
15	get() = _dice2
16	
17	private val _isDouble = MutableLiveData(false)
18	val isDouble : LiveData<Boolean>
19	get() = _isDouble
20	
21	fun rollDice() {
22	val roll1 = Random.nextInt(1,7)
23	val roll2 = Random.nextInt(1,7)
24	_dice1.value = roll1
25	_dice2.value = roll2
26	_isDouble.value = roll2 == roll1
27	}
28	}

3. activity_main.xml

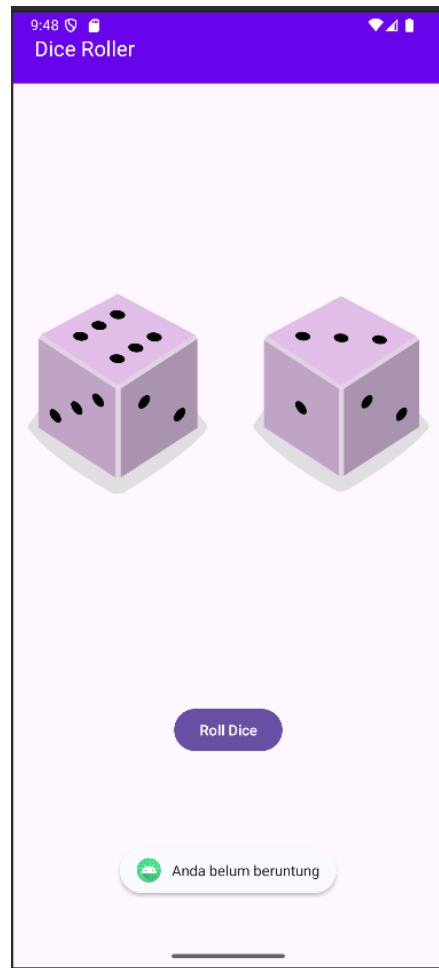
Tabel 3 Source Code activity_main.xml

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	tools:context=".MainActivity">

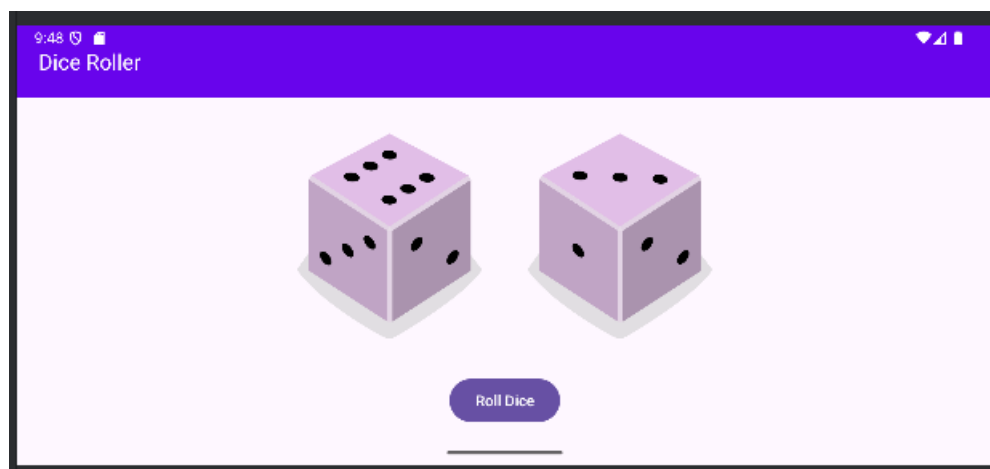
```
10
11     <TextView
12         android:id="@+id/headerTitle"
13         android:layout_width="0dp"
14         android:layout_height="wrap_content"
15         android:background="#6804ec"
16         android:padding="20dp"
17         android:text="@string/app_name"
18         android:textColor="#FFFFFF"
19         android:textSize="20sp"
20         app:layout_constraintEnd_toEndOf="parent"
21         app:layout_constraintHorizontal_bias="0.0"
22         app:layout_constraintStart_toStartOf="parent"
23         app:layout_constraintTop_toTopOf="parent" />
24
25     <LinearLayout
26         android:id="@+id/diceContainer"
27         android:layout_width="wrap_content"
28         android:layout_height="wrap_content"
29         android:orientation="horizontal"
30         android:gravity="center"
31
32     app:layout_constraintTop_toBottomOf="@+id/headerTitle"
33         app:layout_constraintBottom_toTopOf="@+id/button"
34         app:layout_constraintStart_toStartOf="parent"
35         app:layout_constraintEnd_toEndOf="parent"
36         app:layout_constraintVertical_bias="0.5">
37
38     <ImageView
39         android:id="@+id/imageView"
40         android:layout_width="wrap_content"
41         android:layout_height="wrap_content"
42         android:src="@drawable/dice_0" />
43
```

```
44         <ImageView
45             android:id="@+id/imageView2"
46             android:layout_width="wrap_content"
47             android:layout_height="wrap_content"
48             android:src="@drawable/dice_0"
49             android:layout_marginStart="16dp" />
50     </LinearLayout>
51
52     <Button
53         android:id="@+id/button"
54         android:layout_width="wrap_content"
55         android:layout_height="wrap_content"
56         android:text="@string/button_1"
57
58         app:layout_constraintTop_toBottomOf="@+id/diceContainer"
59         app:layout_constraintBottom_toBottomOf="parent"
60         app:layout_constraintStart_toStartOf="parent"
61         app:layout_constraintEnd_toEndOf="parent"
62         app:layout_constraintVertical_bias="0.0"
63         android:layout_marginBottom="32dp" />
64
65 </androidx.constraintlayout.widget.ConstraintLayout>
```

B. Output Program



Gambar 1 Screenshot Vertikal Aplikasi



Gambar 2 Screenshoot Horizontal Aplikasi

C. Pembahasan

1. MainActivity.kt:

Pada line [1] – [8], terdapat package dan library – library yang akan di gunakan

Pada line [10], terdapat main activity yang merupakan wadah dari kode yang akan menjadi wadah untuk kode lainnya.

Pada line [12] – [13] terdapat inisiasi awal dari view binding dan view model

Pada line [15] – [18] terdapat insiasi dari view binding

Pada line [20] – [22] terdapat syntax `binding.button.setOnClickListener`, pada syntax ini, button di set ke listener atau Ketika di klik akan melakukan aksi yang di tentukan, Dimana aksi disini merupakan `rollDice()`, namun pada syntax button tadi terdapat `binding.button` yang secara otomatis mengarah ke button yang terdapat pada `main_activity.xml` tanpa harus menulis sintaks yang panjang. Kemudian pada `rollDice()` juga terdapat `viewModel.rollDice()`, sintaks ini mengarah ke fungsi `rollDice()` pada file `DiceViewModel.kt`, fungsi ini bertujuan untuk mengacak angka dadu dari 1-6.

Pada line [23] – [28] terdapat syntax dengan kerangka seperti `viewModel.dice1.observe(this, Observer{value-> binding.imageView.setImageResource(getDiceImage(value))`, pertama `viewModel.dice1` mengarah pada variable `dice1` yang terdapat di `DiceViewModel.kt`, kemudian terdapat syntax `.observe(this, Observer{value->` syntax ini bertujuan untuk menyatakan pada program bahwa program harus mengamati value pada kasus ini `dice1`, jika value `dice1` pada `DiceViewModel.kt` berubah maka akan terjadi perubahan pada image, selanjutnya terdapat syntax `binding.imageView.setImageResource(getDiceImage(value))`, Dimana syntax ini bertujuan untuk merubah image sesuai dengan jumlah mata dadu yang terdapat pada `dice1`, pada syntax ini terdapat fungsi `getDiceImage`, Dimana fungsi ini berguna untuk mengambil gambar sesuai dengan mata dadu,

setelah itu melakukan `setImageResource` yang bertujuan untuk merubah gambar dari `imageView`.

Pada line [29] – [36] terdapat syntax `getDiceImage()` yang berguna untuk mengecek apakah dadu double atau tidak, pertama program akan mengambil data `isDouble` dari `DiceViewModel.kt`, kemudian program akan melakukan observasi terdapat data `isDouble` tersebut, apabila ternyata data adalah `true`, maka program akan memberi selamat, dan jika ternyata adalah `false` maka program akan memberi celotehan, cara kerja pengiriman pesan tersebut adalah dengan melakukan print notifikasi menggunakan `Toast`, dan pada akhir terdapat `.show()` untuk menunjukkan pesan.

Pada line [38] – [48] terdapat fungsi `getDiceImage()` untuk menentukan gambar mana yang di gunakan berdasar hasil roll dadu.

2. DiceViewModel.kt

Pada line [1] – [6], terdapat package dan library – library yang akan di gunakan

Pada line [8], terdapat `class DiceViewModel : ViewModel()` yang merupakan wadah dari kode yang akan menjadi wadah untuk kode lainnya, dan juga menentukan bahwa class ini merupakan `Viewmodel()` dari proyek ini.

Pada line [9] – [19], terdapat syntax `private val _dice1 = MutableLiveData(1)`, pada syntax ini terdapat `private val _dice1` yang menyebutkan bahwa value `_dice1` hanya bisa di baca di class ini, dan penggunaan `_` pada `_dice1` penamaan ini menandakan bahwa variable ini adalah variable local saja, setelah itu terdapat `MutabilityLiveData(1)`, yang berfungsi agar data bisa di rubah namun hanya pada lingkup local, dan (1) merupakan place holder data itu sendiri, kemudian terdapat `val dice1 : LiveData<Int> get() = _dice1`, Dimana `val dice1 : LiveData<Int>` berguna untuk melakukan deklarasi bahwa `dice1` bersifat unmutable dan hanya bisa di observe atau di lihat aja, kemudian terdapat `get() = _dice1`, syntax ini bertujuan untuk mengisi data `dice1`, menjadi data dari `_dice1` yang akan berubah di fungsi dibawah, dan untuk variable lainnya, cara kerjanya juga sama.

Pada line [21] – [28], terdapat fungsi `fun rollDice()`, fungsi ini bertujuan untuk melakukan rolling pada dadu menggunakan syntax `val roll1 = Random.nextInt(1, 7)`, dan juga menentukan apakah dadu double atau tidak dan di simpan ke dalam Boolean menggunakan `_isDouble.value = roll2 == roll1`.

3. activity_main.xml

Pada line [1], terdapat deklarasi `<?xml version="1.0" encoding="utf-8"?>`, yang merupakan standar deklarasi file XML agar dikenali dengan benar oleh sistem Android.

Pada line [2] – [8], terdapat tag utama yaitu `<androidx.constraintlayout.widget.ConstraintLayout>`, ini adalah root layout yang digunakan untuk menyusun elemen-elemen UI secara fleksibel dengan constraint dan juga layout ini punya `id`, `layout_width`, `layout_height`, dan `tools:context` untuk menunjuk ke `MainActivity`.

Pada line [10] – [20], terdapat komponen `TextView` dengan `id` `headerTitle`, yang berfungsi sebagai header aplikasi.

Pada line [22] – [38], terdapat `LinearLayout` dengan `id` `diceContainer` yang berfungsi sebagai wadah untuk dua gambar dadu.

Pada line [24] – [33], ada dua `ImageView` yang merepresentasikan dua buah dadu

Pada line [40] – [48], terdapat sebuah `Button` dengan `id` `button` yang digunakan untuk me-roll dadu.

D. Tautan Git

<https://github.com/au290/Kuliah/tree/main/Pemro%20Mobile/Modul%201>