**LAPORAN PRAKTIKUM**

**PEMROGRAMAN MOBILE**

**MODUL 1**

****

**ANDROID BASIC WITH KOTLIN**

**Oleh:**

**Bi’ahlil Haq Aulia Akbar Awaludin NIM. 2010817110011**

**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 : Bi’ahlil Haq Aulia Akbar Awaludin

NIM : 2010817110011

| Menyetujui,  Asisten Praktikum  Muhammad Raka Azwar  NIM. 2210817210012 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Andreyan Rizky Baskara, S.Kom., M.Kom.  NIP. 19930703 201903 01 011 |
| --- | --- |

# DAFTAR ISI

[LEMBAR PENGESAHAN 1](#_heading=h.es4b5m3cj30r)

[DAFTAR ISI 2](#_heading=h.1kpbtj5mlzz9)

[DAFTAR GAMBAR 3](#_heading=h.e5rd16w1jcbq)

[DAFTAR TABEL 4](#_heading=h.3w5liagl36v0)

[SOAL 1 5](#_heading=h.k2qoi6g7td89)

[A. Source Code 8](#_heading=h.f62nqc8hqm29)

[Compose 8](#_heading=h.g6i7azjox3ld)

[XML 10](#_heading=h.im5ja0a85r7v)

[B. Output Program 14](#_heading=h.lgqcrdp8ffdz)

[C. Pembahasan 15](#_heading=h.wfvkdc6dsxib)

[Compose 15](#_heading=h.la6dck8fg7d1)

[XML 15](#_heading=h.qw6npw73b4vz)

[D. Tautan Git 16](#_heading=h.qwaannqzhw6j)

# DAFTAR GAMBAR

[Gambar 1. Screenshot Hasil Jawaban Soal 1 Compose 1](#_heading=h.hshgvq6t1o5k)4

[Gambar 2. Screenshot Hasil Jawaban Soal 1 XML 1](#_heading=h.hshgvq6t1o5k)4

# DAFTAR TABEL

[Tabel 1. Source Code Jawaban Soal 1](#_heading=h.g6i7azjox3ld)  8

[Tabel 2. Source Code Jawaban Soal 1 10](#_heading=h.im5ja0a85r7v)

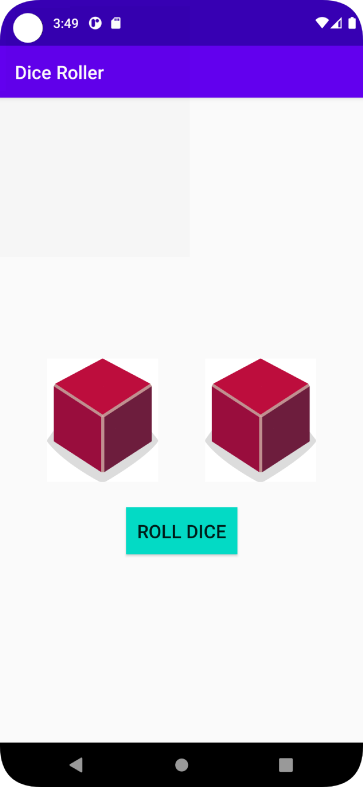
[Tabel 3. Source Code Jawaban Soal 1](#_heading=h.im5ja0a85r7v) 12

# SOAL 1

**Soal Praktikum:**

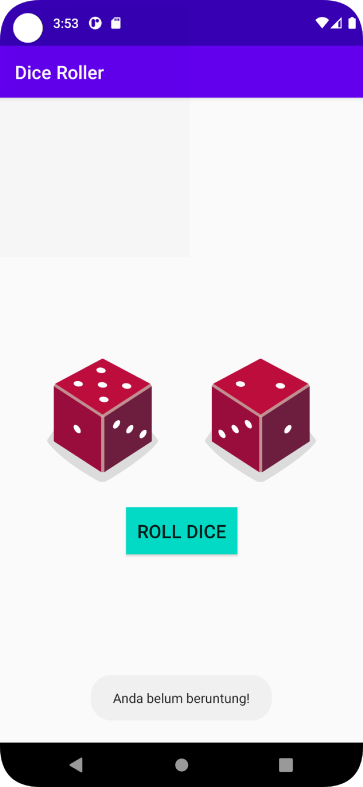
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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



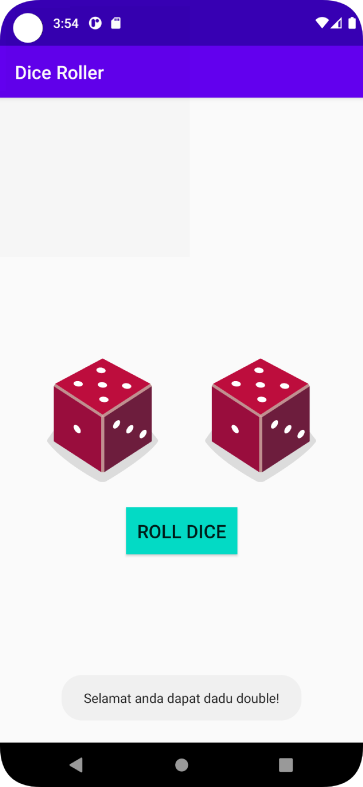
**Gambar 1 Tampilan Awal Aplikasi**

1. Setelah user menekan tombol “Roll” maka masing-masing dadu akan memperlihatkan sisi dadunya dengan angka antara 1 s/d 6. Apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2, maka aplikasi akan menampilkan pesan “Anda belum beruntung!” seperti yang dapat dilihat pada Gambar 2.



**Gambar 2 Tampilan Dadu Setelah Di Roll**

1. 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**

1. Buatlah aplikasi tersebut menggunakan XML dan Jetpack Compose.

1. Upload aplikasi yang telah anda buat ke dalam repository GitHub ke dalam **folder Modul 1 dalam bentuk Project.** Jangan lupa untuk melakukan **Clean Project** sebelum mengupload pekerjaan anda pada repository.

Untuk gambar dadu dapat didownload pada link berikut: [https://drive.google.com/file/d/14V3qXGdFnuoYN4AGd\_9SgFh8kw8X9ySm/view?usp= sharing](https://drive.google.com/file/d/14V3qXGdFnuoYN4AGd_9SgFh8kw8X9ySm/view?usp=)

## Source Code

### 

### Compose

**MainActivity.kt**

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117 | package com.example.diceroller  import android.os.Bundle  import androidx.activity.ComponentActivity  import androidx.activity.compose.setContent  import androidx.activity.enableEdgeToEdge  import androidx.compose.foundation.Image  import androidx.compose.foundation.background  import androidx.compose.foundation.layout.Arrangement  import androidx.compose.foundation.layout.Column  import androidx.compose.foundation.layout.Row  import androidx.compose.foundation.layout.Spacer  import androidx.compose.foundation.layout.fillMaxSize  import androidx.compose.foundation.layout.fillMaxWidth  import androidx.compose.foundation.layout.height  import androidx.compose.foundation.layout.padding  import androidx.compose.foundation.layout.wrapContentSize  import androidx.compose.material3.BottomAppBar  import androidx.compose.material3.Button  import androidx.compose.material3.MaterialTheme  import androidx.compose.material3.Scaffold  import androidx.compose.material3.Surface  import androidx.compose.material3.Text  import androidx.compose.runtime.Composable  import androidx.compose.runtime.getValue  import androidx.compose.runtime.mutableIntStateOf  import androidx.compose.runtime.mutableStateOf  import androidx.compose.runtime.remember  import androidx.compose.runtime.setValue  import androidx.compose.ui.Alignment  import androidx.compose.ui.Modifier  import androidx.compose.ui.graphics.Color  import androidx.compose.ui.res.painterResource  import androidx.compose.ui.res.stringResource  import androidx.compose.ui.tooling.preview.Preview  import androidx.compose.ui.unit.dp  import com.example.diceroller.ui.theme.DiceRollerTheme  class MainActivity : ComponentActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  enableEdgeToEdge()  setContent {  DiceRollerTheme {  Surface(  modifier = Modifier.fillMaxSize(),  ) {  DiceRollerApp()  }  }  }  }  }  @Preview  @Composable  fun DiceRollerApp() {  DiceWithButtonAndImage(modifier = Modifier  .fillMaxSize()  .wrapContentSize(Alignment.Center)  .background(MaterialTheme.colorScheme.background)  )  }  @Composable  private fun DiceWithButtonAndImage(modifier: Modifier = Modifier) {  var dice1 by remember { mutableIntStateOf(0) }  var dice2 by remember { mutableIntStateOf(0) }  val imageResource1 = when (dice1) {  0 -> R.drawable.dice\_0  1 -> R.drawable.dice\_1  2 -> R.drawable.dice\_2  3 -> R.drawable.dice\_3  4 -> R.drawable.dice\_4  5 -> R.drawable.dice\_5  else -> R.drawable.dice\_6  }  val imageResource2 = when (dice2) {  0 -> R.drawable.dice\_0  1 -> R.drawable.dice\_1  2 -> R.drawable.dice\_2  3 -> R.drawable.dice\_3  4 -> R.drawable.dice\_4  5 -> R.drawable.dice\_5  else -> R.drawable.dice\_6  }  Column(  modifier = Modifier,  horizontalAlignment = Alignment.CenterHorizontally  ) {  Spacer(modifier = Modifier.weight(1f))  Row {  Image(painter = painterResource(imageResource1), contentDescription = dice1.toString())  Image(painter = painterResource(imageResource2), contentDescription = dice2.toString())  }  Spacer(modifier = Modifier.height(22.dp))  Button(onClick = {  dice1 = (1..6).random()  dice2 = (1..6).random()  }  ) { Text(stringResource(R.string.roll)) }  Spacer(modifier = Modifier.weight(1f))  if (dice1 != 0 || dice2 != 0)  Text(  stringResource(  if (dice1 == dice2) R.string.twins else R.string.zonk  ),  color = MaterialTheme.colorScheme.background,  style = MaterialTheme.typography.labelMedium,  modifier = Modifier  .fillMaxWidth()  .background(MaterialTheme.colorScheme.primary)  .padding(8.dp)  )  }  } |
| --- | --- |

*Tabel 1. Source Code Jawaban Soal 1*

### XML

**MainActivity.kt**

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70 | package com.example.dicerollerxml  import android.os.Bundle  import android.widget.Button  import android.widget.ImageView  import android.widget.TextView  import android.widget.Toast  import androidx.activity.enableEdgeToEdge  import androidx.appcompat.app.AppCompatActivity  import androidx.core.view.ViewCompat  import androidx.core.view.WindowInsetsCompat  class MainActivity : AppCompatActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  enableEdgeToEdge()  setContentView(R.layout.activity\_main)  val diceImage1: ImageView = findViewById(R.id.dice1)  val diceImage2: ImageView = findViewById(R.id.dice2)  val rollButton: Button = findViewById(R.id.rollButton)  val bottomText: TextView = findViewById(R.id.messageText)  rollButton.setOnClickListener {  val dice1 = (1..6).random()  val dice2 = (1..6).random()  val drawable1 = when (dice1) {  1 -> R.drawable.dice\_1  2 -> R.drawable.dice\_2  3 -> R.drawable.dice\_3  4 -> R.drawable.dice\_4  5 -> R.drawable.dice\_5  else -> R.drawable.dice\_6  }  val drawable2 = when (dice2) {  1 -> R.drawable.dice\_1  2 -> R.drawable.dice\_2  3 -> R.drawable.dice\_3  4 -> R.drawable.dice\_4  5 -> R.drawable.dice\_5  else -> R.drawable.dice\_6  }  diceImage1.setImageResource(drawable1)  diceImage2.setImageResource(drawable2)  bottomText.apply {  text = getString(  if (dice1 == dice2) R.string.twins else R.string.zonk  )  setBackgroundColor(getColor(R.color.purple\_500))  setTextColor(getColor(R.color.white))  setPadding(16, 16, 16, 16)  visibility = TextView.VISIBLE  }  }  ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->  val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  insets  }  }  } |
| --- | --- |

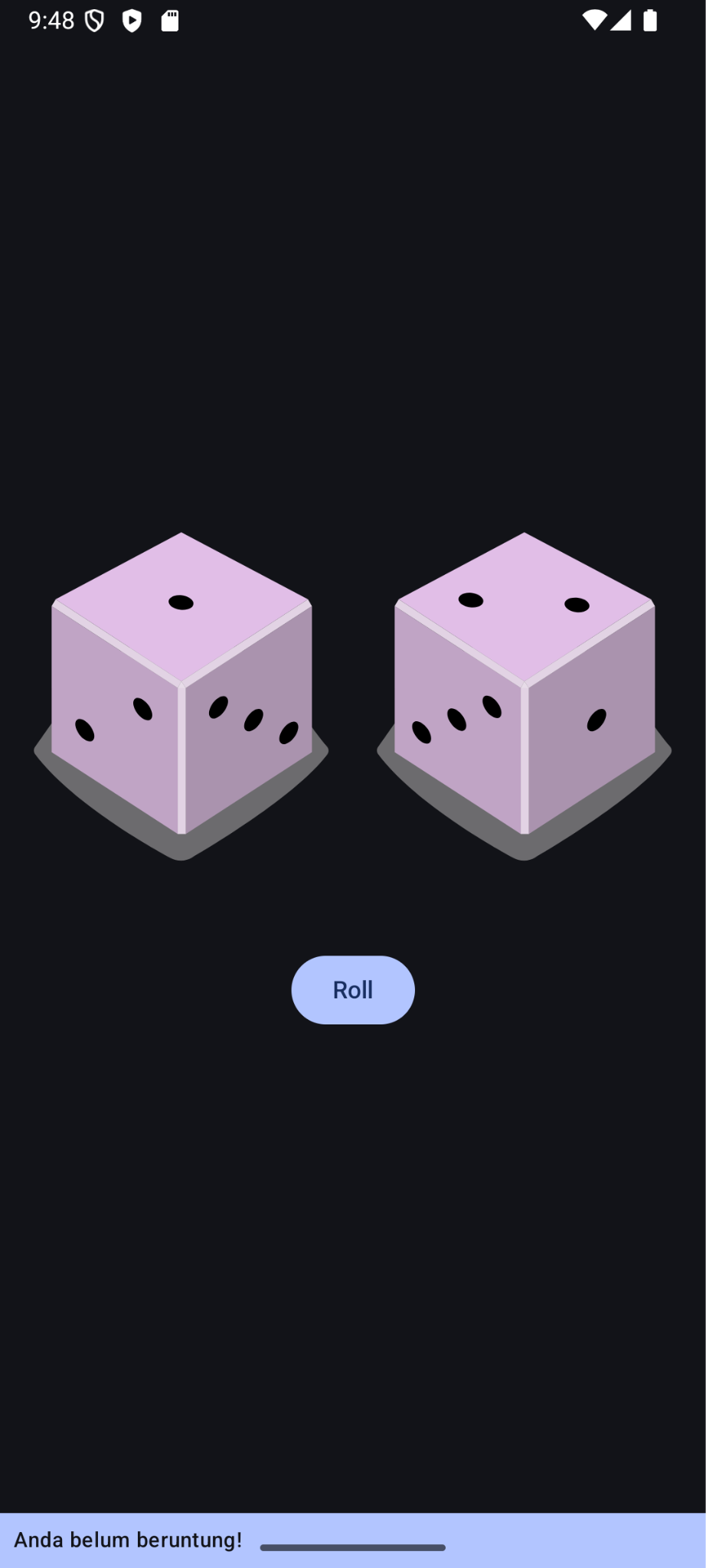
*Tabel 2. Source Code Jawaban Soal 1*

**activity\_main.xml**

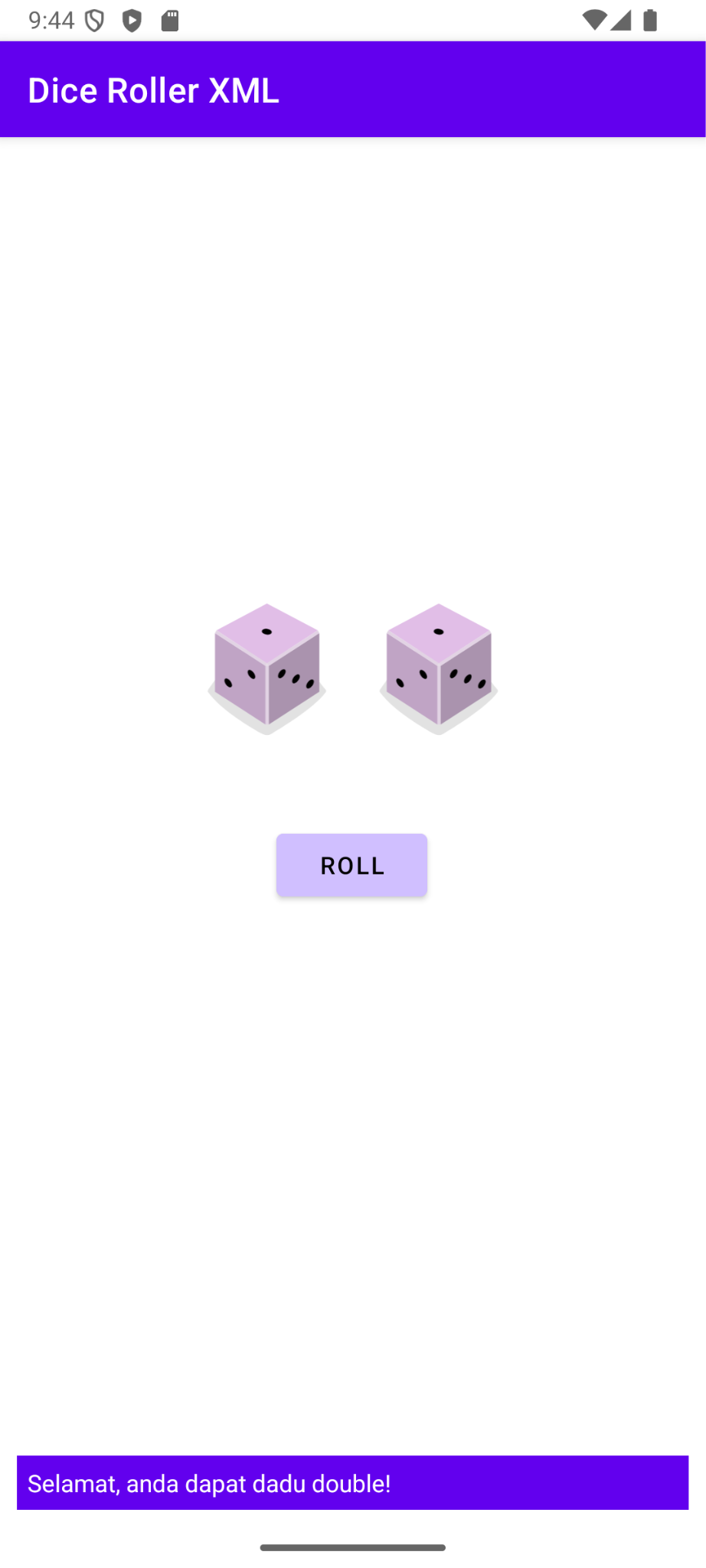
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 | <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:id="@+id/main"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:contentDescription="@string/dice2"  tools:context=".MainActivity"  tools:layout\_editor\_absoluteX="10dp"  tools:layout\_editor\_absoluteY="32dp">  <ImageView  android:id="@+id/dice1"  android:layout\_width="100dp"  android:layout\_height="100dp"  android:layout\_marginTop="260dp"  android:contentDescription="@string/dice1"  android:src="@drawable/dice\_0"  app:layout\_constraintEnd\_toStartOf="@+id/dice2"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintHorizontal\_chainStyle="packed"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintBottom\_toTopOf="@+id/rollButton"  app:layout\_constraintVertical\_bias="0.0"  />  <ImageView  android:id="@+id/dice2"  android:layout\_width="100dp"  android:layout\_height="100dp"  android:contentDescription="@string/dice2"  android:src="@drawable/dice\_0"  app:layout\_constraintBottom\_toBottomOf="@+id/dice1"  app:layout\_constraintBottom\_toTopOf="@+id/rollButton"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_chainStyle="packed"  app:layout\_constraintStart\_toEndOf="@+id/dice1"  app:layout\_constraintTop\_toTopOf="@+id/dice1"  app:layout\_constraintVertical\_bias="0.0" />  <Button  android:id="@+id/rollButton"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="40dp"  android:backgroundTint="#D0BFFF"  android:text="@string/roller"  android:textColor="#000000"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.498"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@id/dice1"  />  <TextView  android:id="@+id/messageText"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_margin="10dp"  android:layout\_marginBottom="8dp"  android:background="#FFFFFF"  android:padding="12dp"  android:text="@string/zonk"  android:textColor="#000000"  android:visibility="invisible"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="1.0"  app:layout\_constraintStart\_toStartOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> |
| --- | --- |

*Tabel 3. Source Code Jawaban Soal 1*

## Output Program



*Gambar 1. Screenshot Hasil Jawaban Soal 1 Compose*



*Gambar 2. Screenshot Hasil Jawaban Soal 1 XML*

## Pembahasan

### Compose

**MainActivity.kt:**

Pada line 44, dideklarasikan theme default yang dibuat oleh compose secara otomatis, didalamnya akan dipanggil fungsi dicerollerapp yang membungkus keseluruhan ui

Pada line 57, dideklarasikan preview sekaligus fungsi dicerollerapp yang membungkus keseluruhan ui

Pada line 65, dideklarasikan fungsi DiceWithButtonAndImage untuk ui image dice dan buttonnya

Pada line 68 - 86, terdapat logic untuk mengatur source dari image painter sesuai dengan state nya, disini juga dideklarasikan variabel untuk menyimpan state dice1 dan dice2

Pada line 67, diimport Column layout milik compose untuk mengatur image dice dan button

Pada line 92, diimport Row layout milik compose untuk mengatur image dice 1 dan dice 2 sehingga bisa bersebelahan

Pada line 103, terdapat logic untuk menampilkan Text milik compose jika state dari dice 1 dan dice 2 berubah

Pada line 105, terdapat logic untuk mengubah stringresource Text milik compose jika value dari dice 1 dan dice 2 sama

### XML

**MainActivity.kt:**

Pada line 7, diimport AppCompatActivity untuk base class activity yang mendukung versi terdahulu Android

Pada line 17, diimport fungsi setContentView untuk menjadikan activity\_main sebagai layout tujuan

Pada line 18 - 23, dideklarasikan variable yang merujuk pada setiap komponen ui yang ada pada layout

Pada line 25, terdapat fungsi yang memuat logic apabila button di klik

Pada line 26 - 45, dideklarasikan variable untuk menyimpan nilai dadu dan logic untuk mengubah tampilan image dadu sesuai dengan nilai dari dadu

Pada line 47 - 58, dipanggil scope fungsi pada bottomText untuk memberikan logic dimana text tidak akan muncul jika button tidak diklik, dan text akan berubah jika dadu 1 dan 2 memiliki nilai sama atau tidak

**activity\_main.xml:**

Pada line 15 - 31, dideklarasikan komponen imageview untuk dadu 1, dengan constraint endtostartof ke dadu 2 dan juga constraintHorizontal\_chainStyle packed

Pada line 33 - 48, dideklarasikan komponen imageview untuk dadu 2, dengan constraint Bottom\_toBottomOf ke dice1, Bottom\_toTopOf ke rollButton, Horizontal\_chainStyle packed

Pada line 50 - 64, dideklarasikan komponen button untuk rollButton, dengan constraint Top\_toBottomOf ke dice1

Pada line 66 - 80, dideklarasikan komponen textview untuk messageText, dengan visibility invisible, dan constraint Bottom\_toBottomOf ke parent, End\_toEndOf ke parent, Start\_toStartOf ke parent

## Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/biahlil/Pemrograman-Mobile.git>