**LAPORAN PRAKTIKUM**

**PEMROGRAMAN MOBILE**

**MODUL 1**

****

**ANDROID BASIC WITH KOTLIN**

**Oleh:**

**Allano Lintang Ertantora NIM. 2310817210004**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMBUNG MANGKURAT**

**APRIL 2025**

# 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 : Allano Lintang Ertantora

NIM : 2310817210004

|  |  |
| --- | --- |
| Menyetujui,  Asisten Praktikum  Zulfa Auliya Akbar  NIM. 2210817210026 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Muti`a Maulida S.Kom M.T.I  NIP. 19881027 201903 20 13 |
|  |  |

# DAFTAR ISI

[LEMBAR PENGESAHAN 2](#_Toc195619036)

[DAFTAR ISI 3](#_Toc195619037)

[DAFTAR GAMBAR 4](#_Toc195619038)

[DAFTAR TABEL 5](#_Toc195619039)

[SOAL 1 6](#_Toc195619040)

[A. Source Code 8](#_Toc195619041)

[B. Output Program 11](#_Toc195619042)

[C. Pembahasan 11](#_Toc195619043)

[D. Tautan Git 13](#_Toc195619044)

# DAFTAR GAMBAR

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

# DAFTAR TABEL

Tabel 1. Source Code Jawaban Soal 1 10

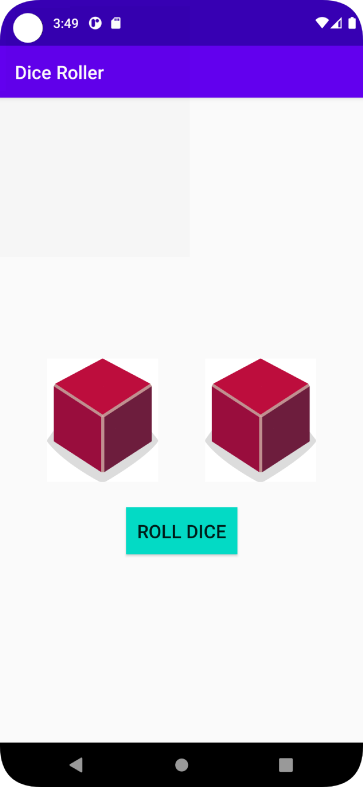
[Tabel 2. Source Code Jawaban Soal 1 10](#_Toc166105944)

# SOAL 1

**Soal Praktikum:**

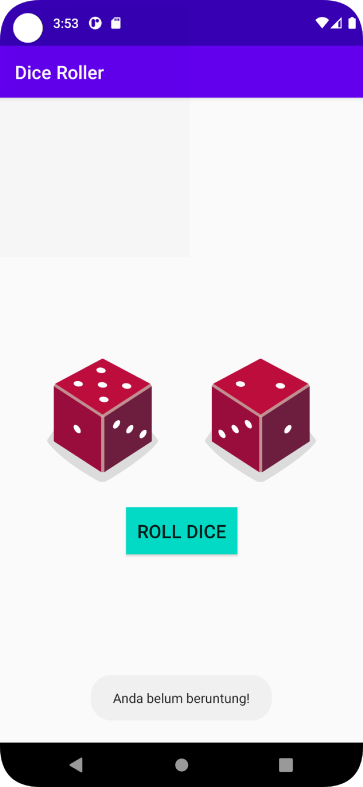
Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

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 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.



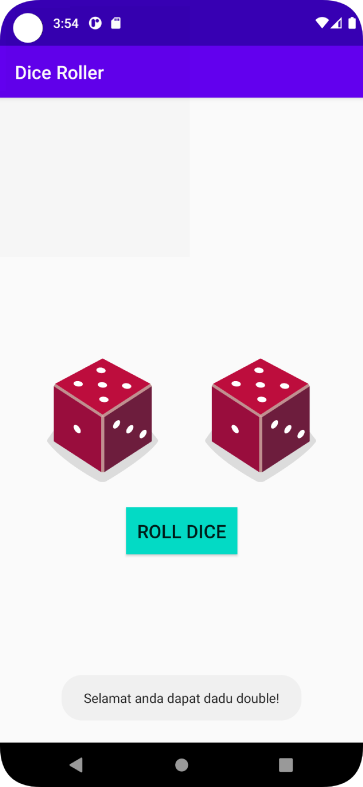
**Gambar 2 Tampilan Dadu Setelah Di Roll**

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.

1. 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.

1. Untuk gambar dadu dapat didownload pada link berikut:

[https://drive.google.com/u/0/uc?id=147HT2lIH5qin3z5ta7H9y2N\_5OMW81Ll&export= download](https://drive.google.com/u/0/uc?id=147HT2lIH5qin3z5ta7H9y2N_5OMW81Ll&export=download)



**Gambar 3 Tampilan Roll Dadu Double**

## Source Code

1. **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 | package com.example.mobileapptest  import android.widget.Toast  import androidx.compose.ui.platform.LocalContext  import android.os.Bundle  import androidx.activity.ComponentActivity  import androidx.activity.compose.setContent  import androidx.activity.enableEdgeToEdge  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.height  import androidx.compose.foundation.layout.padding  import androidx.compose.foundation.layout.wrapContentSize  import androidx.compose.material3.Button  import androidx.compose.material3.Scaffold  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.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.tooling.preview.Preview  import androidx.compose.ui.unit.dp  import com.example.mobileapptest.ui.theme.MobileAppTestTheme  import androidx.compose.foundation.Image as Image2  class MainActivity : ComponentActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  enableEdgeToEdge()  setContent {  MobileAppTestTheme {  DiceRollerApp()  }  }  }  }  @Preview  @Composable  fun DiceRollerApp() {  DiceButtonWithImage()  }  @Composable  fun DiceButtonWithImage(modifier: Modifier = Modifier .fillMaxSize() .wrapContentSize(Alignment.Center)) {  var result1 by remember{mutableStateOf( 0)}  var result2 by remember{mutableStateOf( 0)}  val context = LocalContext.current  val result1Image = getImage(result1)  val result2Image = getImage(result2)  Column (  modifier = modifier,  horizontalAlignment = Alignment.CenterHorizontally  ) {  Row(verticalAlignment = Alignment.CenterVertically, modifier = Modifier){  Image2(  painter = painterResource(id = result1Image),  contentDescription = result1.toString(),  modifier = Modifier.padding(8.dp)  )  Image2(  painter = painterResource(id = result2Image),  contentDescription = result2.toString(),  modifier = Modifier.padding(8.dp)  )  }  Spacer(modifier = Modifier.height(16.dp))  Button(onClick = {  result1 = (1..6).random()  result2 = (1..6).random()  if(result1 == result2){  Toast.makeText(context, "Selamat anda dapat dadu double!", Toast.LENGTH\_SHORT).show()  }else{  Toast.makeText(context, "Anda belum beruntung", Toast.LENGTH\_SHORT).show()  }  }) {  Text(stringResource(R.string.roll))  }  }  }  @Composable  fun getImage(diceValue: Int): Int {  return when (diceValue){  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  6 -> R.drawable.dice\_6  else -> R.drawable.dice\_0  }  }} |

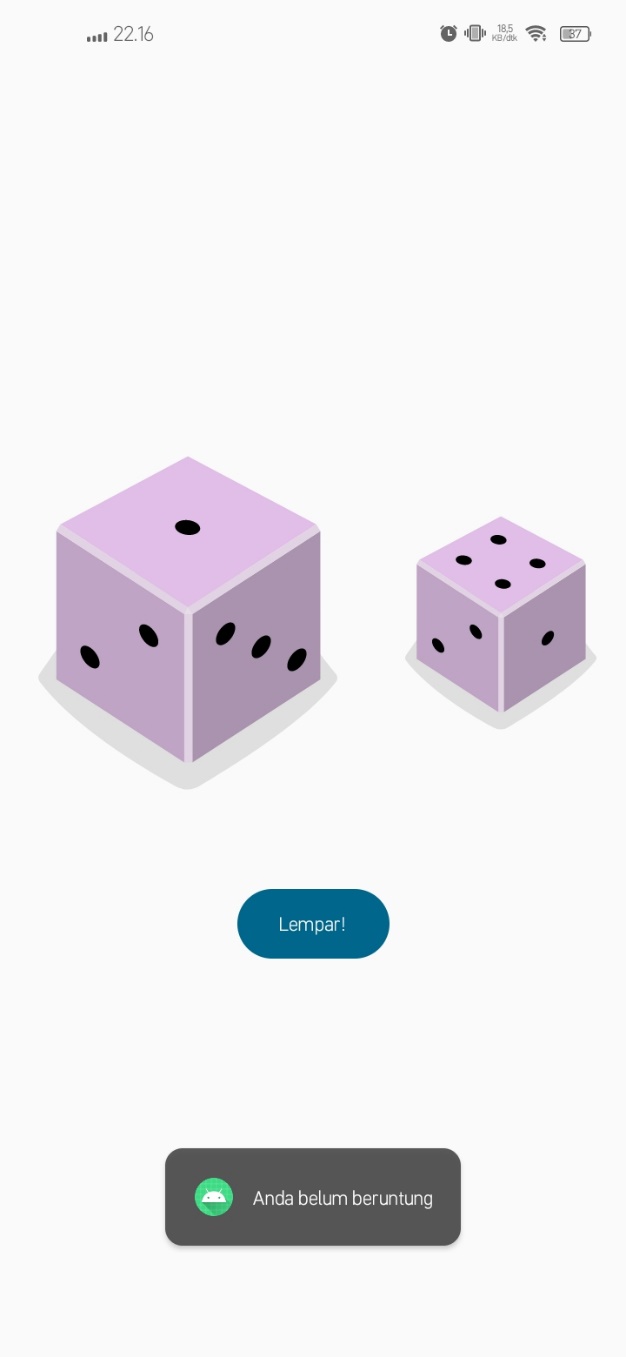
Tabel 1. Source Code Jawaban Soal 1

1. **string.xml**

|  |  |
| --- | --- |
| 1  2  3  4 | <resources>  <string name="app\_name">MobileAppTest</string>  <string name="roll">Lempar!</string>  </resources> |

Tabel 2. Source Code Jawaban Soal 1

## Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

## Pembahasan

1. **MainActivity.kt:**

Pada line 1, dideklarasikan package com.example.mobileapptest sebagai identitas unik dari aplikasi.

Pada line 3, diimport Toast untuk menampilkan notifikasi singkat dalam bentuk pop-up.

Pada line 4, diimport LocalContext untuk mendapatkan konteks aplikasi di dalam komponen Compose.

Pada line 5, diimport Bundle untuk menyimpan dan mengirim data saat lifecycle onCreate.

Pada line 6, diimport ComponentActivity sebagai superclass utama dari MainActivity.

Pada line 7, diimport setContent untuk menampilkan antarmuka pengguna menggunakan Jetpack Compose.

Pada line 8, diimport enableEdgeToEdge agar UI dapat menutupi seluruh area layar.

Pada lines 9–15, diimport berbagai komponen layout seperti Column, Row, Spacer, fillMaxSize, height, padding, dan wrapContentSize untuk mengatur tata letak UI.

Pada lines 16–18, diimport Button, Scaffold, dan Text sebagai elemen UI dari Material 3.

Pada lines 19–23, diimport elemen state Compose seperti @Composable, getValue, mutableStateOf, remember, dan setValue untuk mendukung state management.

Pada lines 26–29, diimport elemen UI tambahan seperti Modifier, painterResource, stringResource, Preview, dan unit dp untuk styling.

Pada line 30, diimport MobileAppTestTheme untuk menerapkan tema khusus aplikasi.

Pada line 31, diimport Image dari foundation dan diberi alias Image2 untuk menghindari konflik nama.

Pada line 33, dideklarasikan kelas MainActivity yang merupakan turunan dari ComponentActivity.

Pada line 34, didefinisikan fungsi onCreate() yang akan dijalankan saat aktivitas dibuat.

Pada line 35, dipanggil super.onCreate() untuk mewarisi logika dasar dari superclass.

Pada line 36, dipanggil enableEdgeToEdge() untuk mengizinkan konten menjangkau batas layar.

Pada line 37, dipanggil setContent {} sebagai awal untuk membangun UI dengan Compose.

Pada line 38, diterapkan tema MobileAppTestTheme agar gaya visual aplikasi konsisten.

Pada line 39, dipanggil fungsi DiceRollerApp() untuk menampilkan UI utama aplikasi.

Pada line 46, digunakan anotasi @Preview untuk menampilkan preview UI di Android Studio.

Pada line 48, didefinisikan fungsi DiceRollerApp() sebagai titik masuk UI, yang memanggil DiceButtonWithImage().

Pada line 54, didefinisikan fungsi composable DiceButtonWithImage() dengan modifier default untuk memenuhi dan memusatkan tampilan.

Pada lines 55–56, dideklarasikan dua variabel state result1 dan result2 dengan nilai awal 0, mewakili hasil dua dadu.

Pada line 57, diambil konteks aplikasi melalui LocalContext.current untuk keperluan Toast.

Pada lines 59–60, dipanggil fungsi getImage() untuk mendapatkan ID gambar berdasarkan nilai result1 dan result2.

Pada line 62, digunakan Column untuk menampilkan komponen secara vertikal.

Pada line 63, modifier diterapkan dan konten disejajarkan ke tengah secara horizontal.

Pada line 66, digunakan Row untuk menampilkan gambar dadu secara horizontal.

Pada lines 67–71, ditampilkan gambar result1 dengan padding 8dp.

Pada lines 72–77, ditampilkan gambar result2 juga dengan padding 8dp.

Pada line 79, ditambahkan Spacer dengan tinggi 16dp sebagai jarak antar elemen.

Pada line 80, dideklarasikan tombol Button dengan logika onClick.

Pada lines 81–82, nilai acak 1–6 diberikan ke result1 dan result2.

Pada lines 83–84, jika nilai dadu sama, ditampilkan pesan “Selamat anda dapat dadu double!” menggunakan Toast.

Pada lines 85–86, jika tidak sama, ditampilkan pesan “Anda belum beruntung”.

Pada line 89, digunakan Text() untuk menampilkan teks tombol dengan string dari R.string.roll.

Pada line 96, didefinisikan fungsi getImage() dengan parameter diceValue bertipe Int.

Pada lines 98–103, digunakan when untuk mencocokkan nilai dadu dengan gambar yang sesuai (dice\_1 sampai dice\_6).

Pada line 104, jika nilai tidak sesuai 1–6, dikembalikan gambar default dice\_0.

1. **string.xml**

Pada line 3, dideklarasikan namespace untuk atribut tombol lempar

## Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/AllanoLintang/MobileAppTest