

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Raudatul Sholehah

NIM. 2310817220002

**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 Praktikum ini dikerjakan oleh:

Nama Praktikan : Raudatul Sholehah
NIM : 2310817220002

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar
NIM. 2210817210026

Muti`a Maulida S.Kom M.T.I
NIP. 19881027 201903 20 13

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code.....	8
B. Output Program	14
C. Pembahasan	16
D. Tautan Git	20

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	14
Gambar 2. Screenshot Hasil Jawaban Soal 1	15
Gambar 3. Screenshot Hasil Jawaban Soal 1	15
Gambar 4. Screenshot Hasil Jawaban Soal 1	16
Gambar 5. Screenshot Hasil Jawaban Soal 1	16

DAFTAR TABEL

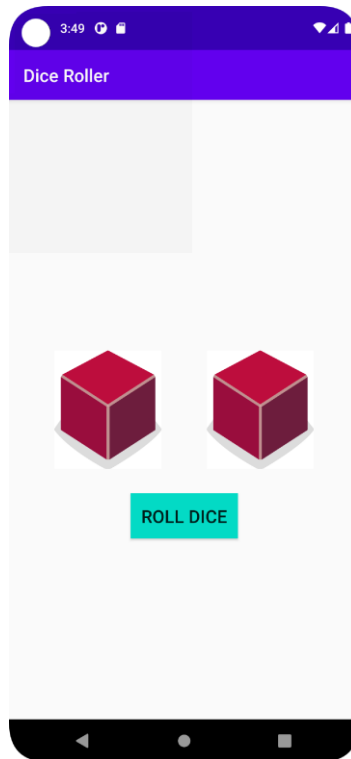
Tabel 1. Source Code Jawaban Soal 1.....	10
Tabel 2. Source Code Jawaban Soal 1.....	11
Tabel 3. Source Code Jawaban Soal 1.....	12
Tabel 4. Source Code Jawaban Soal 1.....	13
Tabel 5. Source Code Jawaban Soal 1.....	13
Tabel 6. Source Code Jawaban Soal 1.....	14
Tabel 7. Source Code Jawaban Soal 1.....	14

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.



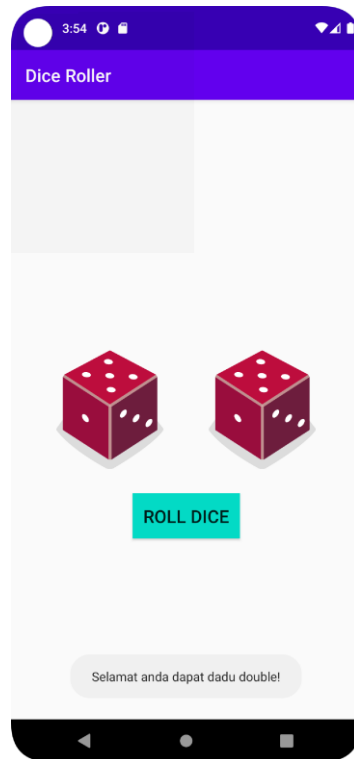
Gambar 1 Tampilan Awal Aplikasi

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=147HT2IIH5qin3z5ta7H9y2N_5OMW81Ll&export=download



Gambar 3 Tampilan Roll Dadu Double

A. Source Code

1. MainActivity.kt

```
1 package com.example.diceroller
2
3 import android.graphics.Color
4 import android.graphics.PorterDuff
5 import androidx.activity.viewModels
6 import androidx.appcompat.app.AppCompatActivity
7 import android.os.Bundle
8 import android.widget.TextView
9 import android.widget.Toast
10 import android.widget.ImageView
11 import com.example.diceroller.databinding.ActivityMainBinding
12 import androidx.lifecycle.ViewModel
13 import androidx.lifecycle.ViewModelProvider
14
15 class MainActivity : AppCompatActivity() {
16
17     private lateinit var binding: ActivityMainBinding
18     private val viewModel: DiceViewModel by viewModels()
19
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22
23         setContentView(R.layout.activity_main)
24         binding = ActivityMainBinding.inflate(layoutInflater)
25         setContentView(binding.root)
26
27         viewModel.rollDice()
28
29         binding.tvResult.text = viewModel.result.toString()
30
31         binding.btnRoll.setOnClickListener {
32             viewModel.rollDice()
33         }
34     }
35 }
```



```

23         binding = ActivityMainBinding.inflate(layoutInflater)
24         setContentView(binding.root)
25
26         binding.letRoll.setOnClickListener { roll() }
27
28         updateDiceImage(binding.diceImage1,
29 viewModel.diceValue1)
30         updateDiceImage(binding.diceImage2,
31 viewModel.diceValue2)
32     }
33
34     private fun roll() {
35         viewModel.diceValue1 = diceroll1()
36         viewModel.diceValue2 = diceroll2()
37
38         if (viewModel.diceValue1 == viewModel.diceValue2) {
39             showCustomToast("Selamat anda dapat dadu double!",
40 Color.parseColor("#F0F0F0"), Color.BLACK)
41         } else {
42             showCustomToast("Anda belum beruntung!",
43 Color.parseColor("#F0F0F0"), Color.BLACK)
44         }
45     }
46
47     private fun diceroll1(): Int {
48         val dice1 = Dice(6)
49         val diceRoll1 = dice1.roll()
50
51         updateDiceImage(binding.diceImage1, diceRoll1)
52         return diceRoll1
53     }
54
55     private fun diceroll2(): Int {
56         val dice2 = Dice(6)
57         val diceRoll2 = dice2.roll()
58
59         updateDiceImage(binding.diceImage2, diceRoll2)
60         return diceRoll2
61     }
62
63     private fun updateDiceImage(imageView: ImageView, diceValue:
64 Int) {
65         val drawableResource = when (diceValue) {
66             1 -> R.drawable.dice_1
67             2 -> R.drawable.dice_2
68             3 -> R.drawable.dice_3
69             4 -> R.drawable.dice_4
70             5 -> R.drawable.dice_5
71             6 -> R.drawable.dice_6
72             else -> R.drawable.empty_dice
73         }
74     }

```

```

70         imageView.setImageResource(drawableResource)
71         imageView.contentDescription = diceValue.toString()
72     }
73
74     private fun showCustomToast(message: String,
75     backgroundColor: Int, textColor: Int) {
76         val toast = Toast.makeText(this, message,
77         Toast.LENGTH_LONG)
78         val view = toast.view
79         view?.background?.setColorFilter(backgroundColor,
80         PorterDuff.Mode.SRC_IN)
81         val text =
82         view?.findViewById<TextView>(android.R.id.message) =
83         text?.setTextColor(textColor)
84         toast.show()
85     }
86 }
87
88 class Dice(private val numSides: Int) {
89     fun roll(): Int = (1..numSides).random()
90 }
91
92 class DiceViewModel : ViewModel() {
93     var diceValue1: Int = 0
94     var diceValue2: Int = 0
95 }

```

Tabel 1. Source Code Jawaban Soal 1

2. activity_main.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3  xmlns:android="http://schemas.android.com/apk/res/android"
4  xmlns:app="http://schemas.android.com/apk/res-auto"
5  xmlns:tools="http://schemas.android.com/tools"
6  android:layout_width="match_parent"
7  android:layout_height="match_parent"
8  android:layout_gravity="center_vertical"
9  android:orientation="vertical"
10 tools:context=".MainActivity">
11     <ImageView
12         android:id="@+id/dice_image1"
13         android:layout_width="150dp"
14         android:layout_height="150dp"
15         android:layout_marginStart="28dp"
16         android:layout_marginTop="300dp"
17         app:layout_constraintStart_toStartOf="parent"
18         app:layout_constraintTop_toTopOf="parent"
19         app:srcCompat="@drawable/empty_dice" />
20
21     <ImageView

```

```

22         android:id="@+id/dice_image2"
23         android:layout_width="150dp"
24         android:layout_height="150dp"
25         android:layout_marginTop="300dp"
26         android:layout_marginEnd="32dp"
27         android:src="@drawable/empty_dice"
28         app:layout_constraintEnd_toEndOf="parent"
29         app:layout_constraintTop_toTopOf="parent" />
30
31     <Button
32         android:id="@+id/let_roll"
33         android:layout_width="wrap_content"
34         android:layout_height="wrap_content"
35         android:layout_marginTop="8dp"
36         android:backgroundTint="#00FFFF"
37         android:text="ROLL DICE"
38         android:textColor="#000000"
39         app:layout_constraintBaseline_toBottomOf="parent"
40         app:layout_constraintEnd_toEndOf="parent"
41         app:layout_constraintStart_toStartOf="parent"
42
43         app:layout_constraintTop_toBottomOf="@id/dice_image1"
44     />
45 </androidx.constraintlayout.widget.ConstraintLayout>

```

Tabel 2. Source Code Jawaban Soal 1

3. activity_main.xml (land)

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <ImageView
11         android:id="@+id/dice_image1"
12         android:layout_width="150dp"
13         android:layout_height="150dp"
14         android:layout_marginEnd="2dp"
15         app:layout_constraintBottom_toTopOf="@+id/let_roll"
16         app:layout_constraintEnd_toStartOf="@+id/dice_image2"
17         app:layout_constraintHorizontal_bias="0.5"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent"
20         app:layout_constraintVertical_bias="0.718"
21         app:srcCompat="@drawable/empty_dice" />
22
23     <ImageView

```

```

23         android:id="@+id/dice_image2"
24         android:layout_width="150dp"
25         android:layout_height="150dp"
26         android:layout_marginStart="2dp"
27         android:src="@drawable/empty_dice"
28         app:layout_constraintBottom_toTopOf="@+id/let_roll"
29         app:layout_constraintEnd_toEndOf="parent"
30         app:layout_constraintHorizontal_bias="0.5"
31         app:layout_constraintStart_toEndOf="@+id/dice_image1"
32         app:layout_constraintTop_toTopOf="parent"
33         app:layout_constraintVertical_bias="0.718" />
34
35     <Button
36         android:id="@+id/let_roll"
37         android:layout_width="wrap_content"
38         android:layout_height="wrap_content"
39         android:layout_marginBottom="88dp"
40         android:backgroundTint="#00FFFF"
41         android:text="ROLL DICE"
42         android:textColor="#000000"
43         app:layout_constraintBottom_toBottomOf="parent"
44         app:layout_constraintEnd_toEndOf="parent"
45         app:layout_constraintStart_toStartOf="parent" />
46 </androidx.constraintlayout.widget.ConstraintLayout>

```

Tabel 3. Source Code Jawaban Soal 1

4. build.gradle (Module :app)

```

1  plugins {
2      id 'com.android.application'
3      id 'kotlin-android'
4  }
5
6  android {
7      namespace 'com.example.diceroller'
8      compileSdkVersion 33
9
10     defaultConfig {
11         applicationId "com.example.diceroller"
12         minSdkVersion 19
13         targetSdkVersion 33
14         versionCode 1
15         versionName "1.0"
16
17         testInstrumentationRunner
18         "androidx.test.runner.AndroidJUnitRunner"
19     }
20     buildFeatures {
21         viewBinding true
22         dataBinding true
23     }

```

24	
25	buildTypes {
26	release {
27	minifyEnabled false
28	proguardFiles getDefaultProguardFile('proguard-
	android-optimize.txt'), 'proguard-rules.pro'
29	}
30	}
31	
32	compileOptions {
33	sourceCompatibility JavaVersion.VERSION_1_8
34	targetCompatibility JavaVersion.VERSION_1_8
35	}
36	
37	kotlinOptions {
38	jvmTarget = "1.8"
39	}
40	}
41	
42	dependencies {
43	def lifecycle_version = "2.6.2"
44	implementation "androidx.lifecycle:lifecycle-viewmodel-
	ktx:\$lifecycle_version"
45	implementation "androidx.lifecycle:lifecycle-livedata-
	ktx:\$lifecycle_version"
46	implementation "androidx.lifecycle:lifecycle-runtime-
	ktx:\$lifecycle_version"
47	implementation fileTree(dir: 'libs', include: ['*.jar'])
48	implementation 'androidx.appcompat:appcompat:1.6.1'
49	implementation
	'androidx.constraintlayout:constraintlayout:2.1.4'
50	testImplementation 'junit:junit:4.13.2'
51	androidTestImplementation 'androidx.test.ext:junit:1.1.5'
52	androidTestImplementation 'androidx.test.espresso:espresso-
	core:3.5.1'
53	implementation "androidx.activity:activity-ktx:1.6.1"
54	}

Tabel 4. Source Code Jawaban Soal 1

5. colors.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<resources>
3	<color name="colorPrimary">#0600a6</color>
4	<color name="colorPrimaryDark">#05016e</color>
5	<color name="colorAccent">#03DAC5</color>
6	</resources>

Tabel 5. Source Code Jawaban Soal 1

6. strings.xml

```
1 <resources>
2   <string name="app_name">Dice Roller</string>
3   <string name="roll">ROLL</string>
4 </resources>
```

Tabel 6. Source Code Jawaban Soal 1

7. styles.xml

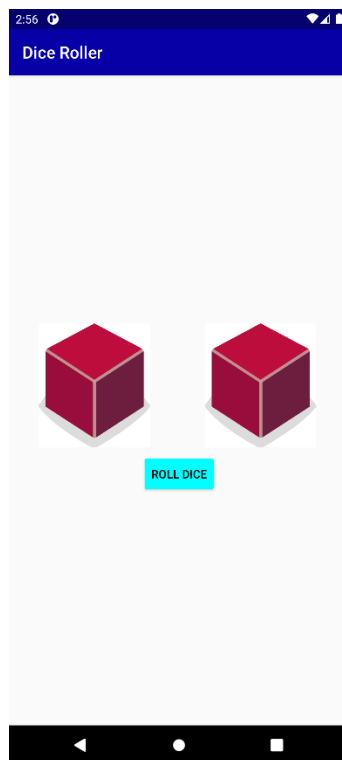
```

1  <resources>
2
3      <style                                name="AppTheme"
parent="Theme.AppCompat.Light.DarkActionBar">
4          <item name="colorPrimary">@color/colorPrimary</item>
5          <item
name="colorPrimaryDark">@color/colorPrimaryDark</item>
6          <item name="colorAccent">@color/colorAccent</item>
7      </style>
8
9  </resources>

```

Tabel 7. Source Code Jawaban Soal 1

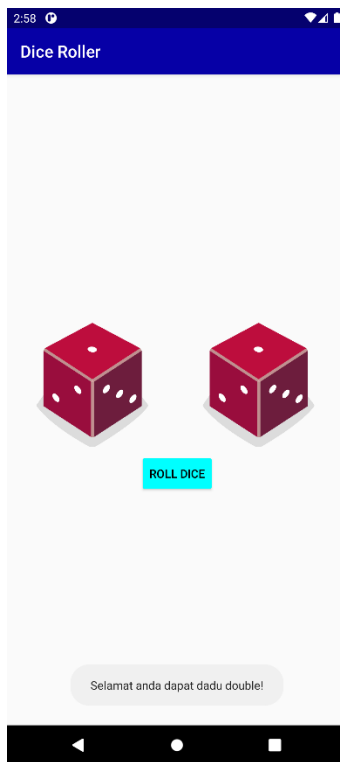
B. Output Program



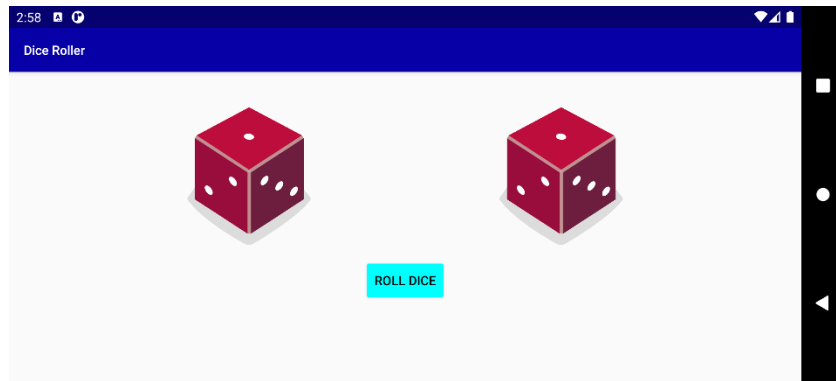
Gambar 1. Screenshot Hasil Jawaban Soal 1



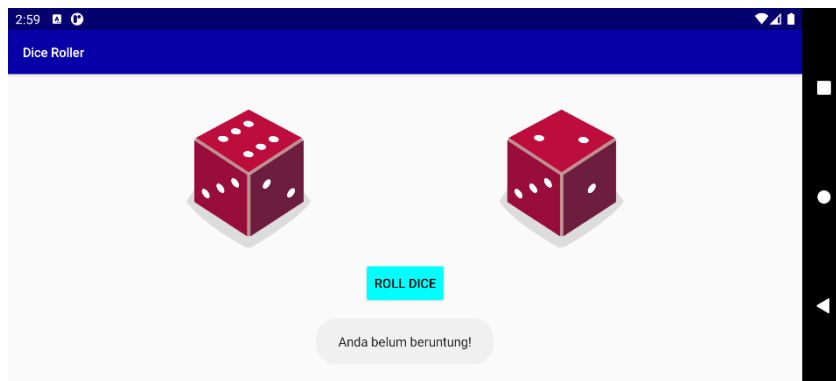
Gambar 2. Screenshot Hasil Jawaban Soal 1



Gambar 3. Screenshot Hasil Jawaban Soal 1



Gambar 4. Screenshot Hasil Jawaban Soal 1



Gambar 5. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt

- Baris 1–10:
Deklarasi package dan import library yang diperlukan. Package `com.example.diceroller` menunjukkan lokasi file dalam struktur proyek. Beberapa import digunakan untuk fitur antarmuka grafis seperti `Toast`, `TextView`, `ImageView`, serta `ViewModel` untuk manajemen data dan `ActivityMainBinding` untuk mengakses komponen UI dari layout secara langsung melalui `ViewBinding`.
- Baris 12–15:
Deklarasi kelas `MainActivity` yang merupakan turunan dari `AppCompatActivity`. Di dalamnya dideklarasikan dua properti: `binding` untuk `ViewBinding` agar bisa mengakses komponen UI secara langsung, dan `viewModel` untuk menyimpan data hasil roll dadu agar tetap bertahan saat konfigurasi berubah (rotasi layar).
- Baris 17–24 (`onCreate`)
Method `onCreate` dipanggil saat activity pertama kali dijalankan. `binding` diinisialisasi dengan `inflate`, lalu layout ditampilkan menggunakan

`setContentView(binding.root)`. Tombol `letRoll` dikaitkan dengan fungsi `roll()` saat diklik. Kemudian, dua fungsi `updateDiceImage` digunakan untuk menampilkan nilai dadu dari `ViewModel` ke dua `ImageView`.

- Baris 26–34 (`roll`)
Fungsi `roll` dipanggil ketika tombol ditekan. Di sini, nilai dadu diperbarui melalui `diceroll1()` dan `diceroll2()` lalu disimpan di `viewModel`. Jika kedua dadu bernilai sama, maka akan muncul pesan `Toast` bahwa pengguna mendapatkan dadu sama. Jika tidak, muncul pesan bahwa pengguna belum beruntung.
- Baris 36–41 (`diceroll1`) dan 43–48 (`diceroll2`)
Kedua fungsi ini membuat objek `Dice` dengan 6 sisi, melakukan roll acak, memperbarui gambar dadu melalui `updateDiceImage`, dan mengembalikan nilai hasil roll. Keduanya bekerja dengan cara yang sama, hanya berbeda pada komponen gambar yang diubah.
- Baris 50–61 (`updateDiceImage`)
Fungsi ini menerima `ImageView` dan nilai dadu. Berdasarkan nilai dadu (1–6), fungsi memilih gambar yang sesuai (misalnya `R.drawable.dice_1`). Gambar ditampilkan pada `ImageView` yang ditentukan, dan konten deskripsinya diperbarui agar aksesibilitas lebih baik.
- Baris 63–69 (`showCustomToast`)
Fungsi ini menampilkan pesan `Toast` dengan tampilan yang dimodifikasi. Latar belakang toast diberi warna kustom, dan teks toast diubah warnanya. Fungsi ini memberi umpan balik visual yang menarik saat pengguna roll dadu.
- Baris 71–73 (`class Dice`)
Kelas `Dice` merupakan class sederhana yang menerima jumlah sisi dadu (dalam kasus ini 6) dan memiliki fungsi `roll()` untuk menghasilkan angka acak antara 1 hingga jumlah sisi.
- Baris 75–77 (`class DiceViewModel`)
Kelas `DiceViewModel` mewarisi `ViewModel` dan menyimpan dua variabel (`diceValue1` dan `diceValue2`) untuk mempertahankan nilai dadu. Dengan menggunakan `ViewModel`, nilai tetap terjaga meskipun terjadi perubahan konfigurasi seperti rotasi layar.

2. `activity_main.xml`

- Baris 1–9:
Bagian awal ini adalah pengaturan dasar layout. Di sini dipakai `ConstraintLayout` yang artinya semua komponen (seperti gambar dan tombol) akan diatur posisinya saling mengait (terhubung satu sama lain). Ukurannya disetel memenuhi layar (`match_parent`), dan `tools:context` memberitahu bahwa layout ini dipakai untuk `MainActivity`.
- Blok `ImageView` pertama (baris 11–18):
Bagian ini menampilkan gambar dadu pertama. Ukurannya dibuat 150x150dp. Letaknya sedikit geser ke kanan (28dp) dan agak turun ke bawah (300dp dari atas

layar). Awalnya, gambar yang ditampilkan adalah `empty_dice` alias dadu kosong.

- Blok `ImageView` kedua (baris 20–27):
Sama seperti yang pertama, ini untuk gambar dadu kedua. Ukuran dan posisinya mirip, tapi gambarnya ditempatkan di sisi kanan layar (dikasih `margin end 32dp`). Gambarnya juga pakai `empty_dice` untuk awal.
- Blok `Button` (baris 29–40):
Ini adalah tombol buat nge-roll dadu. Tombolnya dikasih teks “ROLL DICE”, warnanya biru muda (`#00FFFF`), dan teksnya hitam. Tombol ini diletakkan di bawah gambar dadu, di tengah horizontal, dan posisinya menyesuaikan bagian bawah layout.

3. `activity_main.xml` (land)

- Bagian awal (baris 1–7):
Di bagian pembuka ini, memberitahu bahwa sistem Android jika layout ini pakai `ConstraintLayout`, yaitu jenis layout yang fleksibel untuk mengatur posisi elemen-elemen di layar. Semua elemen akan disusun berdasarkan posisi satu sama lain. Layout ini dipakai di `MainActivity`.
- Gambar Dadu Pertama (baris 9–18):
Ini gambar dadu pertama. Ukurannya `150x150dp`. Letaknya ada di sebelah kiri tengah layar, diatur supaya muncul di atas tombol roll. Posisi horizontalnya agak ditengahin lewat `horizontal_bias`. Gambar awalnya dikasih `empty_dice`.
- Gambar Dadu Kedua (baris 20–30):
Ini dadu kedua, ukurannya sama kayak dadu pertama. Bedanya, dadu ini diletakkan di sebelah kanan dadu pertama, posisinya juga di atas tombol roll dan sejajar. Sama seperti sebelumnya, gambar yang ditampilkan awalnya juga `empty_dice`.
- Tombol Roll Dice (baris 32–40):
Tombol ini muncul di bagian bawah layar. Kalau ditekan, nanti akan melempar dadu. Tulisan di tombolnya "ROLL DICE", warnanya biru terang dan teksnya hitam. Posisinya diatur biar pas di tengah bawah layar dengan margin bawah `88dp` supaya agak naik dikit dari pinggir layar.

4. `build.gradle` (Module :app)

- Blok `plugins` (baris 1–3):
Di bagian ini, berfungsi untuk mengaktifin plugin Android dan Kotlin supaya project ini bisa jalanin kode Android yang ditulis pakai Kotlin.
- Blok `android` (baris 5–41):
Ini bagian pengaturan utama buat project Android-nya.
 - a. `namespace` membuat identitas unik buat aplikasi (kayak nama lengkap aplikasi).
 - b. `compileSdkVersion`, `minSdkVersion`, dan `targetSdkVersion` mengatur versi Android yang didukung (dari Android 4.4 sampai Android 13).

- c. `applicationId` adalah ID aplikasi yang unik, biasanya dipakai buat upload ke Play Store.
- d. `versionCode` dan `versionName` memberitahu info versi app (misal: versi 1.0).
- e. `testInstrumentationRunner` dipakai buat jalankan tes otomatis di Android.

Bagian `buildFeatures` yaitu mengaktifkan `viewBinding` dan `dataBinding`. Intinya, biar kita bisa akses elemen XML di kode Kotlin dengan lebih mudah dan aman, tanpa harus pakai `findViewById()` lagi. Bagian `buildTypes` yaitu pengaturan untuk versi rilis (release). Di sini `minifyEnabled = false`, artinya kita belum pakai fitur untuk mengecilkan ukuran aplikasi (biasanya buat produksi). `compileOptions` dan `kotlinOptions` yaitu bagian yang memberitahu bahwa kita pakai Java 8 dan Kotlin versi yang kompatibel sama Java 8.

- Blok `dependencies` (baris 43–55):

Ini bagian penting buat memberitahu library apa saja yang kita pakai:

- a. `lifecycle-*` untuk mengatur dan menyimpan data `ViewModel` agar tetap hidup walaupun orientasi layar berubah.
- b. `appcompat` dan `constraintlayout` untuk tampilan dan kompatibilitas UI.
- c. `junit` dan `espresso` untuk ngetes aplikasi (unit test dan UI test).
- d. `activity-ktx` agar bisa pakai fitur tambahan Kotlin untuk aktivitas (lebih ringkas dan modern).

5. `colors.xml`

File ini isinya kumpulan warna yang akan dipakai di aplikasi. Jadi daripada nulis kode warna langsung di layout, kita tinggal panggil nama warnanya aja (misal `@color/colorPrimary`).

- a. `colorPrimary` itu warna utama aplikasi (biru tua).
- b. `colorPrimaryDark` dipakai pas status bar atau elemen lain yang lebih gelap.
- c. `colorAccent` biasanya dipakai buat warna tombol atau efek yang mencolok.

6. `strings.xml`

File ini menyimpan teks-teks yang ditampilkan di aplikasi. Jadi kalau mau ganti bahasa atau edit kata-kata, tinggal ubah di sini.

- a. `app_name` adalah nama aplikasinya yang muncul di ikon atau title bar.
- b. `roll` adalah tulisan tombol buat melempar dadu.

7. `styles.xml`

Berguna untuk mengatur tema/tampilan umum aplikasi untuk seluruh tampilan.

- a. `AppTheme` ini tema utama yang kita buat.

- b. Di dalamnya, kita tentuin warna utama (`colorPrimary`), warna gelapnya (`colorPrimaryDark`), dan warna aksennya (`colorAccent`).
- c. `parent="Theme.AppCompat.Light.DarkActionBar"` artinya kita pakai tema bawaan Android yang terang, tapi action bar-nya agak gelap.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/rrdtlsh/Praktikum_Mobile