

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2**



ANDROID LAYOUT

Oleh:

Raudatul Sholehah

NIM. 2310817220002

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2024**

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 : 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	15
C. Pembahasan	17
D. Tautan Git.....	22

DAFTAR GAMBAR

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

DAFTAR TABEL

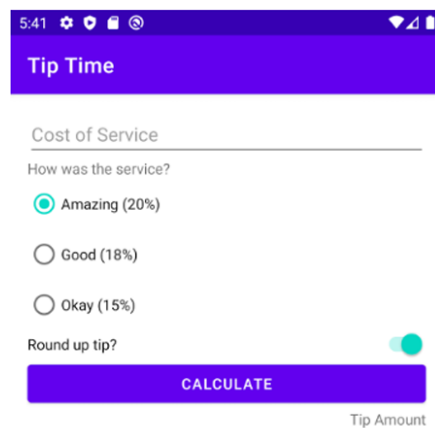
Tabel 1. Source Code Jawaban Soal 1	9
Tabel 2. Source Code Jawaban Soal 1	9
Tabel 3. Source Code Jawaban Soal 1	12
Tabel 4. Source Code Jawaban Soal 1	13
Tabel 5. Source Code Jawaban Soal 1	13
Tabel 6. Source Code Jawaban Soal 1	13
Tabel 7. Source Code Jawaban Soal 1	14
Tabel 8. Source Code Jawaban Soal 1	14

SOAL 1

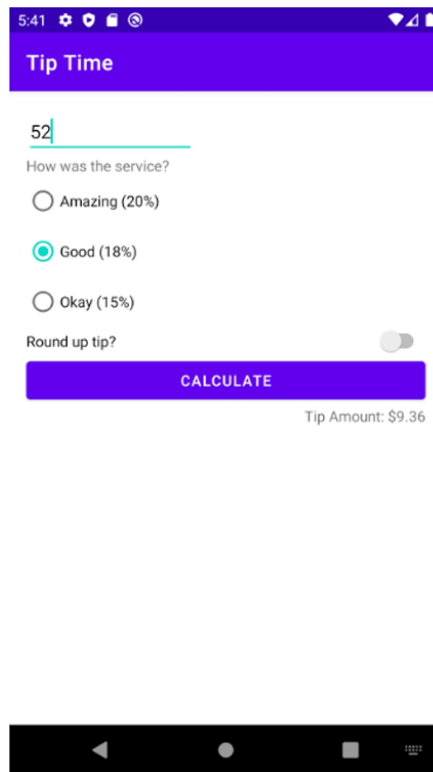
Soal Praktikum:

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



Tampilan Aplikasi Setelah Dijalankan

A. Source Code

1. MainActivity.kt

```
1 package com.example.tiptime
2
3 import android.os.Bundle
4 import android.widget.Toast
5 import androidx.activity.viewModels
6 import androidx.appcompat.app.AppCompatActivity
7 import com.example.tiptime.databinding.ActivityMainBinding
8
9 class MainActivity : AppCompatActivity() {
10
11     private lateinit var binding: ActivityMainBinding
12     private val viewModel: MainViewModel by viewModels()
13
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16
17         binding = ActivityMainBinding.inflate(layoutInflater)
18         setContentView(binding.root)
19
20         binding.costOfService.setText(viewModel.costInput)
21         binding.roundUpSwitch.isChecked = viewModel.roundUp
22
23         viewModel.tipResult.observe(this) { tip ->
24             binding.tipResult.text =
25                 getString(R.string.tip_amount) + ": " + tip
26         }
27
28         if (viewModel.costInput.isNotBlank()) {
29             viewModel.calculateTip()
30         }
31
32         binding.calculateButton.setOnClickListener {
33             val stringInTextField =
34                 binding.costOfService.text.toString()
35             val cost = stringInTextField.toDoubleOrNull()
36
37             if (cost == null || cost == 0.0) {
38                 binding.tipResult.text = ""
39                 binding.costOfService.error = "Masukkan angka
40 yang valid dan lebih dari 0!"
41                 Toast.makeText(this, "Masukkan nilai yang valid
42 dan lebih dari 0!", Toast.LENGTH_SHORT).show()
43                 return@setOnClickListener
44             } else {
45                 binding.costOfService.error = null
46             }
47
48             val tipPercentage = when
49                 (binding.tipOptions.checkedRadioButtonId) {
```


45	<code>R.id.option_twenty_percent -> 0.20</code>
46	<code>R.id.option_eighteen_percent -> 0.18</code>
47	<code>else -> 0.15</code>
48	<code>}</code>
49	
50	<code>viewModel.costInput = stringInTextField</code>
51	<code>viewModel.tipPercentage = tipPercentage</code>
52	<code>viewModel.roundUp = binding.roundUpSwitch.isChecked</code>
53	
54	<code>viewModel.calculateTip()</code>
55	<code>}</code>
56	<code>}</code>
57	<code>}</code>

Tabel 1. Source Code Jawaban Soal 1

2. MainViewModel.kt

1	<code>package com.example.tiptime</code>
2	
3	<code>import androidx.lifecycle.LiveData</code>
4	<code>import androidx.lifecycle.MutableLiveData</code>
5	<code>import androidx.lifecycle.ViewModel</code>
6	<code>import java.text.NumberFormat</code>
7	<code>import kotlin.math.ceil</code>
8	
9	<code>class MainViewModel : ViewModel() {</code>
10	<code> private val _tipResult = MutableLiveData<String>()</code>
11	<code> val tipResult: LiveData<String> = _tipResult</code>
12	
13	<code> var costInput: String = ""</code>
14	<code> var tipPercentage: Double = 0.15</code>
15	<code> var roundUp: Boolean = false</code>
16	
17	<code> fun calculateTip() {</code>
18	<code> val cost = costInput.toDoubleOrNull()</code>
19	<code> if (cost == null) {</code>
20	<code> _tipResult.value = ""</code>
21	<code> return</code>
22	<code> }</code>
23	
24	<code> var tip = tipPercentage * cost</code>
25	<code> if (roundUp) {</code>
26	<code> tip = ceil(tip)</code>
27	<code> }</code>
28	
29	<code> val formattedTip =</code>
30	<code>NumberFormat.getCurrencyInstance().format(tip)</code>
31	<code> _tipResult.value = formattedTip</code>
32	<code> }</code>

Tabel 2. Source Code Jawaban Soal 1

3. 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:importantForAccessibility="no"
9      android:fillViewport="true"
10     tools:context=".MainActivity">
11
12     <androidx.constraintlayout.widget.ConstraintLayout
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:padding="16dp">
16
17         <EditText
18             android:id="@+id/cost_of_service"
19             android:layout_width="0dp"
20             android:layout_height="wrap_content"
21             android:autofillHints=""
22             android:hint="@string/cost_of_service"
23             android:inputType="numberDecimal"
24             android:minHeight="48dp"
25             app:layout_constraintStart_toStartOf="parent"
26             app:layout_constraintEnd_toEndOf="parent"
27             app:layout_constraintTop_toTopOf="parent" />
28
29         <TextView
30             android:id="@+id/service_question"
31             android:layout_width="wrap_content"
32             android:layout_height="wrap_content"
33             android:text="@string/how_was_the_service"
34             app:layout_constraintStart_toStartOf="parent"
35
36             app:layout_constraintTop_toBottomOf="@id/cost_of_service" />
37
38         <RadioGroup
39             android:id="@+id/tip_options"
40             android:layout_width="wrap_content"
41             android:layout_height="wrap_content"
42             android:checkedButton="@id/option_twenty_percent"
43             android:orientation="vertical"
44             app:layout_constraintStart_toStartOf="parent"
45
46             app:layout_constraintTop_toBottomOf="@id/service_question">
47
48             <RadioButton
49                 android:id="@+id/option_twenty_percent"
50                 android:layout_width="wrap_content"
```

```

48         android:layout_height="wrap_content"
49         android:text="@string/amazing_20"
50         android:checked="true"/>
51
52     <RadioButton
53         android:id="@+id/option_eighteen_percent"
54         android:layout_width="wrap_content"
55         android:layout_height="wrap_content"
56         android:text="@string/good_18" />
57
58     <RadioButton
59         android:id="@+id/option_fifteen_percent"
60         android:layout_width="wrap_content"
61         android:layout_height="wrap_content"
62         android:text="@string/okay_15" />
63 </RadioGroup>
64
65     <Switch
66         android:id="@+id/round_up_switch"
67         android:layout_width="0dp"
68         android:layout_height="wrap_content"
69         android:checked="true"
70         android:minHeight="48dp"
71         android:text="@string/round_up_tip"
72         app:layout_constraintEnd_toEndOf="parent"
73         app:layout_constraintStart_toStartOf="@id/tip_options"
74         app:layout_constraintTop_toBottomOf="@id/tip_options"
75         tools:ignore="UseSwitchCompatOrMaterialXml,VisualLintButtonSize"
76     />
77
78     <Button
79         android:id="@+id/calculate_button"
80         android:layout_width="0dp"
81         android:layout_height="wrap_content"
82         android:text="@string/calculate"
83         app:layout_constraintEnd_toEndOf="parent"
84         app:layout_constraintStart_toStartOf="parent"
85         app:layout_constraintTop_toBottomOf="@id/round_up_switch"
86         tools:ignore="VisualLintButtonSize" />
87
88     <TextView
89         android:id="@+id/tip_result"
90         android:layout_width="wrap_content"
91         android:layout_height="wrap_content"
92         android:text="@string/tip_amount"
93         android:layout_marginBottom="24dp"
94         app:layout_constraintEnd_toEndOf="parent"

```

94	app:layout_constraintTop_toBottomOf="@id/calculate_button"
95	tools:text="Tip Amount: \$10" />
96	
97	</androidx.constraintlayout.widget.ConstraintLayout>
98	</ScrollView>

Tabel 3. Source Code Jawaban Soal 1

4. build.gradle (Module :app)

1	plugins {
2	alias(libs.plugins.android.application)
3	alias(libs.plugins.kotlin.android)
4	}
5	
6	android {
7	buildFeatures {
8	viewBinding = true
9	}
10	namespace = "com.example.tiptime"
11	compileSdk = 35
12	
13	defaultConfig {
14	applicationId = "com.example.tiptime"
15	minSdk = 30
16	targetSdk = 35
17	versionCode = 1
18	versionName = "1.0"
19	
20	testInstrumentationRunner =
21	"androidx.test.runner.AndroidJUnitRunner"
22	}
23	buildTypes {
24	release {
25	isMinifyEnabled = false
26	proguardFiles(
27	getDefaultProguardFile("proguard-android-
28	optimize.txt"),
29	"proguard-rules.pro"
30)
31	}
32	compileOptions {
33	sourceCompatibility = JavaVersion.VERSION_11
34	targetCompatibility = JavaVersion.VERSION_11
35	}
36	kotlinOptions {
37	jvmTarget = "11"
38	}
39	}
40	
41	dependencies {

42	
43	<code>implementation(libs.androidx.core.ktx)</code>
44	<code>implementation(libs.androidx.appcompat)</code>
45	<code>implementation(libs.material)</code>
46	<code>implementation(libs.androidx.activity)</code>
47	<code>implementation(libs.androidx.constraintlayout)</code>
48	<code>testImplementation(libs.junit)</code>
49	<code>androidTestImplementation(libs.androidx.junit)</code>
50	<code>androidTestImplementation(libs.androidx.espresso.core)</code>
51	<code>}</code>

Tabel 4. Source Code Jawaban Soal 1

5. colors.xml

1	<code><?xml version="1.0" encoding="utf-8"?></code>
2	<code><resources></code>
3	<code> <color name="purple_200">#FFBB86FC</color></code>
4	<code> <color name="purple_500">#FF6200EE</color></code>
5	<code> <color name="purple_700">#FF3700B3</color></code>
6	<code> <color name="teal_200">#FF03DAC5</color></code>
7	<code> <color name="teal_700">#FF018786</color></code>
8	<code> <color name="black">#FF000000</color></code>
9	<code> <color name="white">#FFFFFFFF</color></code>
10	<code></resources></code>

Tabel 5. Source Code Jawaban Soal 1

6. strings.xml

1	<code><resources></code>
2	<code> <string name="app_name">TipTime</string></code>
3	
4	<code> <string name="cost_of_service">Cost of Service</string></code>
5	<code> <string name="how_was_the_service">How was the</code>
	<code>service?</string></code>
6	<code> <string name="amazing_20">Amazing (20%)</string></code>
7	<code> <string name="good_18">Good (18%)</string></code>
8	<code> <string name="okay_15">Okay (15%)</string></code>
9	<code> <string name="round_up_tip">Round up tip?</string></code>
10	<code> <string name="calculate">Calculate</string></code>
11	<code> <string name="tip_amount">Tip Amount </string></code>
12	<code></resources></code>

Tabel 6. Source Code Jawaban Soal 1

7. themes.xml

1	<code><resources xmlns:tools="http://schemas.android.com/tools"></code>
2	<code> <!-- Base application theme. --></code>
3	<code> <style name="Theme.TipTime"</code>
	<code>parent="Theme.MaterialComponents.DayNight.DarkActionBar"></code>
4	<code> <!-- Primary brand color. --></code>
5	<code> <item name="colorPrimary">@color/purple_500</item></code>
6	<code> <item</code>
	<code>name="colorPrimaryVariant">@color/purple_700</item></code>

7	<item name="colorOnPrimary">@color/white</item>
8	<!-- Secondary brand color. -->
9	<item name="colorSecondary">@color/teal_200</item>
10	<item
	name="colorSecondaryVariant">@color/teal_700</item>
11	<item name="colorOnSecondary">@color/black</item>
12	<!-- Status bar color. -->
13	<item
	name="android:statusBarColor">?attr/colorPrimaryVariant</item>
14	<!-- Customize your theme here. -->
15	</style>
16	</resources>

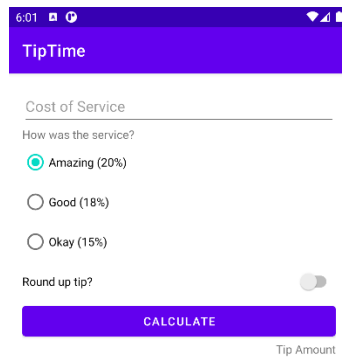
Tabel 7. Source Code Jawaban Soal 1

8. themes.xml (night)

1	<resources xmlns:tools="http://schemas.android.com/tools">
2	<!-- Base application theme. -->
3	<style name="Theme.TipTime"
	parent="Theme.MaterialComponents.DayNight.DarkActionBar">
4	<!-- Primary brand color. -->
5	<item name="colorPrimary">@color/purple_200</item>
6	<item
	name="colorPrimaryVariant">@color/purple_700</item>
7	<item name="colorOnPrimary">@color/black</item>
8	<!-- Secondary brand color. -->
9	<item name="colorSecondary">@color/teal_200</item>
10	<item
	name="colorSecondaryVariant">@color/teal_200</item>
11	<item name="colorOnSecondary">@color/black</item>
12	<!-- Status bar color. -->
13	<item
	name="android:statusBarColor">?attr/colorPrimaryVariant</item>
14	<!-- Customize your theme here. -->
15	</style>
16	</resources>

Tabel 8. Source Code Jawaban Soal 1

B. Output Program



6:01

TipTime

Cost of Service

How was the service?

☒ Amazing (20%)

☐ Good (18%)

☐ Okay (15%)

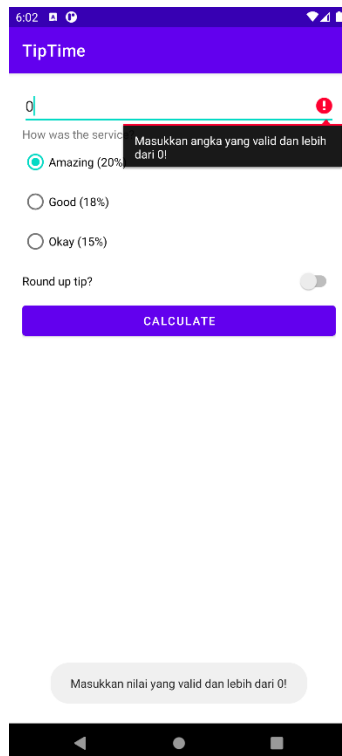
Round up tip? ☐

CALCULATE

Tip Amount



Gambar 1. Screenshot Hasil Jawaban Soal 1



6:02

TipTime

0

How was the service?

☒ Amazing (20%)

☐ Good (18%)

☐ Okay (15%)

Round up tip? ☐

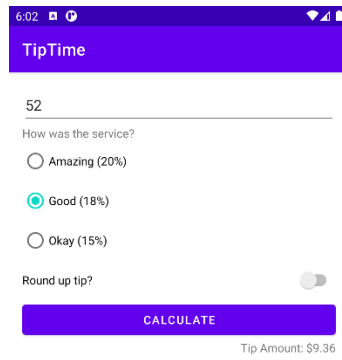
CALCULATE

Masukkan angka yang valid dan lebih dari 0!

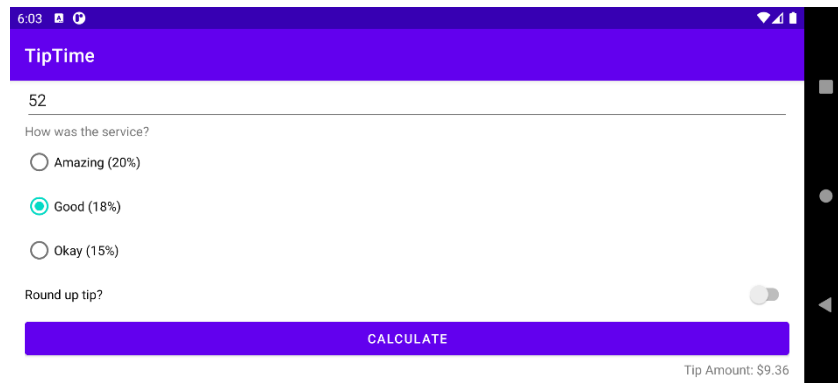
Masukkan nilai yang valid dan lebih dari 0!



Gambar 2. Screenshot Hasil Jawaban Soal 1



Gambar 3. Screenshot Hasil Jawaban Soal 1



Gambar 4. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt

- Baris 1–7:
Deklarasi package dan import library yang diperlukan. `package com.example.tiptime` menunjukkan lokasi file ini dalam struktur proyek. Library yang diimpor mencakup yaitu `Bundle` dan `Toast` dari `Android` untuk fungsi dasar dan notifikasi kepada pengguna. `viewModels` dari `AndroidX` untuk menghubungkan `ViewModel`. `AppCompatActivity` sebagai kelas dasar untuk `activity`. `ActivityMainBinding` dari `ViewBinding` untuk mengakses komponen UI secara langsung tanpa `findViewById`.
- Baris 9–12:
Deklarasi kelas `MainActivity` yang merupakan turunan dari `AppCompatActivity`. Di dalamnya terdapat dua property yaitu `binding`, yang digunakan untuk mengakses view di layout `activity_main.xml`. `viewModel`, instance dari `MainViewModel`, digunakan untuk menyimpan data tip dan bertahan terhadap perubahan konfigurasi seperti rotasi layar.
- Baris 14–18:
Method `onCreate` dipanggil ketika `activity` pertama kali dibuat. `binding` diinisialisasi menggunakan `ActivityMainBinding.inflate(layoutInflater)`. `setContentView(binding.root)` digunakan untuk menampilkan tampilan yang sudah dibinding.
- Baris 20–21:
Mengisi UI dengan data awal dari `ViewModel` yaitu nilai input biaya layanan (`costInput`) di-set ke `EditText`. Status switch untuk pembulatan (`roundUp`) disetel sesuai dengan data di `ViewModel`.
- Baris 23–25:
Mengamati perubahan `tipResult` dari `ViewModel`. Ketika hasil tip berubah, `TextView` akan diperbarui secara otomatis untuk menampilkan jumlah tip dalam format string yang sesuai.
- Baris 27–29:
Jika `costInput` tidak kosong saat `activity` dibuka, maka `calculateTip()` langsung dipanggil untuk menghitung ulang tip berdasarkan nilai yang terakhir dimasukkan. Ini mendukung fitur state persistence pada `ViewModel`.
- Baris 31–54:
Menangani logika ketika tombol "Calculate" diklik. Fungsi pada baris 32–33 untuk mengambil input dan mengubahnya ke `Double?`. Baris 35–39 untuk validasi input, jika kosong atau nol, tampilkan pesan kesalahan dan batalkan proses. Pada baris 40–41 berguna membersihkan error jika input valid. Baris 43–47 menentukan persentase tip berdasarkan radio button yang dipilih. Baris 49–51 menyimpan input, persentase, dan status pembulatan ke `ViewModel` dan pada baris 53 untuk memanggil `calculateTip()` untuk menghitung jumlah tip baru.

2. MainViewModel.kt

- Baris 1-6:
package com.example.tiptime menunjukkan lokasi file dalam struktur proyek. LiveData, MutableLiveData, dan ViewModel digunakan dari Android Jetpack untuk menerapkan arsitektur MVVM. NumberFormat dari Java untuk memformat angka menjadi bentuk mata uang. ceil dari kotlin.math digunakan untuk membulatkan angka ke atas jika opsi pembulatan diaktifkan.
- Baris 8-10:
Deklarasi kelas MainViewModel yang mewarisi ViewModel. Kelas ini bertugas menyimpan dan mengelola data yang digunakan oleh UI, agar tetap bertahan saat terjadi perubahan konfigurasi seperti rotasi layar.
- Baris 9-10:
_tipResult adalah objek MutableLiveData yang dapat diubah dari dalam ViewModel. tipResult adalah LiveData bersifat read-only yang dapat diamati oleh UI. Ini adalah cara yang baik untuk membatasi akses data agar tidak bisa diubah dari luar ViewModel.
- Baris 12-14:
Deklarasi tiga properti utama yaitu costInput untuk menyimpan input biaya layanan dalam bentuk String. tipPercentage menyimpan persentase tip yang dipilih, default-nya 15%. roundup boolean yang menandakan apakah hasil tip dibulatkan ke atas atau tidak.
- Baris 16-25:
Fungsi utama yang digunakan untuk menghitung jumlah tip diantaranya pada baris 17 untuk mengonversi input ke Double. Jika input tidak valid, hasil dikosongkan dan fungsi selesai. Baris 20 untuk menghitung tip awal sebagai hasil dari cost * tipPercentage. Pada baris 21-23 jika opsi roundUp aktif, nilai tip dibulatkan ke atas menggunakan ceil(). Baris 24: Nilai tip diformat ke bentuk mata uang lokal (misalnya Rp, \$, dll). Baris 25: Hasil diformat disimpan ke dalam _tipResult, yang akan dipantau oleh UI.

3. activity_main.xml

- Baris 1-9
Bagian pembuka layout terdiri dari ScrollView yang digunakan sebagai container utama agar tampilan bisa di-scroll saat layar tidak cukup besar. Atribut seperti layout_width="match_parent" dan layout_height="match_parent" memastikan tampilan mengisi seluruh layar. fillViewport="true" memastikan isi tampilan mengisi seluruh tinggi layar meskipun tidak cukup panjang untuk di-scroll. tools:context=".MainActivity" digunakan oleh Android Studio untuk mengetahui activity yang menggunakan layout ini, berguna saat preview UI.
- Baris 11-13:
ConstraintLayout digunakan sebagai layout utama di dalam ScrollView yaitu untuk memberikan kontrol posisi yang fleksibel terhadap elemen UI dan

menggunakan `padding="16dp"` untuk memberi jarak antar elemen dari tepi layar.

- Baris 15–23:
Input teks untuk memasukkan biaya layanan terdiri dari `inputType="numberDecimal"` membatasi input ke angka desimal, `hint="@string/cost_of_service"` menampilkan petunjuk saat belum ada input. Dikonstraint ke atas layout (`Top_toTopOf="parent"`) dan di tengah secara horizontal.
- Baris 25-29:
Berfungsi untuk menampilkan pertanyaan tentang kualitas layanan yaitu teks diambil dari `strings.xml` menggunakan `@string/how_was_the_service` dan ditempatkan di bawah `EditText`.
- Baris 31-49:
Adalah baris grup tombol pilihan untuk menentukan persentase tip yaitu `orientation="vertical"` menampilkan radio button secara vertical ada tiga opsi yang disediakan: 20%, 18%, dan 15%. `option_twenty_percent` disetel sebagai default dengan `android:checked="true"`.
- Baris 51-58
Adalah switch yang berguna untuk mengatur apakah tip perlu dibulatkan yaitu `android:checked="true"` yang menandakan pembulatan aktif secara default. Dikonstraint agar sejajar horizontal dengan `tip_options` dan berada di bawahnya.
- Baris 60-66:
Adalah tombol yang digunakan untuk menghitung jumlah tip yaitu menggunakan `text="@string/calculate"` untuk menampilkan label tombol dan ditempatkan di bawah `Switch`.
- Baris 68-74:
`<TextView android:id="@+id/tip_result" ... />` berguna menampilkan hasil perhitungan tip yaitu teks default diambil dari `@string/tip_amount`. Dikonstraint ke bawah tombol hitung, dan berada di sisi kanan layout.
- Baris 75:
`</androidx.constraintlayout.widget.ConstraintLayout>`,
`</ScrollView>` adalah penutup untuk `ConstraintLayout` dan `ScrollView`.

4. build.gradle (Module :app)

- Baris 1-3 (Plugins):
Adalah blok yang mengatur plugin yang digunakan dalam proyek yaitu `android.application`: Plugin utama untuk proyek Android. `kotlin.android`: berfungsi menyediakan dukungan untuk Kotlin di Android. `alias(libs.plugins...)` memanfaatkan versi plugin yang didefinisikan di `libs.versions.toml` untuk manajemen versi terpusat.
- Baris 5–34 (Blok android):

Bagian ini adalah bagian utama konfigurasi Android.

- Baris 6–8 (buildFeatures):
Mengaktifkan fitur ViewBinding, memungkinkan kamu mengakses view pada layout XML secara aman dan langsung melalui objek binding tanpa `findViewById()`.
- Baris 9-10 (Namespace dan compile SDK):
Namespace untuk menentukan nama package utama dari aplikasi sedangkan `compileSdk = 35` Proyek yang akan dikompilasi menggunakan Android API level 35.
- Baris 12-20 (defaultConfig):
Pada bagian ini berfungsi mengkonfigurasi dasar aplikasi diantaranya `applicationId` yaitu nama unik aplikasi (digunakan saat install di perangkat). `minSdk = 30` adalah minimum versi Android yang didukung. `targetSdk = 35` adalah target versi Android saat aplikasi dijalankan. `versionCode` dan `versionName` adalah informasi versi aplikasi. `testInstrumentationRunner` adalah runner default untuk instrumented test.
- Baris 22–28 (buildTypes):
Bagian ini mengatur tipe build menggunakan `release` yaitu konfigurasi untuk versi rilis aplikasi. `isMinifyEnabled = false` yang artinya tidak mengaktifkan shrinking/proguard. `proguardFiles` untuk menentukan aturan ProGuard yang digunakan jika shrinking diaktifkan.
- Baris 29–32 (compileOptions):
Bagian ini berfungsi menentukan versi Java yang digunakan untuk kompilasi yaitu Java 11 digunakan untuk source dan target compatibility.
- Baris 33–34 (kotlinOptions):
Bagian ini berfungsi untuk mengatur Kotlin agar menggunakan target JVM versi 11, agar kompatibel dengan fitur Java 11.
- Baris 36–45 (Dependencies):
Adalah daftar library yang digunakan dalam proyek yaitu `implementation(...)` library untuk runtime aplikasi, seperti `androidx.core.ktx`, `androidx.appcompat`, `material`, `constraintlayout`. `androidx.activity` untuk komponen Activity modern. `testImplementation(...)` untuk unit test (junit). `androidTestImplementation(...)` untuk instrumented test di Android (junit, espresso).

5. colors.xml

File ini mendefinisikan palet warna aplikasi yang digunakan untuk UI seperti latar belakang, teks, tombol, dan lainnya. Warna-warna ini digunakan di file layout atau tema untuk menjaga konsistensi desain. `purple_200`, `purple_500`,

`purple_700` yaitu warna ungu dalam tiga tingkat kecerahan/ketajaman, sering dipakai untuk elemen utama (`primary`). `teal_200`, `teal_700` yaitu warna aksen (`accent color`) dalam dua variasi, umumnya digunakan untuk tombol atau `highlight`. `black`, `white` yaitu warna dasar teks atau latar belakang, digunakan untuk kontras tinggi. Semua warna disimpan dalam format hexadecimal dengan alpha (FF = full opacity).

6. **strings.xml**

File ini berisi string literal atau teks yang akan ditampilkan ke pengguna. Dengan memisahkannya dari kode sumber, kamu bisa lebih mudah melakukan lokalisasi/multibahasa dan manajemen teks. `app_name` yaitu nama aplikasi yang muncul di launcher dan toolbar. `cost_of_service` yaitu label untuk input biaya layanan. `how_was_the_service` yaitu label untuk menanyakan tingkat pelayanan. `amazing_20`, `good_18`, `okay_15` adalah opsi dalam `RadioGroup` yang menunjukkan kualitas pelayanan dan besaran persentase tip. `round_up_tip` adalah label untuk switch, apakah tip dibulatkan atau tidak. `Calculate` adalah teks pada tombol kalkulasi. `tip_amount` adalah label hasil kalkulasi tip.

7. **themes.xml**

`Theme.TipTime` merupakan tema utama aplikasi dengan gaya dasar dari `MaterialComponents.DayNight.DarkActionBar`. Warna utama (`colorPrimary`) untuk mengatur elemen seperti Toolbar dengan warna `purple_500`. Warna variasi utama (`colorPrimaryVariant`) biasanya digunakan untuk status bar atau bagian yang lebih gelap, di-set ke `purple_700`. `colorOnPrimary` yaitu warna teks atau ikon yang tampil di atas warna utama, menggunakan warna putih agar kontras. Warna sekunder (`colorSecondary`) digunakan untuk elemen tambahan seperti tombol aksi atau `highlight`, diatur ke `teal_200`. `colorSecondaryVariant` yaitu versi gelap dari warna sekunder (`teal_700`), memberikan fleksibilitas visual. `colorOnSecondary` adalah warna teks/ikon di atas warna sekunder (hitam agar kontras). `statusBarColor` yaitu status bar akan mengambil warna dari `colorPrimaryVariant`.

8. **themes.xml (night)**

Mode ini diaktifkan ketika perangkat berada di mode malam (`dark mode`). Warna utama (`colorPrimary`) dipilih lebih terang (`purple_200`) agar tetap terlihat di latar belakang gelap. `colorOnPrimary` yaitu menggunakan hitam, karena `colorPrimary` terang, jadi dibutuhkan warna teks gelap agar tetap terbaca. `colorSecondary` dan variannya tetap `teal_200`, tidak berubah dari mode terang, karena warna ini sudah cukup netral untuk dipakai di kedua mode.

`colorOnSecondary` yaitu masih hitam, karena sekunder tetap terang. Konfigurasi `statusBarColor` tetap mengambil `colorPrimaryVariant`.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/rrdtlsh/Praktikum_Mobile