

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 2**



**ANDROID LAYOUT**

**Oleh:**

**Allano Lintang Ertantora**

**NIM. 2310817210004**

**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 : Zulfa Auliya Akbar  
NIM : 2310817210004

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Allano Lintang Ertantora  
NIM. 2310817210004

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 .....	11
C. Pembahasan .....	13
Tautan Git.....	15

## **DAFTAR GAMBAR**

Gambar 1. Screenshot Hasil Jawaban Soal 1 .....	13
---	----

## DAFTAR TABEL

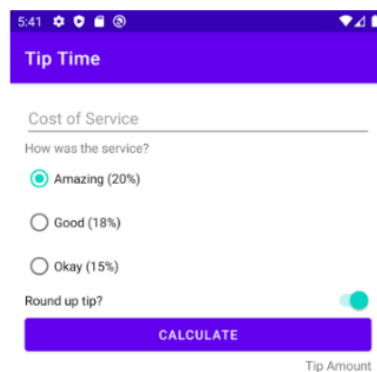
Tabel 1. Source Code Jawaban Soal 1.....	7
Tabel 2. Source Code Jawaban Soal 1.....	8

## SOAL 1

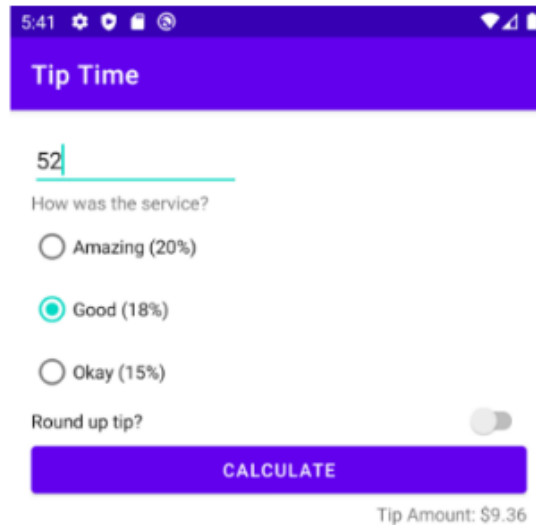
### Soal Praktikum:

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*

1	package com.example.tipcalc
2	
3	import android.os.Bundle
4	import androidx.activity.ComponentActivity
5	import androidx.activity.compose.setContent
6	import androidx.activity.enableEdgeToEdge
7	import androidx.compose.foundation.layout.fillMaxSize
8	import androidx.compose.foundation.layout.padding
9	import androidx.compose.material3.Scaffold
10	import androidx.compose.material3.Text
11	import androidx.compose.runtime.Composable
12	import androidx.compose.ui.Modifier
13	import androidx.compose.ui.tooling.preview.Preview
14	import android.widget.Toast

```

15 import
16     com.example.tipcalc.databinding.ActivityMainBinding
17 import com.example.tipcalc.ui.theme.TipcalcTheme
18
19 class MainActivity : ComponentActivity() {
20     lateinit var binding: ActivityMainBinding
21
22     override fun onCreate(savedInstanceState: Bundle?) {
23         super.onCreate(savedInstanceState)
24         binding =
25             ActivityMainBinding.inflate(layoutInflater)
26         setContentView(binding.root)
27         binding.calculate.setOnClickListener {
28             hitung()
29         }
30         private fun hitung(){
31             val cost =
32             binding.costOfService.text.toString().toDouble()
33             val tip = binding.tipOptions.checkedRadioButtonId
34             val tipPercent = when (tip) {
35                 R.id.option_twenty -> 0.20
36                 R.id.option_eighteen -> 0.18
37                 else -> 0.15
38             }
39             var total = tipPercent * cost
40             val roundUp = binding.roundUp.isChecked
41             if(roundUp){
42                 total = kotlin.math.ceil(total)
43             }
44             if(cost <= 0){
45                 Toast.makeText(this, "Masukkan nilai yang
46                 benar", Toast.LENGTH_SHORT).show()
47             }else{
48                 binding.total.text = total.toString()
49             }
50         }
51     }

```

## 2. activity\_main.xml

Tabel 2. Source Code Jawaban Soal 1

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3
4     xmlns:android="http://schemas.android.com/apk/res/android"

```



```

5      xmlns:app="http://schemas.android.com/apk/res-auto"
6      xmlns:tools="http://schemas.android.com/tools"
7      android:id="@+id/main"
8      android:layout_width="match_parent"
9      android:layout_height="match_parent"
10     android:padding="16dp"
11     tools:context=".MainActivity">
12
13     <EditText
14         android:id="@+id/cost_of_service"
15         android:hint="Cost of Service"
16         android:inputType="number"
17         app:layout_constraintTop_toTopOf="parent"
18         app:layout_constraintLeft_toLeftOf="parent"
19         android:layout_width="match_parent"
20         android:layout_height="48dp"
21     />
22     <TextView
23         android:id="@+id/service_question"
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         app:layout_constraintStart_toStartOf="parent"
27 app:layout_constraintTop_toBottomOf="@id/cost_of_service"
28         android:text="How was the service?"
29     />
30     <RadioGroup
31         android:id="@+id/tip_options"
32         app:layout_constraintStart_toStartOf="parent"
33 app:layout_constraintTop_toBottomOf="@id/service_question"
34         android:checkedButton="@id/option_twenty"
35         android:layout_width="wrap_content"
36         android:layout_height="wrap_content">
37
38         <RadioButton
39             android:id="@+id/option_twenty"
40             android:layout_width="wrap_content"
41             android:layout_height="48dp"
42             android:text="Amazing (20%)"
43             app:layout_constraintTop_toTopOf="parent"
44             app:layout_constraintLeft_toLeftOf="parent"
45         />
46         <RadioButton
47             android:id="@+id/option_eighteen"
48             android:layout_width="wrap_content"
49             android:layout_height="48dp"
50             android:text="Good (18%)"
51             app:layout_constraintTop_toTopOf="parent"

```

```

52         app:layout_constraintLeft_toLeftOf="parent"
53     />
54     <RadioButton
55         android:id="@+id/option_fifteen"
56         android:layout_width="wrap_content"
57         android:layout_height="48dp"
58         android:text="Okay (15%)"
59         app:layout_constraintTop_toTopOf="parent"
60         app:layout_constraintLeft_toLeftOf="parent"
61     />
62 </RadioGroup>
63 <Switch
64     android:id="@+id/round_up"
65     android:text="Round up tip?"
66     app:layout_constraintStart_toStartOf="parent"
67     app:layout_constraintTop_toBottomOf="@id/tip_options"
68     android:layout_width="match_parent"
69     android:layout_height="48dp"
70 />
71 <Button
72     android:id="@+id/calculate"
73     android:text="Calculate"
74     app:layout_constraintStart_toStartOf="parent"
75     app:layout_constraintTop_toBottomOf="@+id/round_up"
76     android:layout_width="match_parent"
77     android:layout_height="wrap_content"/>
78 <TextView
79     android:id="@+id/text"
80     android:text="Tip amount: "
81     app:layout_constraintRight_toLeftOf="@id/total"
82     app:layout_constraintTop_toBottomOf="@+id/calculate"
83     android:layout_width="wrap_content"
84     android:layout_height="wrap_content"
85 />
86 <TextView
87     android:id="@+id/total"
88     android:text=""
89     app:layout_constraintEnd_toEndOf="parent"
90     app:layout_constraintTop_toBottomOf="@+id/calculate"
91     android:layout_width="wrap_content"
92     android:layout_height="wrap_content"
93 />
94 </androidx.constraintlayout.widget.ConstraintLayout>

```

## B. Output Program

The image shows a mobile application interface for calculating a tip. At the top, there is a status bar with signal strength, time (20:51), and battery level. Below the status bar, the title "Cost of Service" is displayed. The main section is titled "How was the service?" and contains three radio button options: "Amazing (20%)", "Good (18%)", and "Okay (5%)". The "Amazing (20%)" option is selected. Below these options, there is a toggle switch for "Round up tip?". A "CALCULATE" button is positioned below the toggle switch. To the right of the button, the text "Tip amount:" is visible. The interface is clean and modern, with a light gray background and dark text.

Cost of Service

How was the service?

☒ Amazing (20%)

☐ Good (18%)

☐ Okay (5%)

Round up tip? ☐

CALCULATE

Tip amount:

20:52



68505

How was the service?

☐ Amazing (20%)

☒ Good (18%)

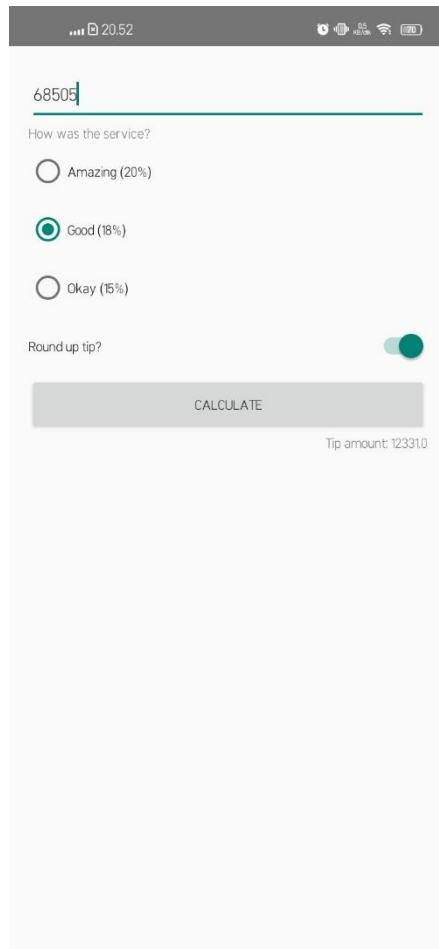
☐ Okay (15%)

Round up tip?



CALCULATE

Tip amount: 12330.9



*Gambar 1. Screenshot Hasil Jawaban Soal 1*

## **C. Pembahasan**

### **1. MainActivity.kt:**

Pada baris [1-14], file dimulai dengan deklarasi package serta berbagai import yang diperlukan untuk menjalankan komponen Android, termasuk elemen-elemen dari Jetpack Compose dan View Binding. Class `MainActivity` dideklarasikan sebagai turunan dari `ComponentActivity`, yang berfungsi sebagai entry point atau titik masuk utama dari aplikasi Android tersebut.

Pada baris [16], terdapat deklarasi variabel binding dengan tipe `ActivityMainBinding` dan modifier `lateinit`, yang artinya variabel ini akan

diinisialisasi nanti, khususnya saat layout sudah tersedia. Variabel ini digunakan untuk mengakses view dari file layout XML yang telah di-inflate.

Pada baris [18-23], method `onCreate()` dioverride untuk menjalankan beberapa proses penting saat activity dibuat. Pertama, aplikasi mengaktifkan tampilan edge-to-edge dengan memanggil `enableEdgeToEdge()`. Selanjutnya, layout XML di-inflate ke dalam objek binding menggunakan `ActivityMainBinding.inflate()`, kemudian root view dari binding diatur sebagai content view aplikasi dengan `setContentView()`. Terakhir, tombol "calculate" pada layout diberikan listener (`setOnClickListener`) yang akan memicu pemanggilan fungsi `hitung()` ketika ditekan.

Pada baris [24-38], fungsi `hitung()` berisi logika utama untuk melakukan perhitungan tip. Pertama, nilai `cost` diambil dari input `EditText` dan dikonversi ke dalam tipe `Double`. Kemudian, pilihan persentase tip diambil dari `RadioGroup` dan ditentukan menggunakan ekspresi `when`, berdasarkan `id` dari `RadioButton` yang dipilih. Setelah itu, nilai tip dihitung dengan mengalikan persentase yang sesuai dengan nilai `cost`. Jika `Switch` bernama `roundUp` dalam keadaan aktif, maka hasil tip akan dibulatkan ke atas menggunakan fungsi `ceil()`. Jika nilai `cost` kurang dari atau sama dengan nol, maka akan ditampilkan pesan kesalahan menggunakan `Toast`. Akhirnya, hasil perhitungan tip ditampilkan pada `TextView` dengan `id total`.

## **2. activity\_main.xml**

Pada baris [1-10], layout utama aplikasi menggunakan `ConstraintLayout` sebagai root view dengan ID `main`, dan diberi padding sebesar 16dp di keempat sisinya untuk memberikan jarak yang nyaman terhadap tepi layar. Layout ini dihubungkan dengan konteks `MainActivity`, yang berarti semua komponen yang berada di dalamnya akan mengikuti logika dan siklus hidup dari activity tersebut.

Pada baris [12-20], terdapat elemen `EditText` yang digunakan untuk input nilai "cost of service". Komponen ini memiliki ID `cost_of_service`, dilengkapi dengan hint text "Cost of Service" sebagai petunjuk bagi pengguna, dan menggunakan `inputType number` agar hanya menerima input berupa angka. Ukuran tinggi ditetapkan sebesar 48dp dan lebar `match_parent` agar memanjang sesuai lebar layar.

Selanjutnya, pada baris [21-27], terdapat `TextView` dengan teks pertanyaan "How was the service?" yang diletakkan dengan constraint tepat di bawah `EditText` sebelumnya, berfungsi sebagai pengantar untuk opsi pemilihan tip.

Pada baris [28-56], terdapat `RadioGroup` yang terdiri dari tiga `RadioButton` untuk memilih persentase tip. Salah satu tombol radio, yaitu `option_twenty` (20%), disetel sebagai default terpilih. Setiap `RadioButton` menampilkan teks dan nilai persentase yang sesuai, dan seluruh grup ini di-constraint agar berada di bawah `TextView` pertanyaan.

Kemudian, pada baris [57-62], terdapat komponen `Switch` dengan teks "Round up tip?" yang memberikan opsi kepada pengguna untuk membulatkan hasil tip ke atas. `Switch` ini juga diposisikan dengan constraint di bawah `RadioGroup`.

Pada baris [63-68], terdapat `Button` dengan ID `calculate`, yang digunakan untuk memicu proses perhitungan tip. Tombol ini memiliki lebar `match_parent` dan dikonfigurasi agar berada tepat di bawah `Switch`.

Terakhir, pada baris [69-82], terdapat dua buah `TextView` yang berfungsi untuk menampilkan hasil perhitungan. `TextView` pertama menampilkan label "Tip amount: ", dan berada di sebelah kiri, sedangkan `TextView` kedua memiliki ID `total` dan digunakan untuk menampilkan nilai hasil perhitungan tip, diletakkan di sebelah kanan. Kedua elemen ini diatur agar berada tepat di bawah tombol `calculate`, menjaga konsistensi dan kejelasan tata letak hasil output.

### **Tautan Git**

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/AllanoLintang/Pemrograman-mobile>