

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 1**



ANDROID BASIC WITH KOTLIN

Oleh:

Sheila Sabina

NIM. 2310817220028

**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 : Sheila Sabina
NIM : 2310817220028

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	11
C. Pembahasan	12
D. Tautan Git.....	18

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	12
---	----

DAFTAR TABEL

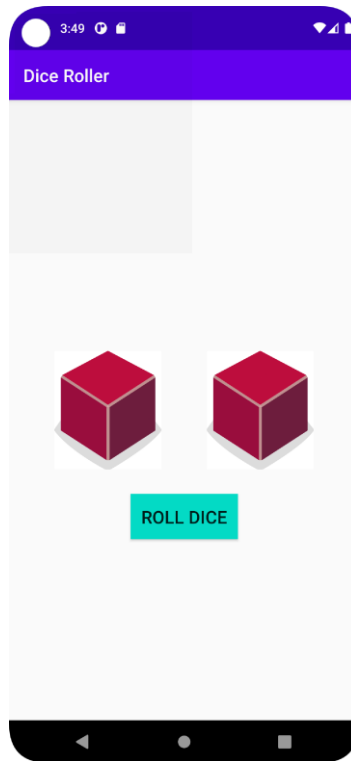
Tabel 1. Source Code Jawaban Soal 1.....	9
Tabel 2. Source Code Jawaban Soal 1.....	11

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.



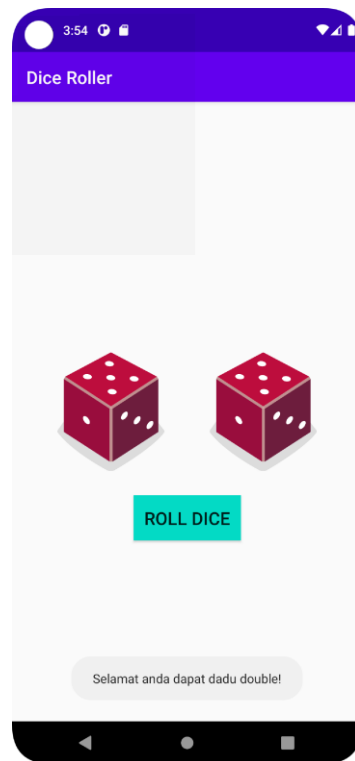
Gambar 1 Tampilan Awal Aplikasi

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_5OMW81LI&export=download



Gambar 3 Tampilan Roll Dadu Double

A. Source Code

1. MainActivity.kt

```

1 package com.example.dice_roller
2
3 import android.os.Bundle
4 import android.widget.Button
5 import android.widget.ImageView
6 import android.widget.Toast
7 import androidx.activity.enableEdgeToEdge
8 import androidx.appcompat.app.AppCompatActivity
9 import androidx.core.view.ViewCompat
10 import androidx.core.view.WindowInsetsCompat
11 import com.example.dice_roller.databinding.ActivityMainBinding
12
13 class MainActivity : AppCompatActivity() {
14
15     private lateinit var binding: ActivityMainBinding
16     override fun onCreate(savedInstanceState: Bundle?) {
17         super.onCreate(savedInstanceState)
18         enableEdgeToEdge()
19         binding = ActivityMainBinding.inflate(layoutInflater)
20         setContentView(binding.root)
21
22         // dice1 = findViewById(R.id.dice_image1)

```



```

23 //          dice2  =  findViewById(R.id.dice_image2)
24 //
25 //          roll_button  =  findViewById(R.id.roll_button)
26
27          binding.rollButton.setOnClickListener{
28              rollDice()
29          }
30      }
31      private          fun          rollDice()          {
32
33          val          randomInt1          =          (1..6).random()
34          val          drawableResource1          =          when          (randomInt1)          {
35              1          ->          R.drawable.dice_1
36              2          ->          R.drawable.dice_2
37              3          ->          R.drawable.dice_3
38              4          ->          R.drawable.dice_4
39              5          ->          R.drawable.dice_5
40              6          ->          R.drawable.dice_6
41              else          ->          R.drawable.dice_0
42          }
43          val          randomInt2          =          (1..6).random()
44          val          drawableResource2          =          when          (randomInt2)          {
45              1          ->          R.drawable.dice_1
46              2          ->          R.drawable.dice_2
47              3          ->          R.drawable.dice_3
48              4          ->          R.drawable.dice_4
49              5          ->          R.drawable.dice_5
50              6          ->          R.drawable.dice_6
51              else          ->          R.drawable.dice_0
52          }
53          binding.diceImage1.setImageResource(drawableResource1)
54          binding.diceImage2.setImageResource(drawableResource2)
55
56          if          (randomInt1          ==          randomInt2)          {
57              Toast.makeText(this, "Selamat Kamu dapat Double!",
58 Toast.LENGTH_SHORT).show()
59          }          else          {
60              Toast.makeText(this, "Anda Kurang Beruntung",
61 Toast.LENGTH_SHORT).show()
62          }
63      }
64  }

```

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:id="@+id/main"

```

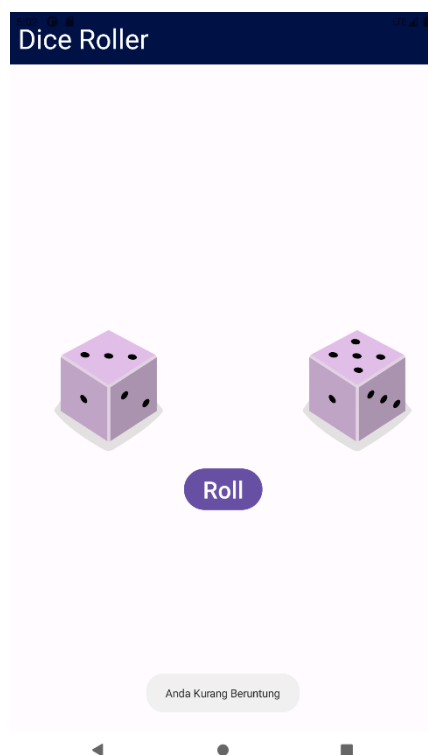
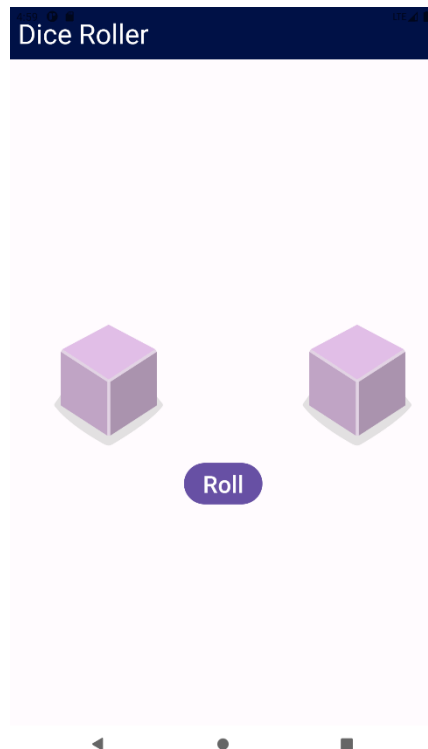
```

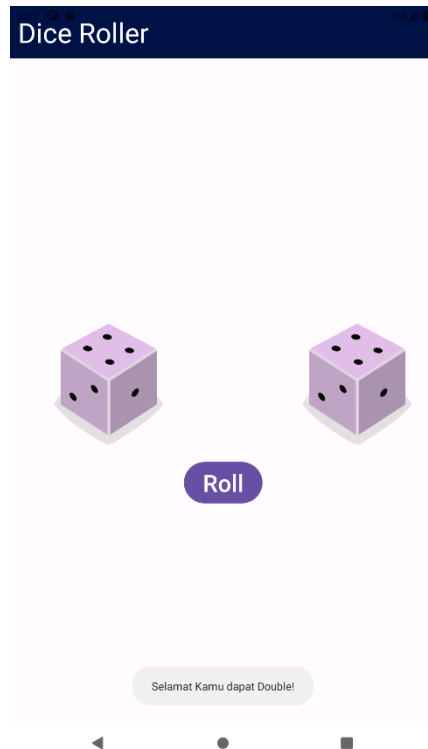
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      tools:context=".MainActivity">
10
11      <TextView
12          android:layout_width="match_parent"
13          android:layout_height="wrap_content"
14          android:text="Dice                                Roller"
15          android:textColor="@color/white"
16          android:textSize="35sp"
17          app:layout_constraintStart_toStartOf="parent"
18          app:layout_constraintTop_toTopOf="parent"
19          android:padding="10dp"
20          android:background="#001146"
21      />
22
23      <ImageView
24          android:id="@+id/dice_image1"
25          android:layout_width="200dp"
26          android:layout_height="200dp"
27          android:layout_marginStart="25dp"
28          android:src="@drawable/dice_0"
29          app:layout_constraintBottom_toBottomOf="parent"
30          app:layout_constraintStart_toStartOf="parent"
31          app:layout_constraintTop_toTopOf="parent"
32      />
33
34      <ImageView
35          android:id="@+id/dice_image2"
36          android:layout_width="200dp"
37          android:layout_height="200dp"
38          android:layout_marginStart="25dp"
39          android:src="@drawable/dice_0"
40          app:layout_constraintBottom_toBottomOf="parent"
41          app:layout_constraintEnd_toEndOf="parent"
42          app:layout_constraintTop_toTopOf="parent"
43      />
44
45      <Button
46          android:id="@+id/roll_button"
47          android:layout_width="wrap_content"
48          android:layout_height="wrap_content"
49          app:layout_constraintBottom_toBottomOf="parent"
50          app:layout_constraintEnd_toEndOf="parent"
51          app:layout_constraintStart_toStartOf="parent"
52          app:layout_constraintTop_toTopOf="parent"
53          android:layout_marginTop="250dp"
54          android:text="Roll"
55          android:textSize="30sp"
56      />
57
58  </androidx.constraintlayout.widget.ConstraintLayout>

```

Tabel 2. Source Code Jawaban Soal 1

B. Output Program





Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt:

Pada baris [1] terdapat fungsi `package com.example.dice_roller` yang digunakan untuk mendeklarasikan bahwa file tersebut termasuk dalam paket bernama `com.example.dice_roller`. Paket (package) berfungsi untuk mengelompokkan kelas-kelas yang saling berhubungan agar kode lebih terstruktur dan mudah dikelola. Pada baris [2] hingga [11] terdapat fungsi `import` yang digunakan untuk mengimpor berbagai komponen yang dibutuhkan dalam pengembangan aplikasi Android. Baris `import android.os.Bundle,` `android.widget.Button,` `android.widget.ImageView,` dan `android.widget.Toast` digunakan untuk mengakses komponen dasar Android seperti tombol, gambar, dan pesan pop-up. Sementara itu, `androidx.activity.enableEdgeToEdge,` `androidx.appcompat.app.AppCompatActivity,` dan `androidx.core.view` digunakan untuk mendukung tampilan antarmuka yang

lebih modern dan kompatibilitas dengan berbagai versi Android. Terakhir, `com.example.dice_roller.databinding.ActivityMainBinding` mengimpor class yang dihasilkan secara otomatis oleh View Binding untuk mengakses elemen-elemen UI di layout `activity_main.xml` dengan lebih efisien dan aman. Pada baris [13] hingga [28] terdapat fungsi bagian dari kelas `MainActivity` dalam aplikasi Android, yang berfungsi sebagai aktivitas utama saat aplikasi dijalankan. Kelas ini mewarisi `AppCompatActivity`, sehingga dapat menggunakan fitur-fitur kompatibilitas Android. Pada fungsi `onCreate`, yang dipanggil saat aktivitas pertama kali dibuat, terdapat beberapa hal penting seperti `enableEdgeToEdge()` yang berfungsi untuk mengatur agar tampilan aplikasi bisa menyesuaikan hingga ke tepi layar, memberikan tampilan yang lebih modern. `binding = ActivityMainBinding.inflate(layoutInflater)` yang digunakan untuk menginisialisasi objek *View Binding* untuk mengakses elemen-elemen UI tanpa perlu `findViewById`. `setContentView(binding.root)` yang digunakan untuk menentukan layout yang akan ditampilkan di layar, yaitu root view dari file `activity_main.xml`. `binding.rollButton.setOnClickListener { rollDice() }` berfungsi untuk mengatur aksi saat tombol "roll" ditekan, yaitu memanggil fungsi `rollDice()` yang nantinya digunakan untuk melempar dadu.

Pada baris [31] hingga [52] terdapat fungsi `rollDice()` bertanggung jawab untuk mensimulasikan pelemparan dua buah dadu secara acak dan menentukan gambar dadu mana yang akan ditampilkan berdasarkan hasil tersebut. Di dalam fungsi ini, pertama-tama dibuat dua variabel `randomInt1` dan `randomInt2` yang masing-masing menghasilkan angka acak antara 1 hingga 6 menggunakan `(1..6).random()`. Angka ini merepresentasikan sisi dadu yang muncul. Setelah angka acak diperoleh, masing-masing angka tersebut dipetakan ke resource gambar (drawable) yang sesuai menggunakan struktur `when`. Misalnya, jika angka yang muncul adalah 3, maka gambar yang digunakan adalah `R.drawable.dice_3`. Jika secara tidak sengaja angka di luar 1–6 muncul (meskipun itu hampir tidak mungkin terjadi dengan range yang sudah ditentukan), maka `else` akan menampilkan gambar default `dice_0`. Hasil dari pemetaan ini disimpan dalam

`drawableResource1` dan `drawableResource2`, yang nantinya bisa digunakan untuk mengatur gambar pada dua `ImageView` dadu di tampilan aplikasi. Meskipun pada bagian kode ini belum ada perintah untuk menampilkan gambar ke tampilan (seperti `imageView.setImageResource(drawableResource1)`), namun logika inti untuk menentukan gambar yang sesuai berdasarkan angka acak sudah lengkap dan siap digunakan untuk animasi atau visualisasi dadu secara dinamis.

Baris [53] dan [54] terdapat fungsi yang digunakan untuk menyelesaikan proses simulasi lempar dadu dengan menampilkan gambar dadu di layar sesuai angka acak yang didapat, sehingga pengguna dapat melihat hasilnya secara visual. `binding.diceImage1` dan `binding.diceImage2` adalah referensi ke dua `ImageView` yang ada di layout XML, dan diakses menggunakan View Binding. `setImageResource(drawableResource1)` dan `setImageResource(drawableResource2)` digunakan untuk mengganti gambar di `ImageView` dengan gambar dadu yang sesuai berdasarkan angka acak yang diperoleh sebelumnya. `drawableResource1` dan `drawableResource2` masing-masing berisi ID gambar (`R.drawable.dice_1`, `dice_2`, dst.) yang sesuai dengan hasil lemparan dadu. Pada baris [56] hingga [61] terdapat fungsi yang digunakan untuk memberikan umpan balik kepada pengguna berupa pesan pop up (Toast) berdasarkan hasil lemparan dua dadu. `if (randomInt1 == randomInt2)` berfungsi untuk mengecek apakah hasil dua dadu sama (misalnya keduanya 4 atau keduanya 6). Jika iya, maka kondisi dianggap "double". `Toast.makeText(...).show()` berguna untuk menampilkan pesan singkat di layar pengguna. Jika angkanya sama, maka muncul pesan: "Selamat Kamu dapat Double!" Jika berbeda, maka muncul pesan: "Anda Kurang Beruntung". `this` mengacu pada konteks `MainActivity`, yang dibutuhkan untuk menampilkan Toast. `Toast.LENGTH_SHORT` menentukan durasi tampilan Toast (sebentar saja).

2. activity_main.xml

Pada baris [1] hingga [9] terdapat fungsi yang merupakan kerangka dasar dari tampilan layar utama aplikasi android menggunakan `ConstraintLayout`, yang siap untuk diisi dengan elemen – elemen UI seperti tombol, gambar, teks, dan lainnya. `<?xml version="1.0" encoding="utf-8"?>` merupakan baris deklarasi XML, menunjukkan bahwa file ini ditulis dalam format XML versi 1.0 dengan encoding UTF-8. `ConstraintLayout` merupakan jenis layout dari Android Jetpack yang fleksibel untuk menyusun tampilan UI secara responsif dengan mengatur posisi tiap elemen berdasarkan constraint (batasan). `xmlns:android`, `xmlns:app`, dan `xmlns:tools` adalah deklarasi namespace yang dibutuhkan agar atribut-atribut XML dikenali dan digunakan dengan benar oleh sistem Android dan tools Android Studio. `android:id="@+id/main"` yang akan memberikan ID pada layout ini, sehingga bisa diakses melalui kode Kotlin menggunakan `View Binding` atau `findViewById`. `android:layout_width="match_parent"` dan `android:layout_height="match_parent"` yang akan mengatur agar layout ini mengisi seluruh lebar dan tinggi layar. `tools:context=".MainActivity"` yang akan memberitahu Android Studio bahwa layout ini digunakan dalam `MainActivity`, sehingga tampilan layout di editor dapat disesuaikan. Pada baris [11] hingga [21] terdapat fungsi yang berguna sebagai judul dari aplikasi dengan tampilan yang jelas dan mencolok di bagian atas layar, memberikan identitas visual bahwa aplikasi ini adalah "Dice Roller". `android:layout_width="match_parent"` digunakan untuk membuat lebar `TextView` mengikuti lebar parent (mengisi penuh dari kiri ke kanan). `android:layout_height="wrap_content"` berfungsi untuk membuat tinggi `TextView` menyesuaikan tinggi teks di dalamnya. `android:text="Dice Roller"` yang akan menentukan teks yang ditampilkan, dalam hal ini adalah "Dice Roller". `android:textColor="@color/white"` digunakan untuk mengatur warna teks menjadi putih, mengambil dari file `colors.xml`. `android:textSize="35sp"` yang berguna mengatur ukuran teks menjadi 35sp (scale-independent pixels), agar tetap terbaca dengan baik di berbagai ukuran layar.

`app:layout_constraintStart_toStartOf="parent"` dan
`app:layout_constraintTop_toTopOf="parent"` yang akan
menempatkan `TextView` di bagian kiri atas layar, dengan constraint ke sisi kiri dan
atas dari layout induk (`ConstraintLayout`). `android:padding="10dp"`
yang akan memberikan ruang di dalam `TextView` sebesar 10dp dari semua sisi agar
teks tidak terlalu menempel ke tepi. `android:background="#001146"`
berguna untuk memberi latar belakang berwarna biru gelap (kode hex #001146).

Pada baris [23] hingga [31] terdapat fungsi yang digunakan untuk
menampilkan gambar dadu di tengah layar. Gambar ini memiliki ukuran 200dp x
200dp, dengan jarak margin 25dp dari sisi kiri. Gambar dadu pertama (`dice_image1`)
ditampilkan dengan gambar default `dice_0` dan diposisikan di tengah layar dengan
`ConstraintLayout`. `android:id="@+id/dice_image1"` berguna untuk
memberikan ID unik untuk `ImageView` ini, yang dapat diakses di kode (misalnya
melalui View Binding atau `findViewById`), untuk mengganti gambar atau
mengatur propertinya. `android:layout_width="200dp"` digunakan untuk
menentukan lebar `ImageView` menjadi 200dp (density-independent
pixels). `android:layout_height="200dp"` yang berguna untuk
menentukan tinggi `ImageView` menjadi 200dp.
`android:layout_marginStart="25dp"` untuk mmberikan jarak margin
sebesar 25dp di sisi kirit dari `ImageView`, memberikan sedikit ruang antara gambar
dan tepi layar. `android:src="@drawable/dice_0"` yang akan menetapkan
gambar sumber yang ditampilkan di `ImageView`, dalam hal ini adalah gambar
`dice_0`, yang kemungkinan besar merupakan gambar dadu kosong atau default.
`app:layout_constraintBottom_toBottomOf="parent"` berguna
untuk menempatkan bagian bawah `ImageView` pada bagian bawah dari parent
layout (`ConstraintLayout`), memastikan gambar berada di bawah layar.
`app:layout_constraintStart_toStartOf="parent"` yang akan
menempatkan sisi kiri `ImageView` pada sisi kiri dari parent layout.
`app:layout_constraintTop_toTopOf="parent"` yang akan
menempatkan sisi atas `ImageView` pada sisi atas dari parent layout.

Pada baris [34] hingga [43] memiliki fungsi kode yang sama dengan blok kode yang sebelumnya namun dengan format gambar yang berbeda yakni gambar dadu pertama (dice_image2). Pada baris [45] hingga [56] terdapat fungsi button yang digunakan untuk memberikan interaksi pengguna, seperti tombol “roll” dadu. `android:id="@+id/roll_button"` berguna untuk memberikan ID unik untuk tombol ini, yang memungkinkan tombol tersebut diakses dalam kode melalui `View` `Binding` atau `findViewById`. `android:layout_width="wrap_content"` yang akan mengatur lebar tombol agar hanya sebesar konten di dalamnya, yaitu teks "Roll". `android:layout_height="wrap_content"` berfungsi untuk mengatur tinggi tombol agar hanya sebesar konten di dalamnya, sesuai dengan ukuran teks. `app:layout_constraintBottom_toBottomOf="parent"` berguna untuk menyusun tombol di bagian bawah layar, dengan mengikat bagian bawah tombol ke bagian bawah parent layout (`ConstraintLayout`). `app:layout_constraintEnd_toEndOf="parent"` yang akan menyusun tombol di sisi kanan layar, dengan mengikat sisi kanan tombol ke sisi kanan parent layout. `app:layout_constraintStart_toStartOf="parent"` yang berguna untuk menyusun tombol di sisi kiri layar, dengan mengikat sisi kiri tombol ke sisi kiri parent layout. `app:layout_constraintTop_toTopOf="parent"` yang akan menyusun tombol di bagian atas layar, dengan mengikat sisi atas tombol ke sisi atas parent layout. `android:layout_marginTop="250dp"` yang akan memberikan jarak margin sebesar 250dp di bagian atas tombol, memastikan tombol tidak terlalu dekat dengan bagian atas layar. `android:text="Roll"` : berguna untuk menentukan teks yang ditampilkan pada tombol, yaitu "Roll". `android:textSize="30sp"` berfungsi untuk menetapkan ukuran teks pada tombol menjadi 30sp (scale-independent pixels), memastikan teks tetap terbaca dengan baik di berbagai ukuran layar. Terakhir, pada baris [58] terdapat fungsi `</androidx.constraintlayout.widget.ConstraintLayout>` yang digunakan untuk sebagai tag penutup yang menandakan bahwa

ConstraintLayout telah selesai dan semua elemen yang berada di dalamnya, seperti TextView, Button, dan ImageView, juga telah selesai didefinisikan. ConstraintLayout adalah jenis layout yang fleksibel dan efisien dalam menyusun tampilan UI di Android. Setiap elemen di dalam ConstraintLayout diatur dengan constraint (batasan) terhadap elemen lain atau terhadap parent layout.

D. Tautan Git

<https://github.com/SheilaSabina/Praktikum-Mobile/tree/master/MODUL1>