

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2**



ANDROID LAYOUT

Oleh:

Jovan Gilbert Natamasindah NIM. 2310817310002

**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 Praktikum ini dikerjakan oleh:

Nama Praktikan : Jovan Gilbert Natamasindah
NIM : 2310817310002

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	11
C. Pembahasan	12
D. Tautan Git.....	17

DAFTAR GAMBAR

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

DAFTAR TABEL

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

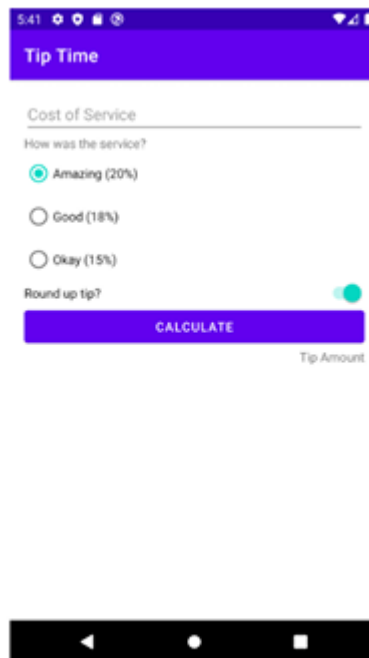
SOAL 1

Soal Praktikum:

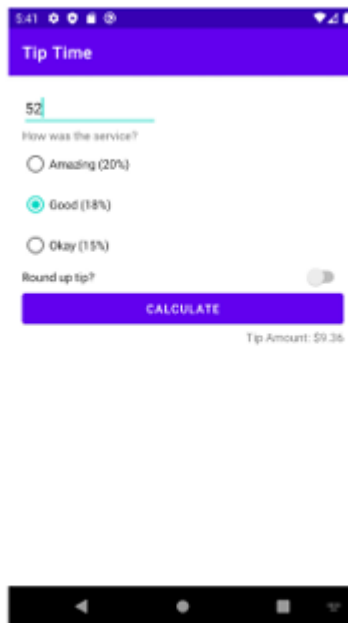
Buatlah sebuah aplikasi kalkulator tip yang dirancang untuk membantu pengguna menghitung tip yang sesuai berdasarkan total biaya layanan yang mereka terima. Fitur-fitur yang diharapkan dalam aplikasi ini mencakup:

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.

Tampilan awal aplikasi setelah dijalankan akan menampilkan 2 buah dadu kosong seperti dapat dilihat pada Gambar 1.



Gambar 1 Tampilan Awal Aplikasi



Gambar 2 Tampilan Aplikasi Setelah Dijalankan

A. Source Code

1. MainActivity.kt

1	package com.android.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.activity.enableEdgeToEdge
8	import androidx.compose.foundation.background
9	import androidx.compose.foundation.layout.Arrangement
10	import androidx.compose.foundation.layout.Column
11	import androidx.compose.foundation.layout.Row
12	import androidx.compose.foundation.layout.fillMaxSize
13	import androidx.compose.foundation.layout.fillMaxWidth
14	import androidx.compose.foundation.layout.padding
15	import androidx.compose.foundation.layout.statusBarsPadding
16	import androidx.compose.foundation.rememberScrollState
17	import androidx.compose.foundation.shape.RoundedCornerShape
18	import androidx.compose.foundation.text.KeyboardOptions
19	import androidx.compose.foundation.verticalScroll
20	import androidx.compose.material3.*
21	import androidx.compose.runtime.*
22	import androidx.compose.runtime.mutableStateOf
23	import androidx.compose.runtime.saveable.rememberSaveable
24	import androidx.compose.ui.Alignment

```

25 import androidx.compose.ui.Modifier
26 import androidx.compose.ui.graphics.Color
27 import androidx.compose.ui.platform.LocalContext
28 import androidx.compose.ui.res.colorResource
29 import androidx.compose.ui.text.input.KeyboardType
30 import androidx.compose.ui.unit.dp
31 import com.android.modul2.ui.theme.Modul2Theme
32
33 class MainActivity : ComponentActivity() {
34     override fun onCreate(savedInstanceState: Bundle?) {
35         super.onCreate(savedInstanceState)
36         enableEdgeToEdge()
37         setContent {
38             Modul2Theme {
39                 Display()
40             }
41         }
42     }
43 }
44
45 @Composable
46 fun Display() {
47     Scaffold(topBar = { TopBar() }) { innerPadding ->
48         TipCalc(modifier = Modifier.padding(innerPadding))
49     }
50 }
51
52 @OptIn(ExperimentalMaterial3Api::class) // agar bisa
menggunakan topAppBar yang bersifat experimental
53 @Composable
54 fun TopBar( ) {
55     TopAppBar(
56         title = { Text("Tip Time") },
57         colors = TopAppBarDefaults.topAppBarColors(
58             containerColor = colorResource(id =
R.color.hijau_dark),
59             titleContentColor = colorResource(id =
R.color.white)
60         ),
61         modifier = Modifier
62             .statusBarsPadding() // menghilangkan padding
otomatis untuk menghindari tabrakan dengan status bar
63     )
64 }
65
66 @OptIn(ExperimentalMaterial3Api::class) // agar bisa
menggunakan topAppBar yang bersifat experimental
67 @Composable
68 fun TipCalc(modifier: Modifier = Modifier) {
69     var inputNilai by rememberSaveable { mutableStateOf("") }
70     var optionSelect by rememberSaveable {
mutableStateOf("Amazing") }

```



```

71     var switchPosition by rememberSaveable {
72         mutableStateOf(true) }
73     var hasil by rememberSaveable { mutableStateOf(0.0) }
74     val context = LocalContext.current
75     Column(
76         modifier = modifier
77             .background(colorResource(id = R.color.abu))
78             .fillMaxSize()
79             .verticalScroll(rememberScrollState())
80             .padding(horizontal = 10.dp)
81     ) {
82         TextField(
83             value = inputNilai,
84             onChange = {inputNilai = it},
85             label = { Text("Cost of Service") },
86             modifier = Modifier
87                 .fillMaxWidth(),
88             colors =
89                 TextFieldDefaults.textFieldColors(containerColor =
90                     Color.Transparent),
91             keyboardOptions = KeyboardOptions(keyboardType =
92                 KeyboardType.Number)
93         )
94         Text("How was the service?")
95
96         Row(
97             verticalAlignment = Alignment.CenterVertically
98         ){
99             RadioButton(
100                 selected = optionSelect == "Amazing",
101                 onClick = { optionSelect = "Amazing"},
102             )
103             Text("Amazing (20%)")
104         }
105         Row(
106             verticalAlignment = Alignment.CenterVertically
107         ){
108             RadioButton(
109                 selected = optionSelect == "Good",
110                 onClick = { optionSelect = "Good"}
111             )
112             Text("Good (18%)")
113         }
114         Row(
115             verticalAlignment = Alignment.CenterVertically
116         ){
117             RadioButton(
118                 selected = optionSelect == "Okay",
119                 onClick = { optionSelect = "Okay"}
120             )
121             Text("Okay (16%)")

```

```

119     }
120
121     Row(
122         modifier = Modifier
123             .fillMaxWidth(),
124         verticalAlignment = Alignment.CenterVertically,
125         horizontalArrangement = Arrangement.SpaceBetween
126     ) {
127         Text("Round up tip?")
128         Switch(
129             checked = switchPosition,
130             onCheckedChange = {isChecked -> switchPosition
= isChecked}
131         )
132     }
133
134     Button(
135         onClick = {
136             if(inputNilai.isEmpty() || inputNilai == "0")
137             {
138                 Toast.makeText(context, "Silahkan masukkan
angka dan bukan nol!", Toast.LENGTH_SHORT).show()
139                 return@Button
140             }
141             if(inputNilai < "0") {
142                 Toast.makeText(context, "jangan masukkan
angka negatif!", Toast.LENGTH_SHORT).show()
143                 return@Button
144             }
145             hasil = tipCalculator(inputNilai.toInt(),
optionSelect, switchPosition)
146             },
147             modifier = Modifier
148                 .fillMaxWidth(),
149             colors = ButtonDefaults.buttonColors(
150                 contentColor = colorResource(R.color.white),
151                 containerColor =
colorResource(R.color.hijau_muda)
152             ),
153             shape = RoundedCornerShape(5.dp)
154         ) { Text("CALCULATE") }
155
156     Row(
157         modifier = Modifier.fillMaxWidth(),
158         horizontalArrangement = Arrangement.End
159     ) {
160         Text("Tip Amount: $")
161         hasil.let { Text("$it") }
162     }
163 }

```

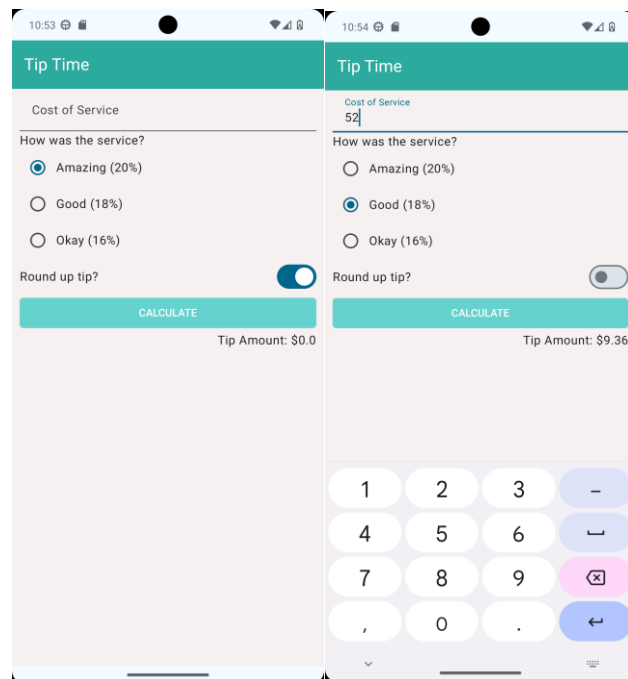
Tabel 1. Source Code Jawaban Soal 1

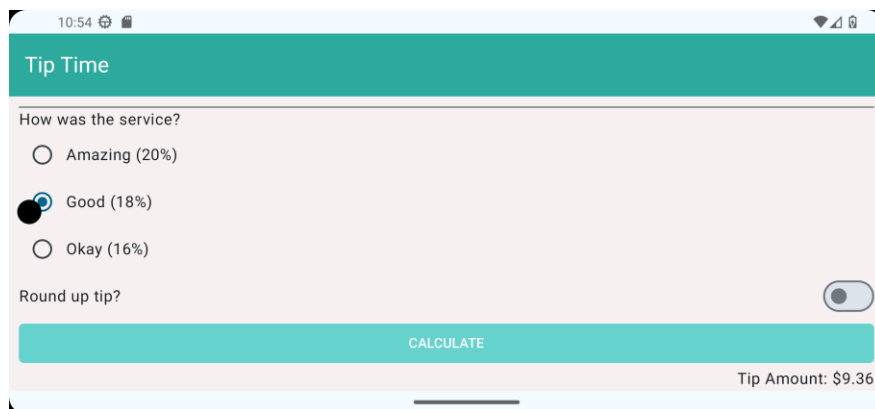
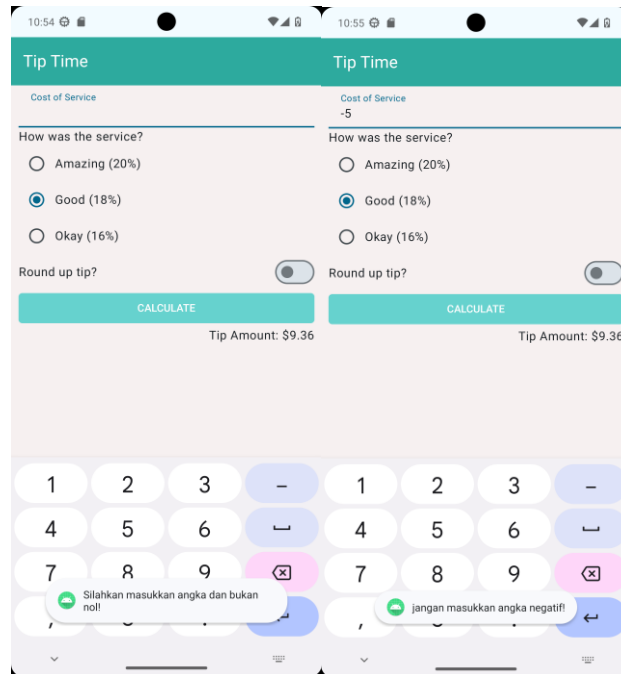
2. tipCalculator.kt

```
1 package com.android.modul2
2
3 import kotlin.math.ceil
4
5 fun tipCalculator( nilai: Int, opsi: String, switch: Boolean):
Double{
6     var hasil = 0.0;
7     when(opsi){
8         "Amazing" -> hasil = nilai * 20.0/100.0
9         "Good" -> hasil = nilai * 18.0/100.0
10        "Okay" -> hasil = nilai * 16.0/100.0
11    }
12    if (switch) {hasil = ceil(hasil) }
13    return hasil
14 }
```

Tabel 2. Source Code Jawaban Soal 1

B. Output Program





Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt:

- Pada baris 1, **package com.android.modul2** pendeklarasian nama package file Kotlin.
- Pada baris 3-31, **import** berfungsi untuk mengimpor package atau kelas untuk mengakses komponen dan fungsi dari android SDK dan jetpack compose.
- Pada baris 33, **class MainActivity : ComponentActivity()**, merupakan titik mula yang menjadi kelas utama dan akan dijalankan pertama kali saat aplikasi dibuka

- Pada baris 34, **override fun onCreate(savedInstanceState: Bundle?)**, berfungsi untuk menimpa (override) fungsi onCreate dari ComponentActivity.
- Pada baris 35, **super.onCreate(savedInstanceState)**, berfungsi untuk memanggil superclass dari fungsi onCreate untuk memastikan bahwa proses inisialisasi standar dari Android dijalankan sebelum logika saya dijalankan.
- Pada baris 36, **enableEdgeToEdge()**, berfungsi agar tampilan aplikasi dapat menggunakan seluruh layar dari status bar sampai navigation bar atau fullscreen layout.
- Pada baris 37, **setContent()**, digunakan untuk menampilkan UI berbasis jetpack compose ke dalam activity
- Pada baris 38, **Modul2Theme**, merupakan fungsi yang berisi tema custom yang membungkus seluruh UI untuk memberikan style yang konsisten.
- Pada baris 39, **Display()**, merupakan fungsi yang dibuat untuk menampilkan aplikasi pengocok dadu ke dalam activity.
- Pada baris 45-67, **@composable**, merupakan annotation untuk menandai bahwa fungsi yang dibuat dibawahnya merupakan fungsi UI di jetpack compose.
- Pada baris 46, **fun Display()**, merupakan fungsi yang dibuat untuk menampung semua component UI yang ingin ditampilkan ke dalam activity, seperti TopBar() dan TipCalc().
- Pada baris 47, **scaffold()**, merupakan komponen layout yang menjadi kerangka dasar dari tampilan yang mempermudah dalam mengatur struktur UI karena sudah disediakan slot-slot bawaannya seperti topBar yang sudah saya isi dengan TopBar() dan content saya isi dengan RollDice().
- Pada baris 47, **TopBar()**, berfungsi untuk memanggil fungsi TopBar() di dalam scaffold dan dimasukkan ke dalam slot topBar untuk ditampilkan dalam activity.
- Pada baris 48, **TipCalc()**, berfungsi untuk memanggil fungsi TipCalc() dengan parameter Modifier.padding(innerPadding) untuk memberikan padding bawaan dari scaffold.
- Pada baris 52 dan 66, **@OptIn(ExperimentalMaterial3Api::class)**, berfungsi untuk memberitahukan compiler bahwa fungsi composable dibawahnya masih menggunakan API yang bersifat eksperimental untuk menghindari error saat kompilasi.
- Pada baris 54, **fun TopBar()**, merupakan fungsi composable yang dibuat untuk menampung komponen UI TopAppBar yang akan ditampilkan ke activity tepatnya dibagian atas layar.

- Pada baris 55, **AppBar()**, komponen yang berfungsi untuk menampilkan elemen UI di bagian atas layar.
 - Pada baris 56, **title**, merupakan parameter dalam AppBar() untuk menampilkan judul .
 - Pada baris 57, **colors** merupakan parameter dalam AppBar() untuk mengatur warna background (containerColor) dan warna teks judul (titleContentColor).
 - Pada baris 61, **modifier** merupakan parameter dalam AppBar() untuk menerapkan modifikasi terhadap tampilan komponen, salah satunya memberikan .statusBarPadding() untuk menambahkan padding secara otomatis di atas layar guna menghindari tabrakan elemen dengan status bar.
- Pada baris 68, **fun TipCalc(modifier: Modifier = Modifier)**, merupakan fungsi composable yang dibuat untuk menampung beberapa logika sederhana mengenai aplikasi perhitungan tip, menerima parameter modifier, dan komponen UI Column, Row, Image, Button, Text yang akan ditampilkan ke activity.
- Pada baris 69-73, **var** merupakan keyword untuk membuat suatu variable bersifat mutable yang menampung nilai dari berbagai tipe data.
- Pada baris 69-72, **rememberSaveable** berfungsi untuk menyimpan dan mengingat nilai dari variable, dalam kasus ini inputNilai, optionSelect, switchPosition, hasil, dan context agar selama komponen composable masih aktif dan jika terjadi perubahan konfigurasi layar misalnya saat dirotasi.
- Pada baris 73, **LocalContext.current** berfungsi untuk mendapatkan context dari activity saat ini dalam fungsi composable dan juga digunakan untuk menampilkan Toast sebagai notifikasi dari peringatan jika user salah input.
- Pada baris 73, **val** merupakan keyword untuk membuat suatu variable bersifat immutable yang menampung nilai dari berbagai tipe data
- Pada baris 75, **Column()** merupakan komponen layout mirip seperti scaffold, hanya saja Column digunakan untuk menyusun elemen UI secara vertical.
 - Pada baris 76, **modifier** merupakan parameter dalam Column() untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.background* untuk memberikan warna background dan *.fillMaxSize* agar column mengisi seluruh ukuran layar.
- Pada baris 82, **TextField** merupakan komponen untuk menerima input dari pengguna

- Pada baris 83, **value** merupakan nilai yang akan ditampilkan di dalam TextField
- Pada baris 84, **onValueChange** berfungsi untuk memperbarui nilai saat pengguna mengetik sesuatu
- Pada baris 85, **label** berfungsi untuk memberikan placeholder saat input blm diklik pengguna, dan akan berpindah keatas saat diklik pengguna
- Pada baris 86, **modifier** merupakan parameter dalam TextField untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.fillMaxWidth* agar column lebar TextField mengisi ukuran layar.
- Pada baris 88, **colors** merupakan parameter dalam TextField untuk mengatur warna background (containerColor).
- Pada baris 89, **keyboardOptions** berfungsi untuk membuat keyboard yang muncul saat TextField ditekan berupa numeric saja.
- Pada baris 91-159, **Text()** merupakan komponen UI yang berfungsi menampilkan teks.
- Pada baris 93-155, **Row** merupakan komponen layout yang berlawanan dari Column, Dimana jika Column menyusun elemen UI secara Vertical, maka Row menyusun elemen UI secara horizontal.
 - Pada baris 94-124, **verticalAlignment** berfungsi untuk menyusun elemen secara vertical.
 - Pada baris 94-124, **modifier** merupakan parameter dalam TextField untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.fillMaxWidth* agar column lebar Row mengisi ukuran layar.
 - Pada baris 94-124, **horizontalArrangement** berfungsi untuk memberi jarak maksimal antar elemen.
- Pada baris 96-114, **RadioButton** merupakan komponen UI yang memungkinkan pengguna memilih satu dari beberapa opsi yang tersedia
 - Pada baris 97-115, **selected** berfungsi untuk menentukan apakah radiobutton dipilih atau tidak
 - Pada baris 98-116, **onClick** merupakan fungsi yang dijalankan saat radio button di tekan
- Pada baris 128, **Switch** merupakan komponen berbentuk switch yang digunakan untuk menyalakan atau mematikan fitur
 - Pada baris 129, **checked** berfungsi untuk menentukan status switch saat ini apakah true atau false.

- Pada baris 130, **onCheckedChange** merupakan fungsi yang dipanggil saat user mengubah posisi switch.
- Pada baris 134, **Button()** merupakan komponen UI yang berfungsi untuk membuat tombol pada aplikasi.
 - Pada baris 135, **onClick** merupakan parameter dalam Button() untuk memberikan aksi yang akan dijalankan ketika tombol ditekan.
 - Pada baris 136 dan 140, **if** merupakan perkondisian Dimana jika kondisi if terpenuhi maka statementnya akan dijalankan.
 - Pada baris 146, **modifier** merupakan parameter dalam TextField untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.fillMaxWidth* agar column lebar Button mengisi ukuran layar.
 - Pada baris 148, **colors** merupakan parameter dalam Button() untuk mengatur warna background (containerColor) dan warna teks (titleContentColor).
 - Pada baris 152, **shape** merupakan parameter dalam Button() untuk mengatur sudut agar dapat membulat.
- Pada baris 160, **.let** berfungsi untuk mengeksekusi blok kode dengan konteks objek yang dipanggil

2. RandomDice.kt

- Pada baris 1, **package com.android.modul2** pendeklarasian nama package file Kotlin.
- Pada baris 3, **import** berfungsi untuk mengimpor package atau kelas untuk mengakses komponen dan fungsi dari android SDK dan jetpack compose
- Pada baris 5, **fun tipCalc(nilai: Int, opsi: String, switch: Boolean): Double** merupakan fungsi untuk menghitung tip yang akan dibayarkan berdasarkan opsi radiobutton yang dipilih dan dapat membulatkan hasilnya keatas
- Pada baris 6, **var** merupakan keyword untuk membuat suatu variable bersifat mutable yang menampung nilai dari berbagai tipe data.
- Pada baris 7, **when** merupakan perkondisian mirip seperti if else hanya saja syntaxnya lebih pendek dan mudah dipahami.
- Pada baris 12, **if** merupakan perkondisian Dimana jika kondisi if terpenuhi maka statementnya akan dijalankan.
- Pada baris 13, **return** merupakan pengembalian terhadap suatu nilai di sebuah fungsi . jika di tipCalculator() akan mengembalikan hasil kalkulasi tip.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/AppleCraft2005/kuliah-santuy/tree/main/semesterIV/Pemrograman-Mobile>