LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1

ANDROID BASIC WITH KOTLIN

Oleh:

Najah Maisyaroh NIM. 2210817120009

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 Prakitkum ini dikerjakan oleh:

Nama Praktikan : Najah Maisyaroh
NIM : 2210817120009

Menyetujui, Asisten Praktikum	Mengetahui, Dosen Penanggung Jawab Praktikum
Muhammad Asandy Putra NIM. 2110817310002	Muti'a Maulida, S.Kom., M.T.I NIP. 1988102 720190 32 013

DAFTAR ISI

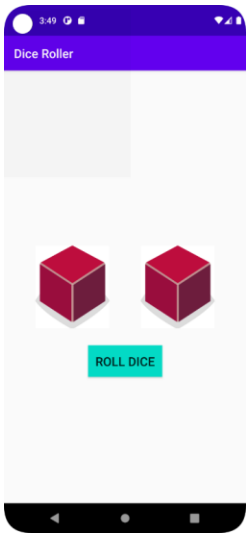
DAFTAR GAMBAR

DAFTAR TABEL

SOAL 1

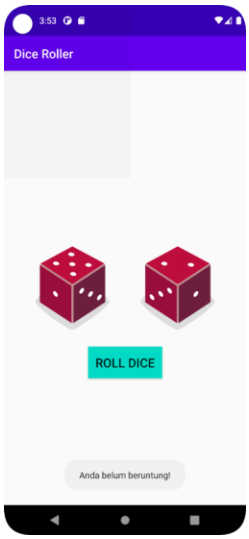
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol "Roll Dice". Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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.

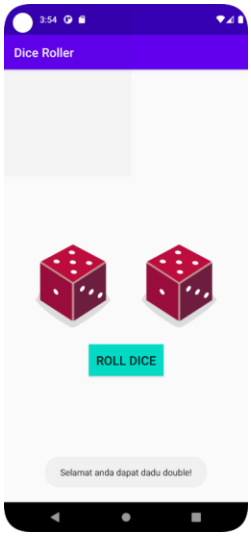


Gambar 2. Tampilan Dadu Setelah Di Roll

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

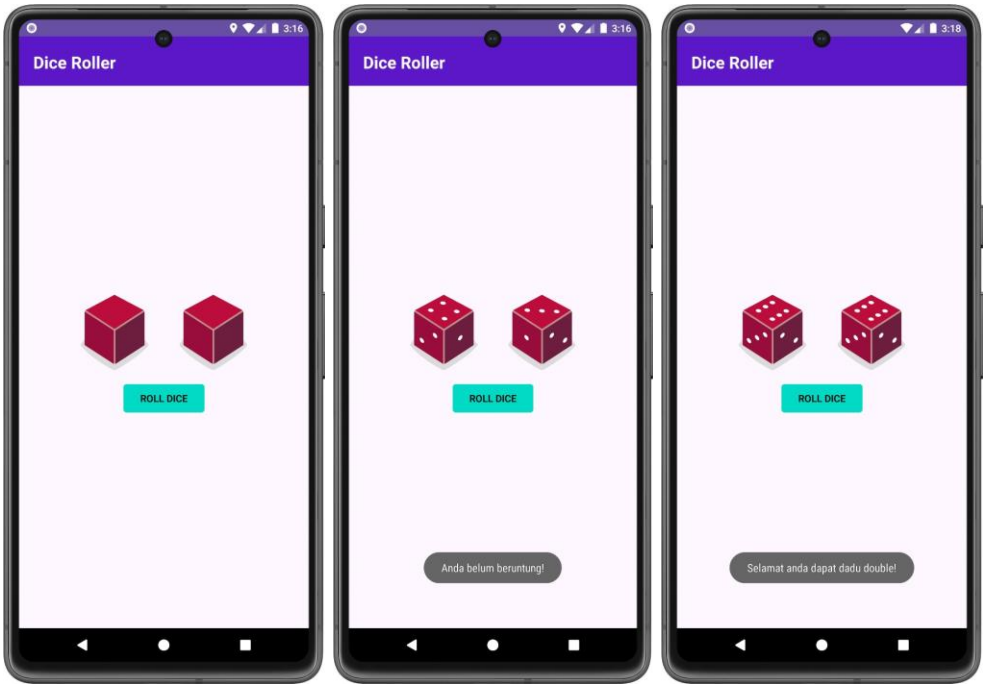
```
1  package com.example.diceroller
2  import android.os.Bundle
3  import android.widget.Toast
4  import android.widget.Button
5  import android.widget.ImageView
6  import androidx.appcompat.app.AppCompatActivity
7
8  class MainActivity : AppCompatActivity() {
9      override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12         val firstDiceImage: ImageView =
   findViewById(R.id.imageView)
13         val secondDiceImage: ImageView =
   findViewById(R.id.imageView2)
```

```
14 firstDiceImage.setImageResource(R.drawable.empty_dice)
15 secondDiceImage.setImageResource(R.drawable.empty_dice)
16         val         rollButton:         Button         =
   findViewById(R.id.button)
17         rollButton.setOnClickListener { rollDices() }
18     }
19     private fun rollDices() {
20         val dice = Dice(6)
21         val firstRoll = dice.roll()
22         val secondRoll = dice.roll()
23         val       firstImageView:       ImageView       =
   findViewById(R.id.imageView)
24         val      secondImageView:      ImageView      =
   findViewById(R.id.imageView2)
25         val firstDrawable = when (firstRoll) {
26             1 -> R.drawable.dice_1
27             2 -> R.drawable.dice_2
28             3 -> R.drawable.dice_3
29             4 -> R.drawable.dice_4
30             5 -> R.drawable.dice_5
31             else -> R.drawable.dice_6
32         }
33         val secondDrawable = when (secondRoll) {
34             1 -> R.drawable.dice_1
35             2 -> R.drawable.dice_2
36             3 -> R.drawable.dice_3
37             4 -> R.drawable.dice_4
38             5 -> R.drawable.dice_5
39             else -> R.drawable.dice_6
40         }
41         firstImageView.setImageResource(firstDrawable)
42 secondImageView.setImageResource(secondDrawable)
43         firstImageView.contentDescription              =
   firstRoll.toString()
44         secondImageView.contentDescription             =
   secondRoll.toString()
45         if (firstRoll != secondRoll) {
46             Toast.makeText(this,       "Anda       belum
   beruntung!", Toast.LENGTH_SHORT).show()
47         } else {
48             Toast.makeText(this,  "Selamat  anda  dapat
   dadu double!", Toast.LENGTH_SHORT).show()
49         }
50     }
```

```
51  }
52  class Dice(private val sides: Int) {
53      fun roll(): Int {
54          return (1..sides).random()
55      }
56  }
```

Table 1.Source Code Jawaban Soal 1

B. Output Program



Gambar 4.Screenshot Hasil Jawaban Soal 1

C. Pembahasan

Baris [1]: package com.example.diceroller: Pada baris ini, mendefinisikan paket (package) di mana kelas MainActivity dan Dice berada. Paket digunakan untuk mengelompokkan kelas dan menghindari konflik nama dengan kelas dari paket lain.

Baris [2]: import android.os.Bundle: Mengimpor kelas Bundle dari paket android.os, yang digunakan untuk menyimpan data dalam aktivitas Android.

Baris [3]: import android.widget.Toast: Mengimpor kelas Toast dari paket android.widget. Toast digunakan untuk menampilkan pesan singkat pada layar.

Baris [4]: import android.widget.Button: Mengimpor kelas Button dari paket android.widget. Button adalah elemen antarmuka pengguna yang merepresentasikan tombol.

Baris [5]: import android.widget.ImageView: Mengimpor kelas ImageView dari paket android.widget. ImageView adalah elemen antarmuka pengguna yang digunakan untuk menampilkan gambar.

Baris [6]: import androidx.appcompat.app.AppCompatActivity: Mengimpor kelas AppCompatActivity dari paket androidx.appcompat.app. AppCompatActivity adalah aktivitas dasar yang dapat digunakan untuk mengimplementasikan antarmuka pengguna.

Baris [8]: class MainActivity : AppCompatActivity(): Pada baris ini, mendefinisikan kelas MainActivity yang mengextends AppCompatActivity. Ini menunjukkan bahwa MainActivity adalah aktivitas yang akan dijalankan saat aplikasi dimulai.

Baris [9]: override fun onCreate(savedInstanceState: Bundle?): Mendefinisikan fungsi onCreate, yang merupakan fungsi siklus hidup aktivitas pertama yang dipanggil saat aktivitas dibuat. savedInstanceState adalah parameter yang digunakan untuk menyimpan data yang tersimpan.

Baris [10]: super.onCreate(savedInstanceState): Memanggil implementasi onCreate dari kelas dasar (AppCompatActivity) untuk menjalankan proses pengaturan standar aktivitas.

Baris [11]: setContentView(R.layout.activity_main): Mengatur tata letak aktivitas menggunakan file XML activity_main.xml.

Baris [12]: val firstDiceImage: ImageView = findViewById(R.id.imageView): Mendefinisikan variabel firstDiceImage sebagai ImageView, dan menghubungkannya dengan elemen ImageView di tata letak dengan ID R.id.imageView.

Baris [13]: val secondDiceImage: ImageView = findViewById(R.id.imageView2): Mendefinisikan variabel secondDiceImage sebagai ImageView, dan menghubungkannya dengan elemen ImageView di tata letak dengan ID R.id.imageView2.

Baris [14]: firstDiceImage.setImageResource(R.drawable.empty_dice): Mengatur gambar kosong untuk firstDiceImage menggunakan sumber daya gambar R.drawable.empty_dice.

Baris [15]: secondDiceImage.setImageResource(R.drawable.empty_dice): Mengatur gambar kosong untuk secondDiceImage menggunakan sumber daya gambar R.drawable.empty_dice.

Baris [16]: val rollButton: Button = findViewById(R.id.button): Mendefinisikan variabel rollButton sebagai Button, dan menghubungkannya dengan elemen Button di tata letak dengan ID R.id.button.

Baris [17]: rollButton.setOnClickListener { rollDices() }: Mengatur OnClickListener pada tombol rollButton sehingga saat tombol ditekan, fungsi rollDices akan dipanggil.

Baris [19]: private fun rollDices(): Mendefinisikan fungsi rollDices yang akan mengimplementasikan logika untuk menggulir dadu.

Baris [20]: val dice = Dice(6): Mendefinisikan variabel dice sebagai objek Dice dengan 6 sisi.

Baris [21]: val firstRoll = dice.roll(): Mendefinisikan variabel firstRoll sebagai hasil dari pengguliran dadu pertama.

Baris [22]: val secondRoll = dice.roll(): Mendefinisikan variabel secondRoll sebagai hasil dari pengguliran dadu kedua.

Baris [23]: val firstImageView: ImageView = findViewById(R.id.imageView): Mendefinisikan variabel firstImageView sebagai ImageView, dan menghubungkannya dengan elemen ImageView di tata letak dengan ID R.id.imageView.

Baris [24]: val secondImageView: ImageView = findViewById(R.id.imageView2): Mendefinisikan variabel secondImageView sebagai ImageView, dan menghubungkannya dengan elemen ImageView di tata letak dengan ID R.id.imageView2.

Baris [25-32]: val firstDrawable = when (firstRoll) {...}: Menentukan sumber daya gambar (drawable) yang sesuai dengan hasil pengguliran dadu pertama (firstRoll).

Baris [33-40]: val secondDrawable = when (secondRoll) {...}: Menentukan sumber daya gambar (drawable) yang sesuai dengan hasil pengguliran dadu kedua (secondRoll).

Baris [41]: firstImageView.setImageResource(firstDrawable): Mengatur gambar yang sesuai dengan firstRoll pada firstImageView.

Baris [42]: secondImageView.setImageResource(secondDrawable): Mengatur gambar yang sesuai dengan secondRoll pada secondImageView.

Baris [43]: firstImageView.contentDescription = firstRoll.toString(): Mengatur deskripsi konten (accessibility content description) dari firstImageView sebagai string yang mewakili hasil firstRoll.

Baris [44]: secondImageView.contentDescription = secondRoll.toString(): Mengatur deskripsi konten (accessibility content description) dari secondImageView sebagai string yang mewakili hasil secondRoll.

Baris [45]: if (firstRoll != secondRoll): Mengecek apakah hasil dari pengguliran dadu pertama berbeda dengan hasil pengguliran dadu kedua.

Baris [46]: Toast.makeText(this, "Anda belum beruntung!", Toast.LENGTH_SHORT).show(): Jika hasil dari pengguliran dadu pertama berbeda dengan hasil pengguliran dadu kedua, menampilkan pesan toast "Anda belum beruntung!" pada layar.

Baris [47]: else: Kondisi untuk penanganan jika firstRoll sama dengan secondRoll.

Baris [48]: Toast.makeText(this, "Selamat anda dapat dadu double!", Toast.LENGTH_SHORT).show(): Jika hasil dari pengguliran dadu pertama sama dengan hasil pengguliran dadu kedua, menampilkan pesan toast "Selamat anda dapat dadu double!" pada layar.

Baris [52]: class Dice(private val sides: Int): Mendefinisikan kelas Dice dengan parameter sides yang mewakili jumlah sisi dadu.

Baris [53]: fun roll(): Int: Mendefinisikan fungsi roll yang menggulir dadu dan mengembalikan hasilnya sebagai integer.

Baris [54]: return (1..sides).random(): Mengembalikan angka acak antara 1 dan sides (jumlah sisi dadu) sebagai hasil pengguliran dadu.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
https://github.com/nanutt/Praktikum_Pemrograman_Mobile/tree/main/Modul%201/DiceRoller