

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2**



Android Basics in Kotlin

Oleh:

Putri Ridha Amalia NIM. 2010817120007

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MARET 2022**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
MODUL 2

Laporan Praktikum Pemrograman Mobile Modul 2: Android Basics in Kotlin ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Putri Ridha Amalia
NIM : 2010817120007

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Rezi Rahadianor
NIM. 1810817210019

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 1 011

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI.....	3
DAFTAR GAMBAR	4
DAFTAR TABEL.....	5
SOAL	6
A. Source Code	8
B. Output Program	12
C. Pembahasan	14
TAUTAN GIT	16

DAFTAR GAMBAR

Gambar 1 Tampilan Awal Aplikasi	6
Gambar 2 Tampilan Dadu Setelah Di Roll	7
Gambar 3 Tampilan Roll Dadu Double	8
Gambar 5 Screenshot Hasil Tampilan Awal Aplikasi	12
Gambar 6 Screenshot Hasil Tampilan Setelah di Roll	13
Gambar 7 Screenshot Hasil Tampilan Roll Dadu Double	13

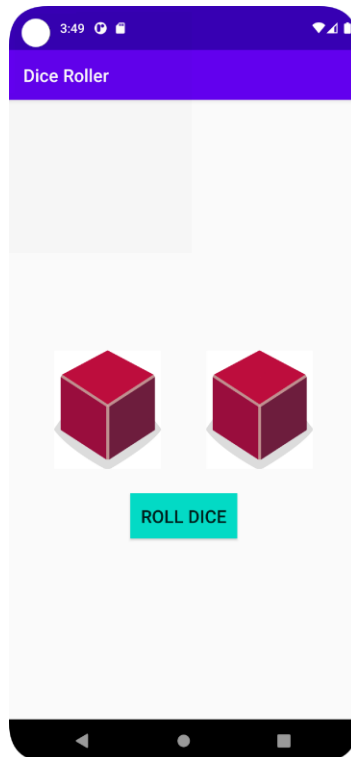
DAFTAR TABEL

Table 1 Source Code colors.xml	8
Table 2 Source Code strings.xml	8
Table 3 Source Code style.xml	9
Table 4 Source Code activity_main.xml	10
Table 5 Source Code MainActivity.kt	12

SOAL

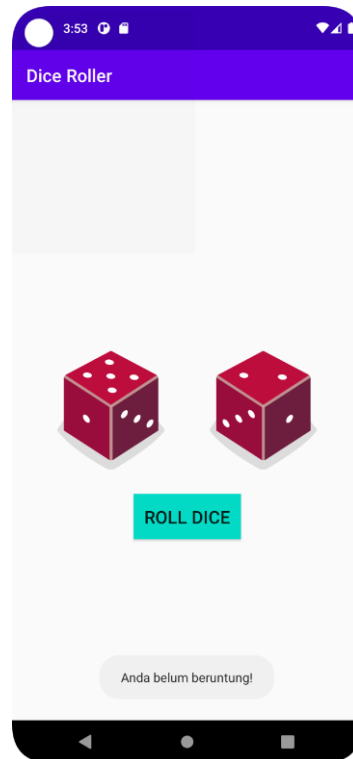
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



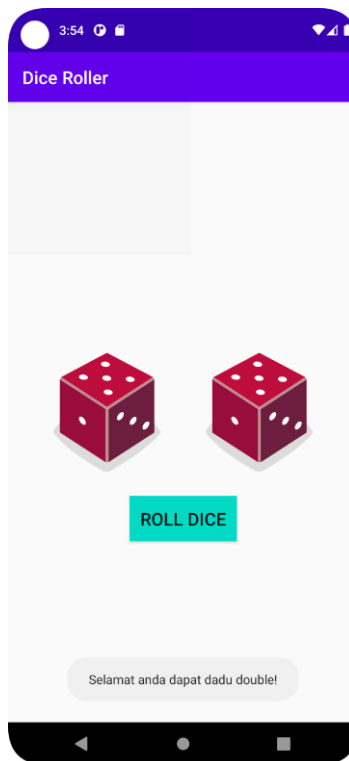
Gambar 1 Tampilan Awal Aplikasi

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_5OMW81Ll&export=download



Gambar 3 Tampilan Roll Dadu Double

A. Source Code

1	<?xml version="1.0" encoding="utf-8"?>
2	<resources>
3	<color name="colorPrimary">#6200EE</color>
4	<color name="colorPrimaryDark">#3700B3</color>
5	<color name="colorAccent">#00CED1</color>
6	</resources>

Table 1 Source Code colors.xml

1	<resources>
2	<string name="app_name">Dice Roller</string>
3	<string name="roll">ROLL DICE</string>
4	</resources>

Table 2 Source Code strings.xml

1	<resources>
2	<item
3	name="colorPrimary">@color/colorPrimary</item>
4	<item
5	name="colorPrimaryDark">@color/colorPrimaryDark</item>
6	<item
7	name="colorAccent">@color/colorAccent</item>
	</style>
	</resources>

Table 3 Source Code style.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	
3	<!-- Layout of the text-based Dice Roller app -->
4	<androidx.constraintlayout.widget.ConstraintLayout
5	
6	xmlns:android="http://schemas.android.com/apk/res/android"
7	xmlns:app="http://schemas.android.com/apk/res-auto"
8	xmlns:tools="http://schemas.android.com/tools"
9	android:layout_width="match_parent"
10	android:layout_height="match_parent"
11	tools:context=".MainActivity">
12	
13	
14	<Button
15	android:id="@+id/button"
16	android:layout_width="wrap_content"
17	android:layout_height="wrap_content"
18	android:layout_marginTop="16dp"
19	android:background="@color/colorAccent"
20	android:text="@string/roll"
21	app:layout_constraintEnd_toEndOf="parent"
22	app:layout_constraintStart_toStartOf="parent"
23	app:layout_constraintTop_toBottomOf="@+id/image1"
24	/>
25	
26	
27	<ImageView
28	android:id="@+id/image1"
29	android:layout_width="100dp"
30	android:layout_height="150dp"
31	android:layout_marginStart="30dp"
32	android:layout_marginTop="265dp"

33	app:layout_constraintEnd_toEndOf="parent"
34	app:layout_constraintHorizontal_bias="0.0"
35	app:layout_constraintStart_toEndOf="@+id/image2"
36	app:layout_constraintTop_toTopOf="parent"
37	tools:srcCompat="@drawable/empty_dice" />
38	
39	<ImageView
40	android:id="@+id/image2"
41	android:layout_width="100dp"
42	android:layout_height="150dp"
43	android:layout_marginStart="80dp"
44	android:layout_marginTop="265dp"
45	app:layout_constraintStart_toStartOf="parent"
46	app:layout_constraintTop_toTopOf="parent"
47	tools:srcCompat="@drawable/empty_dice" />
48	
49	
50	</androidx.constraintlayout.widget.ConstraintLayout>

Table 4 Source Code activity_main.xml

1	
2	package com.example.diceroller
3	
4	import android.os.Bundle
5	import android.widget.Button
6	import android.widget.ImageView
7	import android.widget.Toast
8	import androidx.appcompat.app.AppCompatActivity
9	
10	class MainActivity : AppCompatActivity() {
11	override fun onCreate(savedInstanceState: Bundle?) {
12	super.onCreate(savedInstanceState)
13	setContentView(R.layout.activity_main)
14	
15	
16	val rollButton: Button =
17	findViewById(R.id.button)
18	
19	rollButton.setOnClickListener { rollDice() }
20	
21	val diceImage1: ImageView =
22	findViewById(R.id.image1)
23	val diceImage2: ImageView =
24	findViewById(R.id.image2)
25	
26	

```

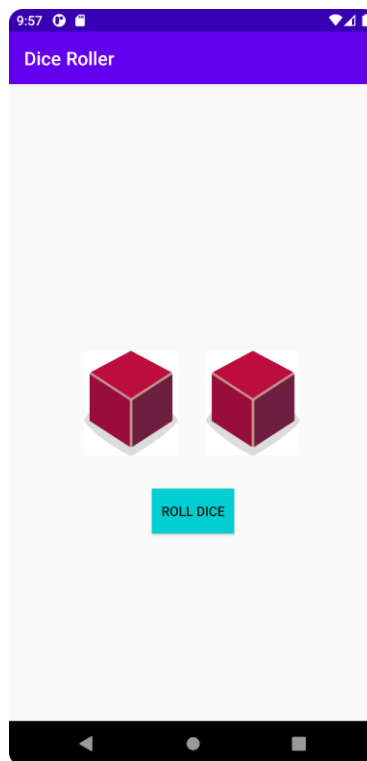
27
28 diceImage1.setImageResource(R.drawable.empty_dice)
29
30 diceImage2.setImageResource(R.drawable.empty_dice)
31     }
32
33     private fun rollDice() {
34         val dice = Dice(6)
35         val diceRoll1 = dice.roll()
36         val diceRoll2 = dice.roll()
37
38         val diceImage1: ImageView =
39 findViewById(R.id.image1)
40         val diceImage2: ImageView =
41 findViewById(R.id.image2)
42
43         val drawableResource1 = when (diceRoll1) {
44             1 -> R.drawable.dice_1
45             2 -> R.drawable.dice_2
46             3 -> R.drawable.dice_3
47             4 -> R.drawable.dice_4
48             5 -> R.drawable.dice_5
49             else -> R.drawable.dice_6
50         }
51
52         val drawableResource2 = when (diceRoll2) {
53             1 -> R.drawable.dice_1
54             2 -> R.drawable.dice_2
55             3 -> R.drawable.dice_3
56             4 -> R.drawable.dice_4
57             5 -> R.drawable.dice_5
58             else -> R.drawable.dice_6
59         }
60
61         diceImage1.setImageResource(drawableResource1)
62         diceImage2.setImageResource(drawableResource2)
63
64         diceImage1.contentDescription =
65 diceRoll1.toString()
66         diceImage2.contentDescription =
67 diceRoll2.toString()
68
69         if (diceRoll1 == diceRoll2) {
70             Toast.makeText(applicationContext, "Selamat
71 anda dapat dadu double!", Toast.LENGTH_LONG).show()
72         } else {
73

```

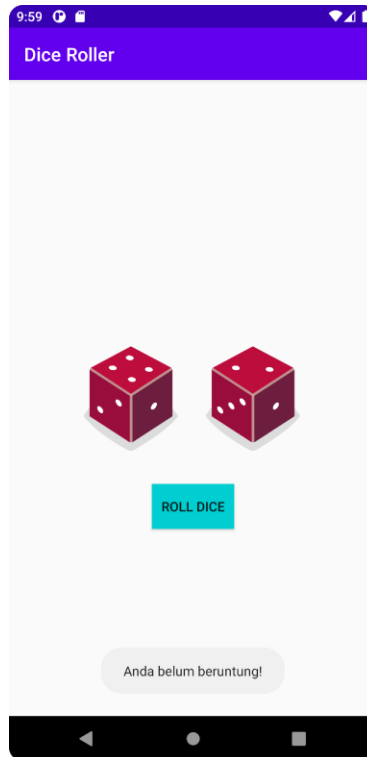
74	Toast.makeText(applicationContext, "Anda
75	belum beruntung!", Toast.LENGTH_LONG).show()
76	
77	}
78	}
79	}
80	
81	class Dice(private val numSides: Int) {
82	fun roll(): Int {
83	return (1..numSides).random()
84	}
	}

Table 5 Source Code MainActivity.kt

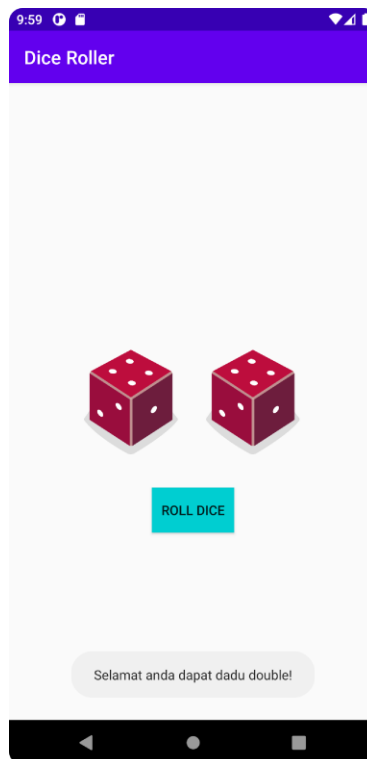
B. Output Program



Gambar 4 Screenshot Hasil Tampilan Awal Aplikasi



Gambar 5 Screenshot Hasil Tampilan Setelah di Roll



Gambar 6 Screenshot Hasil Tampilan Roll Dadu Double

C. Pembahasan

Source Code colors.xml

File colors.xml digunakan untuk mendefinisikan kode warna (background, teks, gambar, dll) pada aplikasi android yang kita buat. Pada baris [5], terdapat warna dengan nama = `colorAccent` dengan code warna hexadesimal `#00CED1` (Dark Turquoise) digunakan untuk memberi warna pada komponen button, dengan cara memanggil id-nya pada `activity_main.xml`.

Source Code strings.xml

File strings.xml digunakan untuk mendefinisikan string yang akan dimasukkan pada aplikasi android yang kita buat. Pada baris [3], terdapat string dengan nama = `roll` berisi teks `ROLL DICE` digunakan untuk menambahkan string pada komponen button, dengan cara memanggil id-nya pada `activity_main.xml`.

Source Code styles.xml

File styles.xml digunakan untuk mendefinisikan styles atau tema pada aplikasi android yang kita buat, dengan cara mengaksesnya pada `activity_main.xml` menggunakan id.

Source Code activity_main.xml

File `activity_main.xml` digunakan untuk mengatur tampilan layout pada aplikasi android yang kita buat. Pada baris [4], terdapat tag `constraint` yang digunakan untuk mengimplementasikan constrain layout pada file `activity_main.xml`. Pada baris [6], terdapat `xmlns:android`, digunakan untuk mendeklarasikan namespace atribut android. sedangkan pada baris [7], `xmlns:app`, untuk deklarasi namespace atribut app, dan pada baris [8], `xmlns:tools`, untuk deklarasi namespace atribut tools. Pada baris [9] dan [10], terdapat `"match_parent"`, digunakan untuk menyesuaikan lebar dan tinggi dari suatu tampilan sesuai dengan layar smartphone android. Pada baris [11], terdapat `tools:context=".MainActivity"`, digunakan sebagai atribut untuk mendeklarasikan class `MainActivity` pada layout.

Pada baris [14], terdapat tag `Button`, digunakan untuk membuat elemen button pada tampilan aplikasi android yang di buat. Pada baris [15], terdapat `android:id="@+id/button"`, digunakan sebagai atribut untuk mengenali button, dengan id bernama button. Pada baris [9] dan [10], terdapat `"wrap_content"`, digunakan untuk menyesuaikan lebar dan tinggi dari suatu tampilan berdasarkan isi/konten yang ada. Pada baris [18], terdapat `android:layout_marginTop`, digunakan untuk mengatur jarak pada bagian atas button. Pada baris [19], terdapat `android:background="@color/colorAccent"`, digunakan untuk mengakses warna background pada file colors.xml. pada baris [20], terdapat `android:text="@string/roll"`, digunakan untuk mengakses string pada file strings.xml.

Pada baris [27], terdapat tag `ImageView`, digunakan untuk membuat widget untuk menampilkan gambar pada tampilan aplikasi android yang kita buat. Pada [37], terdapat `tools:srcCompat`, digunakan untuk mengambil gambar pada folder `drawable`.

Source Code MainActivity.kt

Pada baris [4] – [8], terdapat library yang digunakan pada aplikasi android yang kita buat. Pada baris [10], terdapat `class MainActivity : AppCompatActivity()`, digunakan untuk membuat class pada MainActivity. Pada baris [11], terdapat `onCreate()`, digunakan sebagai kondisi awal saat activity baru diciptakan. Pada baris [13], terdapat `setContentView()`, digunakan untuk menghubungkan activity dengan layout pada file xml. Pada baris [16], terdapat `findViewById(R.id.button)`, digunakan untuk menemukan object button pada file xml. Pada baris [19], terdapat `setOnClickListener()`, digunakan sebagai metode untuk membaca saat tombol button di klik atau touch. Pada baris [21], terdapat `findViewById(R.id.image1)`, digunakan untuk menemukan object gambar pada file xml. Pada baris [28], terdapat `setImageResource()`, digunakan untuk mengatur sumber gambar yang dipakai. Pada baris [43], terdapat `when (diceRoll1)`, digunakan sebagai kondisi apabila mendapat angka 1, maka yang luar gambar `dice_1`, begitu seterusnya sampai angka 5 yang keluar gambar `dice_5`, else yang keluar gambar `dice_6`. Pada baris [64].

Pada baris [69], terdapat kondisi apabila yang keluar gambar dadu 1 == gambar dadu 2 maka akan mengeluarkan text "Selamat anda dapat dadu double!", dan apabila gambar dadu yang dihasilkan berbeda maka akan mengeluarkan text "Anda belum beruntung!".

Pada baris [81], terdapat `class Dice()`, digunakan untuk membuat class dice. Pada baris [82], terdapat `fun roll(): Int`, digunakan untuk membuat fungsi roll, yang didalamnya mengembalikan angka random.

TAUTAN GIT

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/putriridha13/praktikummobile2/tree/master/modul2>