

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2**



Android Layout

Oleh:

Aliya Raffa Naura Ayu

NIM. 2310817120014

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
MODUL 2

Laporan Praktikum Pemrograman Mobile Modul 1: Android Layout ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Aliya Raffa Naura Ayu
NIM : 2310817120014

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.....	19

DAFTAR GAMBAR

Gambar 1. Screenshot Output Tampilan Awal Soal 1.....	14
Gambar 2. Screenshot Output Tampilan Setelah Dijalankan Soal 1	15
Gambar 3. Screenshot Output Tampilan Side Quest 1	15
Gambar 4. Screenshot Output Tampilan Side Quest 2.....	16

DAFTAR TABEL

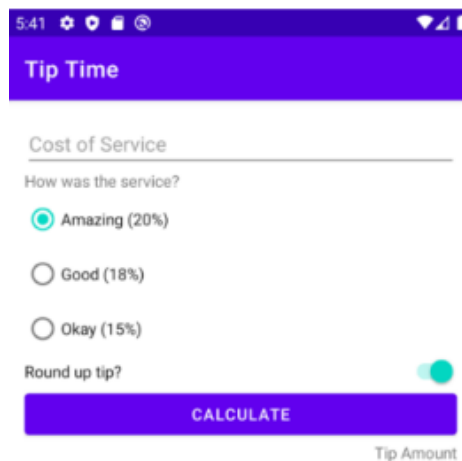
Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt	7
Tabel 2. Source Code Jawaban Soal 1 activity_main.xml	10
Tabel 3. Source Code Jawaban Soal 1 rounded_button.xml	13
Tabel 4. Source Code Jawaban Soal 1 strings.xml	14

SOAL 1

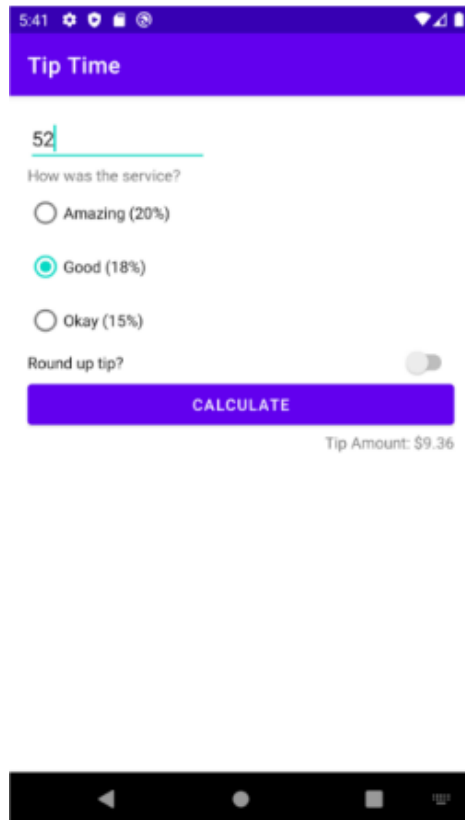
Soal Praktikum:

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



Gambar 2 Tampilan Aplikasi Setelah Dijalankan

A. Source Code

1. MainActivity.kt

Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt

1	<code>package com.example.calculatortip</code>
2	
3	<code>import android.annotation.SuppressLint</code>
4	<code>import android.os.Bundle</code>
5	<code>import android.widget.*</code>
6	<code>import androidx.appcompat.app.AppCompatActivity</code>
7	<code>import kotlin.math.ceil</code>
8	
9	<code>class MainActivity : AppCompatActivity() {</code>
10	
11	<code> private lateinit var costOfServiceEditText: EditText</code>
12	<code> private lateinit var tipOptions: RadioGroup</code>

```

13     @SuppressWarnings("UseSwitchCompatOrMaterialCode")
14     private lateinit var roundUpSwitch: Switch
15     private lateinit var calculateButton: Button
16     private lateinit var tipResultTextView: TextView
17
18     companion object {
19         private const val KEY_COST = "key_cost"
20         private const val KEY_TIP_RESULT = "key_tip_result"
21         private const val KEY_RADIO_ID = "key_radio_id"
22         private const val KEY_SWITCH_STATE = "key_switch_state"
23     }
24
25     @SuppressWarnings("UseSwitchCompatOrMaterialCode", "SetTextI18n")
26     override fun onCreate(savedInstanceState: Bundle?) {
27         super.onCreate(savedInstanceState)
28         setContentView(R.layout.activity_main)
29
30         costOfServiceEditText =
31         findViewById(R.id.cost_of_service_edit_text)
32         tipOptions = findViewById(R.id.tip_options)
33         roundUpSwitch = findViewById(R.id.round_up_switch)
34         calculateButton = findViewById(R.id.calculate_button)
35         tipResultTextView = findViewById(R.id.tip_result)
36
37         if (savedInstanceState != null) {
38
39         costOfServiceEditText.setText(savedInstanceState.getString(KEY_COST,
40         ""))
41
42         tipResultTextView.text =
43         savedInstanceState.getString(KEY_TIP_RESULT, "")
44         tipOptions.check(savedInstanceState.getInt(KEY_RADIO_ID,
45         R.id.amazing_option))
46         roundUpSwitch.isChecked =
47         savedInstanceState.getBoolean(KEY_SWITCH_STATE, false)

```


47	}
48	
49	calculateButton.setOnClickListener {
50	val costInput = costOfServiceEditText.text.toString()
51	val cost = costInput.toDoubleOrNull()
52	
53	if (cost == null cost == 0.0) {
54	costOfServiceEditText.error = "Masukkan angka yang
55	valid dan lebih besar dari 0"
56	Toast.makeText(this, "Input tidak valid! Masukkan
57	angka yang benar.", Toast.LENGTH_SHORT).show()
58	tipResultTextView.text =
59	getString(R.string.tip_amount, 0.0)
60	return@setOnClickListener
61	}
62	
63	val tipPercentage = when
64	(tipOptions.checkedRadioButtonId) {
65	R.id.amazing_option -> 0.20
66	R.id.good_option -> 0.18
67	R.id.okay_option -> 0.15
68	else -> 0.0
69	}
70	
71	var tip = cost * tipPercentage
72	
73	if (roundUpSwitch.isChecked) {
74	tip = ceil(tip)
75	}
76	
77	tipResultTextView.text = "Tip Amount:
78	\$\${"%.2f".format(tip)}"
79	}
80	}

81	
82	override fun onSaveInstanceState(outState: Bundle) {
83	super.onSaveInstanceState(outState)
84	outState.putString(KEY_COST,
85	costOfServiceEditText.text.toString())
86	outState.putString(KEY_TIP_RESULT,
87	tipResultTextView.text.toString())
88	outState.putInt(KEY_RADIO_ID,
89	tipOptions.checkedRadioButtonId)
90	outState.putBoolean(KEY_SWITCH_STATE,
91	roundUpSwitch.isChecked)
92	}
93	}

2. activity_main.xml

Tabel 2. Source Code Jawaban Soal 1 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:tools="http://schemas.android.com/tools"
5	android:layout_width="match_parent"
6	android:layout_height="match_parent"
7	android:fillViewport="true">
8	
9	<LinearLayout
10	android:layout_width="match_parent"
11	android:layout_height="wrap_content"
12	android:orientation="vertical"
13	android:background="#FFFFFF"
14	android:paddingBottom="16dp">
15	
16	<TextView
17	android:layout_width="match_parent"
18	android:layout_height="60dp"

19	android:background="#8000FF"
20	android:gravity="center_vertical start"
21	android:paddingStart="16dp"
22	android:paddingEnd="16dp"
23	android:text="@string/tip_time"
24	android:textColor="#FFFFFF"
25	android:textSize="20sp"
26	android:textStyle="bold" />
27	
28	<EditText
29	android:id="@+id/cost_of_service_edit_text"
30	android:layout_width="match_parent"
31	android:layout_height="wrap_content"
32	android:layout_marginTop="5dp"
33	android:layout_marginHorizontal="16dp"
34	android:hint="@string/cost_of_service"
35	android:inputType="numberDecimal" />
36	
37	<RadioGroup
38	android:id="@+id/tip_options"
39	android:layout_width="match_parent"
40	android:layout_height="wrap_content"
41	android:paddingStart="16dp"
42	android:paddingEnd="16dp">
43	
44	<RadioButton
45	android:id="@+id/amazing_option"
46	android:layout_width="wrap_content"
47	android:layout_height="wrap_content"
48	android:text="@string/amazing_20" />
49	
50	<RadioButton
51	android:id="@+id/good_option"
52	android:layout_width="wrap_content"

```

53         android:layout_height="wrap_content"
54         android:text="@string/good_18" />
55
56     <RadioButton
57         android:id="@+id/okay_option"
58         android:layout_width="wrap_content"
59         android:layout_height="wrap_content"
60         android:text="@string/okay_15" />
61 </RadioGroup>
62
63 <LinearLayout
64     android:layout_width="match_parent"
65     android:layout_height="wrap_content"
66     android:orientation="horizontal"
67     android:gravity="center_vertical"
68     android:paddingHorizontal="16dp">
69
70     <TextView
71         android:layout_width="0dp"
72         android:layout_weight="1"
73         android:layout_height="wrap_content"
74         android:text="@string/round_up_tip" />
75
76     <Switch
77         android:id="@+id/round_up_switch"
78         android:layout_width="wrap_content"
79         android:layout_height="wrap_content"
80         tools:ignore="UseSwitchCompatOrMaterialXml" />
81 </LinearLayout>
82
83 <Button
84     android:id="@+id/calculate_button"
85     android:layout_width="match_parent"
86     android:layout_height="wrap_content"

```

87	android:layout_marginHorizontal="16dp"
88	android:layout_marginTop="16dp"
89	android:background="@drawable/rounded_button"
90	android:text="@string/calculate"
91	android:textAllCaps="true"
92	android:textColor="#FFFFFF" />
93	
94	<TextView
95	android:id="@+id/tip_result"
96	android:layout_width="wrap_content"
97	android:layout_height="wrap_content"
98	android:layout_marginTop="8dp"
99	android:layout_marginEnd="16dp"
100	android:layout_gravity="end"
101	android:text="@string/amount_tip"
102	android:textSize="13sp"
103	android:textColor="#808080" />
104	</LinearLayout>
105	</ScrollView>

3. rounded_button.xml

Tabel 3. Source Code Jawaban Soal 1 rounded_button.xml

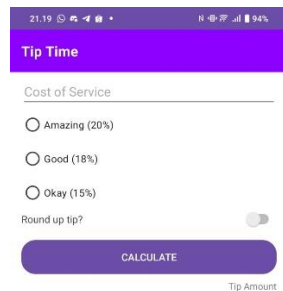
1	<?xml version="1.0" encoding="utf-8"?>
2	
3	<shape
4	xmlns:android="http://schemas.android.com/apk/res/android"
5	android:shape="rectangle">
6	<solid android:color="#8000FF" />
7	<corners android:radius="20dp" />
8	</shape>

4. strings.xml

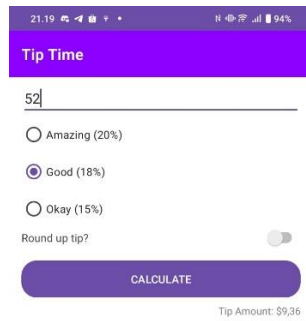
Tabel 4. Source Code Jawaban Soal 1 strings.xml

1	<resources>
2	<string name="app_name">Tip Calculator</string>
3	<string name="tip_time">Tip Time</string>
4	<string name="cost_of_service">Cost of Service</string>
5	<string name="how_was_the_service">How was the
6	service?</string>
7	<string name="amazing_20">Amazing (20%)</string>
8	<string name="good_18">Good (18%)</string>
9	<string name="okay_15">Okay (15%)</string>
10	<string name="round_up_tip">Round up tip?</string>
11	<string name="calculate">CALCULATE</string>
12	<string name="amount_tip">Tip Amount</string>
13	<string name="tip_amount">Tip: %.2f</string>
14	</resources>

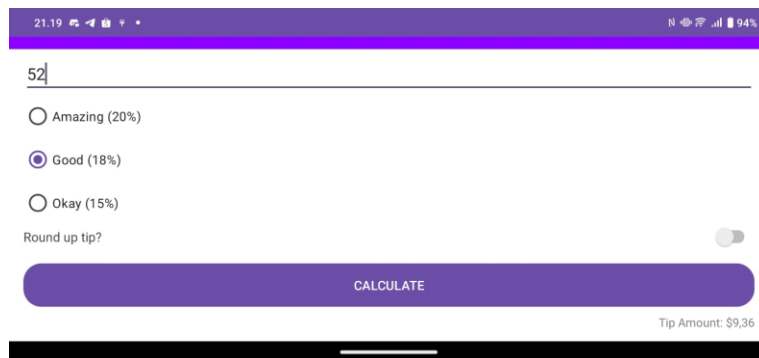
B. Output Program



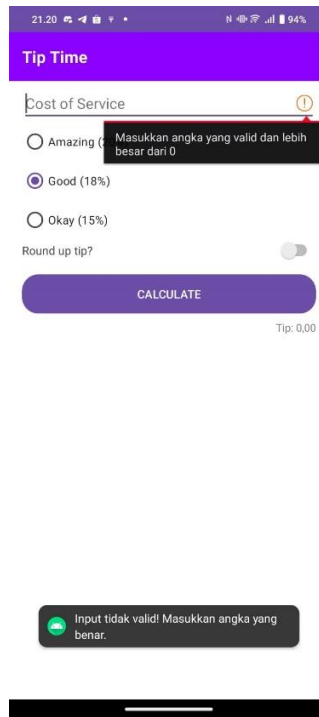
Gambar 1. Screenshot Output Tampilan Awal Soal 1



Gambar 2. Screenshot Output Tampilan Setelah Dijalankan Soal 1



Gambar 3. Screenshot Output Tampilan Side Quest 1



Gambar 4. Screenshot Output Tampilan Side Quest 2

C. Pembahasan

1. MainActivity.kt

Pada baris [1], `package com.example.calculatortip` digunakan untuk menentukan namespace atau nama paket dari aplikasi. Pada baris [3-7], dilakukan import beberapa class dari Android SDK dan Kotlin Standard Library yang diperlukan untuk membangun UI dan logika aplikasi. Pada baris [9], `class MainActivity : AppCompatActivity()` mendefinisikan `MainActivity` sebagai turunan dari `AppCompatActivity`, memungkinkan penggunaan fitur Activity Android. Pada baris [11-16], `private lateinit var` deklarasi variabel view yang akan diinisialisasi nanti menggunakan `lateinit`. Pada baris [18-23], `companion object` companion object menyimpan key konstanta yang digunakan untuk menyimpan dan mengembalikan state saat rotasi layar.

Pada baris [26], `onCreate` adalah fungsi lifecycle utama yang dipanggil saat activity dibuat. Pada baris [27], `super.onCreate(savedInstanceState)` memanggil implementasi `onCreate` dari superclass. Pada baris [28], `setContentView(R.layout.activity_main)` menampilkan layout XML `activity_main`. Pada baris [31-35], menginisialisasi view dengan `findViewById`

berdasarkan ID dari layout XML. Pada baris [37-47], jika `savedInstanceState` tidak null (rotasi layar), maka nilai input pengguna sebelumnya dikembalikan. Pada baris [49], `calculateButton.setOnClickListener` digunakan untuk menetapkan aksi ketika tombol `calculateButton` diklik. Pada baris [50-51], `val costInput = costOfServiceEditText.text.toString()` `val cost = costInput.toDoubleOrNull()` digunakan untuk mengambil input biaya layanan dan mencoba mengonversinya menjadi `Double`. Pada baris [53-61], `if (cost == null || cost == 0.0) {...}` digunakan untuk memeriksa validitas input. Jika tidak valid, tampilkan pesan kesalahan dan hentikan proses perhitungan.

Pada baris [63-69], `val tipPercentage = when (tipOptions.checkedRadioButtonId) {...}` Digunakan untuk menentukan persentase tip berdasarkan radio button yang dipilih. Pada baris [71], `var tip = cost * tipPercentage` digunakan untuk menghitung nilai tip awal berdasarkan input biaya dan persentase. Pada baris [73-75], jika switch `roundUpSwitch` aktif, nilai tip dibulatkan ke atas menggunakan `ceil`. Pada baris [77-78], hasil perhitungan ditampilkan pada `tipResultTextView`. Pada baris [82], override fungsi `onSaveInstanceState` untuk menyimpan state aplikasi saat rotasi atau keadaan lainnya. Pada baris [84-91], menyimpan nilai input, hasil tip, pilihan radio, dan status switch ke dalam bundle `outState`.

2. activity_main.xml

Pada baris [1], `<?xml version="1.0" encoding="utf-8"?>` mendefinisikan versi XML dan encoding yang digunakan. Pada baris [2-7], mendefinisikan root layout sebagai `ScrollView` agar isi layout dapat digulir bila melebihi layar. Pada baris [9-14], `LinearLayout` digunakan sebagai kontainer utama dengan orientasi vertikal dan padding bawah. Pada baris [16-26], `TextView` digunakan sebagai judul dengan latar belakang ungu, teks putih, ukuran 20sp, dan bold. Pada baris [28-35], `EditText` memungkinkan pengguna mengisi biaya layanan. Diset `inputType="numberDecimal"` agar hanya angka desimal yang bisa dimasukkan. Pada baris [32-41], `RadioGroup` berisi tiga pilihan persentase tip, ditampilkan secara vertikal.

Pada baris [44-48], `RadioButton` pertama mewakili tip 20%. Pada baris [50-54], `RadioButton` kedua untuk tip 18%. Pada baris [56-60], `RadioButton` ketiga untuk tip

15%. Pada baris [63-68], `LinearLayout` horizontal untuk menampilkan opsi bulatkan tip. Pada baris [70-74], `TextView` label "Round up tip". Pada baris [76-80], `Switch` memungkinkan pengguna memilih apakah ingin membulatkan nilai tip atau tidak. Pada baris [83-92], `Button` untuk menghitung tip. Menggunakan `drawable` kustom `@drawable/rounded_button` untuk tampilan tombol dan teks kapital semua `textAllCaps="true"`. Pada baris [94-103], `TextView` untuk menampilkan hasil tip yang dihitung. Diletakkan di ujung kanan `layout_gravity="end"` dan teks berwarna abu-abu.

3. `rounded_button.xml`

Pada baris [1], `<?xml version="1.0" encoding="utf-8"?>` menyatakan versi XML dan encoding yang digunakan. Pada baris [3], `<shape>` adalah elemen root yang digunakan untuk membuat `drawable` berbentuk geometris. Pada baris [5], atribut `android:shape="rectangle"` menentukan bahwa bentuk `drawable` adalah persegi panjang. Pada baris [6], `<solid android:color="#8000FF" />` memberikan warna ungu solid sebagai latar belakang tombol. Pada baris [7], `<corners android:radius="20dp" />` memberikan efek sudut melengkung dengan radius 20dp. Pada baris [8], `</shape>` menutup elemen `shape`.

4. `strings.xml`

Pada baris [1], `<resources>` adalah elemen root untuk mendefinisikan semua string yang bisa digunakan di seluruh aplikasi. Pada baris [2], `<string name="app_name">Tip Calculator</string>` mendefinisikan nama aplikasi. Pada baris [3], `<string name="tip_time">Tip Time</string>` adalah teks judul utama aplikasi. Pada baris [4], `<string name="cost_of_service">Cost of Service</string>` digunakan sebagai hint untuk input biaya layanan. Pada baris [5-6], `<string name="how_was_the_service">How was the service?</string>` merupakan pertanyaan untuk menentukan kualitas layanan. Pada baris [7], `<string name="amazing_20">Amazing (20%)</string>` label untuk pilihan radio 20%. Pada baris [8], `<string name="good_18">Good (18%)</string>`

label untuk pilihan radio 18%. Pada baris [9], `<string name="okay_15">Okay (15%)</string>` label untuk pilihan radio 15%. Pada baris [10], `<string name="round_up_tip">Round up tip?</string>` label untuk switch pembulatan tip. Pada baris [11], `<string name="calculate">CALCULATE</string>` label untuk tombol kalkulasi tip. Pada baris [12], `<string name="amount_tip">Tip Amount</string>` label untuk hasil yang ditampilkan. Pada baris [13], `<string name="tip_amount">Tip: %.2f</string>` format string untuk hasil tip dengan dua angka di belakang koma. Pada baris [14], `</resources>` menutup elemen resources.

D. Tautan Git

<https://github.com/aliyarfnaura/Pemrograman-Mobile>