

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Dhea Aprilinda utami NIM. 2210817220019

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 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 Praktikum ini dikerjakan oleh:

Nama Praktikan : Dhea Aprilinda utami
NIM : 2210917220019

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Asandy Putra
NIM. 2110817310002

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 01 011

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	Error! Bookmark not defined.
SOAL 1	5
A. Source Code	7
B. Output Program.....	11
C. Pembahasan	12

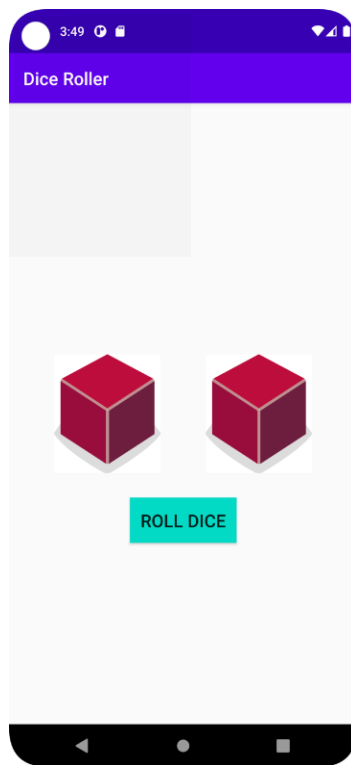
DAFTAR GAMBAR

Gambar 1. Source Code Jawaban Soal 1	10
Gambar 2. Screenshot Hasil Jawaban Soal 1	11

SOAL 1

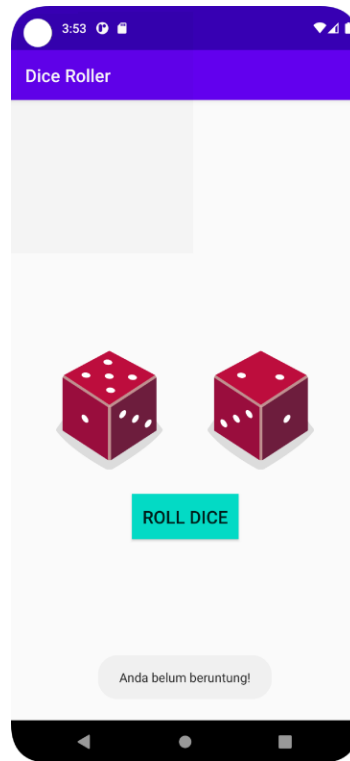
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



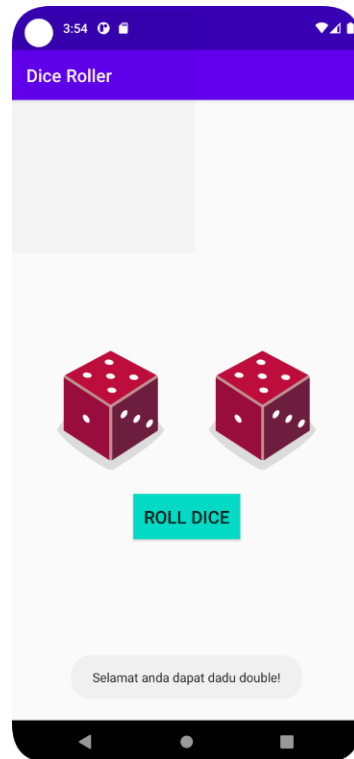
Gambar 1 Tampilan Awal Aplikasi

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

Simpan dengan nama file: PRAK101.kt

A. Source Code

```
1 package com.example.dice
2
3 import android.annotation.SuppressLint
4 import android.os.Bundle
5 import android.widget.Toast
6 import androidx.activity.ComponentActivity
7 import androidx.activity.compose.setContent
8 import androidx.compose.foundation.Image
9 import androidx.compose.foundation.layout.Column
10 import androidx.compose.foundation.layout.Row
11 import androidx.compose.foundation.layout.Spacer
12 import androidx.compose.foundation.layout.fillMaxSize
13 import androidx.compose.foundation.layout.height
14 import androidx.compose.foundation.layout.padding
15 import androidx.compose.foundation.layout.size
16 import androidx.compose.foundation.layout.wrapContentSize
17 import androidx.compose.material3.Button
18 import androidx.compose.material3.ExperimentalMaterial3Api
19 import androidx.compose.material3.Scaffold
20 import androidx.compose.material3.Text
21 import androidx.compose.material3.TopAppBar
22 import
```

```

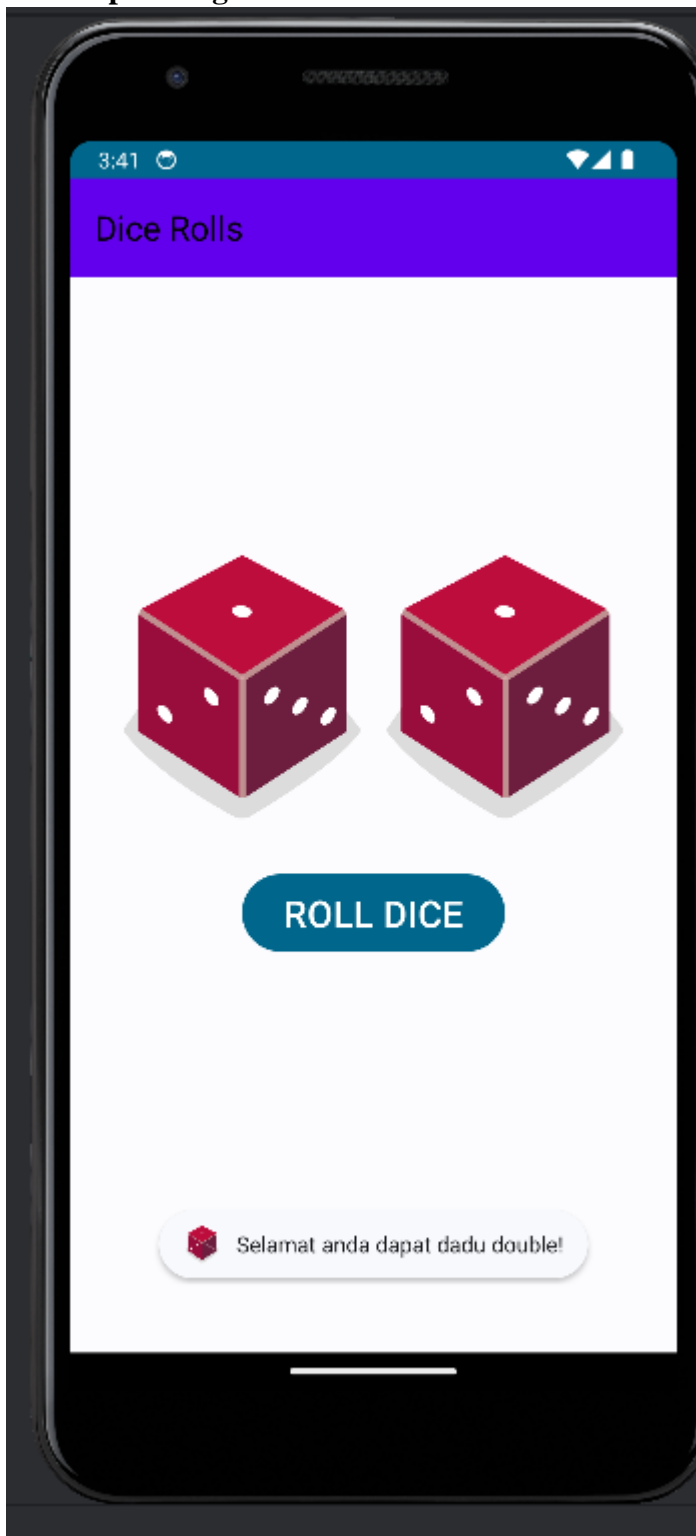
23 androidx.compose.material3.TopAppBarDefaults.topAppBarColors
24 import androidx.compose.runtime.Composable
25 import androidx.compose.runtime.getValue
26 import androidx.compose.runtime.mutableStateOf
27 import androidx.compose.runtime.setValue
28 import androidx.compose.ui.Alignment
29 import androidx.compose.ui.Modifier
30 import androidx.compose.ui.graphics.Color
31 import androidx.compose.ui.platform.LocalContext
32 import androidx.compose.ui.res.painterResource
33 import androidx.compose.ui.res.stringResource
34 import androidx.compose.ui.tooling.preview.Preview
35 import androidx.compose.ui.unit.dp
36 import androidx.compose.ui.unit.sp
37 import com.example.dice.ui.theme.DiceTheme
38
39 class MainActivity : ComponentActivity() {
40     override fun onCreate(savedInstanceState: Bundle?) {
41         super.onCreate(savedInstanceState)
42         setContent {
43             DiceTheme {
44                 DiceRollerApp()
45             }
46         }
47     }
48 }
49
50 @SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
51 @OptIn(ExperimentalMaterial3Api::class)
52 @Preview
53 @Composable
54 fun DiceRollerApp() {
55     Scaffold(
56         topBar = {
57             TopAppBar(
58                 colors = topAppBarColors(
59                     containerColor = Color(0xFF6200EE),
60                     titleContentColor = Color.Black,
61                 ),
62                 title = {
63                     Text("Dice Rolls")
64                 }
65             )
66         },
67         content = {
68             DiceWithButtonAndImage(modifier = Modifier
69                 .fillMaxSize()
70                 .wrapContentSize(Alignment.Center))
71         }
72     )
73 }

```


124	beruntung!", Toast.LENGTH_SHORT).show()
125	}
126) {
127	Text(text = stringResource(R.string.roll),
128	fontSize = 24.sp)
129	}
130	}

Gambar 1. Source Code Jawaban Soal 1

B. Output Program



Gambar 2. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

Line 1 Package tempat menyimpan program
Line 2
Line 3 Inport library
Line 4 Inport library
Line 5 Inport library
Line 6 Inport library
Line 7 Inport library
Line 8 Inport library
Line 9 Inport library
Line 10 Inport library
Line 11 Inport library
Line 12 Inport library
Line 13 Inport library
Line 14 Inport library
Line 15 Inport library
Line 16 Inport library
Line 17 Inport library
Line 18 Inport library
Line 19 Inport library
Line 20 Inport library
Line 21 Inport library
Line 22 Inport library
Line 23 Inport library
Line 24 Inport library
Line 25 Inport library
Line 26 Inport library
Line 27 Inport library
Line 28 Inport library
Line 29 Inport library
Line 30 Inport library
Line 31 Inport library
Line 32 Inport library
Line 33 Inport library
Line 34 Inport library
Line 35 Inport library
Line 36 Inport library
Line 37 Inport library
Line 38
Line 39 class MainActivity : ComponentActivity() { : Ini adalah deklarasi kelas MainActivity, yang merupakan turunan dari kelas ComponentActivity. ComponentActivity adalah kelas dasar yang menyediakan implementasi standar dari LifecycleOwner untuk menggunakan arsitektur komponen Android.

Line 40 `override fun onCreate(savedInstanceState: Bundle?) {`: Ini adalah override dari metode `onCreate()` dari kelas `ComponentActivity`. Metode ini dipanggil ketika aktivitas dibuat.

Line 41 `super.onCreate(savedInstanceState)`: Ini adalah pemanggilan metode `onCreate()` dari kelas induk (`ComponentActivity`) menggunakan kata kunci `super`. Ini memastikan bahwa fungsionalitas dari metode `onCreate()` kelas induk dieksekusi.

Line 42 `setContent { ... }`: Ini adalah ekstensi dari kelas `ComponentActivity` yang digunakan untuk menetapkan tata letak (layout) konten aktivitas dengan menggunakan fungsi `lambda`. Dalam blok `lambda` ini, ditetapkan tata letak konten aktivitas.

Line 43 `DiceTheme { ... }`: Ini adalah pemanggilan komponen `DiceTheme`, yang merupakan tema yang digunakan dalam tata letak konten aktivitas. Tema ini mungkin mengatur tampilan dan gaya visual dari aplikasi.

Line 44 `DiceRollerApp()`: Ini adalah pemanggilan komponen `DiceRollerApp()`, yang mungkin merupakan bagian dari tata letak konten aktivitas. Ini mungkin berisi logika dan tampilan aplikasi yang sebenarnya.

Line 45 `}`

Line 46 `}`

Line 47 `}`

Line 48 `}`

Line 49

Line 50 `@SuppressWarnings("UnusedMaterial3ScaffoldPaddingParameter")`: Ini adalah anotasi yang menandakan bahwa pemeriksaan lint dengan nama `"UnusedMaterial3ScaffoldPaddingParameter"` harus diabaikan untuk fungsi ini. Ini menghilangkan peringatan yang tidak diperlukan dari lint.

Line 51 `@OptIn(ExperimentalMaterial3Api::class)`: Ini adalah anotasi yang menandakan bahwa fungsi ini menggunakan API eksperimental dari `Material Design 3`. Dengan menggunakan anotasi ini, pengembang menyetujui penggunaan API eksperimental dan mengetahui bahwa API tersebut dapat berubah dalam versi yang akan datang.

Line 52 `@Preview`: Ini adalah anotasi yang menandakan bahwa fungsi ini adalah preview dari antarmuka pengguna yang dibuat dengan menggunakan `Jetpack Compose`. Preview ini akan ditampilkan di `Android Studio` sehingga pengembang dapat melihat tampilan antarmuka pengguna tanpa perlu menjalankan aplikasi.

Line 53 `@Composable`: Ini adalah anotasi yang menandakan bahwa fungsi ini adalah sebuah komposisi. Dalam konteks `Jetpack Compose`, komposisi adalah fungsi yang mendefinisikan antarmuka pengguna.

Line 54 `fun DiceRollerApp() { ... }`: Ini adalah deklarasi fungsi `DiceRollerApp`, yang merupakan komposisi untuk membuat antarmuka pengguna aplikasi pemutar dadu.

Line 55 `Scaffold { ... }`: Ini adalah komponen utama yang digunakan dalam komposisi. `Scaffold` adalah komponen yang menyediakan tata letak dasar untuk aplikasi, termasuk bar atas (top bar), area konten utama, dan komponen lainnya seperti floating action button (FAB). Ini memungkinkan pengembang untuk membuat tata letak yang konsisten dengan mudah.

Line 56 `topBar = { ... }`: Ini adalah parameter dari Scaffold yang digunakan untuk menentukan komponen yang akan ditampilkan di bagian atas antarmuka pengguna. Dalam hal ini, kita menggunakan `TopAppBar` sebagai bar atas. `TopAppBar` adalah komponen yang menampilkan bar atas dengan judul dan opsional tombol aksi.

Line 57 `TopAppBar { ... }`: Ini adalah komponen `TopAppBar` yang digunakan untuk menampilkan bar atas dalam antarmuka pengguna. Di dalamnya, kita menentukan warna latar belakang (`colors`), judul (`title`), dan warna teks judul (`titleContentColor`) dengan menggunakan properti-properti yang tersedia.

Line 58 `colors = topAppBarColors { ... }`: Ini adalah properti yang digunakan untuk menentukan warna-warna yang akan digunakan dalam `TopAppBar`. Dalam hal ini, kita menggunakan fungsi `topAppBarColors`

Line 59 menentukan warna latar belakang (`containerColor`)

Line 60 warna teks judul (`titleContentColor`).

Line 61 `)`

Line 62 `title = { Text("Dice Rolls") }`: Ini adalah properti yang digunakan untuk menentukan judul yang akan ditampilkan dalam `TopAppBar`. Dalam hal ini, kita menggunakan komponen `Text` untuk menampilkan teks "Dice Rolls" sebagai judul.

Line 63 `}`

Line 64 `)`

Line 65 `}`

Line 66 `content = { ... }`: Ini adalah parameter dari Scaffold yang menentukan konten utama yang akan ditampilkan di antarmuka pengguna. Konten utama ini dapat berupa satu atau beberapa komponen yang ingin ditampilkan.

Line 67 `DiceWithButtonAndImage(...)`: Ini adalah komponen `DiceWithButtonAndImage` yang akan ditampilkan sebagai konten utama dalam antarmuka pengguna. Komponen ini mengandung elemen-elemen untuk memainkan dadu, termasuk tombol untuk melempar dadu dan gambar dadu yang akan berubah setiap kali dadu dilempar.

Line 68 `modifier = Modifier.fillMaxSize().wrapContentSize(Alignment.Center)`: Ini adalah

Line 69 properti modifier yang digunakan untuk mengatur tata letak komponen `DiceWithButtonAndImage`. Dalam hal ini, kita menggunakan `fillMaxSize` untuk membuat komponen memenuhi seluruh ruang yang tersedia,

Line 70 `wrapContentSize` dengan `Alignment.Center` untuk memposisikan komponen ke tengah secara horizontal dan vertikal.

Line 71 `}`

Line 72 `)`

Line 73 `}`

Line 74

Line 75 `@Composable`: Anotasi yang menandai bahwa fungsi `DiceWithButtonAndImage` adalah sebuah komposable function, yang berarti fungsi ini dapat digunakan untuk membuat UI.

Line 76 `fun DiceWithButtonAndImage(modifier: Modifier = Modifier)`: Ini adalah deklarasi fungsi komposable `DiceWithButtonAndImage` yang mengambil parameter modifier

dari tipe Modifier. Modifier digunakan untuk mengatur properti seperti ukuran, padding, dan tata letak suatu komponen.

- Line 77 `var hasildadu1 by remember { mutableStateOf(0) }`: Ini mendeklarasikan variabel `hasildadu1` yang akan menyimpan hasil dari lemparan dadu pertama. Variabel ini diinisialisasi menggunakan `mutableStateOf(0)`, yang merupakan salah satu cara untuk membuat state yang dapat diubah di dalam fungsi komposable. `remember` digunakan untuk mempertahankan nilai variabel antara rekompilasi komponen, sehingga nilainya tidak akan direset saat fungsi di-recompose.
- Line 78 `var hasildadu2 by remember { mutableStateOf(0) }`: Ini adalah langkah yang serupa dengan `hasildadu1`, tetapi untuk hasil lemparan dadu kedua.
- Line 79 `val context = LocalContext.current`: Ini mendapatkan konteks lokal saat ini menggunakan `LocalContext.current`. Konteks digunakan untuk mengakses sumber daya aplikasi seperti `drawable`, `string`, dan lainnya.
- Line 80 `val dadu1 = when (hasildadu1) { ... }`: Ini adalah ekspresi `when` yang mengatur gambar dadu yang akan ditampilkan berdasarkan nilai `hasildadu1`. Bergantung pada nilai `hasildadu1`, variabel `dadu1` akan diatur dengan ID sumber daya yang sesuai dari `drawable`.
- Line 81 Jika `hasildadu1` adalah 0, maka akan menampilkan gambar dadu kosong (`R.drawable.empty_dice`).
- Line 82 Jika `hasildadu1` adalah 1, akan menampilkan gambar dadu dengan satu titik (`R.drawable.dice_1`).
- Line 83 Jika `hasildadu1` adalah 2, akan menampilkan gambar dadu dengan dua titik (`R.drawable.dice_2`).
- Line 84 Jika `hasildadu1` adalah 3, akan menampilkan gambar dadu dengan dua titik (`R.drawable.dice_3`).
- Line 85 Jika `hasildadu1` adalah 4, akan menampilkan gambar dadu dengan dua titik (`R.drawable.dice_4`).
- Line 86 Jika `hasildadu1` adalah 5, akan menampilkan gambar dadu dengan dua titik (`R.drawable.dice_5`).
- Line 87 Jika `hasildadu1` bukan angka dari 0 hingga 5, maka akan menampilkan gambar dadu nomor 6 (`R.drawable.dice_6`).
- Line 88 `}`
- Line 89 `val dadu2 = when (hasildadu2) { ... }`: Ini adalah ekspresi `when` yang mengatur gambar dadu yang akan ditampilkan berdasarkan nilai `hasildadu2`. Bergantung pada nilai `hasildadu2`, variabel `dadu2` akan diatur dengan ID sumber daya yang sesuai dari `drawable`.
- Line 90 Jika `hasildadu2` adalah 0, maka akan menampilkan gambar dadu kosong (`R.drawable.empty_dice`).
- Line 91 Jika `hasildadu2` adalah 1, akan menampilkan gambar dadu dengan satu titik (`R.drawable.dice_1`).
- Line 92 Jika `hasildadu2` adalah 2, akan menampilkan gambar dadu dengan dua titik (`R.drawable.dice_2`).

Line 93 Jika hasildadu2 adalah 3, akan menampilkan gambar dadu dengan dua titik (R.drawable.dice_3).

Line 94 Jika hasildadu2 adalah 4, akan menampilkan gambar dadu dengan dua titik (R.drawable.dice_4).

Line 95 Jika hasildadu2 adalah 5, akan menampilkan gambar dadu dengan dua titik (R.drawable.dice_5).

Line 96 Jika hasildadu2 bukan angka dari 0 hingga 5, maka akan menampilkan gambar dadu nomor 6 (R.drawable.dice_6).

Line 97 }

Line 98

Line 99 Column(modifier = modifier, horizontalAlignment = Alignment.CenterHorizontally) {Ini adalah sebuah Column, yang merupakan komponen UI dalam Jetpack Compose untuk menempatkan elemen-elemen dalam satu kolom vertikal.modifier = modifier menetapkan modifikasi yang diteruskan dari luar ke komponen Column.horizontalAlignment = Alignment.CenterHorizontally menetapkan alignment horizontal dari komponen di dalam Column agar berada di tengah secara horizontal.

Line 100 Row {Ini adalah sebuah Row, komponen UI yang digunakan untuk menempatkan elemen-elemen dalam satu baris horizontal.

Line 101 Image(Ini adalah komponen untuk menampilkan gambar.

Line 102 modifier = Modifier.size(170.dp) menetapkan ukuran gambar dengan lebar dan tinggi 170 density-independent pixels (dp).

Line 103 painter = painterResource(dadu1) mengambil gambar yang akan ditampilkan, diambil dari resource menggunakan nilai dadu1.

Line 104 contentDescription = hasildadu1.toString() adalah teks deskripsi konten yang digunakan untuk aksesibilitas. Ini diambil dari nilai hasildadu1 yang dikonversi menjadi string.

Line 105)

Line 106 Image(Ini adalah komponen untuk menampilkan gambar.

Line 107 modifier = Modifier.size(170.dp) menetapkan ukuran gambar dengan lebar dan tinggi 170 density-independent pixels (dp).

Line 108 painter = painterResource(dadu2) mengambil gambar yang akan ditampilkan, diambil dari resource menggunakan nilai dadu2.

Line 109 contentDescription = hasildadu2.toString() adalah teks deskripsi konten yang digunakan untuk aksesibilitas. Ini diambil dari nilai hasildadu1 yang dikonversi menjadi string.

Line 110)

Line 111 }.

Line 112 Spacer(modifier = Modifier.height(20.dp))Spacer adalah komponen yang digunakan untuk menambahkan ruang kosong di antara elemen-elemen UI.modifier = Modifier.height(20.dp) menetapkan modifikasi pada spacer, dalam hal ini tinggi sebesar 20 density-independent pixels (dp).

Line 113 Button(komponen untuk menampilkan tombol.

Line 114 modifier = Modifier.padding(vertical = 16.dp).size(width = 170.dp, height = 50.dp)
 menetapkan modifikasi pada tombol:

Line 115 .padding(vertical = 16.dp) menambahkan padding secara vertikal sebesar 16dp.

Line 116 .size(width = 170.dp, height = 50.dp) menetapkan ukuran tombol dengan lebar
 170dp dan tinggi 50dp.

Line 117 onClick = { ... } properti onClick dari komponen Button, yang menentukan
 tindakan yang akan dilakukan ketika tombol diklik.

Line 118 hasildadu1 = (1..6).random() logika untuk menghasilkan angka acak dari 1 hingga 6
 dan menentukannya ke variabel hasildadu1. (1..6) adalah rentang dari 1 hingga 6,
 dan .random() mengambil nilai acak dari rentang tersebut.

Line 119 hasildadu2 = (1..6).random() logika yang serupa dengan hasildadu1, tetapi
 menghasilkan nilai acak untuk hasildadu2.

Line 120 if (hasildadu1 == hasildadu2) { ... }: Memulai blok kode yang akan dijalankan jika
 hasil dadu pertama sama dengan hasil dadu kedua.

Line 121 Toast.makeText(context, "Selamat anda dapat dadu double!",
 Toast.LENGTH_SHORT).show(): Membuat pesan toast yang memberi tahu
 pengguna bahwa mereka mendapat dadu double.

Line 122 else { ... }: Memulai blok kode yang akan dijalankan jika hasil dadu pertama tidak
 sama dengan hasil dadu kedua.

Line 123 Toast.makeText(context, "Anda belum
 beruntung!", Toast.LENGTH_SHORT).show(): Membuat pesan toast yang memberi
 tahu pengguna bahwa mereka belum beruntung.

Line 124 }

Line 125 }

Line 126)

Line 127 Text(text = stringResource(R.string.roll), fontSize = 24.sp): Menampilkan teks
 "Roll" di atas tombol. stringResource(R.string.roll) digunakan untuk mendapatkan
 teks dari sumber daya string yang didefinisikan dalam file resource strings.xml.
 fontSize = 24.sp mengatur ukuran font teks menjadi 24sp.

Line 128 }

Line 129 }

Line 130 }

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/dheaaprilinda01/pemrogramanMobile/tree/main/Modul%202>