

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2**



ANDROID LAYOUT

Oleh:

Muhammad Azwin Hakim

NIM. 2310817310012

**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 : Muhammad Azwin Hakim
NIM : 2310817310012

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

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	7
B. Output Program.....	10
C. Pembahasan	13
D. Tautan Git	14

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	10
Gambar 2. Screenshot Hasil Jawaban Soal 1	11
Gambar 3. Screenshot Hasil Jawaban Soal 1	11
Gambar 4. Screenshot Hasil Jawaban Soal 1	12
Gambar 5. Screenshot Hasil Jawaban Soal 1	12

DAFTAR TABEL

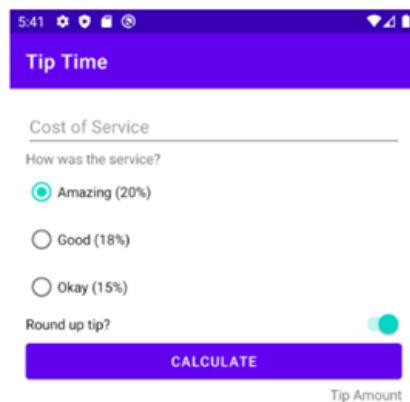
Tabel 1. Source Code Jawaban Soal 1.....	10
--	----

SOAL 1

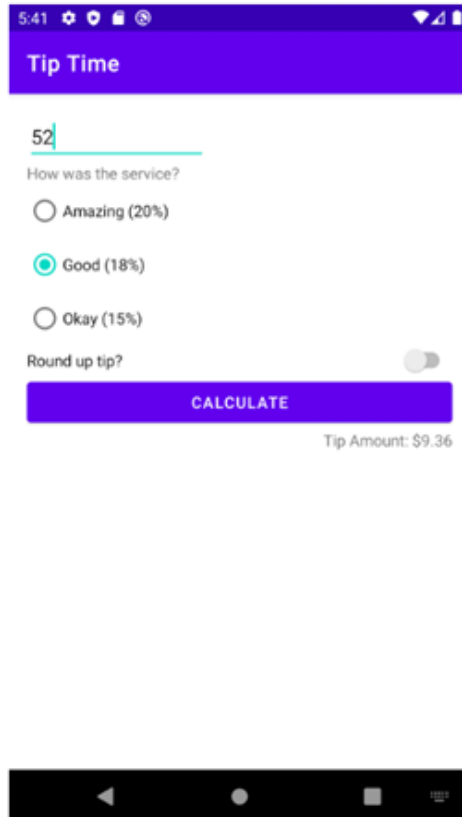
Soal Praktikum:

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 1 Tampilan Aplikasi Setelah Dijalankan

A. Source Code

1. MainActivity.kt

```

1 package com.example.tiptime
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.layout.*
8 import androidx.compose.foundation.text.KeyboardActions
9 import androidx.compose.foundation.text.KeyboardOptions
10 import androidx.compose.material3.*
11 import androidx.compose.runtime.*
12 import androidx.compose.ui.Alignment
13 import androidx.compose.ui.Modifier
14 import androidx.compose.ui.platform.LocalContext
15 import androidx.compose.ui.platform.LocalFocusManager
16 import androidx.compose.ui.text.input.ImeAction
17 import androidx.compose.ui.text.input.KeyboardType
18 import androidx.compose.ui.unit.dp
19 import androidx.compose.foundation.rememberScrollState
20 import androidx.compose.foundation.verticalScroll

```

```

21 import com.example.tiptime.ui.theme.TipTimeTheme
22 import androidx.compose.runtime.saveable.rememberSaveable
23 import kotlin.math.ceil
24
25 class MainActivity : ComponentActivity() {
26     override fun onCreate(savedInstanceState: Bundle?) {
27         super.onCreate(savedInstanceState)
28         setContent {
29             TipTimeTheme {
30                 Surface(
31                     modifier = Modifier.fillMaxSize(),
32                     color = MaterialTheme.colorScheme.background
33                 ) {
34                     TipTimeScreen()
35                 }
36             }
37         }
38     }
39 }
40
41 @Composable
42 fun TipTimeScreen() {
43     var amountInput by rememberSaveable { mutableStateOf("") }
44     var tipPercentage by rememberSaveable {
45         mutableDoubleStateOf(0.20) }
46     var roundUp by rememberSaveable { mutableStateOf(false) }
47     var tipResult by rememberSaveable { mutableStateOf("") }
48
49     val context = LocalContext.current
50     val focusManager = LocalFocusManager.current
51     val scrollState = rememberScrollState()
52
53     Column(
54         modifier = Modifier
55             .fillMaxSize()
56             .padding(16.dp)
57             .verticalScroll(scrollState)
58     ) {
59         OutlinedTextField(
60             value = amountInput,
61             onValueChange = { amountInput = it },
62             label = { Text("Cost of Service") },
63             singleLine = true,
64             keyboardOptions = KeyboardOptions.Default.copy(
65                 keyboardType = KeyboardType.Number,
66                 imeAction = ImeAction.Done
67             ),
68             keyboardActions = KeyboardActions(
69                 onDone = { focusManager.clearFocus() }
70             ),
71             modifier = Modifier.fillMaxWidth()

```



```

71 )
72
73     Spacer(modifier = Modifier.height(16.dp))
74
75     Text("How was the service?")
76     Spacer(modifier = Modifier.height(8.dp))
77
78     val radioOptions = listOf(
79         "Amazing (20%)" to 0.20,
80         "Good (18%)" to 0.18,
81         "Okay (15%)" to 0.15
82     )
83
84     radioOptions.forEach { (label, value) ->
85         Row(
86             verticalAlignment
Alignment.CenterVertically,
87             modifier = Modifier.padding(vertical = 4.dp)
88         ) {
89             RadioButton(
90                 selected = tipPercentage == value,
91                 onClick = { tipPercentage = value }
92             )
93             Text(label)
94         }
95     }
96
97     Spacer(modifier = Modifier.height(16.dp))
98
99     Row(
100         verticalAlignment = Alignment.CenterVertically,
101         horizontalArrangement = Arrangement.SpaceBetween,
102         modifier = Modifier.fillMaxWidth()
103     ) {
104         Text("Round up tip?")
105         Switch(
106             checked = roundUp,
107             onChange = { roundUp = it }
108         )
109     }
110
111     Spacer(modifier = Modifier.height(16.dp))
112
113     Button(
114         onClick = {
115             val amount = amountInput.toDoubleOrNull()
116
117             if (amount == null || amount <= 0) {
118                 Toast.makeText(context, "Harus di isi dan
berupa angka!", Toast.LENGTH_SHORT).show()
119             } else {
120                 var tip = amount * tipPercentage

```

```

121         if (roundUp) {
122             tip = ceil(tip)
123         }
124         tipResult = "Tip Amount: $%.2f".format(tip)
125     }
126 },
127     modifier = Modifier.fillMaxWidth()
128 ) {
129     Text("CALCULATE")
130 }
131
132     Spacer(modifier = Modifier.height(16.dp))
133
134     Box(
135         modifier = Modifier.fillMaxWidth()
136     ) {
137         Text(
138             text = if (tipResult.isEmpty()) "Tip Amount:"
139             else tipResult,
140             style = MaterialTheme.typography.bodySmall,
141             modifier = Modifier.align(Alignment.CenterEnd)
142         )
143     }
144 }

```

Tabel 1. Source Code Jawaban Soal 1

B. Output Program

- Tampilan awal :



Gambar 1. Screenshot Hasil Jawaban Soal 1

- Tampilan Aplikasi Setelah Dijalankan :

16:32

Cost of Service

52

How was the service?

☒ Amazing (20%)

☐ Good (18%)

☐ Okay (15%)

Round up tip? ☒

CALCULATE

Tip Amount: \$10.40

Gambar 2. Screenshot Hasil Jawaban Soal 1

16:32

Cost of Service

52

How was the service?

☐ Amazing (20%)

☐ Good (18%)

☒ Okay (15%)

Round up tip? ☒

CALCULATE

Tip Amount: \$7.80

Gambar 3. Screenshot Hasil Jawaban Soal 1

16:32

Cost of Service

52

How was the service?

☐ Amazing (20%)

☒ Good (18%)

☐ Okay (15%)

Round up tip? ☐

CALCULATE

Tip Amount: \$9.36

Gambar 4. Screenshot Hasil Jawaban Soal 1

- Hasil Pembulatan Tip :

16:32

Cost of Service

52

How was the service?

☐ Amazing (20%)

☒ Good (18%)

☐ Okay (15%)

Round up tip? ☒

CALCULATE

Tip Amount: \$10.00

Gambar 5. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

Kode dimulai dengan mendeklarasikan kelas MainActivity, yang merupakan kelas utama yang mengelola tampilan aplikasi. Pada metode onCreate, aplikasi memanggil fungsi setContentView untuk menetapkan antarmuka pengguna menggunakan Jetpack Compose. Di dalam setContentView, tema aplikasi TipTimeTheme diterapkan, dan tampilan utama dari aplikasi ini adalah hasil dari pemanggilan fungsi komposisi TipTimeScreen. Fungsi ini berisi seluruh logika tampilan dan interaksi pengguna.

Di dalam fungsi TipTimeScreen, terdapat beberapa variabel yang menyimpan data input dan status pengguna. Variabel amountInput digunakan untuk menyimpan nilai yang dimasukkan pengguna sebagai biaya layanan. Variabel tipPercentage menyimpan persentase tip yang dipilih oleh pengguna, seperti 20% untuk layanan yang luar biasa, 18% untuk layanan yang baik, dan 15% untuk layanan yang cukup. Variabel roundUp adalah sebuah boolean yang menunjukkan apakah pengguna ingin membulatkan jumlah tip ke angka terdekat. Hasil perhitungan tip disimpan dalam variabel tipResult, yang akan menampilkan hasil akhir perhitungan tip di layar. Semua variabel ini disimpan dengan menggunakan rememberSaveable, yang memungkinkan data ini tetap bertahan meskipun aplikasi dihentikan atau digulir.

Antarmuka pengguna terdiri dari beberapa elemen UI yang terstruktur dalam kolom. OutlinedTextField digunakan untuk menerima input biaya layanan dari pengguna, di mana hanya angka yang dapat dimasukkan. Keyboard yang muncul saat pengguna mengetik dikonfigurasi untuk hanya menerima angka, dan ada opsi untuk menekan tombol "Done" yang menghapus fokus dari input. Selain itu, ada tiga opsi radio button yang memungkinkan pengguna untuk memilih persentase tip yang sesuai dengan pengalaman layanan mereka, yaitu 20%, 18%, atau 15%. Opsi lain yang disediakan adalah Switch, yang memungkinkan pengguna memilih apakah mereka ingin membulatkan jumlah tip yang dihitung.

Ketika pengguna menekan tombol "CALCULATE", aplikasi memeriksa apakah input biaya layanan yang dimasukkan valid. Jika nilai yang dimasukkan tidak valid, aplikasi akan menampilkan pesan toast yang mengingatkan pengguna untuk memasukkan angka yang valid. Jika input valid, aplikasi kemudian menghitung jumlah tip berdasarkan biaya layanan yang dimasukkan dan persentase yang dipilih. Jika opsi "Round up" diaktifkan, jumlah tip dibulatkan menggunakan fungsi ceil untuk memastikan angka tersebut dibulatkan ke atas. Hasil perhitungan kemudian ditampilkan pada layar, dengan format yang menunjukkan jumlah tip yang perlu diberikan.

Aplikasi ini juga menyediakan mekanisme untuk mengelola fokus pengguna dengan menggunakan LocalFocusManager, yang memungkinkan untuk menghapus fokus dari elemen input setelah pengguna selesai mengisinya. Selain itu, tampilan aplikasi dapat digulirkan menggunakan rememberScrollState dan verticalScroll, yang berguna ketika keyboard muncul dan mengambil ruang layar.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Omelette719/Pemrograman-Mobile.git>