

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Galih Aji Sabdaraya

NIM. 2310817210005

**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 : Galih Aji Sabdaraya
NIM : 2310817210005

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	13
C. Pembahasan	15
Tautan Git.....	19

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 AVD Sebelum Roll Dadu.....	13
Gambar 5. Screenshot AVD Sesudah Roll Dadu	14

DAFTAR TABEL

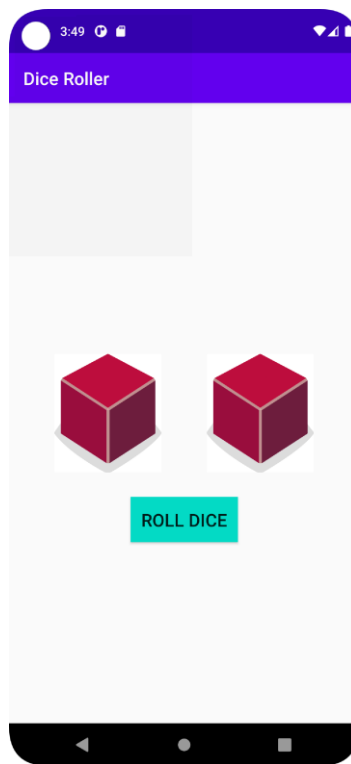
Tabel 1. Source Code MainActivity.kt Soal 1	9
Tabel 2. Source Code DiceViewModel.kt Soal 1	11
Tabel 3. Source activity_main.xml Soal 1	11

SOAL 1

Soal Praktikum:

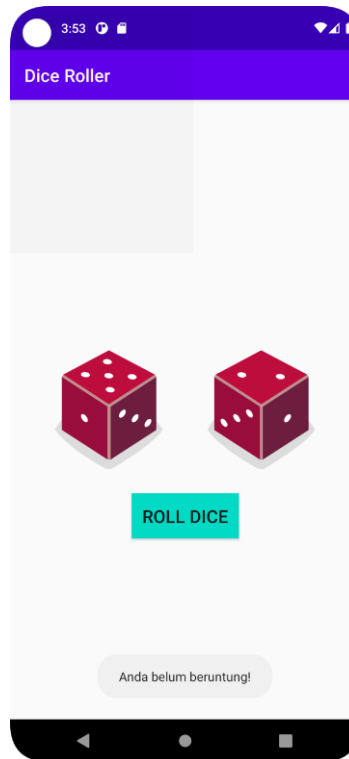
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



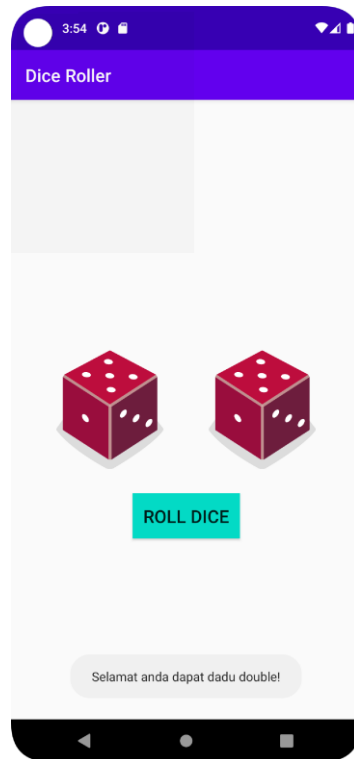
Gambar 1. Tampilan Awal Aplikasi

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.
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=147HT2lIH5qin3z5ta7H9y2N_5OMW81Ll&export=download



Gambar 3. Tampilan Roll Dadu Double

A. Source Code

1. MainActivity.kt

Tabel 1. Source Code MainActivity.kt Soal 1

```
1 package com.example.diceroller
2
3 import android.content.res.Configuration
4 import android.os.Bundle
5 import android.widget.Toast
6 import androidx.activity.enableEdgeToEdge
7 import androidx.activity.viewModels
8 import androidx.appcompat.app.AppCompatActivity
9 import androidx.core.view.ViewCompat
10 import androidx.core.view.WindowInsetsCompat
11 import androidx.lifecycle.Observer
12 import
13     com.example.diceroller.databinding.ActivityMainBinding
14
15 class MainActivity : AppCompatActivity() {
16     private lateinit var binding: ActivityMainBinding
17     private val viewModel : DiceViewModel by viewModels()
18
19     override fun onCreate(savedInstanceState: Bundle?) {
20         super.onCreate(savedInstanceState)
21         enableEdgeToEdge()
22
23         binding =
24             ActivityMainBinding.inflate(layoutInflater)
25         setContentView(binding.root)
26
27         ViewCompat.setOnApplyWindowInsetsListener(binding.topBar)
28         { v, insets ->
29             val topInset =
30                 insets.getInsets(WindowInsetsCompat.Type.systemBars()).top
31             v.setPadding(v.paddingLeft, topInset,
32                 v.paddingRight, v.paddingBottom)
33             insets
34         }
35
36         binding.rollButton.setOnClickListener {
37             viewModel.rollDice()
38
39             val diceLeft = viewModel.diceLeft.value
40             val diceRight = viewModel.diceRight.value
```

```

37
38         if(diceLeft == diceRight){
39             Toast.makeText(this, "Selamat anda dapat
dadu double!", Toast.LENGTH_SHORT).show()
40         } else {
41             Toast.makeText(this, "Anda belum
beruntung!", Toast.LENGTH_SHORT).show()
42         }
43     }
44
45     viewModel.diceLeft.observe(this, Observer { number
->
46 binding.diceLeft.setImageResource(getDiceImage(number))
47     })
48
49     viewModel.diceRight.observe(this, Observer {
number ->
50 binding.diceRight.setImageResource(getDiceImage(number))
51     })
52 }
53
54 private fun getDiceImage(number: Int): Int {
55     return when (number) {
56         1 -> R.drawable.dice_1
57         2 -> R.drawable.dice_2
58         3 -> R.drawable.dice_3
59         4 -> R.drawable.dice_4
60         5 -> R.drawable.dice_5
61         6 -> R.drawable.dice_6
62         else -> throw
IllegalArgumentException("Invalid dice number")
63     }
64 }
65 }

```

2. DiceViewModel.kt

Tabel 2. Source Code DiceViewModel.kt Soal 1

```
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 _diceLeft = MutableLiveData<Int>()
10    private val _diceRight = MutableLiveData<Int>()
11
12    val diceLeft : LiveData<Int> = _diceLeft
13    val diceRight : LiveData<Int> = _diceRight
14
15    fun rollDice() {
16        _diceRight.value = Random.nextInt(1, 7)
17        _diceLeft.value = Random.nextInt(1, 7)
18    }
19 }
```

3. activity_main.xml

Tabel 3. Source activity_main.xml Soal 1

```
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:fitsSystemWindows="true"
10    tools:context=".MainActivity">
11    <TextView
12        android:id="@+id/topBar"
13        android:layout_width="0dp"
14        android:layout_height="50dp"
15        android:background="#7248EF"
16        android:text="@string/app_name"
```

```

17         android:textColor="@android:color/white"
18         android:gravity="center_vertical|start"
19         android:paddingStart="16dp"
20         android:textSize="20sp"
21         app:layout_constraintStart_toStartOf="parent"
22         app:layout_constraintEnd_toEndOf="parent"
23         app:layout_constraintTop_toTopOf="parent" />
24
25     <androidx.constraintlayout.widget.ConstraintLayout
26         android:id="@+id/diceGroup"
27         android:layout_width="wrap_content"
28         android:layout_height="wrap_content"
29         app:layout_constraintTop_toBottomOf="@+id/topBar"
30         app:layout_constraintBottom_toBottomOf="parent"
31         app:layout_constraintStart_toStartOf="parent"
32         app:layout_constraintEnd_toEndOf="parent"
33         android:layout_marginTop="32dp"
34         android:layout_marginBottom="32dp">
35
36         <ImageView
37             android:id="@+id/diceLeft"
38             android:layout_width="176dp"
39             android:layout_height="183dp"
40             android:layout_marginEnd="8dp"
41
42             app:layout_constraintBottom_toTopOf="@+id/rollButton"
43
44             app:layout_constraintEnd_toStartOf="@id/diceRight"
45                 app:layout_constraintTop_toTopOf="parent"
46                 app:srcCompat="@drawable/dice_0" />
47
48         <ImageView
49             android:id="@+id/diceRight"
50             android:layout_width="153dp"
51             android:layout_height="179dp"
52             android:layout_marginStart="8dp"
53
54             app:layout_constraintBottom_toBottomOf="@id/diceLeft"
55
56             app:layout_constraintStart_toEndOf="@id/diceLeft"
57
58             app:layout_constraintTop_toTopOf="@id/diceLeft"
59                 app:srcCompat="@drawable/dice_0" />
60
61         <Button
62             android:id="@+id/rollButton"
63             android:layout_width="wrap_content"

```

```

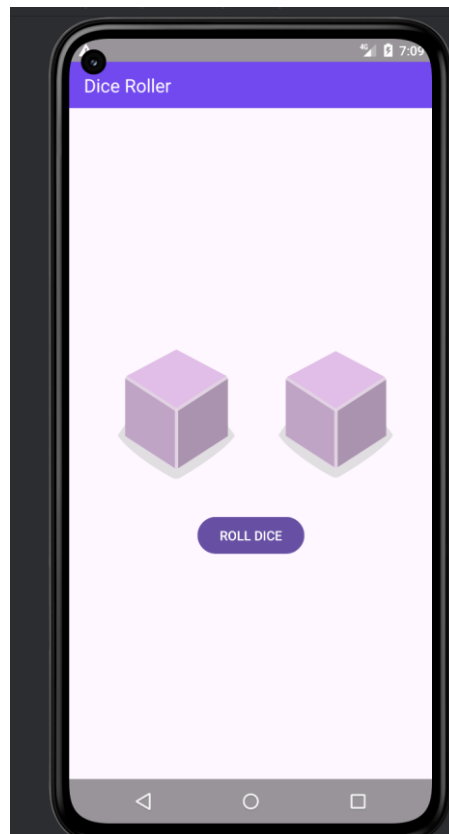
59         android:layout_height="wrap_content"
60         android:text="@string/Roll_AM"

61     app:layout_constraintTop_toBottomOf="@id/diceLeft"

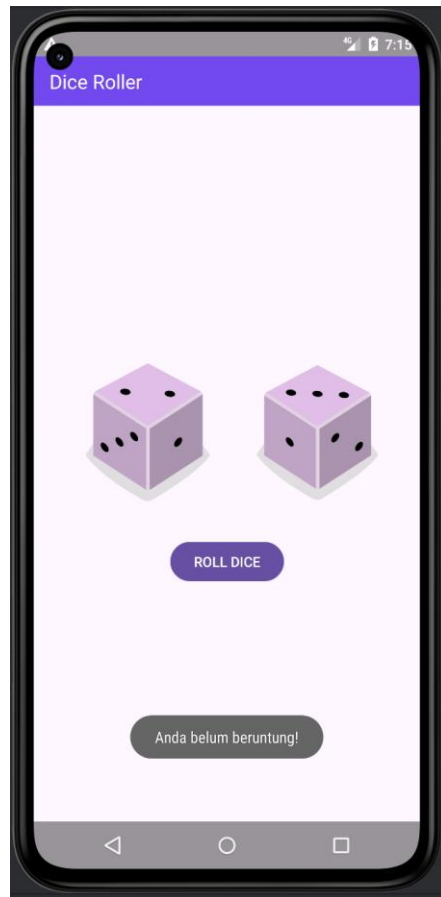
62     app:layout_constraintBottom_toBottomOf="parent"
63         app:layout_constraintStart_toStartOf="parent"
64         app:layout_constraintEnd_toEndOf="parent"
65         android:layout_marginTop="16dp"/>
66     </androidx.constraintlayout.widget.ConstraintLayout>
67
68 </androidx.constraintlayout.widget.ConstraintLayout>

```

B. Output Program



Gambar 4. Screenshot AVD Sebelum Roll Dadu



Gambar 5. Screenshot AVD Sesudah Roll Dadu

C. Pembahasan

1. MainActivity.kt

- Baris [1] adalah deklarasi package. Semua file Kotlin di folder ini akan berada dalam *com.example.diceroller*.
- Baris [3] sampai 12 adalah mengimpor paket – paket yang dibutuhkan dalam kode ini, termasuk fitur UI, ViewModel, data binding, dan window insets.
- Baris [14], class MainActivity : AppCompatActivity() { adalah kelas utama dari aplikasi dan mewarisi dari *AppCompatActivity*.
- Baris [15] sampai baris [16] adalah deklarasi ViewBinding dengan *binding* untuk mengakses elemen UI di activity_main.xml secara langsung tanpa *findViewById()*, dan juga deklarasi ViewModel untuk objek dari kelas *DiceViewModel* yang bertanggung jawab menyimpan state aplikasi (nilai dadu) agar tidak hilang saat berubah orientasi (berubah portrait atau landscape).
- Baris [18] sampai baris [19] adalah *onCreate()* adalah fungsi yang dipanggil saat activity pertama kali atau titik masuk activity.
- Baris [20], *enableEdgeToEdge()* digunakan untuk mengaktifkan edge – to – edge layout yang berarti UI bisa memanjang sampai menyentuh bar navigasi.
- Baris [22] sampai baris [23] adalah menghubungkan layout XML ke kode Kotlin dengan menggunakan ViewBinding. *inflate(layoutInflater)* adalah metode statis yang digunakan untuk membuat instance dari kelas binding dan meng-inflate layout XML menjadi objek tampilan (View). *setContentView(Binding.root)*, adalah merujuk pada root view dari layout XML (*activity_main.xml*).
- Baris [26] sampai baris [30] adalah untuk menyesuaikan padding di bagian atas layout guna tidak tertutup oleh bar status.
- Baris [32] sampai baris [43] adalah ketika saat tombol “ROLL DICE” ditekan (*binding.rollButton.setOnClickListener*), maka akan memutar dadu. Fungsi *rollDice()* di ViewModel akan memutar kedua dadu atau mengacak dari angka 1 sampai 6 untuk kedua dadu. Kemudian, dua variabel bernama *diceLeft* dan *diceRight* diinisialisasi dengan mengambil nilai dari *diceLeft* dan *diceRight* di

ViewModel. Jika nilai *diceLeft* dan *DiceRight* sama, maka akan muncul *toast* “Selamat anda dapat dadu double!”. Jika tidak, maka akan muncul *toast* “Anda belum beruntung!”.

- Baris [45] sampai baris [52] adalah *LiveData* dari ViewModel yang diamati. Ketika nilainya berubah, gambar kedua dadu (dadu kiri dan dadu kanan) di UI akan diperbarui dengan gambar sesuai dengan angka dadu yang telah diputar.
- Baris [54] sampai baris [64] adalah fungsi yang mengembalikan file XML gambar dadu (dadu 1 sampai dadu 6) berdasarkan angka yang dihasilkan dadu. Jika angka tidak valid (1 - 6), akan dilempar *IllegalArgumentException*.

2. DiceViewModel.kt

- DiceViewModel.kt adalah file bagian logika (business logic) dari aplikasi Dice Roller. ViewModel bertugas mengelola dan menyimpan data dadu yang akan digunakan dan diamati oleh *MainActivity*.
- Baris [1] menandakan bahwa kelas file ini adalah bagian dari aplikasi *com.example.diceroller*.
- Baris [3] sampai baris [6] adalah mengimpor paket – paket yang dibutuhkan dalam kode ini, yaitu *Livedata*, *MutableLiveData*, *ViewModel*, dan *Random*.
- Baris [8], kelas *DiceViewModel* mewarisi *ViewModel*, yang dirancang untuk menyimpan dan mengelola data UI secara tahan lama misalnya, saat orientasi layar berubah (portrait ke landscape), data tetap ada.
- Baris [9] sampai baris [10] adalah inisialisasi deklarasi variabel *_diceLeft* dan *_diceRight* sebagai data internal yang menyimpan angka dadu kiri dan dadu kanan menggunakan *MutableLiveData<Int>()*. *MutableLiveData* digunakan untuk menyimpan data yang bisa berubah dan dapat diamati (observable).
- Baris [12] sampai baris [13] adalah dua variabel *diceLeft* dan *diceRight* untuk bisa diakses dari luar kelas ini dan properti read-only yang diberikan ke UI (*MainActivity*) agar bisa mengamati perubahan tapi tidak bisa mengubahnya. *diceLeft* dan *diceRight* mengambil nilai *MutableLiveData* *_diceLeft* dan *_diceRight*, kemudian menyimpannya dengan *LiveData<Int>*.

- Baris [15] sampai baris [18] adalah fungsi *rollDice()* untuk digunakan mengacak angka dadu kiri dan kanan. *Random.nextInt(1, 7)* menghasilkan angka antara 1 sampai 6 (karena batas atas eksklusif) dan nilai tersebut kemudian di-set ke *MutableLiveData* (*_diceLeft* dan *_diceRight*), yang otomatis memicu observer di *MainActivity* untuk memperbarui UI.

3. Activity_main.xml

- Activity_main.xml adalah tampilan utama dari aplikasi Dice Roller.
- Baris [1], terdapat versi XML dan encoding yang digunakan pada file ini.
- Baris [2] sampai baris [9] adalah layout utama menggunakan *ConstraintLayout* yang memungkinkan untuk menyusun elemen UI berdasarkan keterkaitan antar elemen (constraint), bukan sekedar posisi absolut. *match_parent* adalah lebar dan tinggi mengisi layar penuh. *fitsSystemWindows="true"* adalah membuat layout yang sesuai dengan status bar dan navigasi bar.
- Baris [11] sampai baris [23] adalah textView dengan ID *topBar*. Elemen ini adalah header bar di atas layar yang menampilkan nama aplikasi (@string/app_name, yaitu Dice Roller). Header ini berwarna ungu #7248EF dengan teks putih rata kiri dan tengah secara vertikal. Elemen ini memiliki constraint *app:layout_constraintStart_toStartOf="parent"* yang mulai dari sisi kiri layar, *app:layout_constraintEnd_toEndOf="parent"* yang berakhir di sisi kanan, dan *app:layout_constraintTop_toTopOf="parent"* yang berada di atas layar (paling atas).
- Baris [25] sampai baris [66] adalah grup dadu dan tombol yang diletakkan dengan *NestedConstraintLayout*. Grup ini adalah **container tambahan** di tengah layar yang menyimpan kedua dadu dan tombol “ROLL DICE”. Grup ini ditempatkan di tengah layar dengan *Top* di bawah *topBar*, *Bottom* di atas *parent*, dan *start & end* ke *parent*, artinya tengah horizontal.
- Baris [36] sampai baris [44] adalah *imageView* yang menampilkan gambar dadu kiri. Gambarnya memiliki ukuran 176 x 183 dp dan juga ID *diceLeft*. Awal dari *imageView* ini adalah *dice_0*, gambar dadu kosong. Elemen ini

memiliki constraint **Top** yang diletakkan di atas tombol roll dan **end** disambungkan ke **diceRight**, yaitu dadu kanan.

- Baris [46] sampai baris [54] adalah **imageView** yang menampilkan gambar dadu kanan dengan ukuran gambar yang lebih kecil, yaitu 153 x 179 dp. Elemen ini memiliki ID **diceRight**. Sama seperti dadu kiri, gambar awal dari **imageView** adalah **dice_0**. Elemen ini memiliki **constraint Start** disambungkan ke **diceLeft** yang artinya di sebelah kanan dadu kiri dan **Top, Bottom** disejajarkan dengan dadu kiri sehingga tampak lurus secara vertikal.
- Baris [56] sampai baris [65] adalah tombol “ROLL DICE” yang digunakan untuk mengacak dadu saat ditekan. Teks “ROLL DICE” diambil dari **strings.xml** melalui **@string/Roll_AM**. Elemen ini memiliki id **rollButton** dan constraintnya ditempatkan di bawah dadu kiri (**Top_toBottomOf**), ditengah horizontal (**Start & End** ke **parent**), dan diberi margin atas 16dp guna tidak menempel ke dadu.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Kylorts/Pemrograman-mobile>