**LAPORAN PRAKTIKUM**

**PEMROGRAMAN MOBILE**

**MODUL 2**

****

**ANDROID LAYOUT WITH COMPOSEOleh:**

**Muhammad Aufa Fitrianda NIM. 2310817210013**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMBUNG MANGKURAT**

**APRIL 2025**

# LEMBAR PENGESAHAN

**LAPORAN PRAKTIKUM MOBILE**

**MODUL 2**

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

Nama Praktikan : Muhammad Aufa Fitrianda

NIM : 2310817210013

|  |  |
| --- | --- |
| Menyetujui,  Asisten Praktikum  Muhammad Raka Azwar  NIM. 2210817210012 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Ir. Eka Setya Wijaya, S.T., M.Kom  NIP. 198205082008011010 |

# DAFTAR ISI

[LEMBAR PENGESAHAN 2](#_Toc196345480)

[DAFTAR ISI 3](#_Toc196345481)

[DAFTAR TABEL 4](#_Toc196345482)

[SOAL 1 5](#_Toc196345483)

[A. Source Code 5](#_Toc196345484)

[B. Pembahasan 8](#_Toc196345485)

[SOAL 2 8](#_Toc196345486)

[A. Pembahasan 8](#_Toc196345487)

[B. Tautan Git 9](#_Toc196345488)

# DAFTAR TABEL

[Tabel 1. Source Code Jawaban Soal 1 9](#_Toc166105944)

[Tabel 2. Source Code Jawaban Soal 1 10](#_Toc166105944)

# SOAL 1

**Soal Praktikum:**

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

a. Input biaya layanan: Pengguna dapat memasukkan total biaya layanan yang diterima dalam bentuk nominal.

b. Pilihan persentase tip: Pengguna dapat memilih persentase tip yang diinginkan.

c. Pengaturan pembulatan tip: Pengguna dapat memilih untuk membulatkan tip ke angka yang lebih tinggi.

d. Tampilan hasil: Aplikasi akan menampilkan jumlah tip yang harus dibayar secara langsung setelah pengguna memberikan input.

## Source Code

**MainActivity.kt**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70 | package com.example.tipcalculatorxml  import android.os.Bundle import android.text.Editable import android.text.TextWatcher import android.widget.\* import androidx.appcompat.app.AppCompatActivity import com.example.kalkulatortipcml.R import kotlin.math.ceil  class MainActivity : AppCompatActivity() {   private lateinit var serviceCostInput: EditText  private lateinit var tipPercentageSpinner: Spinner  private lateinit var roundTipSwitch: Switch  private lateinit var tipResultText: TextView   private val tipPercentages = *listOf*(10, 15, 20)   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   serviceCostInput = findViewById(R.id.*serviceCostInput*)  tipPercentageSpinner = findViewById(R.id.*tipPercentageSpinner*)  roundTipSwitch = findViewById(R.id.*roundTipSwitch*)  tipResultText = findViewById(R.id.*tipResultText*)   tipPercentageSpinner.*adapter* = ArrayAdapter(this, android.R.layout.*simple\_spinner\_dropdown\_item*, tipPercentages)   // 🛡️ Ubah jadi pakai TextWatcher supaya tidak langsung crash  serviceCostInput.addTextChangedListener(object : TextWatcher {  override fun afterTextChanged(s: Editable?) = updateTip()  override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {}  override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {}  })   tipPercentageSpinner.*onItemSelectedListener* = object : AdapterView.OnItemSelectedListener {  override fun onNothingSelected(parent: AdapterView<\*>?) {}  override fun onItemSelected(parent: AdapterView<\*>?, view: android.view.View?, position: Int, id: Long) {  updateTip()  }  }   roundTipSwitch.setOnCheckedChangeListener **{** \_, \_ **->** updateTip() **}** }   private fun updateTip() {  val cost = serviceCostInput.*text*.toString().*toDoubleOrNull*()  if (cost == null || cost == 0.0) {  tipResultText.*text* = "Jumlah Tip: Rp 0.00"  return  }   val tipPercentage = tipPercentages[tipPercentageSpinner.*selectedItemPosition*]  var tip = cost \* tipPercentage / 100  if (roundTipSwitch.*isChecked*) tip = *ceil*(tip)   tipResultText.*text* = "Jumlah Tip: Rp %.2f".*format*(tip)  } } |

Tabel 1. Source Code Jawaban Soal 1

**activity\_main.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | <?xml version="1.0" encoding="utf-8"?> <androidx.constraintlayout.widget.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:id="@+id/main\_layout"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:padding="24dp">   <EditText  android:id="@+id/serviceCostInput"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:hint="Biaya Layanan"  android:inputType="numberDecimal"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"/>   <Spinner  android:id="@+id/tipPercentageSpinner"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  app:layout\_constraintTop\_toBottomOf="@id/serviceCostInput"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  android:layout\_marginTop="16dp"/>   <Switch  android:id="@+id/roundTipSwitch"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Bulatkan Tip"  app:layout\_constraintTop\_toBottomOf="@id/tipPercentageSpinner"  app:layout\_constraintStart\_toStartOf="parent"  android:layout\_marginTop="16dp"/>   <TextView  android:id="@+id/tipResultText"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Jumlah Tip: Rp 0.00"  android:textSize="18sp"  app:layout\_constraintTop\_toBottomOf="@id/roundTipSwitch"  app:layout\_constraintStart\_toStartOf="parent"  android:layout\_marginTop="24dp"/> </androidx.constraintlayout.widget.ConstraintLayout> |

Tabel 2. Source Code Jawaban Soal 1

## Pembahasan

**MainActivity.kt:**

Terdapat setContentView(R.layout.activity\_main) Menampilkan layout activity\_main.xml.

Terdapat findViewById ...Menghubungkan komponen UI dengan variabel Kotlin.

Terdapat tipPercentageSpinner.adapter ... Mengisi spinner dengan data: [10, 15, 20]

Terdapat serviceCostInput.addTextChangedListener ... Menambahkan listener agar kalkulasi terjadi setiap kali input berubah.

Terdapat tipPercentageSpinner.onItemSelectedListener ... Memanggil updateTip() ketika persentase berubah.

Terdapat roundTipSwitch.setOnCheckedChangeListener ... Memanggil updateTip() jika switch diubah.

**activity\_main.xml:**

Terdapat <androidx.constraintlayout.widget.ConstraintLayout ...> Layout utama aplikasi menggunakan ConstraintLayout yang fleksibel dan powerful.

Terdapat <EditText ...> Untuk input biaya layanan, diberi ID serviceCostInput, dengan input type angka desimal.

Terdapat <Spinner ...> Untuk pilihan persentase tip. ID-nya tipPercentageSpinner.

Terdapat <Switch ...> Untuk mengaktifkan opsi pembulatan. ID-nya roundTipSwitch, labelnya "Bulatkan Tip"

Terdapat <TextView ...> Menampilkan hasil jumlah tip. Diberi ID tipResultText.

# SOAL 2

Jelaskan perbedaan dari implementasi XML dan Jetpack Compose beserta kelebihan dan kekurangan dari masing-masing implementasi.

## Pembahasan

Kelebihan dari xml yaitu mudah dipahami oleh pemula, stabil dan support banyak library sedangkank kekurangannya yaitu kurang reaktif dan sulit mengatur state dan update UI secara dinamis. Untuk kelebihan dari jetpack compose yaitu, reaktif dan dekleratif, lebih cepat untuk iterasi UI, dan tidak perlu findviewbyid(), lebih modern. Untuk kelemahannya yaitu memerlukan android studio versi terbaru,dan bagi dev xml yang lama memerlukan waktu untuk adaptasi.

## Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/AuFaMiReDo/PrakMobile