

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 2**



Android Layout With Compose

Oleh:

Muhammad Raka Azwar NIM. 2210817210012

**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 With Compose Android Basic with Kotlin ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Raymond Hariyono

NIM : 2310817210007

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Salsabila Syifa
NIM. 2010817320004

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 01 011

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR	4
DAFTAR TABEL.....	5
SOAL 1	6
A. Source Code	8
Source Code XML	8
Source Code Compose	Error! Bookmark not defined.
B. Output Program.....	20
C. Pembahasan.....	22
Tautan Git	26

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	11
---	----

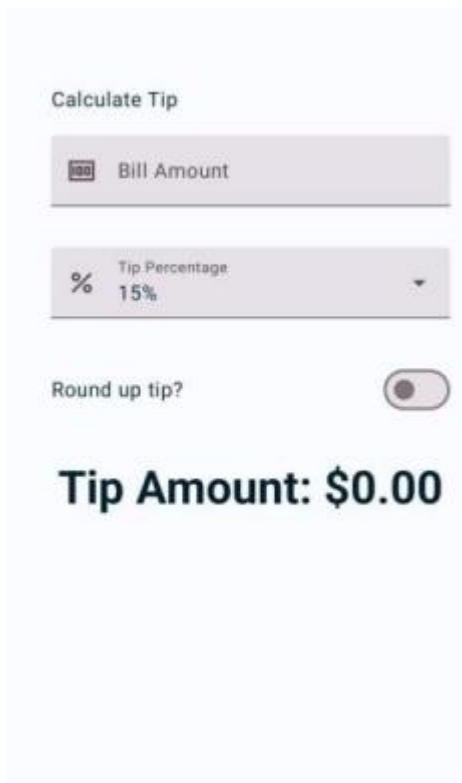
DAFTAR TABEL

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

SOAL 1


Soal Praktikum:

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.



The screenshot shows a mobile application interface titled "Calculate Tip". It features three main input sections: a "Bill Amount" field with a currency icon, a "Tip Percentage" dropdown menu currently set to "15%", and a "Round up tip?" toggle switch which is currently turned off. At the bottom, the "Tip Amount" is displayed as "\$0.00".

Calculate Tip

 Bill Amount
60

% Tip Percentage
15%

15%

18%

20%

Calculate Tip

 Bill Amount
60

% Tip Percentage
18%

Round up tip?



Tip Amount:
\$11.00

A. Source Code

Source Code XML

```
1 package com.example.tipcalculatorxml
2
3 import android.os.Bundle
4 import android.widget.AdapterView
5 import androidx.appcompat.app.AppCompatActivity
6 import
7 com.example.tipcalculatorxml.databinding.ActivityMainBinding
8 import kotlin.math.ceil
9
10 class MainActivity : AppCompatActivity() {
11
12     private lateinit var binding: ActivityMainBinding
13
14     private val tipOptions = arrayOf("15%", "18%", "20%")
15
16     override fun onCreate(savedInstanceState: Bundle?) {
17         super.onCreate(savedInstanceState)
18
19         binding =
20 ActivityMainBinding.inflate(layoutInflater)
21         setContentView(binding.root)
22
23         val adapter = ArrayAdapter(this,
24 android.R.layout.simple_spinner_item, tipOptions)
25
26 adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
27         binding.tipDropdown.setAdapter(adapter)
28
29
30         val updateTip = {
```



```
31         calculateTip()
32     }
33
34     binding.amountInput.setOnEditorActionListener { _,
35     _, _ ->
36         updateTip(); true
37     }
38
39     binding.tipDropdown.setOnItemClickListener { parent,
40     view, position, id ->
41         updateTip()
42     }
43
44     binding.roundUpSwitch.setOnCheckedChangeListener {
45     _, _ ->
46         updateTip()
47     }
48 }
49
50 private fun calculateTip() {
51     val input = binding.amountInput.text.toString()
52     val amount = input.toDoubleOrNull()
53
54     if (amount == null || amount <= 0) {
55         binding.tipResult.text = "Tip Amount: $0.00"
56         return
57     }
58
59     val selectedTip =
60     binding.tipDropdown.text.toString().replace("%",
61     "").toDouble()
62     var tip = amount * (selectedTip / 100)
63
64     if (binding.roundUpSwitch.isChecked) {
```

65	tip = ceil (tip)
66	}
67	
68	binding.tipResult.text = "Tip Amount:
69	\$\${"%.2f".format (tip) }"
70	}
	}

MainActivity.kt

Tabel 1. Source Code Jawaban Soal 1 XML

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <ScrollView
3      xmlns:android="http://schemas.android.com/apk/res/android"
4          xmlns:app="http://schemas.android.com/apk/res-auto"
5          android:padding="24dp"
6          android:layout_width="match_parent"
7          android:layout_height="match_parent">
8
9      <LinearLayout
10         android:orientation="vertical"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content">
13
14         <!-- Judul -->
15         <TextView
16             android:text="Calculate Tip"
17             android:textSize="24sp"
18             android:textStyle="bold"
19             android:layout_marginBottom="16dp"
20             android:layout_width="wrap_content"
21             android:layout_height="wrap_content" />
22
23         <!-- Input Amount -->
24
25         <com.google.android.material.textfield.TextInputLayout
26             android:layout_width="match_parent"
27             android:layout_height="wrap_content"
28             android:layout_marginBottom="16dp"
29             app:boxBackgroundMode="outline"
30             app:startIconDrawable="@drawable/ic_money"
31
32             app:startIconContentDescription="@string/bill_amount">
33
34
```

```

35
36 <com.google.android.material.textfield.TextInputEditText
37     android:id="@+id/amount_input"
38
39     android:background="@drawable/background_purple"
40     android:layout_width="match_parent"
41     android:layout_height="wrap_content"
42     android:hint="Bill Amount"
43     android:inputType="numberDecimal" />
44
45 </com.google.android.material.textfield.TextInputLayout>
46
47     <!-- Dropdown Tip Percentage -->
48
49 <com.google.android.material.textfield.TextInputLayout
50     android:id="@+id/dropdown_layout"
51     android:layout_width="match_parent"
52     android:layout_height="wrap_content"
53     android:layout_marginBottom="16dp"
54     android:background="@drawable/background_purple"
55     app:startIconDrawable="@drawable/ic_percent"
56
57     app:startIconContentDescription="@string/tip_percentage"
58     app:endIconMode="dropdown_menu"
59     app:boxBackgroundMode="outline">
60
61     <AutoCompleteTextView
62         android:id="@+id/tip_dropdown"
63         android:hint="Tip Percentage"
64         android:contentDescription="Select Tip
Percentage"
65
66         android:importantForAccessibility="yes"
67         android:inputType="none"
68         android:layout_width="match_parent"

```

```
        android:layout_height="wrap_content" />

</com.google.android.material.textfield.TextInputLayout>

<!-- Round Up Switch -->
<LinearLayout
    android:orientation="horizontal"
    android:layout_marginBottom="16dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical">

    <TextView
        android:text="Round Up Tip"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

<com.google.android.material.switchmaterial.SwitchMaterial
    android:id="@+id/round_up_switch"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_gravity="end" />
</LinearLayout>

<!-- Tip Result -->
<TextView
    android:id="@+id/tip_result"
    android:text="@string/tip_amount"
    android:textSize="20sp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

	<pre></LinearLayout> </ScrollView></pre>
--	--

Tabel 2. Source Code Jawaban Soal 1 XML

Source Code Compose

MainActivity.kt

```
1 package com.example.tipcalculatorcompose
2
3 import android.os.Bundle
4 import androidx.activity.ComponentActivity
5 import androidx.activity.compose.setContent
6 import androidx.compose.foundation.layout.*
7 import androidx.compose.material3.*
8 import androidx.compose.material3.ExposedDropdownMenuDefaults
9 import androidx.compose.runtime.*
10 import androidx.compose.ui.Modifier
11 import androidx.compose.ui.res.stringResource
12 import androidx.compose.ui.text.font.FontWeight
13 import androidx.compose.ui.tooling.preview.Preview
14 import androidx.compose.ui.unit.dp
15 import androidx.compose.ui.unit.sp
16
17 import
18 com.example.tipcalculatorcompose.ui.theme.TipCalculatorComposeTheme
19 import kotlin.math.ceil
20 import androidx.compose.ui.graphics.Color
21
22
23 class MainActivity : ComponentActivity() {
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContent {
27             TipCalculatorComposeTheme {
28                 TipTimeLayout()
29             }
30         }
31     }
32 }
```

```

33 }
34
35 @OptIn(ExperimentalMaterial3Api::class)
36 @Composable
37 fun TipTimeLayout(modifier: Modifier = Modifier) {
38     var tipAmount by remember { mutableStateOf("") }
39     var roundUp by remember { mutableStateOf(false) }
40
41     val amount = tipAmount.toDoubleOrNull() ?: 0.0
42     val tipPercent = listOf(15.0, 18.0, 20.0)
43
44     var expanded by remember { mutableStateOf(false) }
45     var selectedOption by remember {
46 mutableDoubleStateOf(tipPercent[0]) }
47
48     val tip = calculateTip(amount, selectedOption, roundUp)
49
50     Column(
51         modifier = modifier
52             .padding(16.dp)
53             .fillMaxWidth()
54     ) {
55         Text(
56             text = "Calculate Tip",
57             style = MaterialTheme.typography.titleLarge,
58             modifier = Modifier.padding(bottom = 16.dp)
59         )
60
61         // TextField untuk Bill Amount
62         TextField(
63             value = tipAmount,

```



```

67         onValueChange = { tipAmount = it },
68         label = { Text("Bill Amount") },
69         colors = ExposedDropDownMenuDefaults.textFieldColors(
70             focusedIndicatorColor = Color(0xFFCE93D8),
71             unfocusedIndicatorColor = Color(0xFFCE93D8)
72         ),
73
74         modifier = Modifier
75             .fillMaxWidth()
76             .padding(bottom = 16.dp)
77     )
78
79     // Dropdown untuk Tip Percentage
80     ExposedDropDownMenuBox(
81         expanded = expanded,
82         onExpandedChange = { expanded = !expanded },
83         modifier = Modifier
84             .fillMaxWidth()
85             .padding(bottom = 16.dp)
86     ) {
87         TextField(
88             readOnly = true,
89             value = "${selectedOption.toInt()}%",
90             onValueChange = {},
91             label = { Text(stringResource(id =
92 R.string.the_service)) },
93             trailingIcon = {
94 ExposedDropDownMenuDefaults.TrailingIcon(expanded = expanded)
95         },
96
97
98
99
100

```

```

101         colors =
102 ExposedDropDownMenuDefaults.textFieldColors(
103         focusedIndicatorColor = Color(0xFFCE93D8),
104         unfocusedIndicatorColor = Color(0xFFCE93D8)
105     ),
106     modifier = Modifier
107         .fillMaxWidth()
108         .menuAnchor()
109 )
110 ExposedDropDownMenu(
111     expanded = expanded,
112     onDismissRequest = { expanded = false }
113 ) {
114     tipPercent.forEach { selectionOption ->
115         DropdownMenuItem(
116             text =
117             {
118 Text("${selectionOption.toInt()}%") },
119             onClick = {
120                 selectedOption = selectionOption
121                 expanded = false
122             }
123         )
124     }
125 }
126 }
127 }
128
129 // Switch untuk Round Up
130 Row(
131     modifier = Modifier
132         .fillMaxWidth()
133         .padding(bottom = 16.dp),
134

```

```

135         horizontalArrangement = Arrangement.SpaceBetween
136     ) {
137         Text(text = "Round up tip?")
138         Switch(
139             checked = roundUp,
140             onCheckedChange = { roundUp = it }
141         )
142     }
143
144     // Menampilkan Tip Amount
145     Text(
146         text = "Tip Amount: ${"%2f".format(tip)}",
147         fontSize = 24.sp,
148         fontWeight = FontWeight.Bold
149     )
150 }
151
152 }
153
// Fungsi hitung tip
fun calculateTip(amount: Double, tipPercent: Double, roundUp:
Boolean): Double {
    var tip = tipPercent / 100 * amount
    if (roundUp) {
        tip = ceil(tip)
    }
    return tip
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {

```

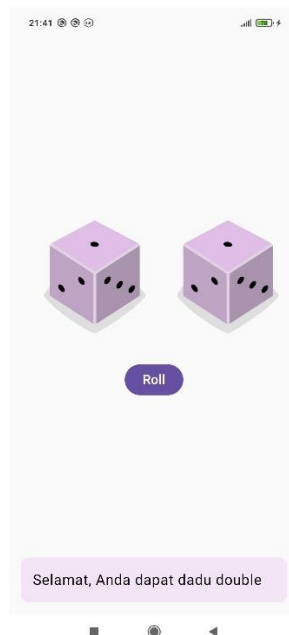
	<pre>TipCalculatorComposeTheme { TipTimeLayout() } }</pre>
--	--

Tabel 3. Source Code Jawaban Soal 1 Compose

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1 XML



Gambar 2. Screenshot Hasil Jawaban Soal 1 Compose

C. Pembahasan

Pembahasan XML

MainActivity.kt:

Line 1: mendeklarasikan file merupakan file XML dengan encoding UTF-8

Line 2 – 6 : membuka ConstraintLayout sebagai layout utama. Yang memiliki tiga namespace yang dideklarasikan android: untuk atribut dasar Android. app: untuk atribut khusus AndroidX dan ConstraintLayout. tools: untuk preview di Android Studio (tidak dibawa ke runtime).

Line 7: Memberikan id pada layout, yaitu @+id/constraintLayout.

Line 9: Tinggi layout juga mengisi seluruh tinggi layar (match_parent).

Line 10: Menyetel context preview ke kelas diceRollerXML,

Line 13-14: Membuat komponen Button dengan ID @+id/buttonRoll agar bisa dipanggil ke mainActivity.

Line 15 – 16: Ukuran tombol mengikuti isi teks (wrap_content).

Line 17: Teks tombol diambil dari file string resource @string/roll.

Line 18: Memberikan latar belakang kustom dari file drawable rounded_border_button.

Line 19-24: Menentukan constraint untuk mengatur posisi tombol berada ditengah dan memiliki jarak.

Line 27 membuat ImageView dengan ID @+id/DiceLeft

Line 28-29 mengatur ukuran gambar dadu dengan panjang dan lebar 200dp x 200 dp

Line 30-35 mengatur posisis ImageView ditengah agak keatas

Line 36: Gambar awal dadu diambil dari @drawable/dice_0.

Line 39: membuat ImageView dengan ID @+id/DiceRight.

Line 40 – 41: mengatur ukuran gambar dadu dengan panjang dan lebar 200dp x 200 dp

Line 42 – 47: Posisi dadu kanan diletakkan di tengah atas

Line 48: Gambar awal dadu diambil dari @drawable/dice_0.

Line 49: tools:ignore digunakan agar Android Studio mengabaikan peringatan kontras pada preview gambar.

Line 52: Membuat TextView dengan ID @+id/resultMessage untuk menampilkan pesan hasil roll.

Line 52: Lebar 0dp berarti akan mengikuti constraint start dan end (match constraint).

Line 54: Menentukan tinggi TextView 40dp

Line 55: teks di align ditengah dari background (gravity="center").

Line 56: Menentukan ukuran teks menjadi 15sp.

Line 57: Menggunakan warna teks hitam dari sistem android

Line 58 – 60: TextView ditempatkan di tengah bawah layar dengan constraint start dan end ke parent.

activity_main.xml:

Line 1: merupakan package file Kotlin yaitu com.example.modul1.

Line 3 – 6: Mengimpor class yang dibutuhkan: Bundle: digunakan saat activity dibuat. Button, ImageView, TextView: komponen UI. AppCompatActivity: kelas dasar dari activity yang digunakan.

Line 8: mendeklarasikan class diceRollerXML yang merupakan turunan dari AppCompatActivity

Line 9 – 11: Override fungsi onCreate, fungsi ini dipanggil saat activity pertama kali dibuat. Baris 10 memanggil fungsi super untuk menjalankan proses bawaan Android. Baris 11 mengatur layout yang digunakan oleh activity ini, yaitu activity_main.xml.

Line 13 – 16: Mendeklarasikan variabel dan menghubungkan dengan elemen UI di XML menggunakan findViewById(): buttonRoll: tombol untuk melempar dadu. diceImageLeft dan diceImageRight: gambar dadu kiri dan kanan. resultMessage: teks hasil lemparan dadu.

Line 19 – 20: Mengatur gambar awal kedua buah dadu default dengan dadu tanpa angka (dice_0).

Line 22: Membuat listerner saat tombol buttonRoll diklik menggunakan buttonRoll.setOnClickListener

Line 23 – 24: Menghasilkan aksi klik pada tombol memakai random()

Line 26 – 33: Pengkondisian when yang mana akan menampilkan angka dadu sebelah kiri sesuai angka acak dan mengambil UI dari drawable

Line 34-41: Pengkondisian when yang mana akan menampilkan angka dadu sebelah kiri sesuai angka acak dan mengambil UI dari drawable

Line 41 – 42 menampilkan gambar dadu kiri dan kanan berdasarkan angka acak yang mengambil imageresource dari drawable

Line 45 – 47: jika kedua angka sama maka akan menampilkan pesan dadu double dan mengatur backgrounds sesuai dengan di drawable yang sudah diatur untuk pesan menang

Line 48 – 51: jika kedua angka sama maka akan menampilkan pesan dadu double dan mengatur backgrounds sesuai dengan di drawable yang sudah diatur untuk pesan kalah

Line 52 – 54 block penutup setOnClickListener, onCreate, dan class.

Pembahasan Compose

MainActivity.kt

Line 1: menunjukkan bahwa file Kotlin berada pada package com.example.diceRollerCompose.

Line 4 – 31: merupakan import statement untuk fungsi dan class dari library android dan jetpack compose

Line 5: Bundle: digunakan untuk menyimpan dan memulihkan data dalam lifecycle Android.

Line 6: ComponentActivity: superclass untuk activity berbasis Compose.

Line 7: setContent: menetapkan konten UI menggunakan Jetpack Compose.

Line 8: enableEdgeToEdge: mengaktifkan tampilan edge-to-edge agar UI menyesuaikan dengan layar penuh (menyentuh notifikasi bar dan navigation bar).

Line 9 – 14: Layout dasar Compose seperti Column, Row, Spacer, Modifier, padding, fillMaxSize, dll.

Line 15: RoundedCornerShape: digunakan untuk membuat sudut membulat pada elemen UI.

Line 16: Button: tombol interaktif dari Material3.

Line 17: Text: untuk menampilkan teks pada layar.

Line 18 – 22: State management: remember, mutableStateOf, mutableIntStateOf

Line 23 – 24: painterResource dan stringResource: untuk memuat gambar dan string dari resource XML.

Line 25: Preview: memungkinkan melihat tampilan UI langsung di Android Studio.

Line 26: dp: satuan ukuran yang digunakan Compose.

Line 27: Mengimpor tema khusus dari folder ui.theme.

Line 32 – 42: Merupakan activity utama

Line 32: Meng-extend ComponentActivity, yang kompatibel dengan Jetpack Compose.

Line 33: Lifecycle method yang dipanggil saat aktivitas dimulai.

Line 34: Memanggil onCreate dari superclass.

Line 35: Mengatur tampilan agar menggunakan seluruh layar.

Line 36- 40: digunakan untuk mengisi UI dengan compose, diceRollerComposeTheme menerapkan styling (warna, font, ukuran), DiceRollerApp() adalah fungsi utama untuk memanggil tampilan aplikasi.

Line 41 – 42: menutup fungsi onCreate() dan MainActivity.

Line 44 – 46: fungsi yang bersifat composable, yang digunakan untuk menghasilkan UI

Line 49 – 114 : Fungsi tampilan utama

Line 49 – 50: Fungsi utama yang menampilkan UI lengkap, dengan 2 gambar dadu, tombol dan pesan.

Line 51 – 53 : resultLeft & resultRight: angka dadu kiri & kanan. messageResult: string pesan saat user menekan tombol. remember: menyimpan nilai selama komposisi ulang.

Line 55 – 63 menentukan gambar dadu kiri berdasarkan angka resultLeft

Line 64 – 72 : menentukan gambar dadu kiri berdasarkan angka resultRight

Line 74 – 113: merupakan kode untuk layout

Line 74 – 81: membuat column atau layout vertical untuk semua layout, mengisi ukuran penuh dengan fillMaxSize, memberi padding agar tidak memenuhi status bar, dan padding 16dp dari tepi

Line 82: Spacer untuk memberi jarak diatas dadu

Line 83- 97: membuat row untuk menampilkan gambar dadu secara horizontal dan berada ditengah

Line 87 – 96: menampilkan gambar dadu dari imageSource dan mengatur ukuran ke 200 dp

Line 98-111: Mengacak dua angka dari 1–6. Menampilkan pesan berdasarkan hasil: jika angka sama (double): menampilkan pesan selamat anda dapat dadu double. Jika beda: user mendapat pesan anda belum beruntung, coba lagi. `stringResource(R.string.roll)`: Mengambil teks tombol dari file `strings.xml`.

Line 113 – 126: Jika `messageResult` tidak kosong, tampilkan pesan ke layar. Jika kosong (belum klik tombol), beri jarak bawah agar layout tetap rapi.

Baris 130 – 136: Fungsi ini digunakan oleh Android Studio untuk melihat tampilan UI secara langsung di editor. Tidak digunakan saat aplikasi dijalankan di device.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/raymondhariyono/mobile-praktikum>