

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2**



ANDROID LAYOUT

Oleh:

Natalie Grace Katiandagho

NIM. 2310817120003

**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 : Natalie Grace Katiandagho
NIM : 2310817120003

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Natalie Grace Katiandagho
NIM. 2310817120003

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	12
C. Pembahasan	12
D. Tautan Git	16

DAFTAR GAMBAR

Tabel 1. Source Code Jawaban Soal 1.....	11
--	----

DAFTAR TABEL

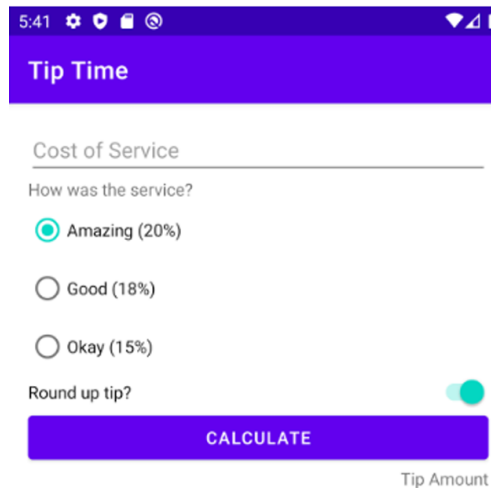
Tabel 1. Source Code Jawaban Soal 1	Error! Bookmark not defined.
Tabel 2. Source Code Jawaban Soal 1	10

SOAL 1

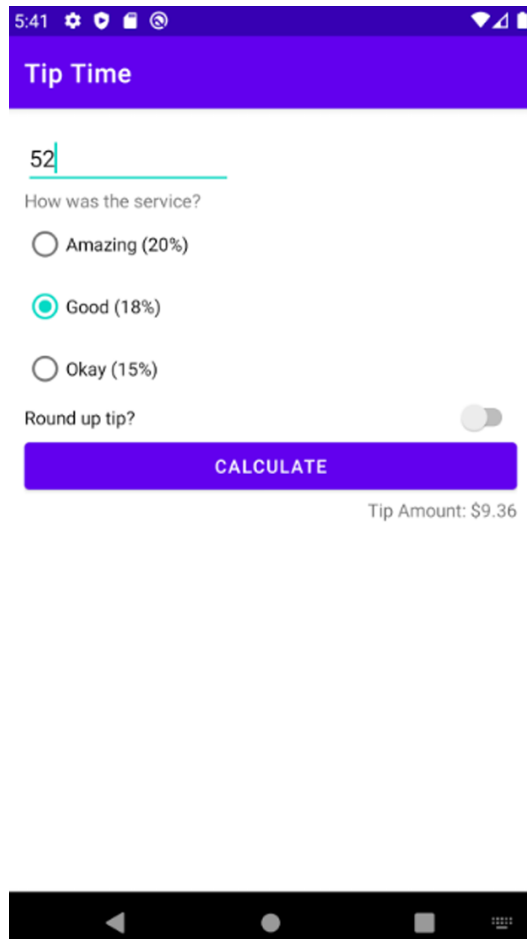
Soal Praktikum:

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 2 Tampilan Aplikasi Setelah Dijalankan

A. Source Code

1. MainActivity.kt

1	package com.example.tipcalculator
2	
3	import android.os.Bundle
4	import androidx.activity.ComponentActivity
5	import androidx.activity.compose.setContent
6	import androidx.annotation.StringRes
7	import androidx.compose.foundation.layout.Arrangement
8	import androidx.compose.foundation.layout.Column
9	import androidx.compose.foundation.layout.Row
10	import androidx.compose.foundation.layout.fillMaxSize
11	import androidx.compose.foundation.layout.fillMaxWidth
12	import androidx.compose.foundation.layout.height
13	import androidx.compose.foundation.layout.padding
14	import androidx.compose.foundation.rememberScrollState
15	import androidx.compose.foundation.text.KeyboardOptions
16	import androidx.compose.foundation.verticalScroll

```

17 import androidx.compose.material3.MaterialTheme
18 import androidx.compose.material3.RadioButton
19 import androidx.compose.material3.Surface
20 import androidx.compose.material3.Switch
21 import androidx.compose.material3.Text
22 import androidx.compose.material3.TextField
23 import androidx.compose.runtime.Composable
24 import androidx.compose.runtime.getValue
25 import androidx.compose.runtime.mutableDoubleStateOf
26 import androidx.compose.runtime.mutableStateOf
27 import androidx.compose.runtime.remember
28 import androidx.compose.runtime.setValue
29 import androidx.compose.ui.Alignment
30 import androidx.compose.ui.Modifier
31 import androidx.compose.ui.res.stringResource
32 import androidx.compose.ui.text.input.ImeAction
33 import androidx.compose.ui.text.input.KeyboardType
34 import androidx.compose.ui.tooling.preview.Preview
35 import androidx.compose.ui.unit.dp
36 import com.example.tipcalculator.ui.theme.TipCalculatorTheme
37 import java.text.NumberFormat
38 import kotlin.math.ceil
39
40 class MainActivity : ComponentActivity() {
41     override fun onCreate(savedInstanceState: Bundle?) {
42         super.onCreate(savedInstanceState)
43         setContent {
44             TipCalculatorTheme {
45                 Surface(modifier = Modifier.fillMaxSize()) {
46                     TipCalculatorLayout()
47                 }
48             }
49         }
50     }
51 }
52
53 @Composable
54 fun TipCalculatorLayout() {
55     var amountInput by remember { mutableStateOf("") }
56     var tipPercent by remember { mutableDoubleStateOf(15.0) }
57     var roundUp by remember { mutableStateOf(false) }
58
59     val amount = amountInput.toDoubleOrNull() ?: 0.0
60     val tip = calculateTip(amount, tipPercent, roundUp)
61
62     Column(
63         modifier = Modifier
64             .padding(32.dp)
65             .verticalScroll(rememberScrollState()),
66         horizontalAlignment = Alignment.CenterHorizontally,
67         verticalArrangement = Arrangement.spacedBy(16.dp)

```



```

68     ) {
69         Text(text = stringResource(R.string.calculate_tip), style =
MaterialTheme.typography.headlineMedium)
70
71         EditNumberField(
72             label = R.string.bill_amount,
73             value = amountInput,
74             onValueChanged = { amountInput = it },
75             keyboardOptions = KeyboardOptions(
76                 keyboardType = KeyboardType.Number,
77                 imeAction = ImeAction.Next
78             )
79         )
80
81         TipOptions(tipPercent) {
82             tipPercent = it
83         }
84
85         RoundTheTipRow(roundUp = roundUp, onRoundUpChanged = { roundUp = it })
86
87         Text(
88             text = stringResource(R.string.tip_amount, tip),
89             style = MaterialTheme.typography.headlineSmall
90         )
91     }
92
93     @Composable
94     fun EditNumberField(
95         value: String,
96         @StringRes label: Int,
97         onValueChanged: (String) -> Unit,
98         keyboardOptions: KeyboardOptions,
99         modifier: Modifier = Modifier
100    ) {
101        TextField(
102            value = value,
103            onValueChange = onValueChanged,
104            singleLine = true,
105            modifier = modifier.fillMaxWidth(),
106            label = { Text(stringResource(label)) },
107            keyboardOptions = keyboardOptions
108        )
109    }
110
111    @Composable
112    fun TipOptions(selectedTip: Double, onTipSelected: (Double) -> Unit) {
113        Column {
114            Text(text = "Tip Percentage:")
115            Row(verticalAlignment = Alignment.CenterVertically) {
                listOf(15.0, 18.0, 20.0).forEach { percent ->

```

```

116         Row(
117             verticalAlignment = Alignment.CenterVertically,
118             modifier = Modifier.padding(end = 16.dp)
119         ) {
120             RadioButton(
121                 selected = selectedTip == percent,
122                 onClick = { onTipSelected(percent) }
123             )
124             Text(text = "${percent.toInt()}%")
125         }
126     }
127 }
128 }
129
130 @Composable
131 fun RoundTheTipRow(
132     roundUp: Boolean,
133     onRoundUpChanged: (Boolean) -> Unit
134 ) {
135     Row(
136         modifier = Modifier
137             .fillMaxWidth()
138             .height(48.dp),
139         verticalAlignment = Alignment.CenterVertically,
140         horizontalArrangement = Arrangement.SpaceBetween
141     ) {
142         Text(
143             text = stringResource(R.string.round_up_tip),
144             modifier = Modifier.weight(1f)
145         )
146         Switch(
147             checked = roundUp,
148             onCheckedChange = onRoundUpChanged
149         )
150     }
151 }
152
153 private fun calculateTip(amount: Double, tipPercent: Double = 15.0, roundUp: Boolean):
154 String {
155     var tip = tipPercent / 100 * amount
156     if (roundUp) {
157         tip = ceil(tip)
158     }
159     return NumberFormat.getCurrencyInstance().format(tip)
160 }
161
162 @Preview(showBackground = true)
163 @Composable
164 fun TipCalculatorPreview() {
165     TipCalculatorTheme {

```

165	TipCalculatorLayout()
166	}
167	}
168	
169	
170	

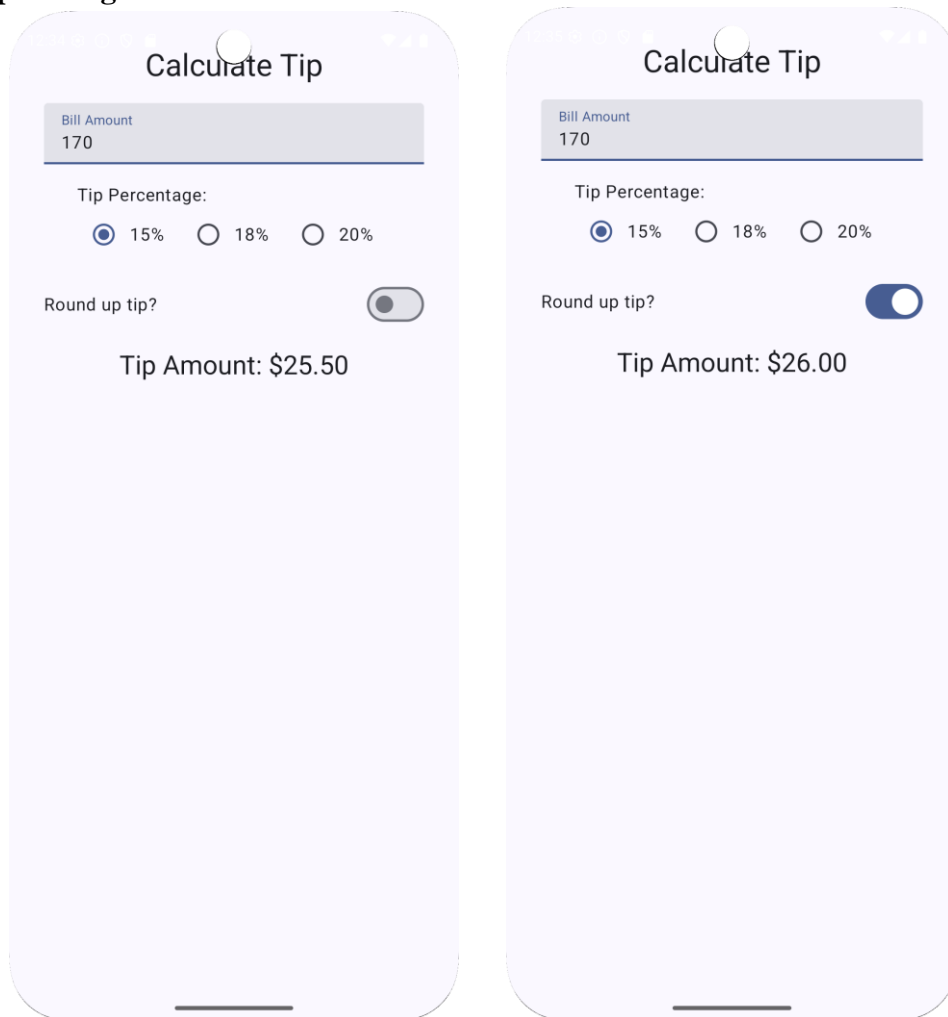
Tabel 1. Source Code Jawaban Soal 1

2. Strings.xml

1	<resources>
2	<string name="app_name">Tip Calculator</string>
3	<string name="calculate_tip">Calculate Tip</string>
4	<string name="bill_amount">Bill Amount</string>
5	<string name="tip">Tip (%)</string>
6	<string name="round_up_tip">Round up tip?</string>
7	<string name="tip_amount">Tip Amount: %1\$s</string>
8	</resources>

Tabel 2. Source Code Jawaban Soal 1

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt:

Pada line 1, dideklarasikan nama package file Kotlin yang pada praktikum kali ini adalah com.example.tipcalculator

Dari line 3-38 fungsinya untuk mengimport :

Pada line 3, Bundle buat komunikasiin data antar aktivitas

Pada line 4, AppCompatActivity sebagai superclass untuk activity yang menggunakan Jetpack Compose

Pada line 5, setContent untuk menampilkan UI berbasis Compose pada activity

Pada line 6, @StringRes jadi anotasi untuk resource ID bertipe string

Pada line 7–13, diimpor beberapa komponen layout :

- Arrangement, untuk pengaturan posisi child dalam layout
- Column, Row untuk struktur tata letak vertikal dan horizontal
- fillMaxSize, fillMaxWidth, height, padding untuk mengatur ukuran dan margin komponen

Pada line 14, rememberScrollState untuk menyimpan dan mengingat posisi scroll

Pada line 15, diimport KeyboardOptions untuk mengatur jenis keyboard dan aksi input

Pada line 16, diimport verticalScroll untuk memungkinkan scroll secara vertikal pada layout

Pada line 17–22, diimpor komponen Material 3:

- MaterialTheme, untuk mengakses tema aplikasi
- RadioButton untuk pemilihan opsi
- Switch untuk tombol on off fitur
- Text menampilkan tulisan ke layar
- TextField, digunakan untuk mengambil input teks dari pengguna

Pada line 23–28, fitur state Compose seperti:

- @Composable, untuk menandai fungsi sebagai composable
- remember, mutableStateOf, mutableDoubleStateOf, untuk mendefinisikan dan mengingat state yang dapat berubah
- getValue, setValue, untuk properti delegasi state

Pada line 29–35 :

- Modifier untuk mengatur tampilan, ukuran, padding komponen
- stringResource untuk mengambil string dari resource menggunakan ID
- ImeAction, KeyboardType untuk pengaturan aksi keyboard dan tipe input
- @Preview untuk pratinjau UI
- dp untuk unit ukuran
- TipCalculatorTheme untuk menggunakan tema khusus aplikasi
- NumberFormat untuk format angka

Pada line 36, untuk menerapkan tema khusus aplikasi yang sudah didefinisikan di folder ui.theme.

Pada line 37, untuk format angka ke dalam format mata uang.

Pada line 38, ceil dari kotlin.math untuk membulatkan nilai ke atas.

Pada line 40, kelas MainActivity yang turunan dari ComponentActivity

Pada line 41, ada fungsi yang dipanggil saat activity pertama kali dibuat. Parameter savedInstanceState digunakan untuk menyimpan data jika activity perlu dibuat ulang.

Pada line 42, memanggil onCreate dari superclass (ComponentActivity) agar fungsi dasar activity tetap berjalan.

Pada line 43, setContent untuk menampilkan UI didalam kurung kurawalnya nanti jadi isi tampilan Activity.

Pada line 44, buat temanya

Pada line 45, surface buat tempat UI bisa dikasih warna dll, fillMaxSize buat ngisi layar penuh

Pada line 46, panggil fungsi TipCalculatorLayout

Pada line 53, buat nandain fungsi UI.

Pada line 54, buat fungsi utama tampilan kalkulator.

Pada line 55, nyimpan input tagihan dari pengguna.

Pada line 56, buat nyimpan presentase tipnya.

Pada line 57, buat nyimpan pilihan tip mau dibulatkan atau engga.

Pada line 59, buat ngubah input ke angka, kalau kosong maka 0.0

Pada line 60, ngitung jumlah tip dari input, persen sama pembulatan tadi.

Pada line 62, buat nyusun semua elemen dari atas ke bawah.

Pada line 63 – 65, buat beri jarak 32dp di isi, terus buat kolom supaya bisa dicroll kalau isinya kepanjangan.

Pada line 66, buat rata tengah horizontal.

Pada line 67, buat ngasih jarak 16dp antar elemen.

Pada line 69, buat ngasih judul : Calculate Tip, uk medium.

Pada line 71-77, buat nampilin input angka di bill amount ada ambil nilai input, diperbarui setiap pengguna mengetik.

Pada line 81-82 buat nampilin pilihan persen tip

Pada line 85 isinya menampilkan switch on off buat tip dibulatkan atau engga, terus ngubah nilai roundUp nya pas diganti.

Pada line 86-88, ada text lagi buat nampilkan perhitungan tip dalam bentuk teks – uk teks kecil.

Pada line 93 ada composible lagi buat tampil UI.

Pada line 94, buat nampilin input angka .

Pada line, 95-99 ada :

- value: String , Nilai input yang sedang diketik.
- label: Int (@StringRes) , ID dari teks label
- onValueChanged: (String) -> Unit fungsi saat pengguna mengetik untuk memperbarui nilai.
- keyboardOptions: KeyboardOptions ngatur jenis keyboard.
- modifier: Modifier untuk pengaturan tampilan tambahan.

Pada TextField (line 101-107) nampilin input teks satu baris, lebarnya fillMaxWidth, label teks dari stringResouce(label), keyboardnya sesuai keyboardOptions.

Pada 111 ada composible lagi yang merupakan fungsi UI.

Pada line 112, nampilkan pilihan persen

Pada 113, text buat label teks

Pada line 114, ada row buat ngasih pilihan secara horizontal, rata tengah vertical.

Pada line 115, buat isi tiga pilihan persen

Di line 116-123, ada setiap pilihan di dalam row, isinya ada tombol pilihan radiobutton, aktif nilainya sama `selectedTip`, pas di klik muncul `onTipSelected(percent)`, text nya buat nampilin nilai persennya.

Pada line 130, sama seperti sebelumnya

Pada line 131-134, buat ngasih fungsi isinya teks sama switch untuk aktif atau nonaktif opsi dari pembulatan

Pada line 135-141, ada row buat elemen secara horizontal, `arrangement.SpaceBetween` buat jarak maz antara text dan switch, `Alignment.CenterVertically` buat mensejajarkan elemen secara vertical Tengah.

Pada line 142-145, buat nampilin text , `Modifier.weight(1f)` buat ngisi ruang yang tersedia.

Pada line 146-148, buat switch on off checked buat status switch ngikutin nilai `roundUp`, `onCheckedChange = onRoundChanged` buat manggil fungsi kalau switchnya diganti.

Pada line 153, merupakan fungsi ngitung jumlah tip berdasarkan input.

- `amount: Double` → Jumlah tagihan.
- `tipPercent: Double (default 15.0)` → Persentase tip.
- `roundUp: Boolean` → Jika true, hasil tip akan dibulatkan ke atas.

Isinya di line 155-158 :

- Hitung tip: `tip = tipPercent / 100 * amount`
- Jika `roundUp == true`, bulatkan ke atas dengan `ceil(tip)`
- Format hasil jadi bentuk uang: `NumberFormat.getCurrencyInstance().format(tip)`

Pada line 161-167, buat pratinjau tampilannya

- `@Preview(showBackground = true)` → Menampilkan preview dengan latar belakang.
- Di dalamnya, memanggil `TipCalculatorLayout()` dengan tema `TipCalculatorTheme`

2. Strings.xml

Pada baris 1, deklarasi resources yang berisi kumpulan string yang digunakan di seluruh aplikasi.

Pada baris 2, berisi nama aplikasi "Tip Calculator" yang akan muncul di layar utama aplikasi.

Pada baris 3, bernama `calculate_tip` untuk judul proses menghitung tip.

Pada baris 4, berisi teks "Bill Amount", digunakan untuk label input jumlah tagihan.

Pada baris 5, berfungsi sebagai label untuk memilih persentase tip, berisi teks "Tip (%)".

Pada baris 6, berisi teks "Round up tip?", digunakan sebagai label untuk opsi apakah ingin membulatkan tip atau tidak.

Pada baris 7, berisi teks "Tip Amount: %1\$s", di mana %1\$s adalah placeholder yang akan diisi dengan jumlah tip yang dihitung saat aplikasi dijalankan.

Pada baris 8, resource ditutup

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

[natnutnot/PrakMobile at master](#)