LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2

ANDROID LAYOUT

Oleh:

Damarjati Suryo Laksono 2310817210014

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 Prakitkum ini dikerjakan oleh:

Nama Praktikan : Damarjati Suryo Laksono
NIM : 2310817210014

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

Soal Praktikum:

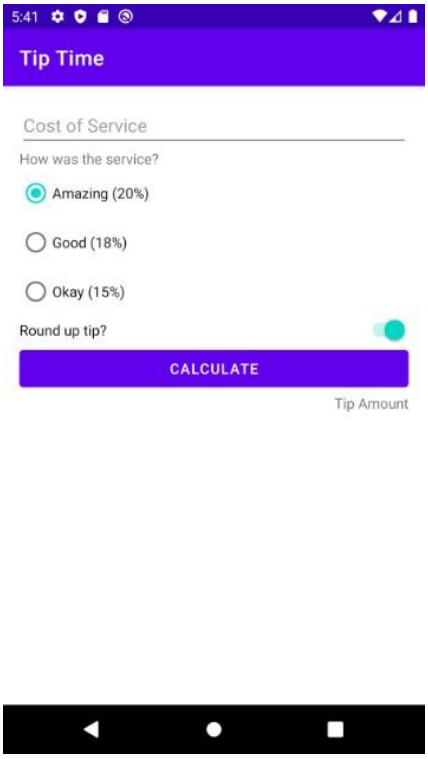
Buatlah sebuah aplikasi kalkulator tip yang dirancang untuk membantu pengguna menghitung tip yang sesuai berdasarkan total biaya layanan yang mereka terima. Fitur-fitur yang diharapkan dalam aplikasi ini mencakup:

1. Input Biaya Layanan: Pengguna dapat memasukkan total biaya layanan yang diterima dalam bentuk nominal.

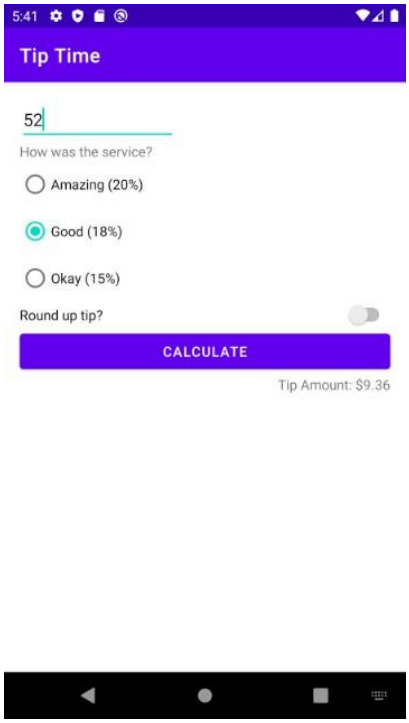
2. Pilihan Persentase Tip: Pengguna dapat memilih persentase tip yang diinginkan dari opsi yang disediakan, yaitu 15%, 18%, dan 20%.

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 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 androidx.appcompat.app.AppCompatDelegate
8  import
9  androidx.core.splashscreen.SplashScreen.Companion.installSpla
   shScreen
   import androidx.lifecycle.Observer
10 import com.example.tiptime.databinding.ActivityMainBinding
11
```

```
12 class MainActivity : AppCompatActivity() {
13
14     private lateinit var binding : ActivityMainBinding
15     private val viewModel: TipViewModel by viewModels()
16
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         installSplashScreen()
20
21
22 AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_
   NIGHT_NO)
23
24         binding = ActivityMainBinding.inflate(layoutInflater)
25         setContentView(binding.root)
26
27         binding.filledButton.setOnClickListener{
28             val input =
   binding.EditText.text?.toString()?.trim()
29             val isRounded = binding.tipround.isChecked
30             val tipPercentage = when
   (binding.radiobuttonGroup.checkedRadioButtonId){
31                 R.id.option1 -> 0.20
32                 R.id.option2 -> 0.18
33                 R.id.option3 -> 0.15
34                 else -> null
35             }
36             viewModel.calculateTip(input, tipPercentage,
   isRounded)
37         }
38
39         viewModel.tipResult.observe(this,Observer { result ->
40             binding.resultText.text = result
41         })
```

```
42          viewModel.errorMessage.observe(this,Observer{ message
    ->
43              message?.let {
44
45  Toast.makeText(this,it,Toast.LENGTH_SHORT).show()
46                  viewModel.errorMessageHandled()
47              }
48          })
49      }
50  }
```

2. TipViewModel.kt

Tabel 2 Source Code TipViewModel.kt

```
1   package com.example.tiptime
2
3   import androidx.lifecycle.LiveData
4   import androidx.lifecycle.MutableLiveData
5   import androidx.lifecycle.ViewModel
6   import kotlin.math.ceil
7
8   class TipViewModel: ViewModel() {
9       private val _tipResult = MutableLiveData<String>()
10      val tipResult: LiveData<String> = _tipResult
11
12      private val _errorMessage = MutableLiveData<String?>()
13      val errorMessage: LiveData<String?> = _errorMessage
14
15      fun calculateTip(ammountInput: String?,tipPercentage:
    Double?, isRounded: Boolean){
16          val ammount = ammountInput?.toDoubleOrNull()
17          if (ammount == null || tipPercentage == null){
18              _errorMessage.value = "Input Harus Berupa Angka"
19              return
```

```
20          }
21          if(ammount < 0){
22              _errorMessage.value = "Angka tidak boleh negatif"
23              return
24          }
25
26          val tip = ammount * tipPercentage
27          val total = if (isRounded) ammount + ceil(tip) else
    ammount + tip
28          _tipResult.value = "Tip Ammount $$total"
29      }
30
31      fun errorMessageHandled() {
32          _errorMessage.value = null
33      }
34
35  }
```

3. activity_main.xml

Tabel 3 Source Code activity_main.xml

```
1   <ScrollView
2   xmlns:android="http://schemas.android.com/apk/res/android"
3       xmlns:app="http://schemas.android.com/apk/res-auto"
4       xmlns:tools="http://schemas.android.com/tools"
5       android:layout_width="match_parent"
6       android:layout_height="match_parent"
7       android:background="@color/white"
8       tools:context=".MainActivity">
9       <androidx.constraintlayout.widget.ConstraintLayout
10          android:id="@+id/main"
11          android:layout_width="match_parent"
12          android:layout_height="match_parent"
13          android:background="@color/white"
14          tools:context=".MainActivity">
```

```
15          <TextView
16              android:id="@+id/headerTitle"
17              android:layout_width="0dp"
18              android:layout_height="wrap_content"
19              android:background="#6804ec"
20              android:padding="20dp"
21              android:text="@string/app_name"
22              android:textColor="#FFFFFF"
23              android:textSize="20sp"
24              app:layout_constraintEnd_toEndOf="parent"
25              app:layout_constraintHorizontal_bias="0.0"
26              app:layout_constraintStart_toStartOf="parent"
27              app:layout_constraintTop_toTopOf="parent" />
28
29
30  <com.google.android.material.textfield.TextInputLayout
31              android:id="@+id/textField"
32              android:layout_width="match_parent"
33              android:layout_height="wrap_content"
34              android:paddingTop="16dp"
35              android:paddingLeft="8dp"
36              android:paddingRight="8dp"
37              android:hint="@string/label"
38              app:helperTextEnabled="true"
39              app:helperText="@string/helper_text"
40              app:endIconMode="clear_text"
41              app:errorEnabled="true"
42
43  app:layout_constraintTop_toBottomOf="@id/headerTitle"
44              app:layout_constraintStart_toStartOf="parent"
45              app:layout_constraintEnd_toEndOf="parent">
46
47
48  <com.google.android.material.textfield.TextInputEditText
```

```
            android:id="@+id/EditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            />

</com.google.android.material.textfield.TextInputLayout>

        <RadioGroup
            android:id="@+id/radiobutton_group"
            android:checkedButton="@+id/enabled_selected"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            app:layout_constraintWidth_percent="0.95"

app:layout_constraintTop_toBottomOf="@+id/textField"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent">

            <RadioButton
                android:id="@+id/option1"
                android:checked="true"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:enabled="true"
                android:paddingStart="@dimen/padding_start"
                android:paddingEnd="@dimen/padding_start"
                android:text="@string/radiobutton_text1"
                android:tag="0.20"/>
            <RadioButton
                android:id="@+id/option2"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:checked="false"
                android:enabled="true"
```

```
                android:paddingStart="@dimen/padding_start"
                android:paddingEnd="@dimen/padding_start"
                android:text="@string/radiobutton_text2"
                android:tag="0.18"/>
            <RadioButton
                android:id="@+id/option3"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:checked="false"
                android:enabled="true"
                android:paddingStart="@dimen/padding_start"
                android:paddingEnd="@dimen/padding_start"
                android:text="@string/radiobutton_text3"
                android:tag="0.15"/>

        </RadioGroup>

        <TextView
            android:id="@+id/tipLabel"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:text="@string/label_1"
            android:gravity="start"
            android:textSize="14dp"
            android:layout_marginTop="1dp"
            android:layout_marginEnd="8dp"
            app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/radiobutton_group"

app:layout_constraintBottom_toBottomOf="@id/tipround"

app:layout_constraintEnd_toStartOf="@id/tipround"
            android:paddingStart="10dp"/>
```

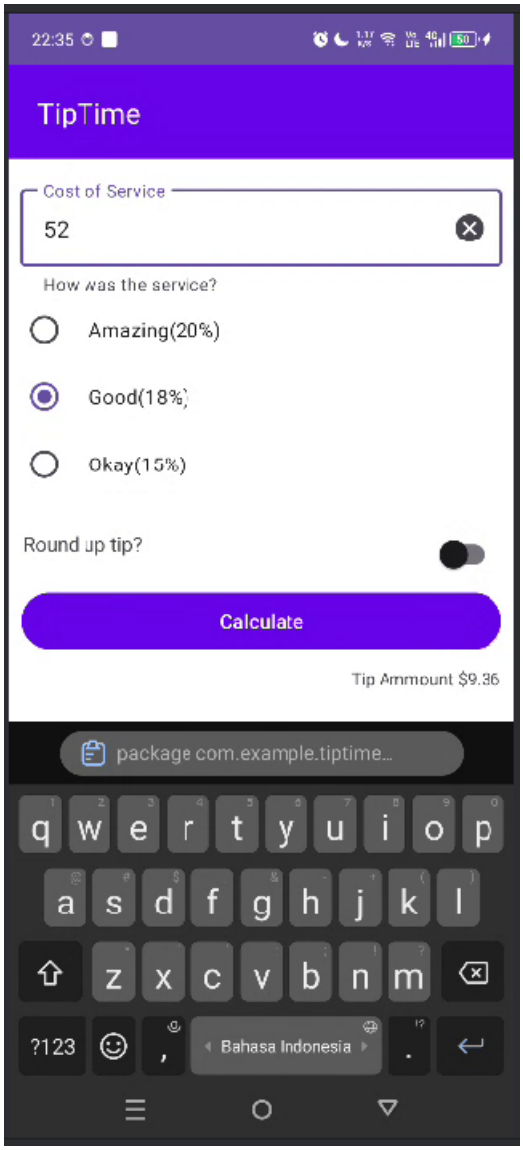
```
114
115
116 <com.google.android.material.switchmaterial.SwitchMaterial
117             android:id="@+id/tiproundㅤ"
118             android:layout_width="wrap_content"
119             android:layout_height="wrap_content"
120             android:layout_marginTop="16dp"
121             android:paddingRight="10dp"
122
123 app:layout_constraintBottom_toTopOf="@id/filledButton"
            app:layout_constraintEnd_toEndOf="parent"
124
125 app:layout_constraintTop_toBottomOf="@id/radiobutton_group"
    />
126
127         <Button
128             android:backgroundTint="#6804ec"
129             android:id="@+id/filledButton"
130             android:layout_width="0dp"
131             android:layout_height="wrap_content"
132             android:text="@string/button_text"
133             app:layout_constraintWidth_percent="0.95"
134             app:layout_constraintStart_toStartOf="parent"
135             app:layout_constraintEnd_toEndOf="parent"
136
137 app:layout_constraintTop_toBottomOf="@+id/tiproundㅤ" />
138
139         <TextView
140             android:id="@+id/result_text"
141             android:layout_width="150dp"
142             android:layout_height="33dp"
143             android:textSize="12dp"
144             android:layout_marginTop="8dp"
145             android:paddingEnd="10dp"
```

```
            android:gravity="right"
            android:text="@string/result_text"

app:layout_constraintTop_toBottomOf="@+id/filledButton"
            app:layout_constraintEnd_toEndOf="parent"
            android:textFontWeight="600"/>

    </androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>
```

B. Output Program



Gambar 1 Screenshot Vertikal Aplikasi

Gambar 2 Screenshoot Horizontal Aplikasi

C. Pembahasan

1. MainActivity.kt:

Pada line [1] – [10], terdapat package dan library – library yang akan di gunakan

Pada line [12], terdapat main activity yang merupakan wadah dari kode yang akan

Pada line [14] – [15] terdapat inisasi awal dari view binding dan view model

Pada line [19] terdapat inisiasi splashscreen untuk android dengan sdk 31+

Pada line [21] terdapat fungsi untuk memaksa pengguna nightmode agar tetap menggunakan theme light

Pada line [23] – [24] terdapat insiasi dari view binding

Pada line [26] – [36] terdapat syntax `binding.button.setOnClickListener`, pada syntax ini, button di set ke listener atau Ketika di klik akan melakukan aksi yang di tentukan, Dimana aksi disini akan menjalankan fungsi `calculateTip` pada `viewModel` dengan parameter `input`, `tipPercentage`, dan `isRounded`, dan juga pada line ini juga terdapat variable seperti `val input` di gunakan untuk menyimpan secara sementara input dari user,kemudian juga terdapat variable `isRounded` yang di gunakan untuk mengecek apakah user melakukan klik pada round option pada interface, dan juga terdapat `tipPercentage` yang di gunakan untuk menyimpan pilihan tip yang akan di berikan oleh user.

Pada line [38] – [40] terdapat fungsi seperti berikut

```
viewModel.tipResult.observe(this,Observer { result ->
    binding.resultText.text = result
```

Fungsi diatas ditujukan untuk melakukan checking apakah terdapat data di dalam variable `tipResult` yang terdapat pada `viewModel`, jika terdapat maka xml yang berkode `resultText` akan di rubah sesuai dengan `result` yang tersedia pada `viewModel`.

Pada line [41] – [46] terdapat fungsi sebagai berikut

```
viewModel.errorMessage.observe(this,Observer{ message ->
    message?.let                                        {

Toast.makeText(this,it,Toast.LENGTH_SHORT).show()
        viewModel.errorMessageHandled()
    }
})
```

Selanjutnya pada fungsi diatas digunakan untuk melakukan toast ketika `errorMessage` pada `viewModel` terisi, dan kemudian jika `Toast` sudah di tampilkan, maka `errorMessage` akan di bersihkan dengan fungsi `errorMessageHandled()` pada `view model`

2. TipViewModel.kt

Pada line [1] – [6], terdapat package dan library – library yang akan di gunakan

Pada line [8], terdapat `class TipViewModel: ViewModel()` yang merupakan wadah dari kode yang akan menjadi wadah untuk kode lainnya, dan juga menentukan bahwa class ini merupakan `Viewmodel()` dari projek ini.

Pada line [9] – [13], terdapat syntax `private val _tipResult = MutableLiveData<String>()`, pada syntax ini terdapat `private val _tipResult` yang menyebutkan bahwa value `_tipResult` hanya bisa di baca di class ini, dan penggunaan _ pada `_tipResult` penamaan ini menunjukan secara eksplisit bahwa variable ini adalah variable local, setelah itu terdapat `MutableLiveData<String>()`, yang berfungsi agar data bisa di rubah namun

hanya pada lingkup local. kemudian terdapat `val tipResult: LiveData<String> = _tipResult`, Dimana `val val tipResult: LiveData<String>` berguna untuk melakukan deklarasi bahwa `tipResult` bersifat unmutable dan hanya bisa di observe atau di lihat aja, begitu juga dengan syntax sejenisnya.

Pada line [15] – [29], terdapat fungsi `fun calculateTip()`, dengan parameter `ammountInput`, `tipPercentage`, dan `isRounded`, dan data yang dimasukan bisa bersifat `null` atau `kosong` untuk data `ammountInput` dan `tipPercentage`. fungsi ini bekerja dengan cara pertama pada line [16] `ammountTip` didefinisikan menjadi entah `double` atau `null`, kemudian pada baris [17] – [24] terdapat validasi apakah `ammount` merupakan angka atau bukan dan apakah `amount` merupakan bilangan negative atau tidak, jika terindikasi benar maka `errorMessage` akan di isi dengan text yang sudah disediakan, selanjutnya pada baris [26] –[29] terdapat logika sederhana untuk melakukan kalkulasi tip itu sendiri.

Pada line [31] – [33] terdapat fungsi `errorMessageHandled()`, yang jika di panggil maka akan mengganti value dari `errorMessage` menjadi `null` atau kosong.

3. activity_main.xml

Pada line [1], terdapat scroll view yang digunakan untuk melakukan scrolling pada aplikasi

Pada line [8] – [143] terdapat `layout constraint` yang digunakan sebagai layout utama pada modul ini, dan kemudian pada baris [14]-[26] terdapat `textView` yang digunakan sebagai `header`, selanjutnya pada baris [28] –[49] terdapat `textfield.TextInputLayout` yang digunakan sebagai wadah untuk `TextInputEditText` yang berguna sebagai input field pada layout ini, kemudian pada baris [51] –[92] terdapat `RadioGroup` dan `RadioButton`, dan pada baris [94] –[117] terdapat `textView` dan custom switch dari google, dan pada baris [119] –[141] terdapat `button` dan `textView`.

D. Additional feature

- viewBinding
- Toast when error is present
- Splashscreen animation
- Logo

E. Tautan Git

https://github.com/au290/Kuliah/tree/main/Pemro%20Mobile/Modul%202