

LAPORAN PRAKTIKUM
PRAKTIUM 7:
PENGENALAN ANDROID STUDIO



Disusun oleh:
Muhamad Ridwan Ash'shidqi
24060121130075

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

KATA PENGANTAR

Segala puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya yang diberikan, sehingga laporan praktikum “Pengenalan Android Studio” ini dapat terselesaikan dengan baik. Adapun laporan ini kami susun sebagai bagian dari penugasan praktikum Pengembangan Berbasis Platform.

Adapun maksud dan tujuan adanya laporan praktikum ini adalah dapat membuat aplikasi sederhana pada Android dengan menggunakan Android Studio.

Dengan ini, saya menyadari bahwa laporan praktikum ini masih jauh dari kata sempurna. Untuk itu, saya dengan sangat terbuka menerima kritik dan saran dari para pembaca. Semoga laporan praktikum ini bermanfaat untuk para pembaca dan saya khususnya dalam usaha meningkatkan pemahaman mengenai aplikasi Android Studio dalam proses pembuatan aplikasi berbasis Android.

Semarang, 06 November 2023

Muhamad Ridwan Ash'shidqi

BAB I

PENDAHULUAN

1.1. Tujuan

- Membuat aplikasi android untuk menghitung volume dari input lebar, panjang, dan tinggi.
- Menambahkan fitur untuk menghitung luas permukaan dan keliling balok pada projek aplikasi tersebut.

1.2. Rumusan Masalah

- 1.2.1. Pembuatan projek aplikasi sesuai tahapan yang diberikan pada praktikum**
- 1.2.2. Menambahkan fitur untuk menghitung luas permukaan dan keliling balok dan menampilkannya pada bagian hasil**

BAB II

LANDASAN TEORI

2.1. Android Studio

Android Studio adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android. Berbasis editor kode dan alat developer yang andal dari IntelliJ IDEA.

Android Studio menawarkan lebih banyak fitur yang mampu meningkatkan produktivitas Anda saat mem-build aplikasi Android, seperti:

- a. Emulator yang cepat dan kaya fitur.
- b. Lingkungan terpadu.
- c. Edit live untuk mengupdate *composable* di emulator dan perangkat fisik secara *real time*.
- d. *Template* kode dan integrasi GitHub..
- e. Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan lainnya.
- f. Dukungan C++ dan NDK.
- g. Dukungan bawaan untuk Google Cloud Platform yang memudahkan integrasi Google Cloud Messaging dan App Engine.

2.2. Kotlin

Bahasa Kotlin adalah salah satu bahasa yang digunakan dalam pembuatan aplikasi Android. Kotlin diluncurkan sejak tahun 2017 lalu dan hingga saat ini masih populer di kalangan bahasa pemrograman Android. Setiap bahasa tentu memiliki karakteristik sendiri-sendiri.

Kotlin adalah bahasa pemrograman populer yang dikembangkan oleh JetBrains. Bahasa ini sifatnya open source dan dapat digunakan untuk berbagai platform baik pembuatan aplikasi web, server, maupun Android. Tetapi, bahasa ini lebih populer untuk membuat aplikasi Android.

BAB III

PEMBAHASAN

3.1. activity_volume.xml

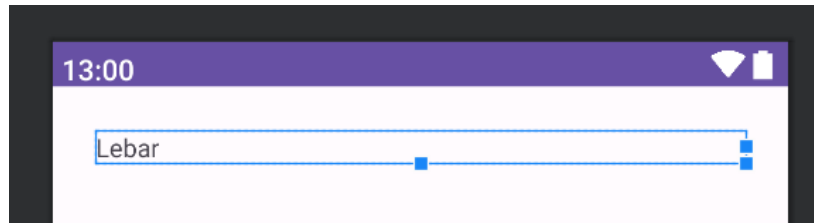
```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:orientation="vertical"
        android:padding="24dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

Penjelasan:

Pertama-tama, akan dibuat file .xml untuk mengatur tampilan aplikasi yang akan dibuat. File .xml ini dibuat dalam folder layout yang ada dalam folder res, dengan nama activity_volume.xml. Pada baris-baris awal file .xml ini, akan dideklarasikan tag yang menyatakan versi XML adalah 1.0 dan *encoding* karakternya adalah UTF-8.

ScrollView merupakan komponen yang membuat kontennya dapat digulir jika melebihi ukuran tampilan yang tersedia. xmlns mendefinisikan namespace android yang digunakan dalam dokumen XML. LinearLayout adalah elemen anak dari ScrollView dan merupakan tata letak yang berisi elemen-elemen lainnya. Pada LinearLayout, tampilan aplikasi nya nanti adalah *vertical*, dengan *padding* sebesar 24 dp dari batas elemen LinearLayout ke kontennya, dan lebar LinearLayout yang bernilai match_parent dan tingginya yang bernilai wrap_content

Untuk pengaturan ukuran, terdapat dua ukuran yang akan banyak digunakan dalam kodingan, yaitu match_parent yang berarti ukuran yang diatur akan mengikuti ukuran *handphone*, dan wrap_content yang berarti ukuran yang diatur akan mengikuti ukuran teks atau apapun yang ada di dalamnya. Berikut contoh ukuran match_parent dan wrap_content secara berurutan.



Gambar 1. match_parent



Gambar 2. wrap_content

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Lebar" />
<EditText
    android:id="@+id/et_width"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:inputType="numberDecimal"/>
```

Penjelasan:

Menampilkan label input beserta inputnya. Teks yang ditampilkan adalah “Lebar” untuk membuat input, digunakan tag EditText. Tag ini memiliki id yang akan dipakai untuk perhitungan nantinya, ukuran lebar yang mengikuti lebar handphone, tinggi yang membungkus konten saja, marginBottom untuk mengatur ukuran jarak margin ke teks atau konten setelahnya, serta inputType yang membatasi jenis input hanya angka desimal.

Untuk pembuatan tampilan Panjang dan Tinggi, format penulisannya sama seperti penulisan Lebar.

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Panjang" />
<EditText
    android:id="@+id/et_length"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:inputType="numberDecimal" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tinggi" />
<EditText
    android:id="@+id/et_height"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:inputType="numberDecimal" />
<Button
    android:id="@+id/btn_calculate"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="hitung volume" />

```

```

<Button
    android:id="@+id/btn_permukaan"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="hitung permukaan balok" />
<Button
    android:id="@+id/btn_keliling"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="hitung keliling balok" />
<TextView
    android:id="@+id/tv_result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="18dp"
    android:text="Hasil"
    android:textAlignment="center"
    android:textSize="18sp"
    android:textStyle="bold" />
</LinearLayout>
</ScrollView>

```

Penjelasan:

Baris atau bagian terakhir dari activity_volume.xml ini adalah pembuatan Button untuk menampilkan tombol pada aplikasi. Akan dibuat tiga Button, yaitu Hitung Volume, Hitung Permukaan Balok, dan Hitung Keliling Balok. Ketiganya memiliki id yang berbeda untuk proses perhitungan.

Selain itu, akan ada bagian `TextView` yang akan menampilkan hasil komputasi jika `Button` telah ditekan dan input telah diisi.

3.2. VolumeActivity.kt

```
package com.example.myapplication

import android.os.Bundle
import android.os.PersistableBundle
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 etLength: EditText
    private lateinit var etHeight: EditText
    private lateinit var btnCalculate: Button
    private lateinit var btnPermukaan: Button
    private lateinit var btnKeliling: Button
    private val KEY_RESULT = "key_result"
```

File `VolumeActivity.kt` yang disimpan pada folder `com.example.myapplication` yang ada dalam folder `java`, bersamaan dengan file `MainActivity.kt`. File ini menggunakan bahasa Kotlin sebagai *backend* nya.

Pada bagian awal file `VolumeActivity.kt`, melakukan *import* kelas yang akan dipakai. Setelahnya, akan dibuat kelas `VolumeActivity` dan menambahkan ekstensi kelas `ComponentActivity()` yang digunakan dalam Android Jetpack, yang menyediakan banyak fitur dan *tools* untuk membantu pengembangan aplikasi Android yang lebih modern. Selain itu, akan ditambahkan juga `View.OnClickListener` yang menunjukkan bahwa kelas ini memiliki metode `onClick` yang harus diimplementasikan.

Dalam kelas `VolumeActivity`, akan dideklarasikan berbagai variabel dan *value* yang akan dipakai didalamnya, beserta tipe nya.


```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView{
        MyApplicationTheme {
            setContentView(R.layout.activity_volume)
            tvResult = findViewById<TextView>(R.id.tv_result)
            etWidth = findViewById<EditText>(R.id.et_width)
            etLength = findViewById<EditText>(R.id.et_length)
            etHeight = findViewById<EditText>(R.id.et_height)
            btnCalculate = findViewById<Button>(R.id.btn_calculate)
            btnPermukaan = findViewById<Button>(R.id.btn_permukaan)
            btnKeliling = findViewById<Button>(R.id.btn_keliling)

            btnCalculate.setOnClickListener(this)
            btnPermukaan.setOnClickListener(this)
            btnKeliling.setOnClickListener(this)
            if (savedInstanceState != null){
                val result = savedInstanceState.getString(KEY_RESULT)
                tvResult.text = result
            }
        }
    }
}

```

Penjelasan:

Fungsi onCreate dipanggil saat aktivitas pertama kali dibuat. Di dalamnya, tata letak aktivitas diatur dengan menggunakan setContentView untuk menghubungkan tata letak dari file .xml dengan nama activity_volume. Element tampilan seperti TextView, EditText, dan Button diinisialisasi dengan menggunakan findViewById. Tiga tombol btnCalculate juga dihubungkan dengan *click listener*. Terakhir, jika ada perubahan, maka hasilnya akan mengubah nilai tampilan teks dari tvResult pada bagian .xml.

```

override fun onClick(view: View?) {
    if(view != null){
        if(view.id == R.id.btn_calculate){

            val inputLength: String = etLength.text.toString().trim();
            val inputWidth: String = etWidth.text.toString().trim();
            val inputHeight: String = etHeight.text.toString().trim();

            var isEmptyFields = false
            if (inputLength.isEmpty()){
                isEmptyFields = true
                etLength.error = "Field ini tidak boleh kosong"
            }
            if (inputWidth.isEmpty()){
                isEmptyFields = true
                etWidth.error = "Field ini tidak boleh kosong"
            }
            if (inputHeight.isEmpty()){
                isEmptyFields = true
                etHeight.error = "Field ini tidak boleh kosong"
            }

            if(!isEmptyFields){
                val volume: Double = inputLength.toDouble() * inputWidth.toDouble() * inputHeight.toDouble();

                tvResult.text = String.format("Volume: %s", volume.toString());
            }
        }
    }
}

```

```

if(view.id == R.id.btn_permukaan){
    val inputLength: String = etLength.text.toString().trim();
    val inputWidth: String = etWidth.text.toString().trim();
    val inputHeight: String = etHeight.text.toString().trim();

    var isEmptyFields = false
    if (inputLength.isEmpty()){
        isEmptyFields = true
        etLength.error = "Field ini tidak boleh kosong"
    }
    if (inputWidth.isEmpty()){
        isEmptyFields = true
        etWidth.error = "Field ini tidak boleh kosong"
    }
    if (inputHeight.isEmpty()){
        isEmptyFields = true
        etHeight.error = "Field ini tidak boleh kosong"
    }
    // rumus luas permukaan
    if(!isEmptyFields){
        val permukaan: Double = 2 * ((inputLength.toDouble() * inputWidth.toDouble()) + (inputLength.toDouble() * inputHeight.toDouble())
            + (inputWidth.toDouble() * inputHeight.toDouble()));

        tvResult.text = String.format("Permukaan: %s", permukaan.toString());
    }
}

```

```

if(view.id == R.id.btn_keliling){
    val inputLength: String = etLength.text.toString().trim();
    val inputWidth: String = etWidth.text.toString().trim();
    val inputHeight: String = etHeight.text.toString().trim();

    var isEmptyFields = false
    if (inputLength.isEmpty()){
        isEmptyFields = true
        etLength.error = "Field ini tidak boleh kosong"
    }
    if (inputWidth.isEmpty()){
        isEmptyFields = true
        etWidth.error = "Field ini tidak boleh kosong"
    }
    if (inputHeight.isEmpty()){
        isEmptyFields = true
        etHeight.error = "Field ini tidak boleh kosong"
    }
    if(!isEmptyFields){
        val keliling: Double = 4 * (inputLength.toDouble() + inputWidth.toDouble() + inputHeight.toDouble());
        tvResult.text = String.format("Keliling: %s", keliling.toString());
    }
}

```

Penjelasan:

Pertama, akan diambil nilai yang diinputkan pengguna lalu akan disimpan ppada variabel inputLength, inputWidth, dan inputHeight. Kemudian akan melakukan validasi yang telah dibuat mengenai input yang tidak boleh kosong.

Jika tombol Hitung Volume yang diklik, maka akan menghitung volume dari nilai input dan menampilkannya ke halaman aplikasi dengan mengganti teks yang ada pada TextView pada hasil, begitu juga pada button Hitung Luas Permukaan dan Hitung Keliling, Ketika di pencet akan menghasilkan hasil pada TextView yang ditentukan untuk menampilkan hasil sesuai rumus yang dibuat

```

}

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

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

```

Penjelasan:

Fungsi onSaveInstanceState digunakan untuk menyimpan keadaan aktivitas jika aktivitas di terminate

3.3. build.gradle

```
android {  
    namespace = "com.example.myapplication"  
    compileSdk = 34  
  
    defaultConfig {  
        applicationId = "com.example.myapplication"  
        minSdk = 24  
        targetSdk = 34  
        versionCode = 1  
        versionName = "1.0"  
    }  
}
```

Penjelasan:

Mengubah Sdk yang awalnya bernilai 33 menjadi 34.

3.4. AndroidManifest.xml

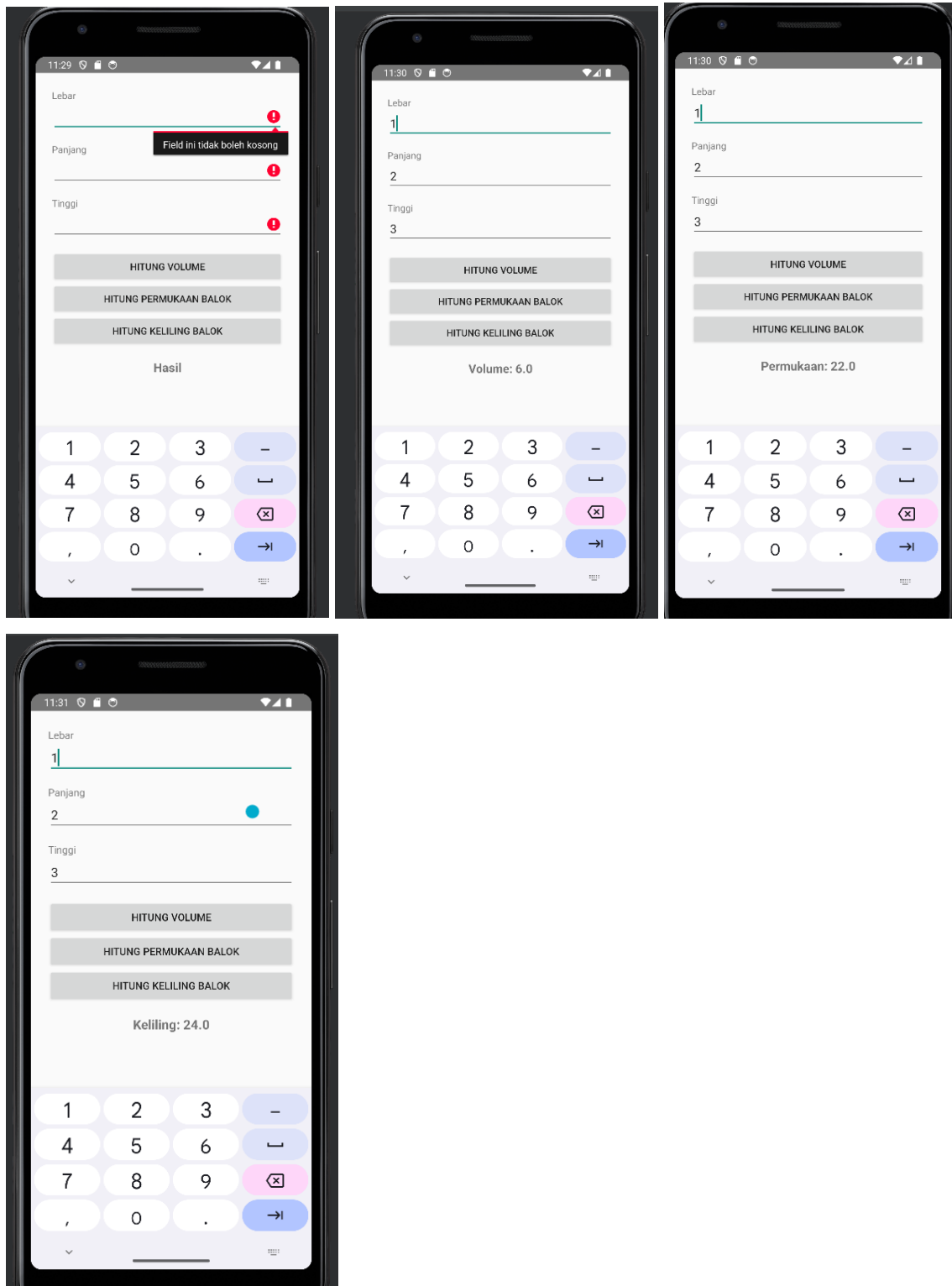
```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools">  
  
    <application  
        android:allowBackup="true"  
        android:dataExtractionRules="@xml/data_extraction_rules"  
        android:fullBackupContent="@xml/backup_rules"  
        android:icon="@mipmap/ic_launcher"  
        android:label="My Application"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportRtl="true"  
        android:theme="@style/Theme.MyApplication"  
        tools:targetApi="31">  
        <activity  
            android:name=".MainActivity"  
            android:exported="false" />  
        <activity  
            android:name=".VolumeActivity"  
            android:exported="true"  
            android:label="My Application"  
            android:theme="@style/Theme.MyApplication">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```

Penjelasan:

Menambah activity baru bernama VolumeActivity, dan mengubah exported pada activity MainActivity menjadi false dan VolumeActivity menjadi true, serta

memindahkan intent-filter ke activity VolumeActivity, tidak ada perubahan lain yang dibutuhkan.

3.5. Screenshot Aplikasi MyApplication



BAB IV

PENUTUP

4.1. Kesimpulan

Laporan ini berisi hasil dokumentasi praktikum dalam mengembangkan aplikasi Android menggunakan Android Studio. Selama praktikum, dapat menggali berbagai konsep dan komponen kunci dalam pengembangan aplikasi Android. Selain itu, dapat memahami cara menggunakan XML untuk mengatur tampilan antarmuka pengguna (UI) dan menggunakan Kotlin sebagai bahasa pemrograman untuk mengelola logika aplikasi.

Dapat mempelajari cara mengkonfigurasi tampilan UI melalui XML dan menghubungkannya dengan kode Kotlin menggunakan metode `findViewById`. Kami juga memahami teknik penanganan klik tombol dan menerapkan `View.OnClickListener` untuk merespons interaksi pengguna terhadap elemen UI. Selain itu, kami memahami dasar-dasar penanganan keadaan (state handling) dalam aplikasi Android, termasuk penggunaan metode `onSaveInstanceState` untuk menyimpan dan mengembalikan keadaan aktivitas saat diperlukan. Dapat membuat aplikasi dengan tiga tombol yang memungkinkan perhitungan volume, luas permukaan, dan keliling objek, seperti balok.

DAFTAR PUSTAKA

Android Studio, 2023. *Mengenal Android Studio* [Online] Available at: <https://developer.android.com/studio/intro?hl=id>. [Accessed 04 November 2023].

Dicoding Intern, 2020. *Apa Itu Kotlin? Kenapa Kita Harus Mempelajari Kotlin?* [Online] Available at: <https://www.dicoding.com/blog/apa-itu-kotlin-kenapa-kita-harus-mempelajari-kotlin/>. [Accessed 04 November 2023].

Modul Praktikum 7 PBP.