

**LAPORAN PRAKTIKUM**  
**PRAKTIKUM 7:**  
**“PENGENALAN ANDROID STUDIO\_ACTIVITY”**



**Disusun Oleh :**

Elisabeth Zefanya Putri  
240601211220032

**PRAKTIKUM PENGEMBANGAN BERBASIS PLATFORM**  
**LAB A2**

**DEPARTEMEN ILMU KOMPUTER / INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**  
**SEMARANG**  
**2023**

# BAB I

## PENDAHULUAN

### A. Rumusan Masalah

1. Tambahkan fitur untuk menghitung luas permukaan dan keliling balok pada project aplikasi yang sudah dikerjakan pada latihan sebelumnya. Tampilkan hasil perhitungan pada tv\_result.
2. Tuliskan laporan praktikum yang berisi penjelasan *source code* fitur menghitung luas permukaan dan keliling balok yang diimplementasikan pada project.
3. Export project dengan menggunakan menu File > Export > Export to ZIP File, kemudian kumpulkan pada link pengumpulan yang sudah ditentukan.

The screenshot shows a mobile application interface with a light gray background. At the top, there are three input fields labeled 'Lebar' (Width), 'Panjang' (Length), and 'Tinggi' (Height). Below these fields are three blue buttons with white text: 'HITUNG VOLUME', 'HITUNG LUAS PERMUKAAN', and 'HITUNG KELILING'. At the bottom, there is a label 'Hasil' (Result) followed by a large empty space for the output.

### B. Tujuan

Tujuan dari praktikum ini antara lain:

1. Dapat memahami cara menggunakan android studio
2. Dapat mengimplementasikan fitur-fitur yang ada pada android studio

## BAB II

### PEMBAHASAN

#### A. Penjelasan Code

Pembuatan fitur untuk hitung luas permukaan dan hitung keliling balok diimplementasikan pada dua file saja yaitu `activity_volume.xml` yang mengatur *frontend* dan `ActivityVolume.kt` yang mengatur backend dari fitur-fitur tersebut. Berikut ini merupakan penjelasan lengkap mengenai *source code* dan bagian-bagian di dalamnya:

##### Pada file `activity_volume.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:orientation="vertical"
        android:padding="24dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/lebar"
            android:labelFor="@+id/et_width"
            />

        <EditText
            android:id="@+id/et_width"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:autofillHints="Masukkan lebar"
            android:inputType="numberDecimal"
            android:minHeight="48dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/panjang"
            android:labelFor="@+id/et_length"/>

        <EditText
            android:id="@+id/et_length"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:autofillHints="Masukkan panjang"
            android:inputType="numberDecimal"
            android:minHeight="48dp" />

        <TextView
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tinggi"
        android:labelFor="@+id/et_height"/>

<EditText
    android:id="@+id/et_height"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:autofillHints="Masukkan Tinggi"
    android:inputType="numberDecimal"
    android:minHeight="48dp" />

<Button
    android:id="@+id/btn_calculate"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/hitung_volume" />

<Button
    android:id="@+id/btn_calculate2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/hitung_luas_permukaan" />

<Button
    android:id="@+id/btn_calculate3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/hitung_keliling" />

<TextView
    android:id="@+id/tv_result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="18dp"
    android:text="@string/hasil"
    android:textAlignment="center"
    android:textSize="18sp"
    android:textStyle="bold" />
</LinearLayout>
</ScrollView>

```

### Penjelasan:

XML ini adalah tampilan layout untuk suatu aktivitas di Android yang memungkinkan pengguna untuk menghitung *volume*, luas permukaan, dan keliling balok berdasarkan input yang mereka masukkan. Berikut ini merupakan penjelasan detail mengenai komponen-komponen utamanya:

1. `<ScrollView>`: Ini adalah tampilan yang memungkinkan kontennya dapat digulir jika terlalu besar untuk layar. Ini berguna jika konten dalam layout melebihi ukuran layar sehingga pengguna dapat menggulir untuk melihat seluruh kontennya.

2. `<LinearLayout>`: Ini adalah tampilan kontainer yang digunakan untuk mengelompokkan elemen-elemen UI lainnya secara vertikal. `android:orientation="vertical"` digunakan untuk menata elemen-elemen dalam susunan vertikal.
3. `<TextView>`: Ini adalah elemen teks yang digunakan untuk menampilkan label seperti "Lebar," "Panjang," "Tinggi," dan "Hasil."
4. `<EditText>`: Ini adalah elemen input teks yang digunakan untuk memasukkan nilai-nilai seperti lebar, panjang, dan tinggi balok. `android:id` digunakan untuk mengidentifikasi setiap `EditText`, dan `android:labelFor` mengaitkan label (`TextView`) dengan `EditText` untuk aksesibilitas.
5. `<Button>`: Ini adalah elemen tombol yang digunakan untuk menghitung volume, luas permukaan, dan keliling balok. Setiap tombol memiliki `android:id` yang membedakannya dari yang lain dan teks yang sesuai seperti "Hitung *Volume*," "Hitung Luas Permukaan," dan "Hitung Keliling.". Penjelasan mendetail mengenai ketiga tombol tersebut adalah sebagai berikut:
  - a. Tombol "*Volume*":
    - i. `<Button android:id="@+id/btn_calculate">`: Ini adalah tombol yang memiliki ID `btn_calculate`, yang akan digunakan untuk mengidentifikasi tombol ini dalam kode Java/Kotlin.
    - ii. `android:layout_width="match_parent"`: Ini mengatur lebar tombol agar mengisi lebar dari parent kontainer (*LinearLayout*).
    - iii. `android:layout_height="wrap_content"`: Ini mengatur tinggi tombol sesuai dengan kontennya.
    - iv. `android:text="@string/hitung_volume"`: Ini mengatur teks pada tombol. Teks ini diambil dari sumber string yang didefinisikan dalam sumber daya string (`strings.xml`).
  - b. Tombol "Luas Permukaan":
    - i. `<Button android:id="@+id/btn_calculate2">`: Ini adalah tombol yang memiliki ID `btn_calculate2`, yang akan digunakan untuk mengidentifikasi tombol ini dalam kode Java/Kotlin.

- ii. `android:layout_width="match_parent"`: Ini mengatur lebar tombol agar mengisi lebar dari parent container (*LinearLayout*).
- iii. `android:layout_height="wrap_content"`: Ini mengatur tinggi tombol sesuai dengan kontennya.
- iv. `android:text="@string/hitung_luas_permukaan"`: Ini mengatur teks pada tombol. Teks ini diambil dari sumber string yang didefinisikan dalam sumber daya string (`strings.xml`).

c. Tombol "Keliling":

- i. `<Button android:id="@+id/btn_calculate3">`: Ini adalah tombol yang memiliki ID `btn_calculate3`, yang akan digunakan untuk mengidentifikasi tombol ini dalam kode Java/Kotlin.
- ii. `android:layout_width="match_parent"`: Ini mengatur lebar tombol agar mengisi lebar dari parent container (*LinearLayout*).
- iii. `android:layout_height="wrap_content"`: Ini mengatur tinggi tombol sesuai dengan kontennya.
- iv. `android:text="@string/hitung_keliling"`: Ini mengatur teks pada tombol. Teks ini diambil dari sumber string yang didefinisikan dalam sumber daya string (`strings.xml`).

Penggunaan tombol-tombol ini akan diatur melalui kode Java/Kotlin yang terkait dengan aktivitas yang menggunakan layout ini. Ketika pengguna mengeklik salah satu tombol, perhitungan yang sesuai akan dijalankan berdasarkan nilai-nilai yang dimasukkan pengguna (lebar, panjang, tinggi), dan hasil perhitungan akan ditampilkan pada elemen `TextView` yang memiliki ID `tv_result`.

- 6. `<TextView>` dengan `android:id="@+id/tv_result"`: Ini adalah elemen teks yang digunakan untuk menampilkan hasil perhitungan, seperti volume, luas permukaan, atau keliling balok. `android:text` diinisialisasi dengan teks "Hasil," dan teks ini akan diperbarui dengan hasil perhitungan yang sesuai melalui kode Java/Kotlin.

Selain itu, beberapa atribut seperti ``android:layout_width``, ``android:layout_height``, ``android:layout_marginBottom``, ``android:autofillHints``, ``android:inputType``, ``android:minHeight``, ``android:textAlignment``, ``android:textSize``, dan ``android:textStyle`` digunakan untuk mengatur tampilan dan perilaku elemen-elemen tersebut. Dalam konteks aplikasi Android, layout XML ini akan digunakan oleh aktivitas untuk menampilkan UI kepada pengguna dan mengambil input yang mereka masukkan untuk melakukan perhitungan.

### Pada file VolumeActivity.kt

```
package com.example.myapplication

import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import com.example.myapplication.ui.theme.MyApplicationTheme

class VolumeActivity : ComponentActivity(), View.OnClickListener {
    private lateinit var tvResult: TextView
    private lateinit var etWidth: EditText
    private lateinit var etHeight: EditText
    private lateinit var etLength: EditText
    private lateinit var btnCalculate: Button
    private lateinit var btnCalculate2: Button
    private lateinit var btnCalculate3: Button

    // State
    private val KEY_RESULT = "key_result"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            MyApplicationTheme {
                setContentView(R.layout.activity_volume)
                tvResult = findViewById(R.id.tv_result)
                etWidth = findViewById(R.id.et_width)
                etLength = findViewById(R.id.et_length)
                etHeight = findViewById(R.id.et_height)
                btnCalculate = findViewById(R.id.btn_calculate)
                btnCalculate2 = findViewById(R.id.btn_calculate2)
                btnCalculate3 = findViewById(R.id.btn_calculate3)

                btnCalculate.setOnClickListener(this)
                btnCalculate2.setOnClickListener(this)
                btnCalculate3.setOnClickListener(this)
            }
        }

        if (savedInstanceState != null) {
```

```

        val result = savedInstanceState.getString(KEY_RESULT)
        tvResult.text = result
    }
}

override fun onClick(view: View?) {
    if (view != null) {
        val inputLength: String =
etLength.text.toString().trim()
        val inputWidth: String =
etWidth.text.toString().trim()
        val inputHeight: String =
etHeight.text.toString().trim()

        try {
            val length = inputLength.toDouble()
            val width = inputWidth.toDouble()
            val height = inputHeight.toDouble()

            when (view.id) {
                R.id.btn_calculate -> {
                    // Hitung volume balok
                    val volume: Double = length * height *
width
                    // Tampilkan hasil perhitungan di TextView
-> tvResult

                    val resultText = "Volume: %s"
                    tvResult.text = String.format(resultText,
volume.toString())
                }
                R.id.btn_calculate2 -> {
                    // Hitung luas permukaan balok
                    val luas: Double = 2 * (length * width +
length * height + height * width)
                    // Tampilkan hasil perhitungan di TextView
-> tvResult

                    val resultText = "Luas Permukaan: %s"
                    tvResult.text = String.format(resultText,
luas.toString())
                }
                R.id.btn_calculate3 -> {
                    // Hitung keliling balok
                    val keliling: Double = 4 * (length +
height + width)
                    // Tampilkan hasil perhitungan di TextView
-> tvResult

                    val resultText = "Keliling: %s"
                    tvResult.text = String.format(resultText,
keliling.toString())
                }
            }
        } catch (e: NumberFormatException) {
            // Tangani kesalahan jika input tidak valid
            tvResult.text = "Input tidak valid"
        }
    }
}

override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
}

```



```

        val calculationResult = tvResult.text.toString()
        outState.putString(KEY_RESULT, calculationResult)
    }
}

```

## Penjelasan:

Kode di atas adalah kode Kotlin untuk sebuah aktivitas Android yang menghitung volume, luas permukaan, dan keliling dari sebuah balok berdasarkan input yang dimasukkan oleh pengguna. Berikut ini merupakan penjelasan detail mengenai kode di atas:

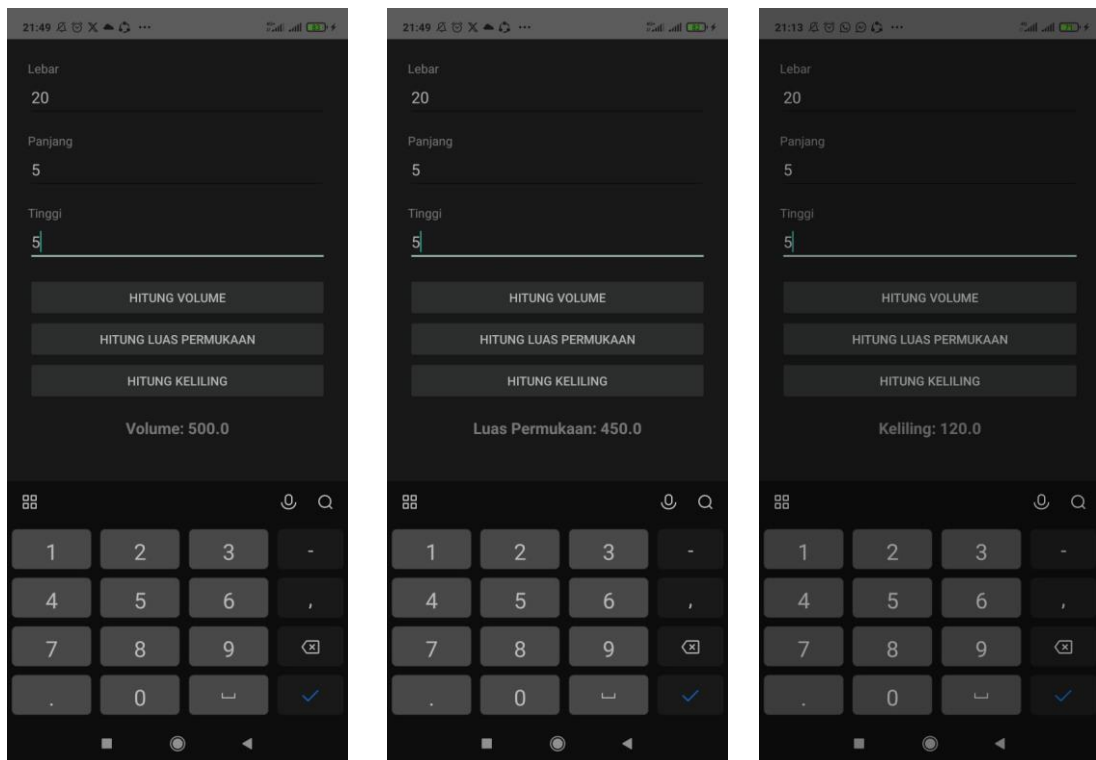
1. ``class`                      VolumeActivity:                      ComponentActivity(),  
View.OnClickListener``: Ini adalah deklarasi kelas ``VolumeActivity`` yang merupakan subkelas dari ``ComponentActivity`` dan mengimplementasikan ``View.OnClickListener``. Artinya, kelas ini adalah aktivitas Android dan juga meng-handle klik pada elemen View.
2. Deklarasi Variabel:
  - a. ``private lateinit var tvResult: TextView``: Ini adalah deklarasi variabel untuk elemen `TextView` yang akan digunakan untuk menampilkan hasil perhitungan.
  - b. ``private lateinit var etWidth: EditText`, `private  
lateinit var etHeight: EditText`, `private lateinit var  
etLength: EditText``: Ini adalah deklarasi variabel untuk elemen `EditText` yang digunakan untuk memasukkan lebar, tinggi, dan panjang balok.
  - c. ``private lateinit var btnCalculate: Button`, `private  
lateinit var btnCalculate2: Button`, `private lateinit var  
btnCalculate3: Button``: Ini adalah deklarasi variabel untuk tiga elemen tombol yang digunakan untuk menghitung volume, luas permukaan, dan keliling balok.
3. ``private val KEY_RESULT = "key_result"``: Ini adalah deklarasi variabel dengan nilai konstan `"key_result"` yang akan digunakan sebagai kunci penyimpanan dalam `Bundle` saat menyimpan hasil perhitungan.

4. `onCreate(savedInstanceState: Bundle?)`: Ini adalah metode yang dipanggil saat aktivitas ini dibuat. Di dalamnya dapat melakukan hal-hal berikut:
  - a. `super.onCreate(savedInstanceState)`: Memanggil metode `onCreate` dari kelas induk.
  - b. `setContent { MyApplicationTheme { ... } }`: Menetapkan tampilan menggunakan `setContent`. `MyApplicationTheme` adalah tema yang digunakan untuk tampilan.
  - c. Menginisialisasi elemen-elemen UI seperti `tvResult`, `etWidth`, `etLength`, `etHeight`, `btnCalculate`, `btnCalculate2`, dan `btnCalculate3` dengan menghubungkannya dengan elemen-elemen dalam tampilan layout XML menggunakan `findViewById`.
  - d. Mengatur `OnClickListener` untuk tombol-tombol sehingga aktivitas ini akan merespons klik pada tombol-tombol tersebut.
  - e. Memeriksa apakah ada data yang disimpan dalam `savedInstanceState` (`Bundle` yang digunakan untuk menyimpan data saat konfigurasi berubah). Jika ada, maka hasil perhitungan sebelumnya akan dipulihkan dan ditampilkan kembali di `tvResult`.
5. `onClick(view: View?)`: Ini adalah metode yang meng-handle klik pada elemen `View`, yaitu tombol. Dalam metode ini dapat melakukan hal-hal berikut:
  - a. Memeriksa elemen `View` yang diklik menggunakan `view.id`.
  - b. Mengambil nilai lebar, panjang, dan tinggi dari elemen `EditText`.
  - c. Mencoba mengonversi input pengguna menjadi nilai numerik (`double`).
  - d. Berdasarkan tombol yang diklik, akan dihitung volume, luas permukaan, atau keliling balok. Volume balok dihitung dengan rumus:  $\text{Volume} = \text{Panjang} \times \text{Lebar} \times \text{Tinggi}$ . Luas permukaan balok dihitung dengan rumus:  $\text{Luas Permukaan} = 2 \times (\text{Panjang} \times \text{Lebar} + \text{Panjang} \times \text{Tinggi} + \text{Lebar} \times \text{Tinggi})$ . Keliling balok dihitung dengan rumus:  $\text{Keliling} = 4 \times (\text{Panjang} + \text{Lebar} + \text{Tinggi})$ .
  - e. Mengatur teks pada `tvResult` sesuai dengan hasil perhitungan. Jika input tidak valid (misalnya, tidak dapat diubah menjadi nilai numerik), pesan kesalahan akan ditampilkan di `tvResult`.
6. `onSaveInstanceState(outState: Bundle)`: Ini adalah metode yang dipanggil saat aktivitas ini perlu menyimpan data saat berubah (misalnya, saat

perangkat diputar). Di dalamnya, hasil perhitungan yang ada di ``tvResult`` disimpan dalam Bundle dengan kunci `"key_result"` untuk memungkinkan pemulihan hasil perhitungan saat aktivitas dibuat ulang.

Kode ini menggambarkan aktivitas sederhana yang memungkinkan pengguna untuk menghitung berbagai fitur dari sebuah balok berdasarkan input yang mereka masukkan. Hasil perhitungan ditampilkan di elemen `TextView` (``tvResult``).

## B. Hasil Source Code



**Volume**

**Luas Permukaan**

**Keliling**

Ketiga gambar di atas adalah hasil *run code* yang dijalankan pada HP Redmi Note 9 Pro. Ketiga tombol yaitu hitung *volume*, hitung luas permukaan, dan hitung keliling dapat berjalan dengan baik terbukti dari ketiganya dapat mengeluarkan hasil yang valid untuk perhitungan volume, luas permukaan, dan keliling balok dengan inputan panjang, lebar, dan tinggi yang dimasukkan oleh pengguna.

## **BAB III**

### **PENUTUP**

#### **A. Kesimpulan**

Kode VolumeActivity adalah aplikasi sederhana yang memungkinkan pengguna untuk menghitung volume, luas permukaan, dan keliling balok berdasarkan input yang dimasukkan. Dengan mengandalkan konsep dasar pemrograman Android dan matematika, kode ini memungkinkan pengguna untuk menjalankan perhitungan dengan mudah. Dengan sedikit peningkatan dalam validasi input dan umpan balik kepada pengguna, kode ini dapat menjadi aplikasi yang lebih responsif dan informatif.