LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2

ANDROID LAYOUT

Oleh:

Noviana Nur Aisyah NIM. 2310817120005

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2025

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
MODUL 2

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

Nama Praktikan : Noviana Nur Aisyah
NIM : 2310817120005

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

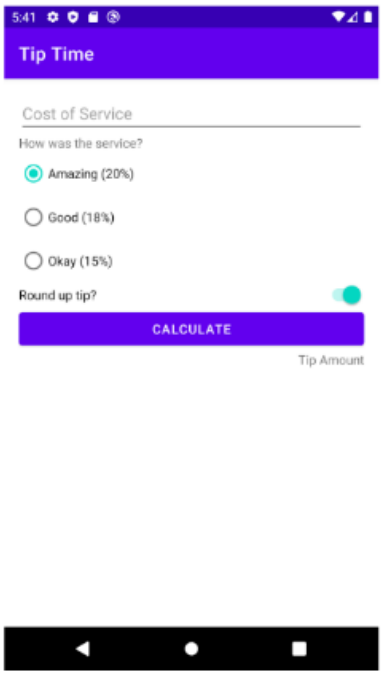
DAFTAR GAMBAR

DAFTAR TABEL

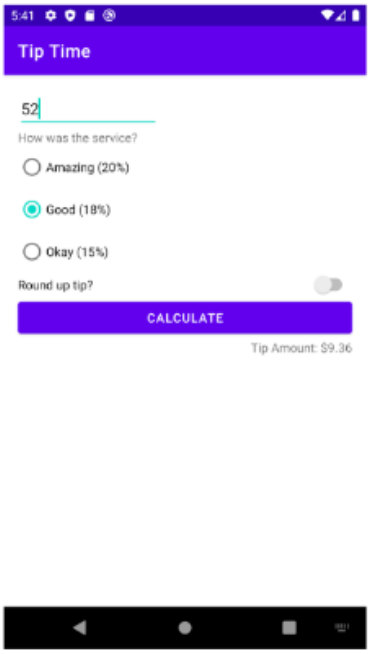
SOAL 1

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.calculatortip
2
3  import android.os.Bundle
4  import androidx.activity.enableEdgeToEdge
5  import androidx.appcompat.app.AppCompatActivity
6  import androidx.core.view.ViewCompat
7  import androidx.core.view.WindowInsetsCompat
8  import androidx.lifecycle.ViewModelProvider
9  import androidx.lifecycle.get
10 import
11 com.example.calculatortip.databinding.ActivityMainBinding
12 import android.view.View
13 import android.os.Handler
```

```
14 import android.os.Looper
15 import android.widget.Toast
16 import
17 androidx.core.splashscreen.SplashScreen.Companion.install
18 SplashScreen
19
20 class MainActivity : AppCompatActivity() {
21     private lateinit var binding: ActivityMainBinding
22
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26
27         Thread.sleep(2000)
28         installSplashScreen()
29
30         binding                                          =
31 ActivityMainBinding.inflate(layoutInflater)
32         setContentView(binding.root)
33
34         var                  viewModel                   =
35 ViewModelProvider(this).get(MainActivityViewModel::class.
36 java)
37
38         viewModel.tipResult.observe(this) { tip ->
39             binding.tipAmount.text = "Tip Amount: $tip"
40             binding.tipAmount.visibility = View.VISIBLE
41         }
42
43         binding.calculateButton.setOnClickListener {
44
```

```
            val inputEditText =
binding.inputText.text.toString()
            val radioGroup =
binding.radioGroup.checkedRadioButtonId
            val switchRound =
binding.switchRound.isChecked

            val inputValue =
inputEditText.toDoubleOrNull()
            if (inputValue == null) {
                Toast.makeText(this, "Mohon isi Cost Of
Value dengan angka", Toast.LENGTH_SHORT).show()
                binding.tipAmount.text = ""
                binding.tipAmount.visibility = View.GONE
                return@setOnClickListener
            } else if (inputValue == 0.0) {
                Toast.makeText(this, "Mohon isi dengan
angka yang lebih besar", Toast.LENGTH_SHORT).show()
                binding.tipAmount.text = ""
                binding.tipAmount.visibility = View.GONE
                return@setOnClickListener
            }

            viewModel.calculate(inputEditText, radioGroup,
switchRound)
        }

    }
}
```

2. MainActivityViewModel.kt

Tabel 2. Source Code Jawaban Soal 1

```
1  package com.example.calculatortip
2
3  import androidx.lifecycle.ViewModel
4  import androidx.lifecycle.LiveData
5  import androidx.lifecycle.MutableLiveData
6  import
7  com.google.android.material.textfield.TextInputEditText
8  import java.text.NumberFormat
9  import kotlin.math.ceil
10
11 class MainActivityViewModel: ViewModel() {
12     private val _tipResult = MutableLiveData<String>()
13     val tipResult: LiveData<String> = _tipResult
14
15     fun calculate(inputEditText: String?, radioGroup: Int,
16 switchRound: Boolean) {
17         val cost = inputEditText?.toDoubleOrNull()
18         if (cost == null) {
19             _tipResult.value = ""
20             return
21         }
22
23         val tipPercentage = when (radioGroup) {
24             R.id.amazing_button -> 0.20
25             R.id.good_button -> 0.18
26             else -> 0.15
27         }
28
29         var tip = tipPercentage * cost
30
31         if (switchRound) {
```

```
32              tip                 =                 ceil(tip)
33          }
34
35          val                 formattedTip                =
36  NumberFormat.getCurrencyInstance().format(tip)
37
38          _tipResult.value          =          formattedTip
39      }
40
41
42  }
```

3. Activity_main.xml

Tabel 3. Source Code Jawaban Soal 1

```
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:id="@+id/toolBarText"
13          android:layout_width="match_parent"
14          android:layout_height="wrap_content"
15          android:background="@android:color/white"
16          android:backgroundTint="#547792"
17          android:fontFamily="sans-serif"
```

```
        android:paddingTop="20dp"
        android:paddingBottom="15dp"
        android:paddingLeft="15dp"
        android:paddingRight="15dp"
        android:text="Tip Time"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        />

    <EditText
        android:id="@+id/input_text"
        android:layout_width="380dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:inputType="text"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/toolBarText"
        android:hint="Cost of Service"
        android:textColorHint="#aaa7ad"
        />

    <TextView
        android:id="@+id/pertanyaan"
        android:layout_width="wrap_content"
```

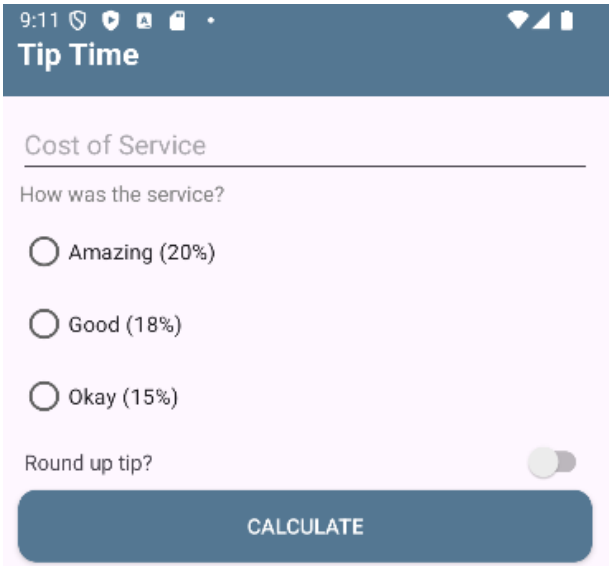
```
49          android:layout_height="wrap_content"
50          android:text="How was the service? "
51          android:textColor="#808080"
52          app:layout_constraintHorizontal_bias="0.06"
53          app:layout_constraintEnd_toEndOf="parent"
54          app:layout_constraintStart_toStartOf="parent"
55
56  app:layout_constraintTop_toBottomOf="@id/input_text" />
57
58      <RadioGroup
59          android:id="@+id/radio_group"
60          android:layout_width="wrap_content"
61          android:layout_height="wrap_content"
62          android:orientation="vertical"
63
64  app:layout_constraintTop_toBottomOf="@id/pertanyaan"
65          app:layout_constraintStart_toStartOf="parent"
66          app:layout_constraintEnd_toEndOf="parent"
67          app:layout_constraintHorizontal_bias="0.06"
68          android:layout_marginTop="5dp"
69          >
70
71          <RadioButton
72              android:id="@+id/amazing_button"
73              android:layout_width="wrap_content"
74              android:layout_height="wrap_content"
75              android:text="Amazing (20%)"
76
77  app:buttonTint="@drawable/radio_button_selector"/>
78
79          <RadioButton
```

```
            android:id="@+id/good_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Good (18%)"

app:buttonTint="@drawable/radio_button_selector"/>

        <RadioButton
            android:id="@+id/okay_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Okay (15%)"

app:buttonTint="@drawable/radio_button_selector"/>
    </RadioGroup>

    <TextView
        android:id="@+id/round_up"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Round up tip? "
        android:layout_marginTop="10dp"
        app:layout_constraintHorizontal_bias="0.06"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/radio_group"
        />

    <Switch
        android:id="@+id/switch_round"
```

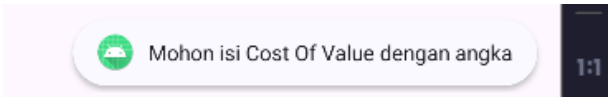
```
111         android:layout_width="wrap_content"
112         android:layout_height="wrap_content"
113         android:layout_marginEnd="15dp"
114
115 app:layout_constraintBottom_toBottomOf="@id/round_up"
116         app:layout_constraintEnd_toEndOf="parent"
117         app:layout_constraintTop_toTopOf="@id/round_up"
118         app:thumbTint="#eeeeee"
119         app:trackTint="#547792"
120         />
121
122     <android.widget.Button
123         android:id="@+id/calculate_button"
124         android:layout_width="380dp"
125         android:layout_height="wrap_content"
126         android:background="@drawable/rounded_border"
127         app:layout_constraintEnd_toEndOf="parent"
128         app:layout_constraintStart_toStartOf="parent"
129
130 app:layout_constraintTop_toBottomOf="@id/switch_round"
131         android:text="Calculate"
132         android:layout_margin="5dp"
133         android:textColor="@color/white"
134         />
135
136     <TextView
137         android:id="@+id/tip_amount"
138         android:layout_width="wrap_content"
139         android:layout_height="wrap_content"
140         android:text="apa saja brok "
141         android:layout_marginTop="5dp"
```

```
142         app:layout_constraintHorizontal_bias="0.94"
143         app:layout_constraintEnd_toEndOf="parent"
144         app:layout_constraintStart_toStartOf="parent"
145
146 app:layout_constraintTop_toBottomOf="@id/calculate_button"
147         android:visibility="gone"
148         />
149
150 </androidx.constraintlayout.widget.ConstraintLayout>
```

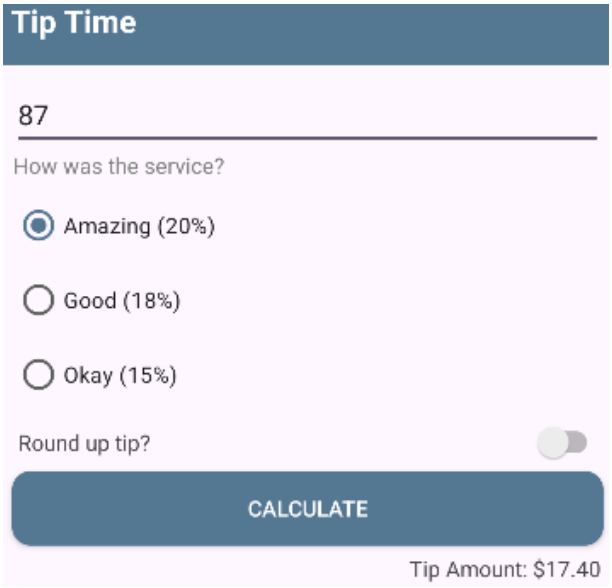
B. Output Program



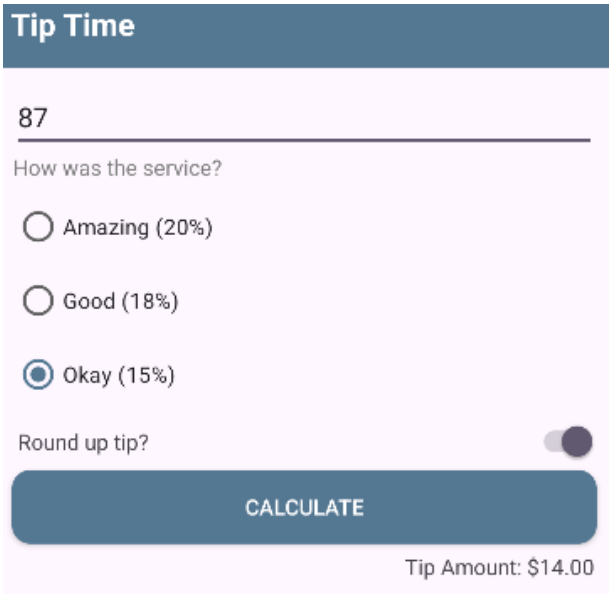
Gambar 1. Screenshot Hasil Jawaban Soal 1



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

File ini merupakan logika utama dari aplikasi untuk menampilkan UI dan mengatur interaksi pengguna. Terdapat beberapa bagian penting, yaitu:

a) Splash Screen

```
Thread.sleep(2000)
installSplashScreen()
```

Digunakan untuk menampilkan splash screen saat aplikasi pertama kali dibuka. `Thread.sleep(2000)` artinya program memberi delay selama 2 detik agar splash terlihat cukup lama, lalu `installSplashScreen()` menampilkan splash screen dari android.

b) View Binding

```
binding = ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)
```

Digunakan untuk menghubungkan layout activity_main.xml ke MainActivity agar elemen UI dapat diakses langsung melalui `binding`.

c) ViewModel & LiveData

```
var viewModel = ViewModelProvider(this).get(MainActivityViewModel::class.java)

viewModel.tipResult.observe(this) { tip ->
    binding.tipAmount.text = "Tip Amount: $tip"
    binding.tipAmount.visibility = View.VISIBLE
}
```

`ViewModelProvider` digunakan untuk menghubungkan MainActivity ke MainActivityViewModel. Kemudian, `tipResult` merupakan LiveData berisi hasil kalkulasi. Saat nilai `tipResult` berubah, `tipAmount` akan menampilkan hasilnya.

d) Event Button

```
binding.calculateButton.setOnClickListener {
    ...
    viewModel.calculate(inputEditText, radioGroup, switchRound)
}
```

Ketika button `calculateButton` diklik, maka akan:

- Mengambil input nilai biaya (`inputEditText`);
- Mengecek radio button yang dipilih;

- Mengecekk apakah pembulatan / round up aktif (switch);
- Jika input kosong atau 0, akan muncul toast sebagai peringatan;
- Jika valid, memanggil fungsi `calculate()` dari View Model untuk memprosesnya.

e) Validasi Input

```
val inputValue = inputEditText.toDoubleOrNull()
if (inputValue == null || inputValue == 0.0) {
    Toast.makeText(this, ...).show()
    binding.tipAmount.text = ""
    binding.tipAmount.visibility = View.GONE
    return@setOnClickListener
}
```

Jika input tidak valid, maka akan muncul peringatan berupa toast, kemudian teks hasil tip disembunyikan.

f) Tampilan Hasil Tip

```
binding.tipAmount.text = "Tip Amount: $tip"
binding.tipAmount.visibility = View.VISIBLE
```

Jika kalkulasi berhasil, maka hasil tip akan ditampilkan di `textView tipAmount`.

2. MainActivityViewModel.kt

File ini memuat ViewModel dari aplikasi kalkulator tip untuk memproses logika perhitungan tip berdasarkan input pengguna. View Model ini akan memberikan hasil ke UI melalui LiveData. Terdapat beberapa bagian, yaitu:

a) LiveData dan MutableLiveData

private val _tipResult = MutableLiveData<String>()

val tipResult: LiveData<String> = _tipResult

Fungsi dari `_tipResult` adalah untuk menyimpan nilai hasil kalkulasi tip dan bisa diubah dari dalam ViewModel. Kemudian, `tipResult` adalah versi read-only dari `_tipResult` yang digunakan oleh Activity untuk mengamati perubahan dan menampilkannya di UI.

b) Fungsi `calculate()`

```
fun calculate(inputEditText: String?, radioGroup: Int,
switchRound: Boolean)
```

Fungsi ini dapat menerima tiga parameter dari UI, yaitu `inputEdittext`, `radioGroup`, dan `switchRound`.

c) Validasi Input

```
val cost = inputEditText?.toDoubleOrNull()
if (cost == null) {
    _tipResult.value = ""
    return
}
```

Jika input kosong atau bukan merupakan angka, maka hasil tidak akan ditampilkan.

d) Menentukan Persentase Tip

```
val tipPercentage = when (radioGroup) {
    R.id.amazing_button -> 0.20
    R.id.good_button -> 0.18
    else -> 0.15
}
```

Bagian ini berfungsi untuk menentukan persentase tip berdasarkan pilihan radio button.

e) Perhitungan dan Pembulatan

```
var tip = tipPercentage * cost
if (switchRound) {
    tip = ceil(tip)
}
```

Bagian ini berfungsi untuk mengalikan nilai `cost` dengan `tipPercentage`. Jika opsi pembulatan (`switchRound`) aktif, maka hasil dibulatkan ke atas dengan `ceil()`.

f) Format Mata Uang dan Update UI

```
val formattedTip = NumberFormat.getCurrencyInstance().format(tip)
_tipResult.value = formattedTip
```

Hasil tip diformat ke format uang lokal (misalnya "`Rp12.000,00`" atau "`$5.00`"), lalu diberikan ke `_tipResult`, yang akan langsung dikirim ke UI melalui observer di Activity.

3. activity_main.xml

a) ConstraintLayout

Digunakan untuk menatur posisi angtar elemen secara efisien dan responsive.

b) textView – toolBarText

Digunakan sebagai header aplikasi yang menampilkan judul, di mana bagian ini diberi warna background, padding dan teks bold agar terlihat seperti toolbar.

c) EditText – input_text

Digunakan untuk memasukkan tip.

d) TextView (pertanyaan)

Berisi pertanyaan kepada pengguna tentang seberapa bagus layanan yang diterima.

e) RadioGroup + RadioButton

Merupakan pilihan tingkat layanan, yaiitu Amazing, Good dan Okay. Digunakan untuk menentukan persentasi tip.

f) TextView + Switch – Round Up

Digunakan untuk membulatkan hasil tip ke atas. Switch digunakan agar pengguna dapat mengaktifkan atau menonaktifkan fitur ini.

g) Button – Calculate

Digunakan untuk menghitung tip berdasarkan input dan pilihan pengguna.

h) TextView – tip_amount

Digunakan untuk menampilkan hasil tip yang dihitung. Pada awalnya tidak terlihat, namun akan ditampilkan setelah tombol ditekan dan hasil tersedia.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat https://github.com/Noviana21/Pemrograman-Mobile/tree/main/modul%202/calculator%20tip.