

BAB II

Android: Activity

1. Tujuan

Mahasiswa mampu membuat Aplikasi hitung volume sederhana.

2. Tools

Android Studio

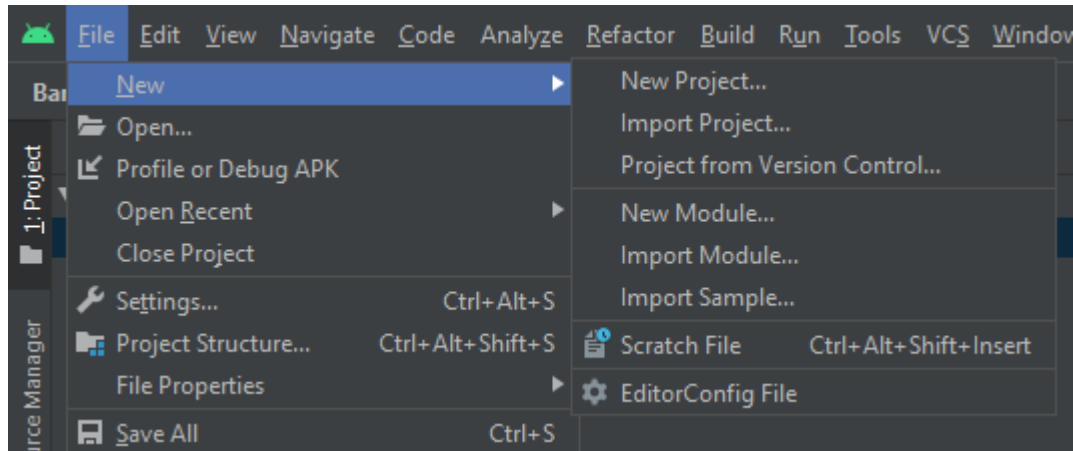
3. Dasar Teori

1. Activity merupakan sebuah komponen di Android yang berfungsi untuk menampilkan user interface ke layar handset Android pengguna. Ini seperti pada saat Anda melihat daftar percakapan pada aplikasi chat atau daftar email pada aplikasi Gmail di ponsel Android Anda.
2. Umumnya dalam sebuah aplikasi terdapat lebih dari satu activity yang saling terhubung dengan tugas yang berbeda-beda.
3. Activity merupakan salah satu komponen penting Android yang memiliki daur hidup (life cycle) dalam sebuah stack pada virtual sandbox yang disiapkan oleh Dalvik Virtual Machine (DVM) atau Android Runtime (ART) yang bersifat last in first out.
4. Pada implementasinya, activity selalu memiliki satu layout user interface dalam bentuk berkas xml.
5. Suatu aplikasi Android bisa memiliki lebih dari satu activity dan harus terdaftar di berkas AndroidManifest.xml sebagai sub aplikasi.
6. Sebuah class Java dinyatakan sebuah activity jika mewarisi (extends) superclass Activity atau turunannya seperti AppCompatActivity dan FragmentActivity.

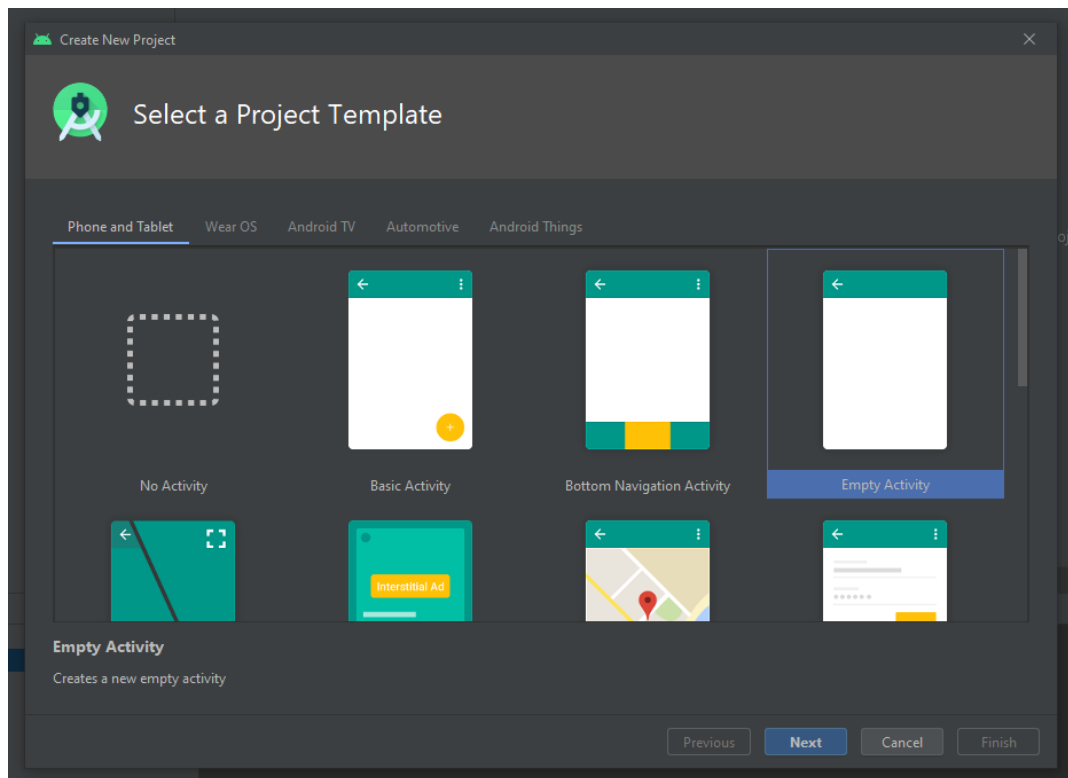
4. Langkah Praktikum

a. Membuat Proyek Baru

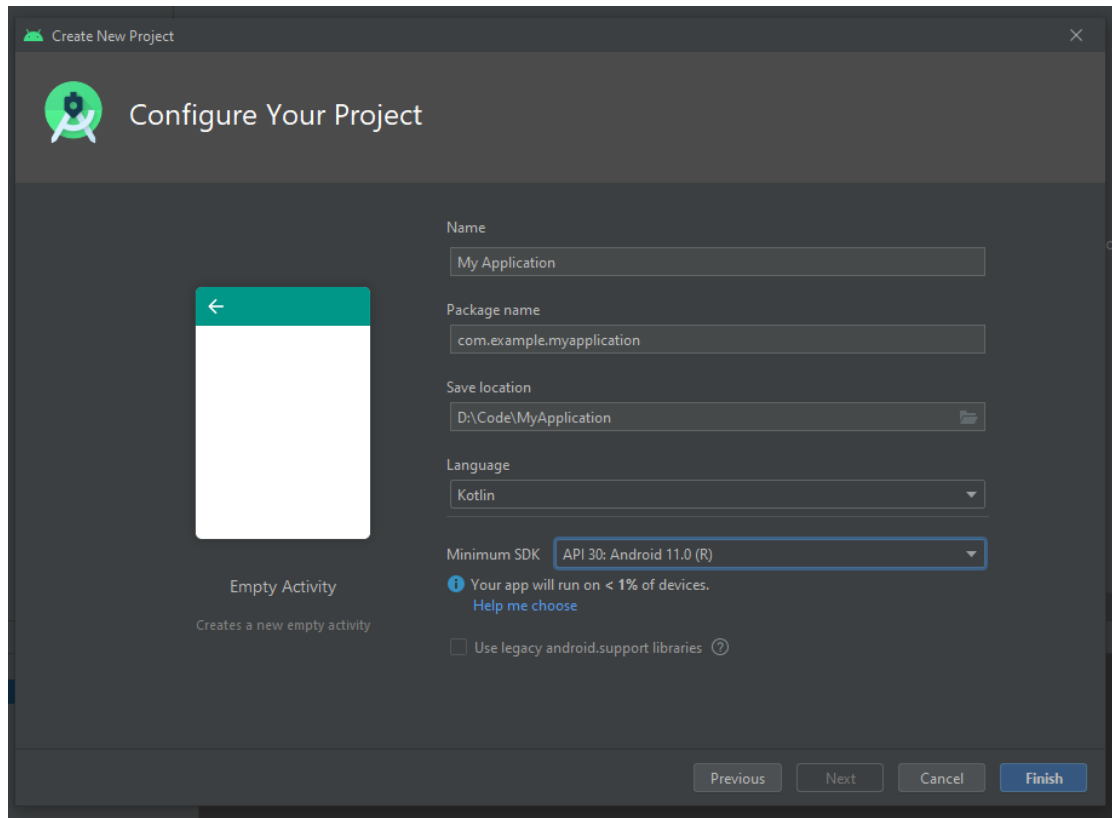
1. Buat proyek baru dengan klik File → New → New Project pada Android Studio Anda atau Anda bisa memilih Start a new Android Studio project di bagian dashboard.



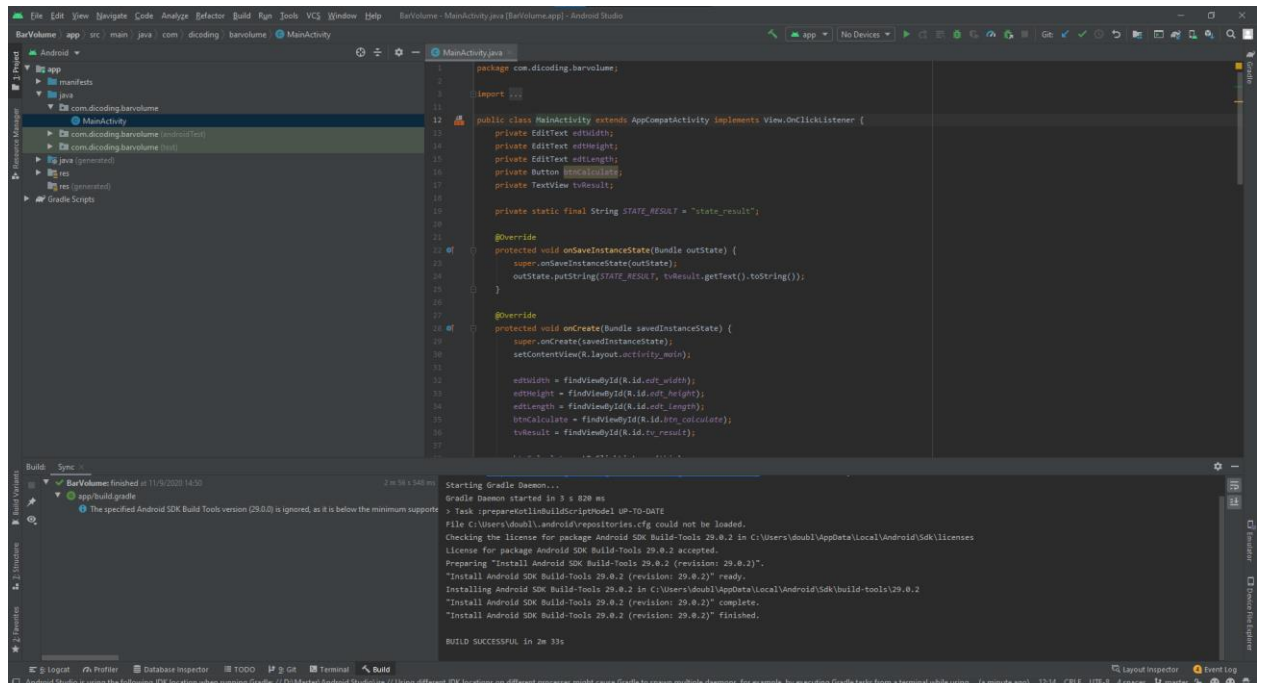
2. Pilih project template “Empty Activity”



3. Selanjutnya masukkan nama aplikasi dan nama package aplikasi Anda. Sebaiknya jangan sama dengan apa yang ada di contoh, karena ini berfungsi sebagai id dari aplikasi yang Anda buat. Pilihan target Android SDK akan mempengaruhi banyaknya peranti yang dapat menggunakan aplikasi. Untuk praktikum PBP ini akan menggunakan Bahasa Java

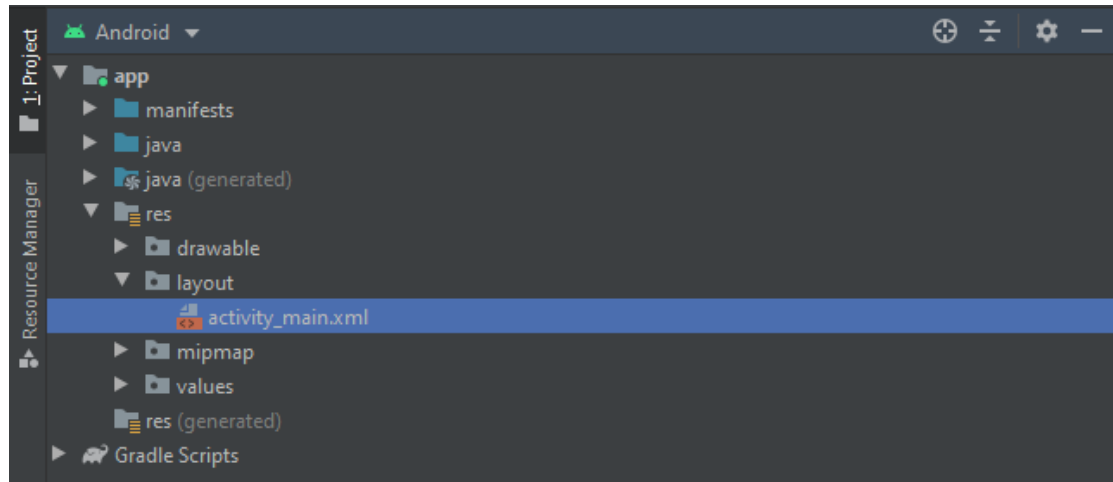


4. Tampilan layar Anda akan seperti contoh dibawah ini

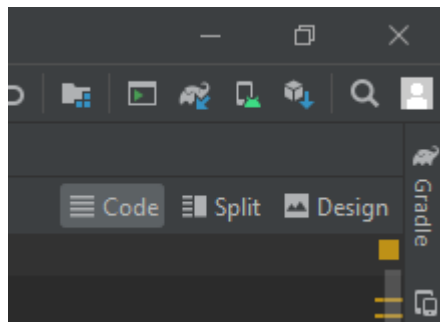


b. Layouting

1. Pilih file **activity_main.xml** pada workspace Anda (**res/layout/activity_main.xml**)



2. Pilih tab **Code** di sebelah pojok kanan atas



3. Masukkan kode berikut ini

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Panjang" />

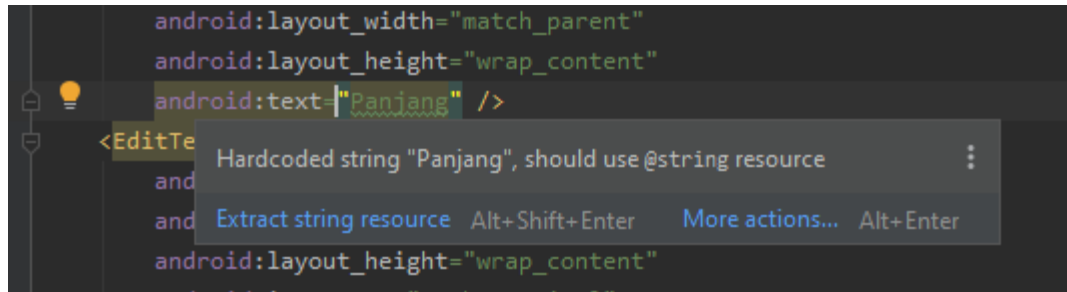
    <EditText
        android:id="@+id/edt_length"
        android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:lines="1" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lebar" />
<EditText
    android:id="@+id/edt_width"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:lines="1" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tinggi" />
<EditText
    android:id="@+id/edt_height"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:lines="1" />
<Button
    android:id="@+id/btn_calculate"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Hitung" />
<TextView
    android:id="@+id/tv_result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Hasil"
    android:textSize="24sp"
    android:textStyle="bold" />
</LinearLayout>

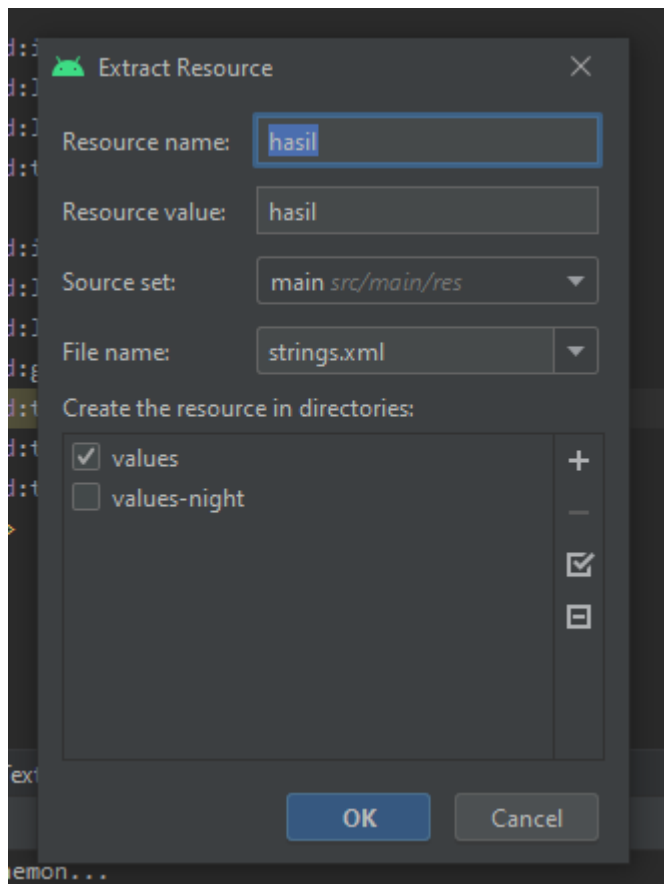
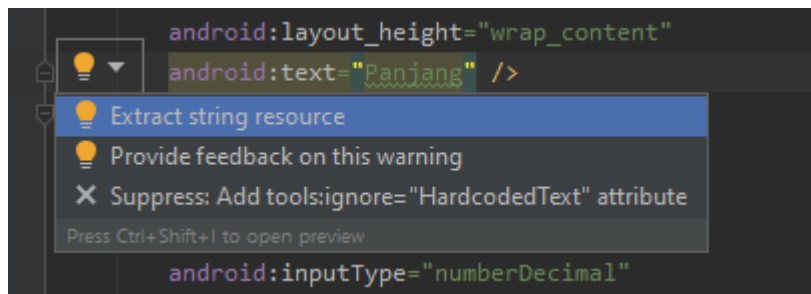
```

4. Maka akan muncul *warning* pada atribut **android:text** pada layout tersebut

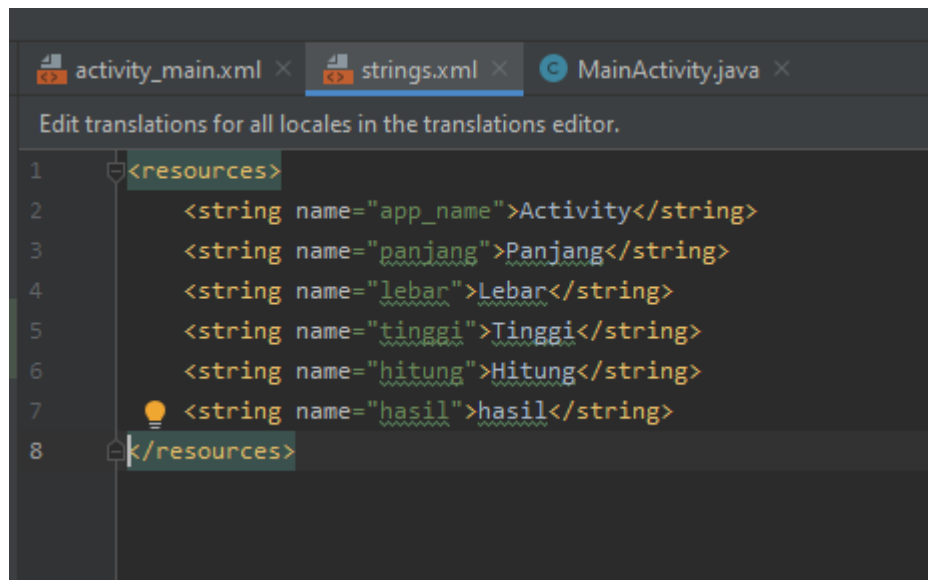


Ini karena kita melakukan *hardcoding* pada nilai *string*-nya. Hilangkan *code warning* tersebut dengan menekan `Alt+Enter` atau menekan lampu kuning di sebelah atribut tersebut.

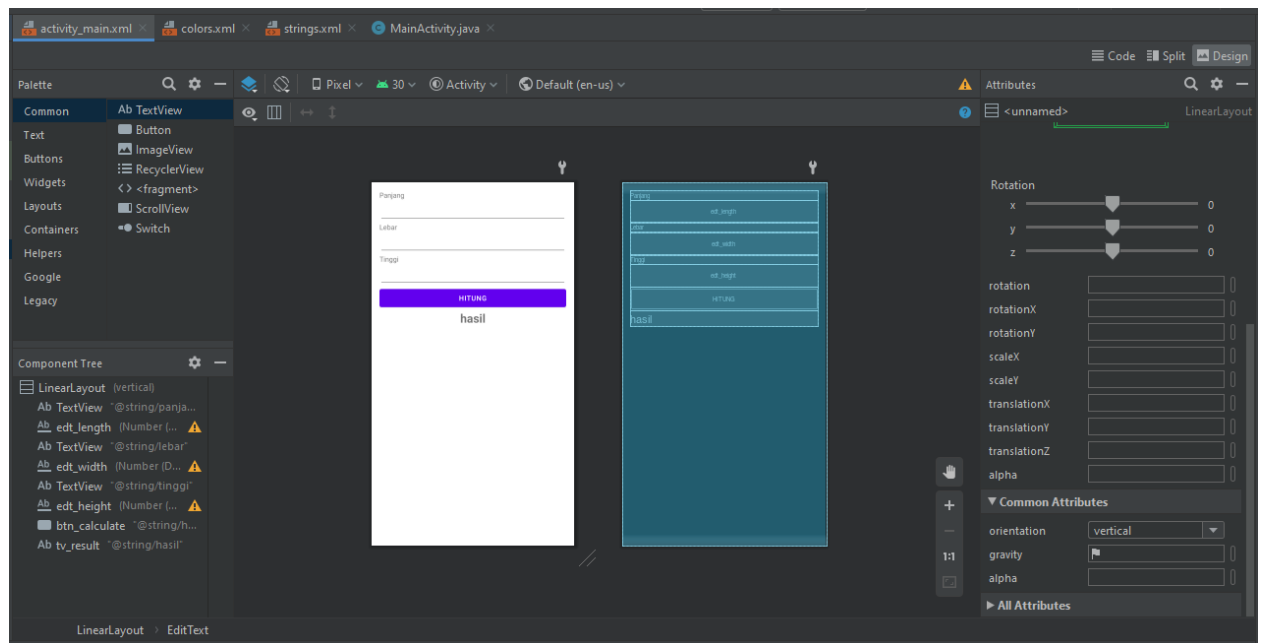
Akan muncul dialog seperti ini, pilih **extract string resource**



5. Fungsi extract string resource akan secara otomatis menambahkan nilai dari android:text ke dalam berkas res → values → strings.xml.



6. Maka hasil layout sementara akan menjadi seperti ini

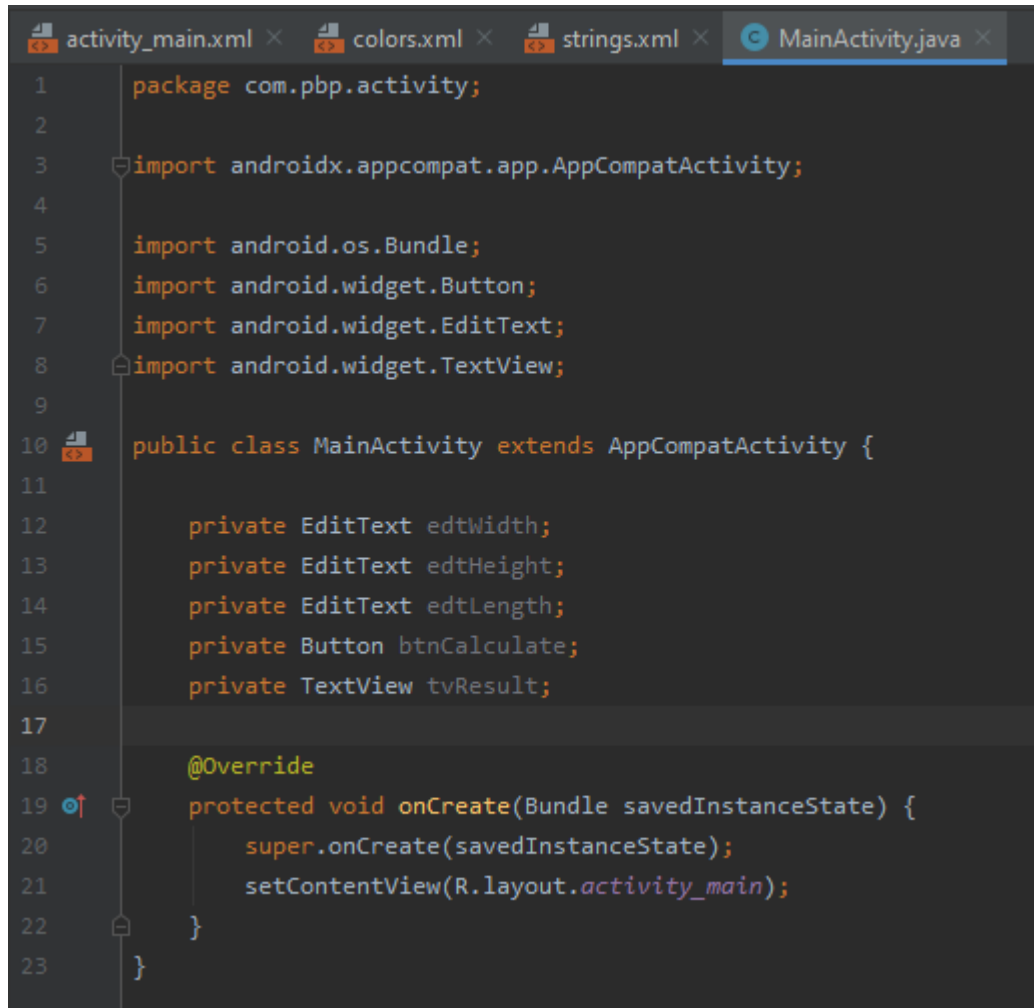


c. Activity

1. Tambahkan kode berikut pada **MainActivity**. Disarankan mengetik secara manual agar komponen dapat di-import secara otomatis.

```
private EditText edtWidth;
private EditText edtHeight;
```

```
private EditText edtLength;  
private Button btnCalculate;  
private TextView tvResult;
```

A screenshot of an IDE window showing the MainActivity.java file. The window has tabs for activity_main.xml, colors.xml, strings.xml, and MainActivity.java. The code is as follows:

```
1 package com.pbp.activity;  
2  
3 import androidx.appcompat.app.AppCompatActivity;  
4  
5 import android.os.Bundle;  
6 import android.widget.Button;  
7 import android.widget.EditText;  
8 import android.widget.TextView;  
9  
10 public class MainActivity extends AppCompatActivity {  
11  
12     private EditText edtWidth;  
13     private EditText edtHeight;  
14     private EditText edtLength;  
15     private Button btnCalculate;  
16     private TextView tvResult;  
17  
18     @Override  
19     protected void onCreate(Bundle savedInstanceState) {  
20         super.onCreate(savedInstanceState);  
21         setContentView(R.layout.activity_main);  
22     }  
23 }
```

2. Kemudian inisiasi variabel yang telah kita buat dengan menambahkan kode berikut di dalam metode **onCreate**.

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    edtWidth = findViewById(R.id.edt_width);  
    edtHeight = findViewById(R.id.edt_height);  
    edtLength = findViewById(R.id.edt_length);  
    btnCalculate = findViewById(R.id.btn_calculate);  
    tvResult = findViewById(R.id.tv_result);
```

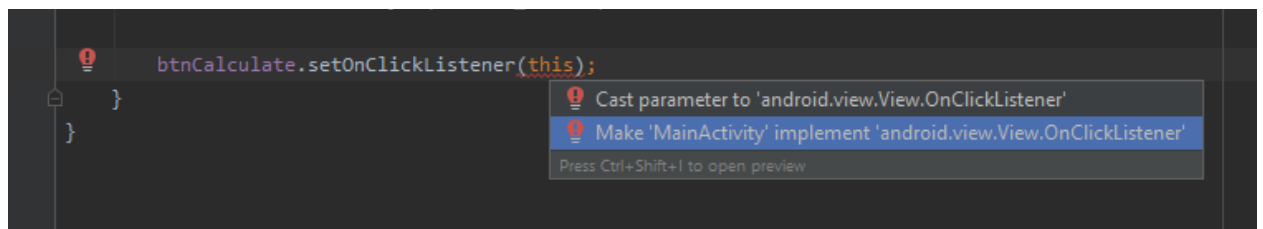


```

        btnCalculate.setOnClickListener(this);
    }
}

```

3. Maka akan muncul baris merah pada **this**. Hal ini karena kita belum menambahkan interface di kelas MainActivity. Setelah memilih aksi tersebut maka akan secara otomatis ditambahkan metode **onClick** dan interface **View.OnClickListener** pada class **MainActivity**



4. Masukkan kode berikut pada metode **onClick**

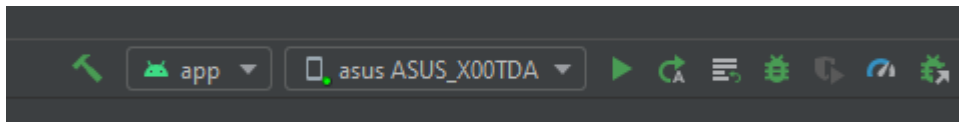
```

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_calculate) {
        String inputLength = edtLength.getText().toString().trim();
        String inputWidth = edtWidth.getText().toString().trim();
        String inputHeight = edtHeight.getText().toString().trim();
        double volume = Double.valueOf(inputLength) *
            Double.valueOf(inputWidth) * Double.valueOf(inputHeight);
        tvResult.setText(String.valueOf(volume));
    }
}

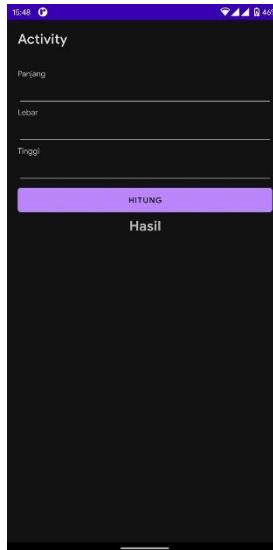
```

d. Run App

1. Jalankan aplikasi dengan memilih menu Run → Run 'app' dari menu bar atau melalui icon run berikut.



2. Pilih OK untuk menjalankan dan tunggu hingga proses building dan instalasi APK selesai. Jika sudah, seharusnya hasilnya akan seperti ini:



3. Namun masih ada sedikit masalah di sini, yaitu Anda tetap melakukan proses perhitungan walaupun salah satu nilainya kosong. Hal ini akan menyebabkan aplikasi force close karena perhitungan tidak dapat diproses. Maka untuk mengatasinya Anda akan menggunakan percabangan untuk mengecek apakah masing-masing EditText kosong atau tidak.
4. Buka kembali kelas **MainActivity**. Tambahkan kode berikut ke dalam metode **onClick** sebelum melakukan perhitungan.

```
@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_calculate) {
        String inputLength = edtLength.getText().toString().trim();
        String inputWidth = edtWidth.getText().toString().trim();
        String inputHeight = edtHeight.getText().toString().trim();
        boolean isEmptyFields = false;
        if (TextUtils.isEmpty(inputLength)) {
            isEmptyFields = true;
            edtLength.setError("Field ini tidak boleh kosong");
        }
        if (TextUtils.isEmpty(inputWidth)) {
            isEmptyFields = true;
            edtWidth.setError("Field ini tidak boleh kosong");
        }
        if (TextUtils.isEmpty(inputHeight)) {
            isEmptyFields = true;
            edtHeight.setError("Field ini tidak boleh kosong");
        }
    }
}
```

```

    }
    if (!isEmptyFields) {
        double volume = Double.valueOf(inputLength) *
Double.valueOf(inputWidth) * Double.valueOf(inputHeight);
        tvResult.setText(String.valueOf(volume));
    }
}
}

```

5. Untuk mengatasi nilai hasil yang hilang ketika orientasi device berubah maka tambahkan metode **onSaveInstanceState** pada **MainActivity**.

```

private static final String STATE_RESULT = "state_result";
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putString(STATE_RESULT,
tvResult.getText().toString());
}

```

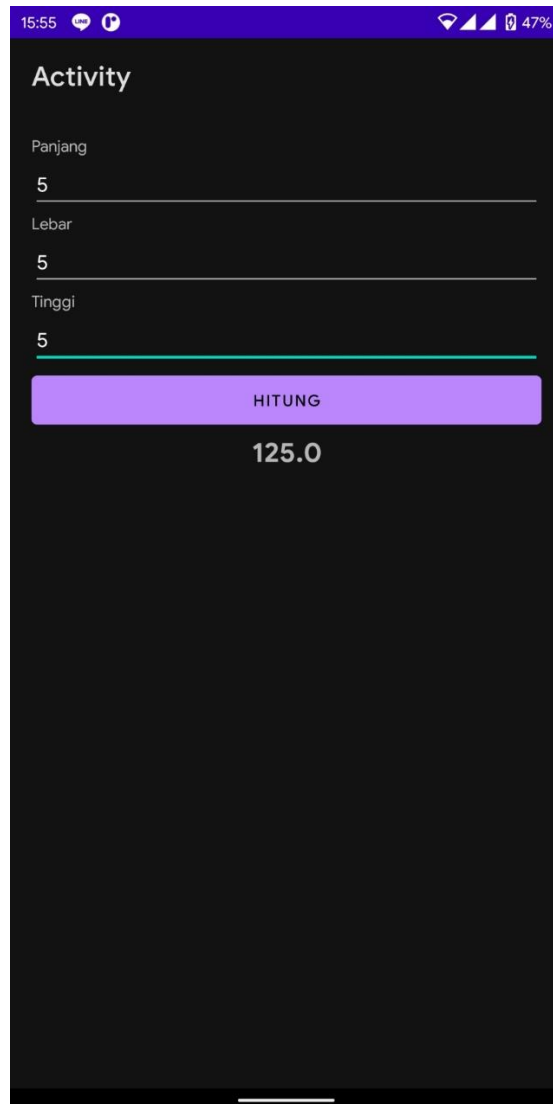
Dan juga beberapa baris kode pada baris terakhir metode **onCreate**.

```

if (savedInstanceState != null) {
    String result = savedInstanceState.getString(STATE_RESULT);
    tvResult.setText(result);
}

```

6. Maka hasil akhir Aplikasi hitung volume akan menjadi seperti ini.



5. Tugas

1. Membuat Aplikasi sesuai dengan yang ada di modul
2. Membuat Video yang berisi tentang Penjelasan kode dan praktikum minggu ini