

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Noviana Nur Aisyah

NIM. 2310817120005

**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 : Noviana Nur Aisyah
NIM : 2310817120005

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	11
C. Pembahasan	12
Tautan Git	17

DAFTAR GAMBAR

Tabel 1. Source Code Jawaban Soal 1.....	9
--	---

DAFTAR TABEL

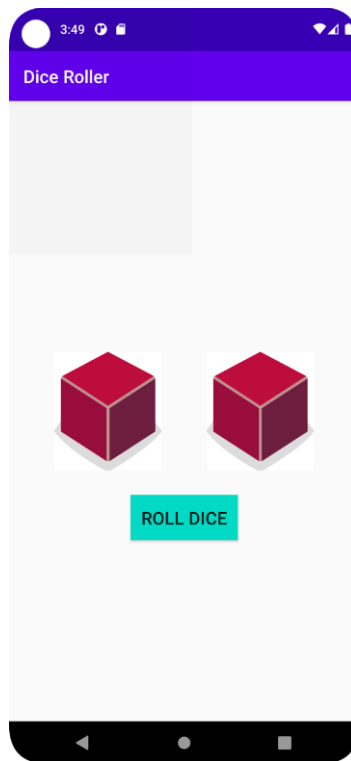
Tabel 1. Source Code Jawaban Soal 1.....	9
Tabel 2. Source Code Jawaban Soal 1.....	11

SOAL 1

Soal Praktikum:

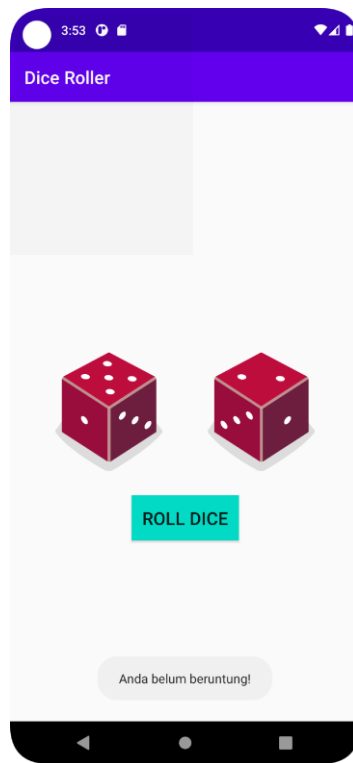
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



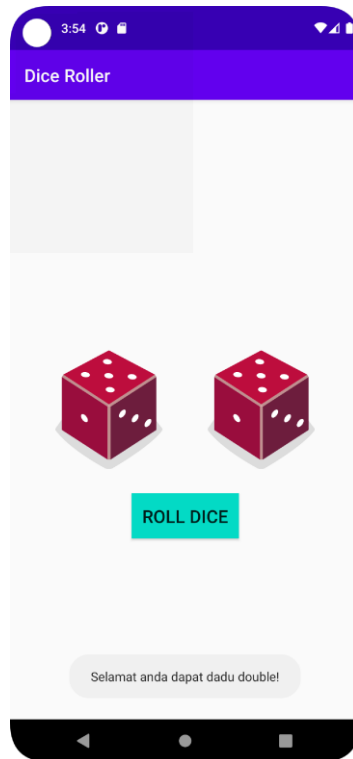
Gambar 1 Tampilan Awal Aplikasi

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.



Gambar 3 Tampilan Roll Dadu Double

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/file/d/14V3qXGdFnuoYN4AGd_9SgFh8kw8X9ySm/view

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		
9	class	MainActivity : AppCompatActivity() {
10	private	var firstDice: Int? = null
11	override fun	onCreate(savedInstanceState: Bundle?) {
12		super.onCreate(savedInstanceState)
13		setContentView(R.layout.activity_main)
14		
15		val rollButton: Button = findViewById(R.id.button)
16		rollButton.setOnClickListener {
17		rollDice()
18		rollDice2()
19		}
20		}

Tabel 1. Source Code Jawaban Soal 1

2. MainActivityViewModel.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		
9	class	MainActivity : AppCompatActivity() {
10	private	var firstDice: Int? = null
11	override fun	onCreate(savedInstanceState: Bundle?) {
12		super.onCreate(savedInstanceState)
13		setContentView(R.layout.activity_main)
14		

15	val rollButton: Button = findViewById(R.id.button)
16	rollButton.setOnClickListener {
17	rollDice()
18	rollDice2()
19	}
20	}

3. Activity_main.xml

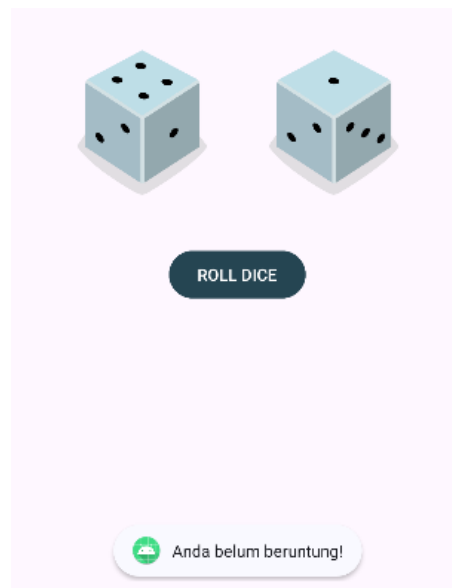
1	package com.example.diceroller
2	
3	import android.os.Bundle
4	import android.widget.Button
5	import android.widget.ImageView
6	import android.widget.Toast
7	import androidx.appcompat.app.AppCompatActivity
8	
9	class MainActivity : AppCompatActivity() {
10	private var firstDice: Int? = null
11	override fun onCreate(savedInstanceState: Bundle?) {
12	super.onCreate(savedInstanceState)
13	setContentView(R.layout.activity_main)
14	
15	val rollButton: Button = findViewById(R.id.button)
16	rollButton.setOnClickListener {
17	rollDice()
18	rollDice2()
19	}
20	}

4. tool_bar.xml

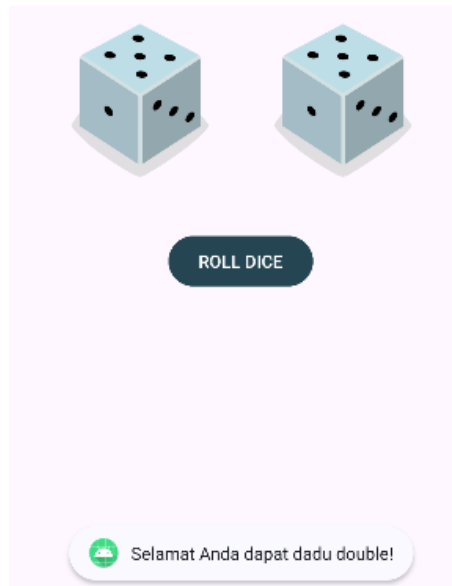
1	<?xml	version="1.0"	encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout		
3	xmlns:android="http://schemas.android.com/apk/res/android"		
4	xmlns:app="http://schemas.android.com/apk/res-auto"		
5	xmlns:tools="http://schemas.android.com/tools"		
6	android:id="@+id/main"		
7	android:layout_width="match_parent"		
8	android:layout_height="match_parent"		
9	tools:context=".MainActivity">		
10			
11	</androidx.constraintlayout.widget.ConstraintLayout>		

Tabel 2. Source Code Jawaban Soal 1

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1



Gambar 2. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt

File ini merupakan logika utama dari aplikasi untuk menampilkan UI dan mengatur interaksi pengguna. Terdapat beberapa bagian penting, yaitu:

a) View Binding

```
binding = ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)
```

Digunakan untuk menghubungkan file `activity_main.xml` (`ActivityMainBinding`).

b) Set gambar Dadu (setiap membuka aplikasi)

```
binding.ivDice.setImageResource(R.drawable.dice_0)
binding.ivDice2.setImageResource(R.drawable.dice_0)
```

Digunakan untuk menampilkan gambar dadu kosong setiap kali aplikasi dibuka.

c) Event Button

```
binding.btnRoll.setOnClickListener {
    viewModel.rollDice()
}
```

Ketika button diklik, maka akan meminta ViewModel untuk menghasilkan dua angka acak dan update Live Data `dice1` dan `dice2`.

d) LiveData

```
viewModel.dice1.observe(this) { hasil ->
    binding.ivDice.setImageResource(hasil)
}
viewModel.dice2.observe(this) { hasil ->
    binding.ivDice2.setImageResource(hasil)
    checkIfDouble()
}
```

Jika terdapat perubahan nilai `dice1` atau `dice2`, gambar dadu akan ter-update sesuai hasilnya, kemudian hasil ini akan dicek apakah sama (`checkIfDouble`).

`viewModel.dice1` adalah LiveData dari ViewModel, bersifat *read-only*.

`.observe(this)` adalah cara Android menyambungkan LiveData ke komponen UI seperti Activity, `this` di sini berarti konteks Activity saat ini.

`{ hasil -> ... }` merupakan fungsi yang dipanggil setiap nilai `dice1` berubah dan `hasil` adalah nilai baru dari `dice1`.

`Binding.ivDice.setImageResource(hasil)` adalah jika `dice1` berubah (misalnya menjadi `R.drawable.dice_4`), maka gambar di ImageView bernama `ivDice` akan langsung berubah ke gambar dadu 4.

e) Cek Double Dadu

```
private fun checkIfDouble() {
    val d1 = viewModel.dice1.value
    val d2 = viewModel.dice2.value

    if (d1 == d2) {
        Toast.makeText(this, "Selamat Anda dapat dadu
double!", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(this, "Anda belum beruntung!",
Toast.LENGTH_SHORT).show()
    }
}
```

Digunakan untuk menampilkan toast jika dua dadu sama (misalnya 6 dengan 6), yang artinya “*double*”.

f) Menyesuaikan Padding Toolbar Terhadap Status Bar

```
val statusBarHeight = resources.getDimensionPixelSize(  
    resources.getIdentifier("status_bar_height",  
    "dimen", "android")  
)
```

```
val toolbar: Toolbar = findViewById(R.id.toolBar)  
toolbar.setPadding(0, statusBarHeight, 0, 0)
```

Digunakan untuk mengatur padding atas toolbar agar tidak tumpang tindih dengan status bar perangkat.

2. MainActivityViewModel.kt

File ini memuat logika untuk generate dadu random dan mengatur LiveData.

```
private val _dice1 = MutableLiveData<Int>()  
val dice1: LiveData<Int> get() = _dice1
```

Bagian ini digunakan untuk membedakan siapa yang boleh mengubah data dan siapa yang hanya dapat mengamati data.

`MutableLiveData<Int>()` artinya dapat bisa diubah, namun hanya dapat diakses dan diubah dari dalam `MainActivityViewMode.kt` karena `private`.

```
val dice1: LiveData<Int> get() = _dice1
```

 artinya adalah *read-only* dari versi `_dice1`. Tipe `LiveData` artinya tidak dapat diubah dari luar `MainActivityViewModel.kt`, namun dapat diamati/dibaca oleh `MainActivity.kt` (`viewModel.dice1.observe(this)`).

```
fun rollDice() {  
    val dice = listOf(  
        R.drawable.dice_1,  
        R.drawable.dice_2,
```

```

        R.drawable.dice_3,
        R.drawable.dice_4,
        R.drawable.dice_5,
        R.drawable.dice_6
    )

    _dice1.value = dice.random()
    _dice2.value = dice.random()
}

```

Saat fungsi rollDice() dipanggil, kedua gambar dadu akan dipilih secara acak dan meng-update LiveData nya. UI akan secara otomatis ter-update karena observer di MainActivity.kt.

3. activity_main.xml

`<ConstraintLayout>` digunakan untuk layout utama.

Merupakan layout fleksibel untuk mengatur posisi komponen berdasarkan constraints (posisi relatif).

`<include .../>` digunakan untuk menyisipkan layout toolbar.

Digunakan untuk layout tool_bar.xml agar digunakan di sini, seperti *copy-paste* isi layout toolbar ke activity_main.xml.

`<ImageView/>` digunakan untuk gambar dadu 1.

`ivDice` untuk menampilkan gambar dadu pertama.

`<ImageView/>` digunakan untuk gambar dadu 2.

`ivDice2` digunakan untuk menampilkan gambar dadu kedua.

`<Button/>` digunakan untuk tombol roll dadu.

Merupakan tombol untuk rolling dadu, di mana ketika diklik akan memanggil fungsi rollDice() di ViewModel.

4. tool_bar.xml

File ini merupakan layout khusus toolbar yang akan dimasukkan ke layout utama menggunakan `<include>`.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat
<https://github.com/Noviana21/Praktikum-Pemrograman-Mobile/tree/main/dadu>.