

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2**



ANDROID LAYOUT

Oleh:

Alysa Armelia

NIM. 2310817120009

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2025**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN I
MODUL 2

Laporan Praktikum Pemrograman Mobile Modul 2: Android Layout ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Alysa Armelia
NIM : 2310817120009

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.....	7
B. Output Program	14
C. Pembahasan	16
D. Tautan Git	22

DAFTAR GAMBAR

Gambar 1 Screenshot Hasil Jawaban Soal 1 Awalan	14
Gambar 2 Screenshot Hasil Jawaban Soal 1	15
Gambar 3 Screenshot Hasil Jawaban Soal 1	15

DAFTAR TABEL

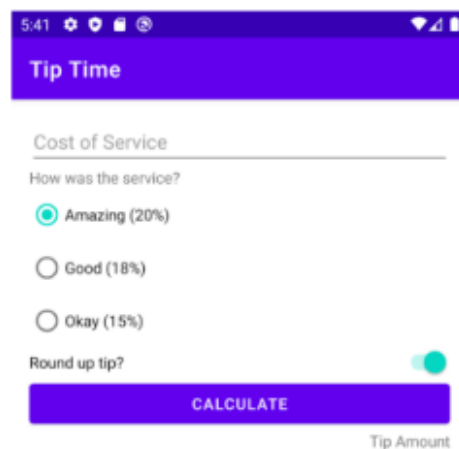
Tabel 1. 1 Source Code Jawaban Soal 1	8
Tabel 1. 2 Source Code Jawaban Soal 1	9
Tabel 1. 3 Source Code Jawaban Soal 1	9
Tabel 1. 4 Source Code Jawaban Soal 1	12
Tabel 1. 5 Source Code Jawaban Soal 1	13

SOAL 1

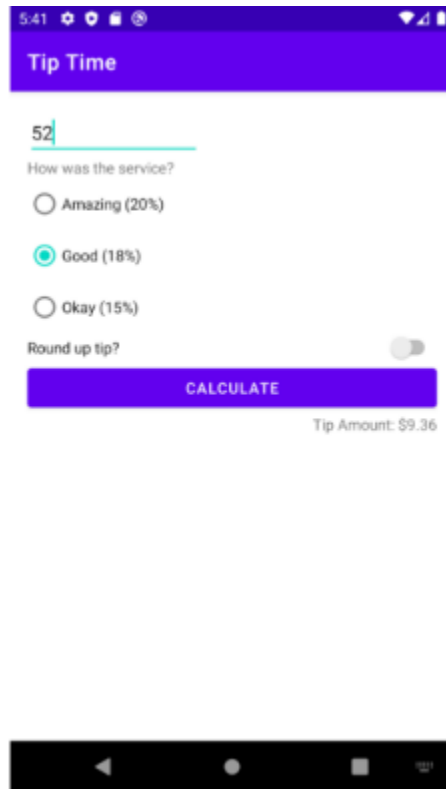
Soal Praktikum:

Buatlah sebuah aplikasi kalkulator tip yang dirancang untuk membantu pengguna menghitung tip yang sesuai berdasarkan total biaya layanan yang mereka terima. Fitur-fitur yang diharapkan dalam aplikasi ini mencakup:

1. Input Biaya Layanan: Pengguna dapat memasukkan total biaya layanan yang diterima dalam bentuk nominal.
2. Pilihan Persentase Tip: Pengguna dapat memilih persentase tip yang diinginkan dari opsi yang disediakan, yaitu 15%, 18%, dan 20%.
3. Pengaturan Pembulatan Tip: Pengguna dapat memilih untuk membulatkan tip ke angka yang lebih tinggi.
4. Tampilan Hasil: Aplikasi akan menampilkan jumlah tip yang harus dibayar secara langsung setelah pengguna memberikan input.



Gambar 1 Tampilan Awal Aplikasi



Gambar 2 Tampilan Aplikasi Setelah Dijalankan

A. Source Code

1. MainActivity.kt

```

1 package com.example.kalkulatortip
2
3 import android.os.Bundle
4 import android.widget.Toast
5 import androidx.appcompat.app.AppCompatActivity
6 import androidx.lifecycle.ViewModelProvider
7 import
8     com.example.kalkulatortip.databinding.ActivityMainBinding
9 import kotlin.math.ceil
10
11 class MainActivity : AppCompatActivity() {
12     private lateinit var binding: ActivityMainBinding
13     private lateinit var viewModel: TipViewModel
14
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17
18         binding = ActivityMainBinding.inflate(layoutInflater)
19         setContentView(binding.root)

```

20	
21	viewModel =
	ViewModelProvider(this) [TipViewModel::class.java]
22	
23	viewModel.lastTipResult?.let {
24	binding.textLala.text = it
25	}
26	
27	binding.button.setOnClickListener {
28	calculateTip()
29	}
30	}
31	
32	private fun calculateTip() {
33	val costInput = binding.costInput.text.toString()
34	val cost = costInput.toDoubleOrNull()
35	
36	if (cost == null cost == 0.0) {
37	Toast.makeText(this, "Input tidak valid, silakan
	masukkan angka yang benar", Toast.LENGTH_SHORT).show()
38	val errorResult = getString(R.string.tip_result,
	0.0)
39	binding.textLala.text = errorResult
40	viewModel.lastTipResult = errorResult
41	return
42	}
43	
44	val tipPercentage = when
	(binding.radioGroup.checkedRadioButtonId) {
45	R.id.radio_pirates -> 0.20
46	R.id.radio_ninjas -> 0.18
47	R.id.radio_Capucino -> 0.15
48	else -> 0.15
49	}
50	
51	var tip = cost * tipPercentage
52	if (binding.saveSwitch.isChecked) {
53	tip = ceil(tip)
54	}
55	
56	val result = getString(R.string.tip_result, tip)
57	binding.textLala.text = result
58	viewModel.lastTipResult = result
59	}
60	}

Tabel 1. 1 Source Code Jawaban Soal 1

2. SplashActivty.kt

```
1 package com.example.kalkulatortip
2
3 import android.annotation.SuppressLint
4 import android.content.Intent
5 import android.os.Bundle
6 import android.os.Handler
7 import android.os.Looper
8 import androidx.appcompat.app.AppCompatActivity
9 import
10 com.example.kalkulatortip.databinding.ActivitySplashBinding
11
12 @SuppressLint("CustomSplashScreen")
13 class SplashActivity : AppCompatActivity() {
14
15     private lateinit var binding: ActivitySplashBinding
16
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19
20         binding =
21 ActivitySplashBinding.inflate(layoutInflater)
22         setContentView(binding.root)
23
24         Handler(Looper.getMainLooper()).postDelayed({
25             goToMainActivity()
26         }, 3000L)
27
28     private fun goToMainActivity(){
29         Intent(this, MainActivity::class.java).also{
30             startActivity(it)
31             finish()
32         }
33     }
34 }
```

Tabel 1. 2 Source Code Jawaban Soal 1

3. SplashActivty.kt

```
1 package com.example.kalkulatortip
2
3 import androidx.lifecycle.ViewModel
4
5 class TipViewModel : ViewModel() {
6     var lastTipResult: String? = null
7 }
```

Tabel 1. 3 Source Code Jawaban Soal 1

4. activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <ScrollView
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:background="@color/white"
9      tools:context=".MainActivity">
10     <androidx.constraintlayout.widget.ConstraintLayout
11         android:id="@+id/main"
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         tools:context=".MainActivity">
15
16         <TextView
17             android:id="@+id/textView"
18             android:layout_width="0dp"
19             android:layout_height="wrap_content"
20             android:background="#6804ec"
21             android:padding="20dp"
22             android:text="@string/app_name"
23             android:textColor="#FFFFFF"
24             android:textSize="20sp"
25             app:layout_constraintEnd_toEndOf="parent"
26             app:layout_constraintHorizontal_bias="0.0"
27             app:layout_constraintStart_toStartOf="parent"
28             app:layout_constraintTop_toTopOf="parent" />
29
30         <com.google.android.material.textfield.TextInputLayout
31             android:id="@+id/username_input"
32             android:layout_width="375dp"
33             android:layout_height="wrap_content"
34             android:layout_marginTop="24dp"
35             android:hint="@string/app_Nono"
36             app:helperTextEnabled="true"
37             app:helperText="@string/app_Nene"
38             app:endIconMode="clear_text"
39             app:layout_constraintEnd_toEndOf="parent"
40             app:layout_constraintHorizontal_bias="0.497"
41             app:layout_constraintStart_toStartOf="parent"
42             app:layout_constraintTop_toBottomOf="@+id/textView">
43
44         <com.google.android.material.textfield.TextInputEditText
45             android:id="@+id/cost_input"
46             android:layout_width="match_parent"
47             android:layout_height="wrap_content"
48             android:inputType="text" />
49     </com.google.android.material.textfield.TextInputLayout>
```

```

50
51
52     <RadioGroup
53         android:id="@+id/radioGroup"
54         android:layout_width="375dp"
55         android:layout_height="wrap_content"
56         android:orientation="vertical"
57         app:layout_constraintEnd_toEndOf="parent"
58         app:layout_constraintStart_toStartOf="parent"
59         app:layout_constraintTop_toBottomOf="@+id/username_input">
60
61         <RadioButton
62             android:id="@+id/radio_pirates"
63             android:layout_width="wrap_content"
64             android:layout_height="wrap_content"
65             android:text="@string/app_legam" />
66
67         <RadioButton
68             android:id="@+id/radio_ninjas"
69             android:layout_width="wrap_content"
70             android:layout_height="wrap_content"
71             android:text="@string/app_white" />
72
73         <RadioButton
74             android:id="@+id/radio_Capucino"
75             android:layout_width="wrap_content"
76             android:layout_height="wrap_content"
77             android:text="@string/app_coklat" />
78
79     </RadioGroup>
80
81     <com.google.android.material.materialswitch.MaterialSwitch
82         android:id="@+id/save_switch"
83         android:layout_width="375dp"
84         android:layout_height="wrap_content"
85         android:checked="false"
86         android:text="@string/app_Nunu"
87         app:layout_constraintEnd_toEndOf="parent"
88         app:layout_constraintHorizontal_bias="0.497"
89         app:layout_constraintStart_toStartOf="parent"
90         app:layout_constraintTop_toBottomOf="@+id/radioGroup"
91     />
92
93     <com.google.android.material.button.MaterialButton
94         android:id="@+id/button"
95         android:layout_width="375dp"
96         android:layout_height="wrap_content"
97         android:text="@string/app_Medium"
98         android:backgroundTint="#6804ec"
99         android:textColor="@color/white"
100        app:cornerRadius="5dp"
        app:layout_constraintEnd_toEndOf="parent"

```

101	app:layout_constraintHorizontal_bias="0.497"
102	app:layout_constraintStart_toStartOf="parent"
103	app:layout_constraintTop_toBottomOf="@+id/save_switch"
104	android:layout_marginBottom="32dp" />
105	
106	<TextView
107	android:id="@+id/textLala"
108	android:layout_width="0dp"
109	android:layout_height="wrap_content"
110	android:paddingEnd="20dp"
111	android:text="@string/app_Nana"
112	android:textColor="#4f4f4f"
113	android:textSize="14sp"
114	app:layout_constraintEnd_toEndOf="parent"
115	app:layout_constraintHorizontal_bias="0.0"
116	app:layout_constraintTop_toBottomOf="@+id/button"/>
117	
118	</androidx.constraintlayout.widget.ConstraintLayout>
119	</ScrollView>

Tabel 1. 4 Source Code Jawaban Soal 1

5. activity_splash.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
	xmlns:android="http://schemas.android.com/apk/res/android"
3	android:layout_width="match_parent"
4	android:layout_height="match_parent"
5	xmlns:app="http://schemas.android.com/apk/res-auto"
6	xmlns:tools="http://schemas.android.com/tools"
7	tools:context=".SplashActivity"
8	android:id="@+id/splash"
9	android:background="@color/white">
10	
11	<androidx.constraintlayout.widget.ConstraintLayout
12	android:layout_width="wrap_content"
13	android:layout_height="wrap_content"
14	app:layout_constraintStart_toStartOf="parent"
15	app:layout_constraintEnd_toEndOf="parent"
16	app:layout_constraintTop_toTopOf="parent"
17	app:layout_constraintBottom_toBottomOf="parent">
18	<androidx.appcompat.widget.AppCompatImageView
19	android:id="@+id/iv_logo"
20	android:layout_width="200dp"
21	android:background="@color/white"
22	android:layout_height="200dp"
23	app:srcCompat="@drawable/amer_kalkulator"
24	app:layout_constraintStart_toStartOf="parent"
25	app:layout_constraintEnd_toEndOf="parent"
26	app:layout_constraintTop_toTopOf="parent"

27	app:layout_constraintBottom_toBottomOf="parent"/>
28	
29	</androidx.constraintlayout.widget.ConstraintLayout>
30	
31	<ProgressBar
32	style="@style/Widget.AppCompat.ProgressBar.Horizontal"
33	android:layout_width="200dp"
34	android:layout_height="32dp"
35	android:indeterminate="true"
36	app:layout_constraintStart_toStartOf="parent"
37	app:layout_constraintEnd_toEndOf="parent"
38	app:layout_constraintBottom_toBottomOf="parent"
39	android:indeterminateTint="#013B43"
40	tools:layout_editor_absoluteX="106dp"
41	tools:layout_editor_absoluteY="611dp" />
42	
43	</androidx.constraintlayout.widget.ConstraintLayout>

Tabel 1. 5 Source Code Jawaban Soal 1

B. Output Program

5:32

Kalkulator Tip

Cost of Service

How was the service?

☐ Amazing (20%)

☐ Good (18%)

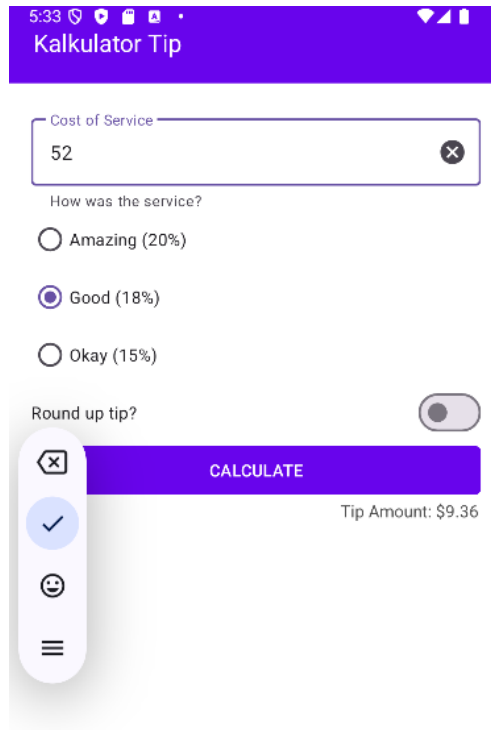
☐ Okay (15%)

Round up tip? ☐

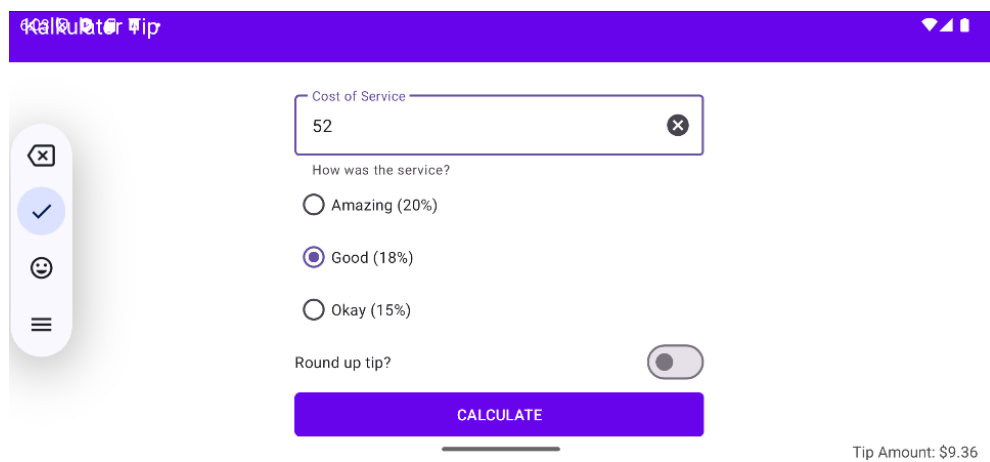
CALCULATE

Tip Amount

Gambar 1 Screenshot Hasil Jawaban Soal 1 Awalan



Gambar 2 Screenshot Hasil Jawaban Soal 1



Gambar 3 Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt

Pada line [1] – [8] kita perlu melakukan deklarasi nama package file Kotlin agar dapat menentukan lokasi file sesuai struktur project. Dimana disini ada `import` yang berguna saat kita ingin mengimpor kelas-kelas dibutuhkan seperti, `Toast` untuk menampilkan pesan pop-up, `ViewModelProvider` berguna untuk menghubungkan `ViewModel` ke `activity`, `ActivityMainBinding` untuk mengakses view lewat `View Binding`, ceil bagian dari fungsi Kotlin untuk membulatkan angka ke atas.

Pada di line [10], [12], [13], dan [60] deklarasi juga inisialisasi ada di baris ini karena `MainActivity` turunan `AppCompatActivity` adalah komponen utama layar Android. Selain itu, ada binding berguna untuk mengakses elemen UI dari layout `activity_main.xml`. Serta, `viewModel` membantu dalam menyimpan data seperti hasil tip agar tetap tersedia saat rotasi layar.

Pada di line [15], [16], dan [41] ada `onCreate()` karena fungsi ini dipanggil saat `activity` dibuat pertama kali dengan `savedInstanceState` berguna untuk menyimpan data saat kita melakukan rotasi layar, dll. Setelah itu, di line [18] – [19] perlu melakukan inisialisasi layout karena menggunakan `View Binding` untuk mempermudah akses elemen UI jangan lupa disini ada `setContentView(binding.root)` berguna saat menampilkan layout XML ke layar. Selanjutnya, di line [21] ada `viewModel = ViewModelProvider(...)` berguna saat ingin menghubungkan `ViewModel` agar data tetap ada meskipun aktivitas sudah direstart. Berikutnya, di line [23] – [25] ada `viewModel.lastTipResult?.let` ketika ada hasil tip sebelumnya yang akan tampil di `textLala`. Serta, di line [27] – [29] `binding.button.setOnClickListener` digunakan saat tombol kita tekan nantinya bisa menjalankan fungsi `calculateTip()`.

Pada line [32] – [59] adalah fungsi `calculateTip()`. Kemudian, di line [32] – [34] berguna untuk mengambil input yang kita masukkan dan ubah ke tipe

Double? jika tidak valid nantinya menjadi `null`. Lajutan ada di line [36] – [42] untuk bagian validasi input apabila kosong atau 0, tampilkan toast dan set hasil ke 0. Lalu, di line [49] – [49] untuk kita bisa pilih presentase tip sesuai `radio button` yang sudah dipilih. Selanjutnya, [51] – [54] adalah bagian bagaimana menghitung tip biaya `x persen` dengan kondisi jika switch "save" aktif nantinya bulatkan ke atas. Berikutnya, di line [56] – [58] kit berhasil membuat format hasil tip dengan string dari `strings.xml` yang nantinya ditampilkan ke UI dan simpan ke bagian `ViewModel`.

2. **Splash_Activity.kt**

Pada line [1] – [9] kita perlu melakukan deklarasi nama package file Kotlin agar dapat menentukan lokasi file sesuai struktur project. Dimana disini ada `import` yang berguna saat kita ingin mengimpor kelas-kelas dibutuhkan seperti, `SuppressLint` berguna att ingin menonaktifkan warning tertentu dari Android Studio, `Intent` untuk memudahkan ketika kita berpindah dari satu activity ke activity lain, `Handler` dan `Looper` untuk menjalankan kode dengan penundaan (delay), serta `ActivtySplashBinding` biasanya binding secara otomatis ke layout XML splash screen.

Pada line [12], [13], [15], dan [34] adalah deklarasi class juga bagian view binding menggunakan `SplashActivity` karena class yang ditampilkan pertama saat aplikasi dibuka dengan `@SuppressLint("CustomSplashScreen")` agar bisa menonaktifkan peringatan dari Android 12 ke atas punya Splash API bawaan, tapi di sini kita bikin manual melibatkann binding untuk bisa mengakses elemen UI di layout `activty_splash.xml`.

Pada line [17] – [26] adalah fungsi `onCreate()` yang dipanggil saat kita menggunakan activity pertama kali dibuat melibatkan `binding = ...inflate(layoutInflater)` agar bisa menghubungkan layout splash screen menuju bagian activity dengan `setContentView(binding.root)` memudahkan saat menampilkan tampilan splash ke layar. Kemudian, `Handler(...)` berguna saat ingin menjalankan kode setelah delay (3 detik /

3000ms). Lalu, ada `postDelayed({ ... }, 3000L)` berguna untuk menunda fungsi `goToMainActivity()` selama 3 detik.

Pada line [28] – [33] adalah fungsi `goToMainActivity()` dengan membuat `Intent` berguna untuk berpindah dari `SplashActivity` menuju `MainActivity` dengan `startActivity(it)` agar bisa memulai activity utama dan ada `finish()` untuk menutup `SplashActivity` supaya tidak kembali ke splash saat kita menekan tombol back.

3. TipViewModel

Pada line [1] dan [3] kita perlu melakukan deklarasi nama package file Kotlin agar dapat menentukan lokasi file sesuai struktur project. Dimana disini ada `import androidx.lifecycle.ViewModel` berguna untuk mengimpor `ViewModel` dari Android Jetpack, bagian dari Android Architecture Components.

Pada line [5] – [7] adalah class `TipViewModel` karena disini ada class `TipViewModel : ViewModel()` berguna saat membuat class `TipViewModel` turunan dari class `ViewModel` melibatkan `var lastTipResult: String? = null` sebagai properti dalam menyimpan hasil terakhir kalkulasi tip atau perhitungan tip dengan tipe datanya `String?`, artinya bisa bernilai teks atau `null` hasilnya.

4. activity_main.xml

Pada line [1] – [8] juga [119] adalah bagian dari deklarasi awal dan `ScrollView` dimana ada `<?xml ... ?>` untuk deklarasi XML standar dengan `<ScrollView>` sebagai layout vertikal yang bisa di scroll, digunakan agar semua konten di dalamnya tetap bisa diakses karena ada bagian yang tidak terlihat di layar saat kita rotate meskipun lebih tinggi dari layar tersebut. Lalu, di line [5] – [6] `android:layout_width="match_parent"` berguna sebagai lebar mengikuti layar juga ada `android:layout_height="match_parent"` untuk tingginya bisa mengikuti layar. Kemudian,

`android:background="@color/white"` menjadi latar belakang putih. Serta, `tools:context=".MainActivity"` untuk menandakan file ini digunakan di `MainActivity` saat preview di Android Studio.

Pada line [10] – [14] juga [118] terdapat struktur utama layout terdiri dari `ConstraintLayout` merupakan Layout fleksibel yang bisa memberikan kemungkinan ke kita bagaimana cara mengatur posisi elemen UI berdasarkan hubungan antar elemen melibatkan `match_parent` untuk lebar dan tinggi layout mengikuti ukuran layar penuh yang menyesuaikan isi di dalamnya. Juga bagian ini cocok untuk desain kompleks dan responsif.

Pada line [16] – [28] adalah bagian judul aplikasi (`TextView`) untuk menampilkan nama aplikasi. Dimana ada `android:background="#6804ec"` sebagai background warna ungu tua dengan `android:textColor="#FFFFFF"` agar menjadi teks putih yang ukuran teksnya `20sp` melibatkan `app:layout_constraintTop_toTopOf="parent"` biasanya ditempatkan di bagian atas layar dengan semua ukuran sisi padding `20dp`.

Pada line [30] – [49] adalah bagian input Harga (`TextInputLayout` + `TextInputEditText`) karena ada komponen material design untuk form input lebih rapi dengan `TextInputLayout` dapat memberi fitur tambahan seperti, hint, helper text, dan ikon (`endIconMode="clear_text"` ikon silang untuk menghapus teks dengan cepat). Kemudian, disini ada `TextInputEditText` sebagai field tempat kita memasukkan nilai biaya nantinya dihitung tip-nya. Lalu, ada Hint & helper text diambil dari `strings.xml` (`@string/app_Nono` dan `@string/app_Nene`) untuk teks bantu di bawah input.

Pada line [52] – [79] adalah bagian `RadioGroup`: Pilihan Tip. Ada `RadioGroup` berguna untuk kita atau pengguna memilih salah satu persentase tip dari 3 pilihan seperti, `radio_pirates = 20%`, `radio_ninjas = 18%`, dan `radio_Capucino = 15%` dengan hanya bisa memilih satu dari tiga radio button ini. Lalu, label teks diatur nantinya dari `strings.xml`, misal

@string/app_legam, @string/app_white, dan
@string/app_coklat.

Pada line [81] – [90] adalah bagian `materialSwitch` atau biasanya disebut Switch Bulat Modern (on/off) karena Switch ini digunakan untuk menentukan apakah tip perlu dibulatkan ke atas atau tidak oleh `ceil()` disini dengan berisi properti seperti, `android:checked="false"` untuk default-nya saat posisi off, `android:text="@string/app_Nunu"` sebagai label switch-nya, dan Layout-nya berada di bawah `RadioGroup`.

Pada line [92] – [104] adalah bagian `MaterialButton` atau tombol hitung tip untuk memicu jalannya kalkulasi tip (`button.setOnClickListener()`) di kode kotlin kita buat dengan di dalamnya ada tampilan `android:text="@string/app_Medium"` sebagai label tombol, `android:backgroundTint="#6804ec"` agar menjadi warna ungu dengan `android:textColor="@color/white"` sebagai teks putih, dan `app:cornerRadius="5dp"` untuk sudutnya bisa agak membulat atau tombol sedikit melengkung.

Pada line [106] – [116] adalah bagian `TextView` hasil tip untuk menampilkan hasil tip setelah tombol kita klik karena digunakan sebagai catatan atau disclaimer letaknya di bawah tombol melibatkan `android:textSize="14sp"` dan `android:textColor="#4f4f4f"` menjadi tampilan teks dengan warna abu-abu tua (`#4f4f4f`), ukuran kecil (`14sp`), teks ini dari `strings.xml`.

5. activity_splash.xml

Pada line [1] – [9] juga [43] adalah bagian `Root Layout: ConstraintLayout` merupakan Layout fleksibel yang bisa memberikan kemungkinan ke kita bagaimana cara mengatur posisi elemen UI berdasarkan hubungan antar elemen berdasarkan constraint melibatkan `match_parent` untuk lebar dan tinggi layout mengikuti ukuran layar penuh dengan `android:background="@color/white"` sebagai latar belakang layar splash

putih serta ada `tools:context=".SplashActivity"` agar menunjukkan layout ini digunakan oleh `SplashActivity`.

Pada line [11] – [17] juga [29] adalah bagian `Inner ConstraintLayout` yang menjadi pusat konten karena layout ini digunakan agar bisa menempatkan logo yang sudah kita buat di tengah layar secara horizontal dan vertical melibatkan `wrap_content` berarti ukuran layout akan menyesuaikan kontennya dengan *Di-constraint* ke semua sisi (Top, Bottom, Start, End) ke parent, sehingga berada di tengah layar.

Pada line [18] – [27] adalah bagian logo aplikasi (`AppCompatActivity`) untuk menampilkan gambar atau logo aplikasi dengan ukuran `200x200dp` melibatkan `app:srcCompat="@drawable/amer_kalkulator"` agar bisa menampilkan gambar bernama `amer_kalkulator.png` atau `.xml` dari folder `res/drawable/` dimana background digunakan putih. Serta perlu *di-constraint* ke semua sisi layout atasnya, sehingga logo tetap berada di tengah layar.

Pada line [31] – [41] adalah bagian `ProgressBar` atau loading. Dimana `ProgressBar` horizontal berguna untuk menunjukkan bahwa aplikasi sedang ada di proses memuat. Kemudian, `indeterminate="true"` untuk tidak menunjukkan progress tertentu dengan hanya ada animasi loading terus menerus. Lalu, `indeterminateTint="#013B43"` sebagai warna loading bar-nya itu hijau toska dengan ditempatkan di bawah layar atau bottom constraint menuju parent.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/alysaarmelia/AlysaArmelia_2310817120009_Pemrograman_Mobile/tree/afc64f78956262f45e9c546d93c755f8b4e2d36c/PRAK_MODUL2