LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2

ANDROID LAYOUT

Oleh:

Rifky Putra Mahardika NIM. 2310817210023

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 Praktikum ini dikerjakan oleh:

Nama Praktikan : Rifky Putra Mahardika
NIM : 2310817210023

Menyetujui, Asisten Praktikum	Mengetahui, Dosen Penanggung Jawab Praktikum
Zulfa Auliya Akbar NIM. 2210817210026	Muti`a Maulida S.Kom M.T.I NIP. 198810272019032013

DAFTAR ISI

DAFTAR GAMBAR

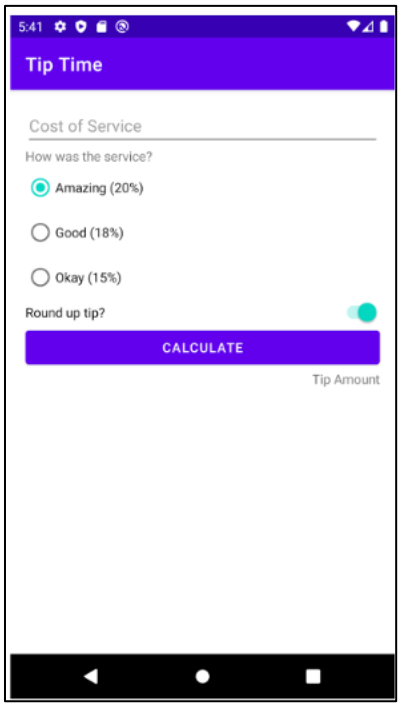
DAFTAR TABEL

SOAL 1

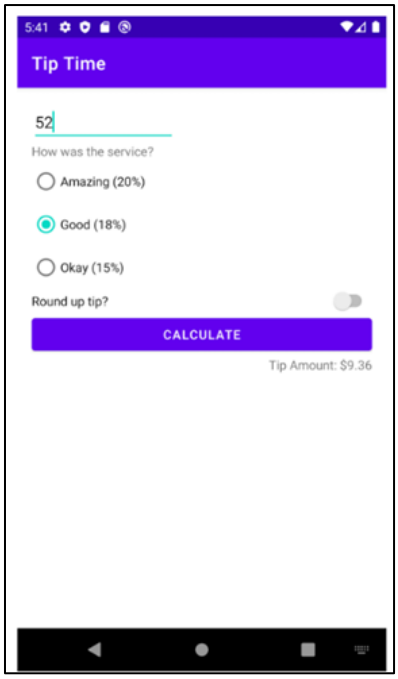
Soal Praktikum:

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

```
1   package com.presca.modul2
2
3   import android.os.Bundle
4   import android.widget.Toast
5   import androidx.activity.ComponentActivity
6   import androidx.activity.compose.setContent
7   import androidx.compose.foundation.background
8   import androidx.compose.foundation.layout.*
9   import androidx.compose.foundation.rememberScrollState
10  import androidx.compose.foundation.selection.selectable
11  import androidx.compose.foundation.shape.RoundedCornerShape
12  import androidx.compose.foundation.text.KeyboardOptions
13  import androidx.compose.foundation.verticalScroll
14  import androidx.compose.material3.*
15  import androidx.compose.runtime.*
```

```
16  import androidx.compose.runtime.saveable.rememberSaveable
17  import androidx.compose.ui.Alignment
18  import androidx.compose.ui.Modifier
19  import androidx.compose.ui.graphics.Color
20  import androidx.compose.ui.platform.LocalContext
21  import androidx.compose.ui.text.input.KeyboardType
22  import androidx.compose.ui.text.style.TextAlign
23  import androidx.compose.ui.unit.dp
24  import androidx.compose.ui.unit.sp
25  import com.presca.modul2.ui.theme.Modul2Theme
26
27  class MainActivity : ComponentActivity() {
28      override fun onCreate(savedInstanceState: Bundle?) {
29          super.onCreate(savedInstanceState)
30          setContent {
31              Modul2Theme {
32                  Surface(
33                      modifier = Modifier.fillMaxSize(),
34                      color =
    MaterialTheme.colorScheme.background
35                  ) {
36                      TipCalculatorScreen()
37                  }
38              }
39          }
40      }
41  }
42
43  @Composable
44  fun TipCalculatorScreen() {
45      var costInput by rememberSaveable { mutableStateOf("") }
46      var selectedOption by rememberSaveable {
    mutableStateOf(20) }
47      var roundUp by rememberSaveable { mutableStateOf(false)
```

```
    }
48      var tipResult by rememberSaveable { mutableStateOf("Tip
    Amount") }
49
50      val context = LocalContext.current
51
52      fun calculateTip() {
53          val cost = costInput.toDoubleOrNull()
54
55          if (cost == null || cost <= 0) {
56              Toast.makeText(
57                  context,
58                  "Masukkan angka positif dan bukan nol!",
59                  Toast.LENGTH_SHORT
60              ).show()
61              return
62          }
63
64          val tip = TipCalculator.calculateTip(
65              cost = cost,
66              percentage = selectedOption,
67              roundUp = roundUp
68          )
69          tipResult = "Tip Amount: \$${"%.2f".format(tip)}"
70      }
71
72      Column(
73          modifier = Modifier
74              .verticalScroll(rememberScrollState())
75              .fillMaxWidth()
76      ) {
77
78          Box(
79              modifier = Modifier
```

```
                .fillMaxWidth()
                .background(Color(0xFF6200EE))
                .padding(16.dp)
        ) {
            Text(
                text = "Tip Time",
                color = Color.White,
                fontSize = 20.sp,
                modifier =
Modifier.align(Alignment.CenterStart)
            )
        }

        Column(
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth(),
            verticalArrangement =
Arrangement.spacedBy(16.dp)
        ) {
            OutlinedTextField(
                value = costInput,
                onValueChange = { costInput = it },
                label = { Text("Cost of Service") },
                modifier = Modifier.fillMaxWidth(),
                keyboardOptions =
KeyboardOptions(keyboardType = KeyboardType.Number),
                shape = RoundedCornerShape(8.dp),
                colors = TextFieldDefaults.colors(
                    focusedIndicatorColor =
Color(0xFF6200EE),
                    unfocusedIndicatorColor = Color.Gray,
                    focusedContainerColor = Color.White,
                    unfocusedContainerColor = Color.White,
```

```
                    focusedLabelColor = Color(0xFF6200EE),
                    unfocusedLabelColor = Color.Gray
                ),
                singleLine = true
            )

            Text(
                text = "How was the service?",
                color = Color.Gray
            )

            val options = listOf(
                "Amazing (20%)" to 20,
                "Good (18%)" to 18,
                "Okay (15%)" to 15
            )

            options.forEach { (label, value) ->
                Row(
                    verticalAlignment =
Alignment.CenterVertically,
                    modifier = Modifier
                        .fillMaxWidth()
                        .selectable(
                            selected = (selectedOption ==
value),
                            onClick = { selectedOption =
value }
                        )
                        .padding(4.dp)
                ) {
                    RadioButton(
                        selected = (selectedOption ==
value),
```

```
                        onClick = { selectedOption = value }
                    )
                    Text(text = label)
                }
            }

            Row(
                verticalAlignment =
Alignment.CenterVertically
            ) {
                Text("Round up tip?")
                Spacer(modifier = Modifier.weight(1f))
                Switch(
                    checked = roundUp,
                    onCheckedChange = { roundUp = it }
                )
            }

            Button(
                onClick = { calculateTip() },
                modifier = Modifier
                    .fillMaxWidth()
                    .height(48.dp),
                colors = ButtonDefaults.buttonColors(
                    containerColor = Color(0xFF6200EE)
                ),
                shape = RoundedCornerShape(0.dp)
            ) {
                Text("CALCULATE", color = Color.White,
fontSize = 16.sp)
            }

            Text(
                text = tipResult,
```

```
172                 modifier = Modifier
173                     .fillMaxWidth()
174                     .padding(top = 4.dp),
175                 textAlign = TextAlign.End,
176                 color = Color.Gray
177             )
178         }
179     }
180 }
```

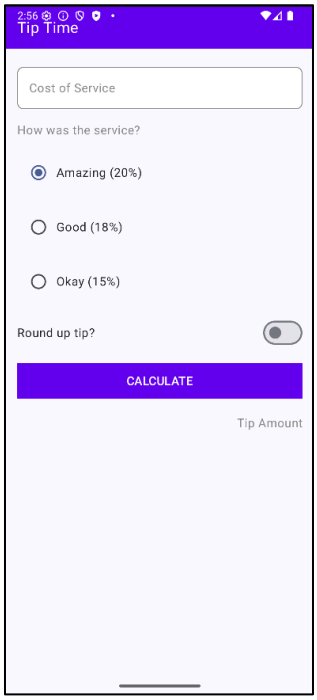
Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt

2. TipCalculator.kt

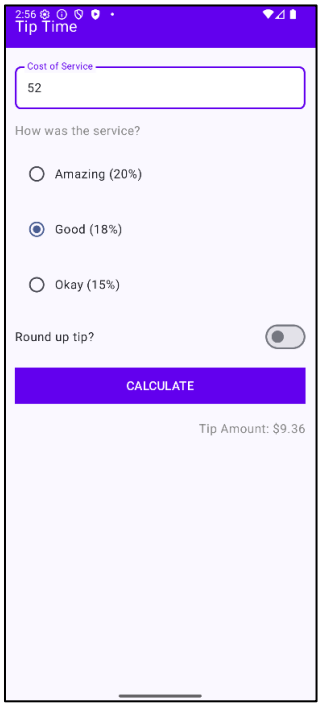
```
1   package com.presca.modul2
2
3   import kotlin.math.ceil
4
5   object TipCalculator {
6       fun calculateTip(cost: Double, percentage: Int, roundUp:
    Boolean): Double {
7           var tip = cost * percentage / 100
8           if (roundUp) tip = ceil(tip)
9           return tip
10      }
11  }
```

Tabel 2. Source Code Jawaban Soal 1 TipCalculator.kt

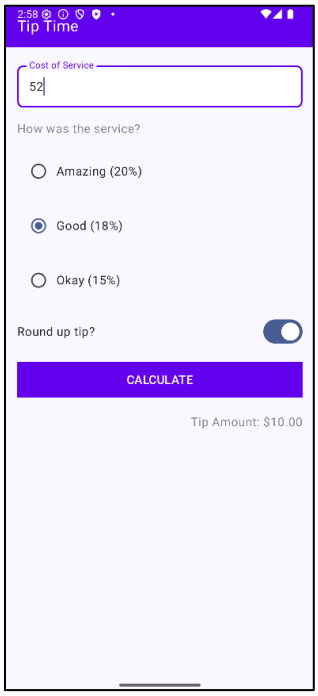
B. Output Program



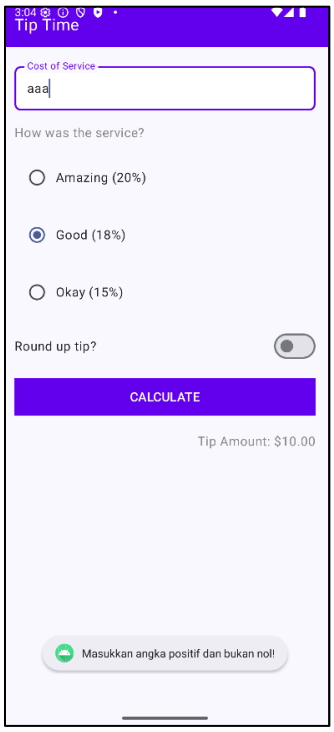
Gambar 3. Screenshot Hasil Jawaban Soal 1 Tampilan Awal



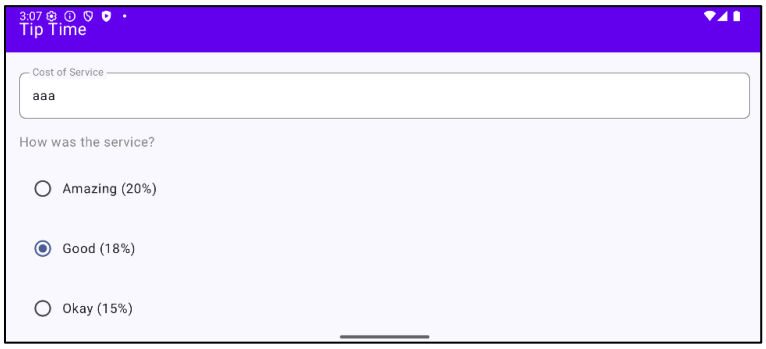
Gambar 4. Screenshot Hasil Jawaban Soal 1 Saat Tip Dihitung



Gambar 5. Screenshot Hasil Jawaban Soal 1 Saat Tip Dibulat



Gambar 6. Screenshot Hasil Jawaban Soal 1 Toast



Gambar 7. Screenshot Hasil Jawaban Soal 1 Mode Landscape

C. Pembahasan

1. MainActivity.kt:

- Pada baris [1], dideklarasikan nama package file Kotlin yang dikelompokkan file ini ke dalam `package com.presca.modul2`.
- Pada baris [3] hingga [25], `import` adalah perintah yang digunakan untuk mengimpor kelas, fungsi, atau objek dari package lain tanpa harus menyebutkan path lengkapnya.
- Pada baris [27], `class MainActivity : ComponentActivity()` ini digunakan sebagai titik awal aplikasi yang akan mengatur tampilan aplikasi.
- Pada baris [28], `onCreate()` merupakan siklus hidup (lifecycle) dari sebuah activity, yang dimana fungsi ini dipanggil pertama kali saat Activity dibuat.
- Pada baris [29], `super.onCreate(savedInstanceState)` ini memanggil `onCreate()` agar proses inisialisasi standar tetap berjalan.
- Pada baris [30], `setContent{…}` digunakan untuk menetapkan tampilan UI dari aplikasi, dan diterapkan tampilan dari `Prakmodul1Theme`.
- Pada baris [31], diterapkan tema khusus aplikasi dari `Modul2Theme`.
- Pada baris [32], `Surface` digunakan untuk wadah yang menampung UI.
- Pada baris [36], `TipCalculatorScreen()` dipanggil untuk menampilkan antarmuka kalkulator tip.

- Pada baris [43], @Composable merupakan anotasi dari Compose, yang dimana ini sebagai penanda bahwa `TipCalculatorScreen` adalah fungsi yang bisa digunakan dalam UI deklaratif Jetpack Compose.
- Pada baris [44], `fun TipCalculatorScreen()` bagian ini bisa digunakan untuk menampilkan UI, mengatur input, dan berinteraksi dengan state.
- Pada baris [50], `val context` ini akan mengambil Contect dari sistem Android yang digunakan untuk kebutuhan non-UI.
- Pada baris [52] dan [53], pada `calculateTip()` ini akan mengunggah costInput (string input teks) menjadi Double, jika gagal maka input tidak valid (bukan angka positif).
- Pada baris [55] hingga [61], dilakukan pengkondisian `if`, divalidasikan jika input yang dimasukkan pengguna kosong (null) atau kurang dari nol, maka akan muncul pesan "Masukkan angka positif dan bukan nol!"
- Pada baris [64] hingga [67], `val tip = TipCalculator.calculateTip()` ini akan memangil fungsi `TipCalculator.calculateTip` untuk menghitung nilai tip berdasarkan input biaya, persentase tip, dan pembulatannya.
- Pada baris [69], `tipResult` akan menampilkan hasil dari perhitungan tip dan akan `{"%.2f".format(tip)}` ini akan memberi format dua angka desimal di belakang koma.
- Pada baris [72], `Column` ini untuk menyusun komponen UI secara vertikal.
- Pada baris [74], `verticalScroll()` digunakan agar kolom pada bagian ini bisa discroll jika kontennya melebihi tinggi layar.
- Pada baris [75], `fillMaxWidth()` digunakan agar kolom memenuhi lebar layar.
- Pada baris [78] hingga [82], `Box` pada bagian ini digunakan untuk menyusun teks agar bisa rata kiri.
- Pada baris [84] hingga [88], `Text` ini digunakan untuk menampilkan teks "Tip Time" pada box.

- Pada baris [92] hingga [96], `Column` pada bagian ini digunakan untuk memberi jarak dari tepi layar (`padding`), digunakan `fillMaxWidth()` agar lebarnya memenuhi layar, dan `Arrangement.spacedBy(16.dp)` untuk memberikan jarak 16dp antar komponen di dalam kolom.
- Pada baris [98] hingga [113], `OutlinedTextField` ini untuk mengatur input teks dalam biaya layanan. `onValueChange = { costInput = it }` berfungsi untuk memperbarui `costInput` saat user mengetik, `keyboardOptions` digunakan agar input ini hanya menerima input angka (`KeyboardType.Number`), shape digunakan untuk membulatkan sudut border, `colors` untuk menyesuaikan warna, dan `singleLine = true` digunakan agar input tetap 1 baris.
- Pada baris [116] hingga [118], `Text` pada baris ini digunakan untuk menampilkan dan mengatur teks “How was the service?”.
- Pada baris [121] hingga [124], dibuat radio button untuk memilihi persentase tip.
- Pada baris [127], dibuat iterasi `forEach` pada `options.forEach` untuk opsi di dalamnya.
- Pada baris [128] hingga [136], `Row` pada bagian ini untuk isi (RadioButton dan Teks) dalam bentuk horizontal. `selected = (selectedOption == value)` digunakan untuk menentukan apakah baris ini dipilih, `onClick = { selectedOption = value }` digunakan pada saat radio button diklik, ubah pilihan ke value yang sesuai.
- Pada baris [138] hingga [142], dibuat fungsi dari RadioButton dan labelnya. Pada saat radio button diklik, maka radio button akan tercentang (fungsi dari `selected = (selectedOption == value)`) dan `onClick = { selectedOption = value }` digunakan pada saat radio diklik, ubah nilai `selectedOption`. `Text` digunakan untuk menampilkan teks dari pilihan.
- Pada baris [146], `Row` ini untuk membuat baris horizontal yang berisi teks dan switch.
- Pada baris [149], `Text("Round up tip?")` ini untuk menampilkan teks Round up tip?”.

- Pada baris [150], `Spacer(modifier = Modifier.weight(1f))` ini digunakan agar switch round up ada di bagian paling kanan.
- Pada baris [151] hingga [153], `Switch` ini adalah tombol on/off, jika switch statusnya on maka `roundUp` bernilai `true`.
- Pada baris [157] hingga [167], `Button` ini digunakan untuk mengatur tombol utama dari "Calculate". Ketika tombol ini ditekan, maka fungsi `calculateTip()` akan dipanggil untuk menghitung tip.
- Pada baris [170] hingga [176], `Text` pada bagian ini digunakan untuk menampilkan dan mengatur style hasil dari perhitungan tip.

2. TipCalculator.kt

- Pada baris [1], dideklarasikan nama package file Kotlin yang dikelompokkan file ini ke dalam `package com.presca.modul2`.
- Pada baris [2], `import` adalah perintah yang digunakan untuk mengimpor kelas, fungsi, atau objek dari package lain tanpa harus menyebutkan path lengkapnya. Import pada baris ini akan mengimpor fungsi `ceil()` yang digunakan untuk membulatkan angka ke atas.
- Pada baris [5], `object` ini menandakan bahwa aka nada satu instance dari TipCalculator.
- Pada baris [6], `cost: Double` digunakan untuk nilai biaya layanan yang sudah dipastikan valid dan juga dikonversi dari input pengguna, `percentage: Int` untuk mengatur persentase tip yang dipilih, dan `roundup: Boolean` digunakan untuk memastikan apakah hasil tip perlu dibulatkan ke atas atau tidak.
- Pada baris [7], `var tip = cost * percentage / 100` ini digunakan untuk menghitung jumlah tip awal sebelum dibulatkan.
- Pada baris [8], `if (roundUp) tip = ceil(tip)` ini merupakan pengkondisian jika pengguna ingin membulatkan tip, maka tip akan dibulatkan ke atas.
- Pada baris [9], `return tip` ini untuk mengembalikan hasil perhitungan tip ke pemanggil fungsinya.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/Prescaa/Kuliah/tree/master/Praktikum%20Mobile/MODUL%202/Modul2