**LAPORAN PRAKTIKUM**

**PEMROGRAMAN MOBILE**

**MODUL 1**

****

**ANDROID BASIC WITH KOTLIN**

**Oleh:**

**Muhammad Bukhari Fitri NIM. 2310817210015**

**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 : Muhammad Bukhari Fitri

NIM : 2310817210015

|  |  |
| --- | --- |
| 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 2](#_Toc166105938)

[DAFTAR ISI 3](#_Toc166105939)

[DAFTAR GAMBAR 4](#_Toc166105940)

[DAFTAR TABEL 5](#_Toc166105941)

[SOAL 1 6](#_Toc166105942)

[A. Source Code 8](#_Toc166105943)

[B. Output Program 11](#_Toc166105944)

[C. Pembahasan 11](#_Toc166105945)

[D. Tautan Git 13](#_Toc166105946)

# DAFTAR GAMBAR

[Gambar 1. Screenshot Hasil Jawaban Soal 1](#_Toc166098723) [11](#_Toc166098723)

# DAFTAR TABEL

[Tabel 1. Source Code Jawaban Soal 1 9](#_Toc166105944)

[Tabel 2. Source Code Jawaban Soal 1 10](#_Toc166105944)

# SOAL 1

**Soal Praktikum:**

Buatlah sebuah aplikasi yang dapat menampilkan 2 buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll”. 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 yang dapat dilihat pada Gambar 3.



Gambar 3. Tampilan Roll Dadu Double

1. Buatlah aplikasi tersebut menggunakan XML dan Jetpack Compose.
2. 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.
3. Untuk gambar dadu dapat didownload pada link berikut:

<https://drive.google.com/file/d/14V3qXGdFnuoYN4AGd_9SgFh8kw8X9ySm/view?usp=sharing>

## Source Code

**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 | package com.example.dicexml  import android.os.Bundle  import androidx.activity.enableEdgeToEdge  import androidx.appcompat.app.AppCompatActivity  import com.example.dicexml.databinding.ActivityMainBinding  class MainActivity : AppCompatActivity() {  private lateinit var binding: ActivityMainBinding  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  enableEdgeToEdge()  binding = ActivityMainBinding.inflate(layoutInflater)  setContentView(binding.root)  binding.button.setOnClickListener {  diceRollerApp(binding)  }  }  } |

Tabel 4. Source Code Jawaban Soal 1

**DiceRollerApp.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 | package com.example.dicexml  import com.example.dicexml.databinding.ActivityMainBinding  import com.google.android.material.snackbar.Snackbar  fun diceRollerApp(binding: ActivityMainBinding) {  val result1 = randomDice()  val result2 = randomDice()  val imageResource1 = getDiceImage(result1)  val imageResource2 = getDiceImage(result2)  binding.imageView.setImageResource(imageResource1)  binding.imageView2.setImageResource(imageResource2)  if (result1 == result2) {  Snackbar.make(binding.button, "Selamat, anda dapat dadu double", Snackbar.LENGTH\_LONG).show()  }  else if (result1 != result2) {  Snackbar.make(binding.button, "Anda belum beruntung!", Snackbar.LENGTH\_LONG).show()  }  }  fun randomDice() : Int {  return (1..6).random()  }  fun getDiceImage(result: Int) : Int {  return when (result) {  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  }  } |

**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 | <?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:background="#2B2A2A"  tools:context=".MainActivity">  <Button  android:id="@+id/button"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="162dp"  android:layout\_marginTop="450dp"  android:layout\_marginEnd="162dp"  android:layout\_marginBottom="234dp"  android:backgroundTint="#D460ED"  android:text="@string/roll"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />  <ImageView  android:id="@+id/imageView"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="171dp"  android:layout\_marginEnd="4dp"  android:layout\_marginBottom="29dp"  android:contentDescription="@string/image\_description"  app:layout\_constraintBottom\_toTopOf="@+id/button"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toEndOf="@+id/imageView2"  app:layout\_constraintTop\_toTopOf="parent"  app:srcCompat="@drawable/dice\_0" />  <ImageView  android:id="@+id/imageView2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="171dp"  android:layout\_marginBottom="29dp"  android:contentDescription="@string/image\_description"  app:layout\_constraintBottom\_toTopOf="@+id/button"  app:layout\_constraintEnd\_toStartOf="@+id/imageView"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:srcCompat="@drawable/dice\_0" />  </androidx.constraintlayout.widget.ConstraintLayout> |

Tabel 2. Source Code Jawaban Soal 1

**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 | package com.example.dicecompose  import android.os.Bundle  import androidx.activity.ComponentActivity  import androidx.activity.compose.setContent  import androidx.activity.enableEdgeToEdge  import androidx.compose.foundation.layout.fillMaxSize  import androidx.compose.material3.Scaffold  import androidx.compose.ui.Modifier  import com.example.dicecompose.component.DiceRollerApp  import com.example.dicecompose.ui.theme.DiceComposeTheme  class MainActivity : ComponentActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  enableEdgeToEdge()  setContent {  DiceComposeTheme {  Scaffold( modifier = Modifier.fillMaxSize() )  { innerPadding ->  DiceRollerApp()  }  }  }  }  } |

Tabel 4. Source Code Jawaban Soal 1

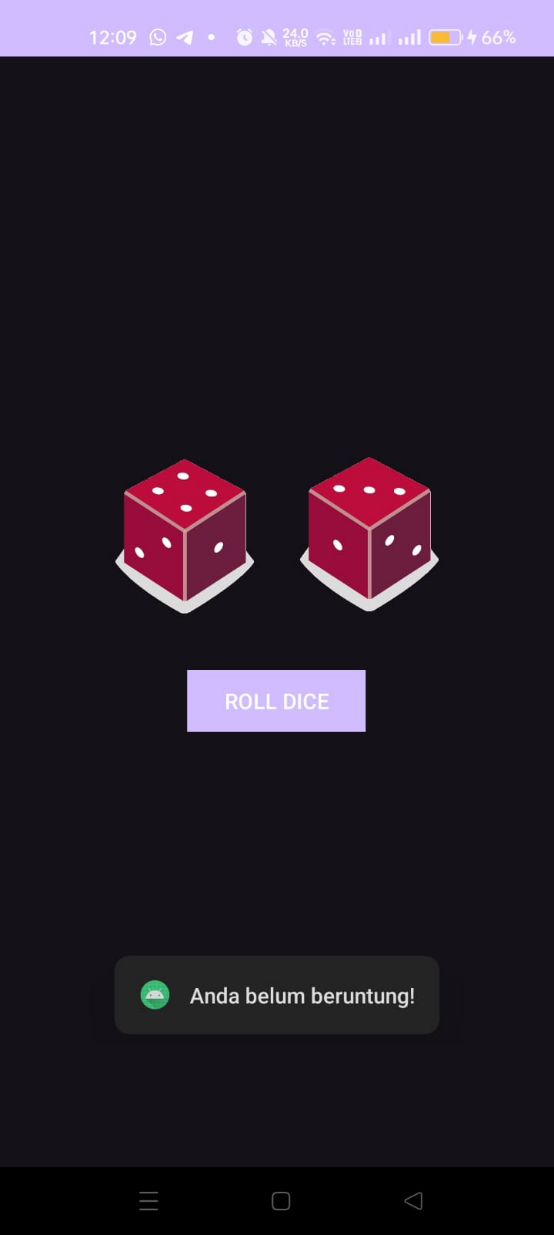
**DiceRollerApp.kt**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | package com.example.dicecompose.component  import androidx.compose.foundation.layout.\*  import androidx.compose.runtime.Composable  import androidx.compose.ui.Alignment  import androidx.compose.ui.Modifier  import androidx.compose.ui.tooling.preview.Preview  @Composable  fun DiceRollerApp() {  DiceWithButtonAndImage(  modifier = Modifier  .fillMaxSize()  .wrapContentSize(Alignment.Center)  )  }  @Preview(showBackground = true)  @Composable  fun DiceRollerAppPreview() {  DiceRollerApp()  } |

**DiceWithButtonAndImage.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 | package com.example.dicecompose.component  import androidx.compose.foundation.Image  import androidx.compose.foundation.layout.Column  import androidx.compose.foundation.layout.Row  import androidx.compose.foundation.layout.Spacer  import androidx.compose.foundation.layout.height  import androidx.compose.material3.Button  import androidx.compose.material3.Scaffold  import androidx.compose.material3.SnackbarDuration  import androidx.compose.material3.SnackbarHost  import androidx.compose.material3.SnackbarHostState  import androidx.compose.material3.Text  import androidx.compose.runtime.Composable  import androidx.compose.runtime.getValue  import androidx.compose.runtime.mutableStateOf  import androidx.compose.runtime.remember  import androidx.compose.runtime.rememberCoroutineScope  import androidx.compose.runtime.setValue  import androidx.compose.ui.Alignment  import androidx.compose.ui.Modifier  import androidx.compose.ui.res.painterResource  import androidx.compose.ui.res.stringResource  import androidx.compose.ui.unit.dp  import com.example.dicecompose.R  import kotlinx.coroutines.launch  @Composable  fun DiceWithButtonAndImage(modifier: Modifier = Modifier) {  var result1 by remember { mutableStateOf(0) }  var result2 by remember { mutableStateOf(0) }  val imageResource1 = getDiceImage(result1)  val imageResource2 = getDiceImage(result2)  val scope = rememberCoroutineScope()  val snackbarHostState = remember { SnackbarHostState()}  Scaffold(  snackbarHost = { SnackbarHost(hostState = snackbarHostState) }  ) {  paddingValues ->  Column (  modifier = modifier,  horizontalAlignment = Alignment.CenterHorizontally  ) {  Row {  Image(  painter = painterResource(imageResource1),  contentDescription = result1.toString()  )  Image(  painter = painterResource(imageResource2),  contentDescription = result2.toString()  )  }  Spacer(modifier = Modifier.height(12.dp))  Button(onClick = {  result1 = (1..6).random()  result2 = (1..6).random()  scope.launch { snackbarHostState.currentSnackbarData?.dismiss()  val message = if (result1 == result2) {  "Selamat, anda dapat dadu double!"  } else {  "Anda belum beruntung!"  }  snackbarHostState.showSnackbar(  message,  duration = SnackbarDuration.Short  )  }  }) {  Text(stringResource(R.string.roll))  } }  }  }  fun getDiceImage(result: Int) : Int {  return when (result) {  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  }  } |

## Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

## Pembahasan

**MainActivity.kt:**

Pada line 1, dideklarasikan nama package file Kotlin

Pada line 3, diimport Bundle untuk mengkomunikasikan data antar aktivitas

Pada line 4, diimport Button untuk mengatur Button dalam kode

Pada line 5, diimport ImageView untuk mengatur dan menunjukkan gambar pada ImageView dalam kode

Pada line 6, diimport Toast untuk menunjukkan pop-up message yang bersifat singkat

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

Pada line 9, dideklarasikan kelas bernama MainActivity yang extend dari AppCompatActivity

Pada line 10, terdapat variabel integer firstDice yang pada awalnya akan di-set sebagai null

Pada line 11, terdapat override pada fungsi onCreate yang dipanggil saat activity dimulai

Pada line 12, akan dipanggil superclass dari onCreate

Pada line 13, content view dari aktivitas ini berasal dari file XML “activity\_main.xml”

Pada line 15, dideklarasikan variabel rollButton yang merupakan Button dengan id “button”

Pada line 16, akan diatur click listener yang akan berfungsi saat rollButton ditekan

Pada line 17, fungsi rollDice() dipanggil

Pada line 18, fungsi rollDice2() dipanggil

Pada line 22, dideklarasikan fungsi private rollDice

Pada line 23, dideklarasikan variabel dice yang merupakan Dice dengan nilai 6

Pada line 24, dideklarasikan variabel diceRoll yang akan memanggil variabel dice ke dalam fungsi roll

Pada line 25, dideklarasikan variabel diceImage yang merupakan ImageView dengan id imageView

Pada line 26, dideklarasikan variabel drawableResource dengan kondisional when diceRoll

Pada line 27, jika nilai sama dengan 1, maka akan dipanggil gambar dice\_1 dari drawable

Pada line 28, jika nilai sama dengan 2, maka akan dipanggil gambar dice\_2 dari drawable Pada line 29, jika nilai sama dengan 3, maka akan dipanggil gambar dice\_3 dari drawable Pada line 30, jika nilai sama dengan 4, maka akan dipanggil gambar dice\_4 dari drawable

Pada line 31, jika nilai sama dengan 5, maka akan dipanggil gambar dice\_5 dari drawable

Pada line 32, jika nilai bukan dari angka di atas, maka akan dipanggil gambar dice\_6 dari drawable

Pada line 34, akan diatur nilai diceImage yang nantinya akan ditunjukkan di dalam imageView sebagai gambar yang terpilih di atas

Pada line 35, variabel firstDice bernilai setara dengan nilai variabel diceRoll

Pada line 38 dideklarasikan fungsi private rollDice2

Pada line 39, dideklarasikan variabel dice yang merupakan Dice dengan nilai 6

Pada line 40, dideklarasikan variabel diceRoll yang akan memanggil variabel dice ke dalam fungsi roll

Pada line 41, dideklarasikan variabel diceImage yang merupakan ImageView dengan id imageView

Pada line 42, dideklarasikan variabel drawableResource dengan kondisional when diceRoll

Pada line 43, jika nilai sama dengan 1, maka akan dipanggil gambar dice\_1 dari drawable

Pada line 44, jika nilai sama dengan 2, maka akan dipanggil gambar dice\_2 dari drawable

Pada line 45, jika nilai sama dengan 3, maka akan dipanggil gambar dice\_3 dari drawable Pada line 46, jika nilai sama dengan 4, maka akan dipanggil gambar dice\_4 dari drawable

Pada line 47, jika nilai sama dengan 5, maka akan dipanggil gambar dice\_5 dari drawable

Pada line 48, jika nilai bukan dari angka di atas, maka akan dipanggil gambar dice\_6 dari drawable

Pada line 50, akan diatur nilai diceImage yang nantinya akan ditunjukkan di dalam imageView2 sebagai gambar yang terpilih di atas

Pada line 52, terdapat kondisional if, di mana jika diceRoll pada fungsi nilainya sama dengan variabel firstDice

Pada line 53, akan mengeksekusi kondisional di atas, dengan mengeluarkan pesan “Selamat, anda dapat dadu double!” menggunakan Toast.

Pada line 55, terdapat kondisional else

Pada line 56, akan mengeksekusi kondisional di atas, dengan mengeluarkan pesan “Anda belum beruntung!” menggunakan Toast.

Pada line 61, terdapat kelas Dice dengan variabel private numSides yang nilainya integer

Pada line 62, terdapat fungsi roll yang nilainya integer

Pada line 63, fungsi roll akan mengembalikan hasil random antara 1 dan numSides

**activity\_main.xml:**

Pada line 1, terdapat versi XML dan encoding yang digunakan

Pada line 2, dideklarasikan root dari file layout, yakni ConstraintLayout dan namespace XML untuk atribut Android

Pada line 3, dideklarasikan namespace untuk atribut kustom

Pada line 4, dideklarasikan namespace untuk atribut tools dalam aplikasi

Pada line 5, terdapat id dari ConstraintLayout, yakni main

Pada line 6, lebar ConstraintLayout diatur untuk match dengan parent

Pada line 7, tinggi ConstraintLayout diatur untuk match dengan parent

Pada line 8, berisi context ConstraintLayout sebagai “.MainActivity”, yang berhubungan dengan file MainActivity.kt

Pada line 10, terdapat tag deklarasi element Button

Pada line 11, terdapat id dari Button, yakni main

Pada line 12, lebar Button mengikuti besar konten di dalamnya

Pada line 13, tinggi Button mengikuti besar konten di dalamnya

Pada line 14, didefinisikan start margin Button sebesar 147dp

Pada line 15, didefinisikan end margin Button sebesar 148dp

Pada line 16, didefinisikan bottom margin Button sebesar 278dp

Pada line 17, didefinisikan teks di dalam Button mengikuti string resource “roll”

Pada line 18, didefinisikan warna teks Button sebagai “#FFFFFF” atau putih

Pada line 19, didefinsikan radius ujung Button sebesar 0dp, yang membuat Button terlihat seperti persegi panjang.

Pada line 20, layout constraint ujung bawah Button adalah parent atau ConstraintLayout

Pada line 21, layout constraint ujung akhir Button adalah parent atau ConstraintLayout

Pada line 22, layout constraint ujung awal Button adalah parent atau ConstraintLayout

Pada line 23, dideklarasikan agar tools menghiraukan check pada text color di Button

Pada line 25, terdapat tag deklarasi element ImageView

Pada line 26, terdapat id dari ImageView, yakni imageView

Pada line 27, lebar ImageView adalah sebesar 100sp

Pada line 28, tinggi ImageView adalah sebesar 100sp

Pada line 29, didefinisikan start margin ImageView sebesar 70dp

Pada line 30, didefinisikan top margin ImageView sebesar 268dp

Pada line 31, didefinisikan bottom margin ImageView sebesar 40dp

Pada line 32, layout constraint ujung bawah ImageView adalah Button di atas

Pada line 33, layout constraint ujung akhir ImageView adalah parent atau ConstraintLayout

Pada line 34, layout constraint ujung awal ImageView adalah parent atau ConstraintLayout

Pada line 35, akan diatur gambar empty\_dice sebagai gambar yang tertampil di ImageView

Pada line 37, terdapat tag deklarasi element ImageView

Pada line 38, terdapat id dari ImageView, yakni imageView2

Pada line 39, lebar ImageView adalah sebesar 100sp

Pada line 40, tinggi ImageView adalah sebesar 100sp

Pada line 41, didefinisikan top margin ImageView sebesar 265dp

Pada line 42, didefinisikan end margin ImageView sebesar 70dp

Pada line 43, didefinisikan bottom margin ImageView sebesar 40dp

Pada line 44, layout constraint ujung bawah ImageView adalah Button di atas

Pada line 45, layout constraint ujung akhir ImageView adalah parent atau ConstraintLayout Pada line 46, layout constraint ujung atas ImageView adalah parent atau ConstraintLayout

Pada line 47, akan diatur gambar empty\_dice sebagai gambar yang tertampil di ImageView

Pada line 49 menjadi penutup dari tag elemen root ConstraintLayout

## Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/awan-ama/praktikummobile/tree/main/DiceRoller