

MODUL IV

APLIKASI SEDERHANA

I. TUJUAN

- A. Mahasiswa mengetahui apa itu *Activity* pada Aplikasi *Android*
- B. Mahasiswa Mampu memahami tahapan pengimplementasian komponen *Activity* pada aplikasi *Android* sederhana.

II. ALAT DAN BAHAN

- A. Laptop
- B. Aplikasi *Android Studio*
- C. Modul

III. TEORI DASAR

Dalam modul ini kita akan belajar tentang komponen-komponen dasar yang digunakan untuk membuat aplikasi *android* yang sederhana. Beberapa komponen diantaranya adalah:

A. *Activity*

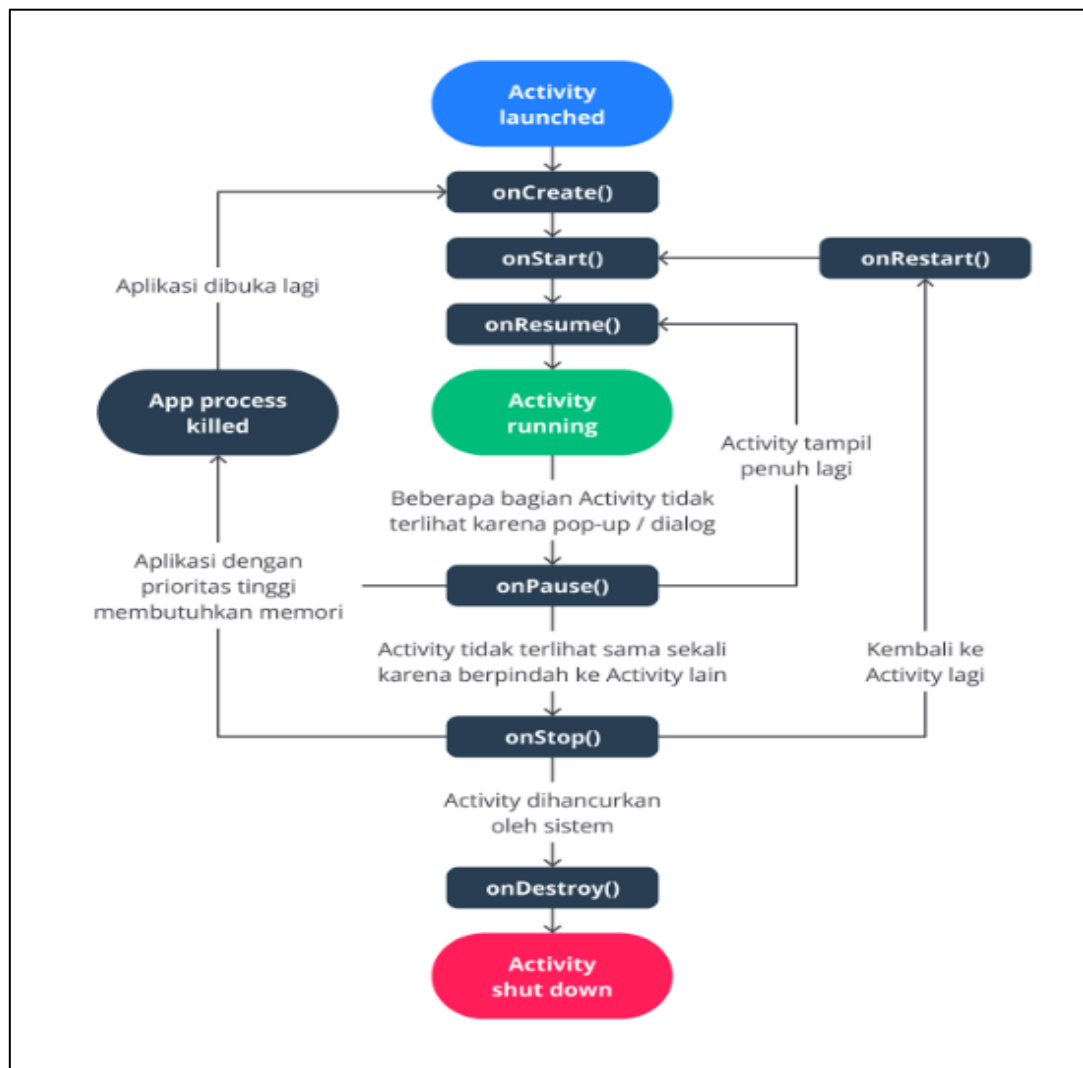
Activity merupakan salah satu komponen penting *Android* yang berfungsi untuk menampilkan *user interface* ke layar pengguna. Ini seperti pada saat Anda melihat daftar percakapan pada aplikasi *chat* atau daftar *email* pada aplikasi *Gmail* di ponsel *Android* Anda. Di dalamnya Anda dapat berinteraksi dengan aplikasi Anda, baik dengan menekan tombol atau menampilkan *list*. Seperti ketika Anda membuat project baru di *Android Studio*, biasanya akan ada dua berkas yang sudah tercipta, yaitu *MainActivity* dan *activity_main.xml*. *MainActivity* ini disebut sebagai class *Activity* karena mewarisi (*extends*) *superclass* *Activity*. Tugasnya yaitu menampilkan layout *activity_main.xml* dan mengelola interaksi yang ada di dalamnya.

Umumnya dalam sebuah aplikasi terdapat lebih dari satu *Activity* yang saling terhubung dengan tugas yang berbeda-beda. Yang perlu diperhatikan yaitu setiap *Activity* harus terdaftar di *AndroidManifest.xml*. Secara *default*, ia akan didaftarkan jika Anda membuat *Activity* baru dengan cara otomatis. Caranya yaitu klik kanan pada nama package → New → *Activity* → pilih template *Activity* yang tersedia.

Siklus hidup *Activity* melibatkan beberapa metode utama seperti *onCreate()*, *onStart()*, *onResume()*, *onPause()*, *onStop()*, *onDestroy()*, dan *onRestart()*. Ini memungkinkan pengembang untuk mengelola berbagai tindakan sesuai dengan tahapan siklus hidup *Activity*.

1. *onCreate()*: Metode pertama yang dipanggil saat *Activity* dibuat. Inisialisasi umumnya dilakukan di sini.
2. *onStart()*: Dipanggil ketika *Activity* dimulai dan tampak bagi pengguna.
3. *onResume()*: Aktivitas berada dalam fokus pengguna.
4. *onPause()*: Aktivitas kehilangan fokus, tetapi masih terlihat.

5. `onStop()`: Aktivitas tidak lagi terlihat.
6. `onDestroy()`: Aktivitas dihancurkan dan dibebaskan dari memori.
7. `onRestart()`: Dipanggil saat Activity dimulai lagi setelah dihentikan.

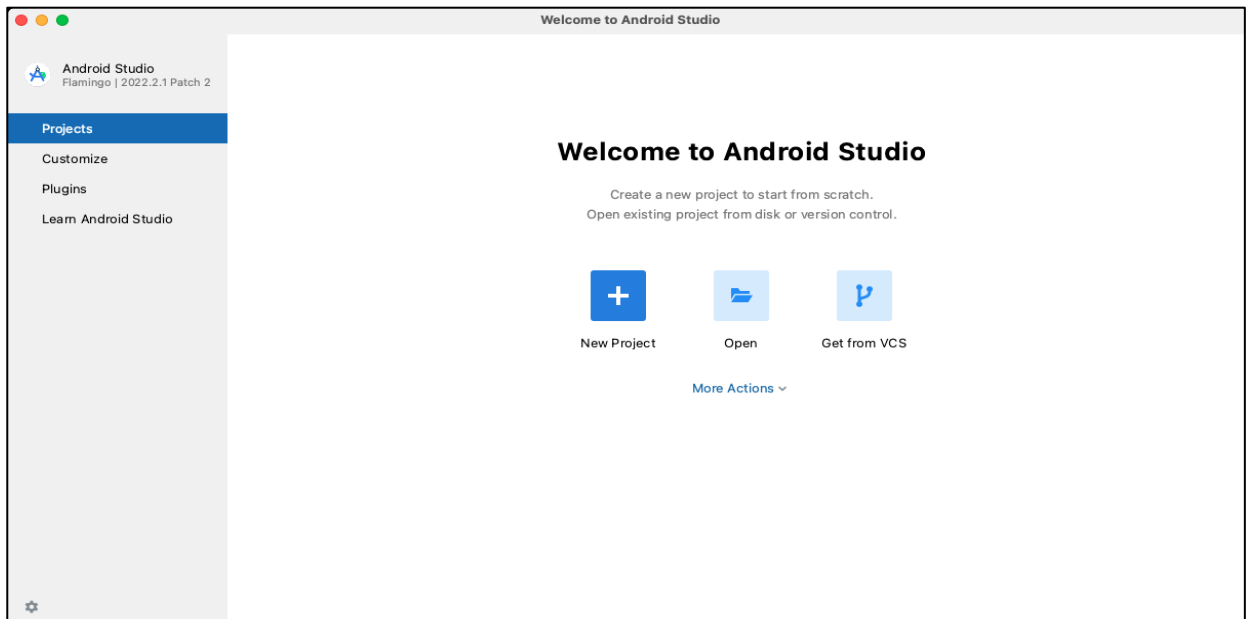


- A. Activity merupakan sebuah komponen di *Android* yang berfungsi untuk menampilkan *user interface* ke layar *handset Android* pengguna. Ini seperti pada saat Anda melihat daftar percakapan pada aplikasi *chat* atau daftar *email* pada aplikasi Gmail di ponsel *Android* Anda. Activity adalah komponen pada aplikasi *Android* yang menampilkan halaman interaksi.
- B. Umumnya dalam sebuah aplikasi terdapat lebih dari satu *activity* yang saling terhubung dengan tugas yang berbeda-beda.
- C. 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*.
- D. Pada implementasinya, *activity* selalu memiliki satu *layout user interface* dalam bentuk berkas *xml*.
- E. Suatu aplikasi *Android* bisa memiliki lebih dari satu *activity* dan harus terdaftar di berkas *AndroidManifest.xml* sebagai sub aplikasi.
- F. Sebuah *class Java* dinyatakan sebuah *activity* jika mewarisi (*extends*) superclass *Activity* atau turunannya seperti *AppCompatActivity* dan *FragmentActivity*.

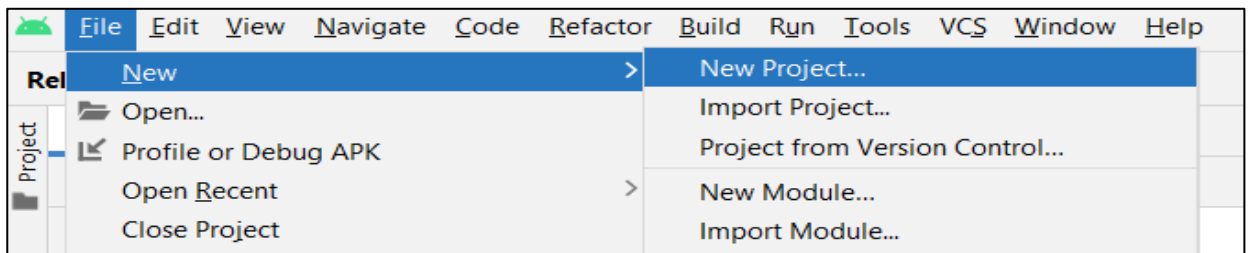
IV. LANGKAH KERJA

A. Codelab membuat proyek baru

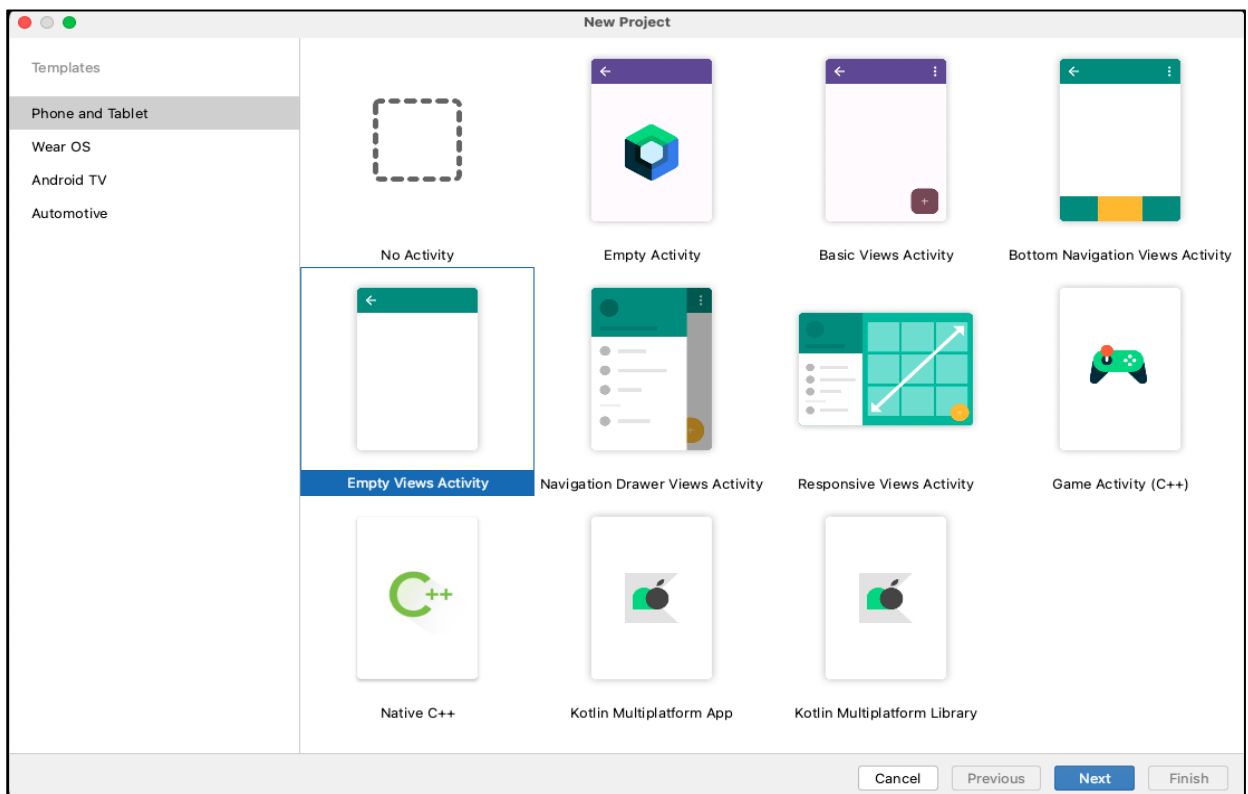
1. Jika Anda berada di halaman **dashboard**, pilih **New Project** pada Android Studio.



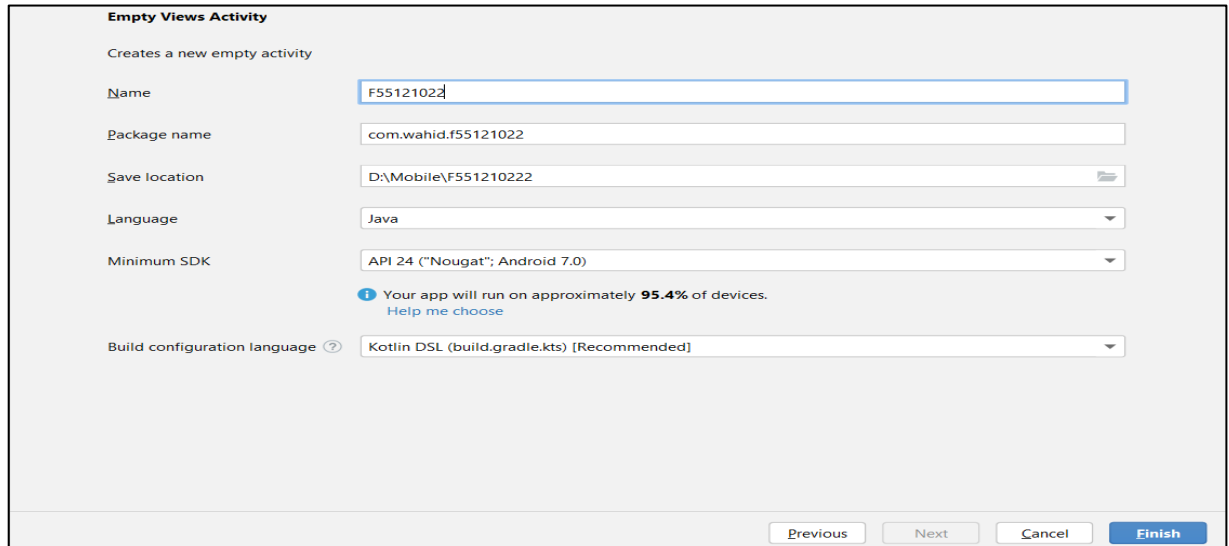
2. Buat proyek baru dengan klik *File* → *New* → *New Project* pada *Android Studio* Anda atau Anda bisa memilih *Start a new Android Studio project* di bagian *dashboard*.



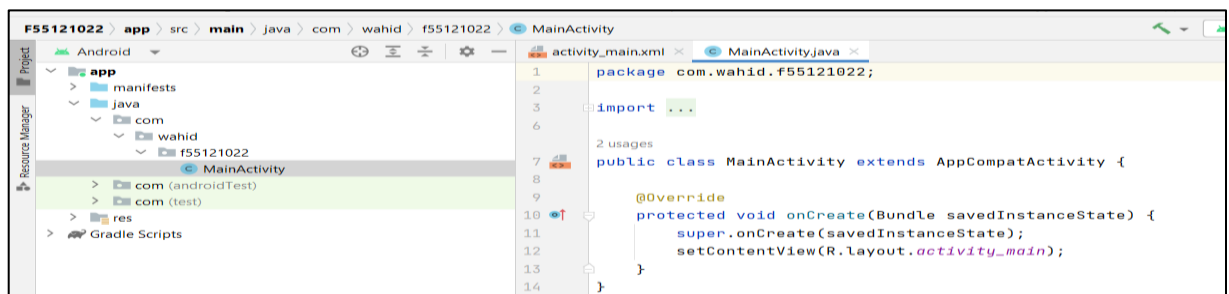
3. Pada bagian ini kita akan memilih tipe activity awal dari *template* yang telah disediakan.



4. 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. Kemudian Anda bisa menentukan lokasi proyek yang akan Anda buat. Setelah itu pilih tipe gawai/peranti (*device*) untuk aplikasi beserta target minimum SDK yang akan digunakan. Pilihan target Android SDK akan mempengaruhi banyaknya peranti yang dapat menggunakan aplikasi. Di sini kita memilih nilai minimum SDK kita pasang ke **Level 24 (Nougat)**. Klik **Finish** untuk melanjutkan.

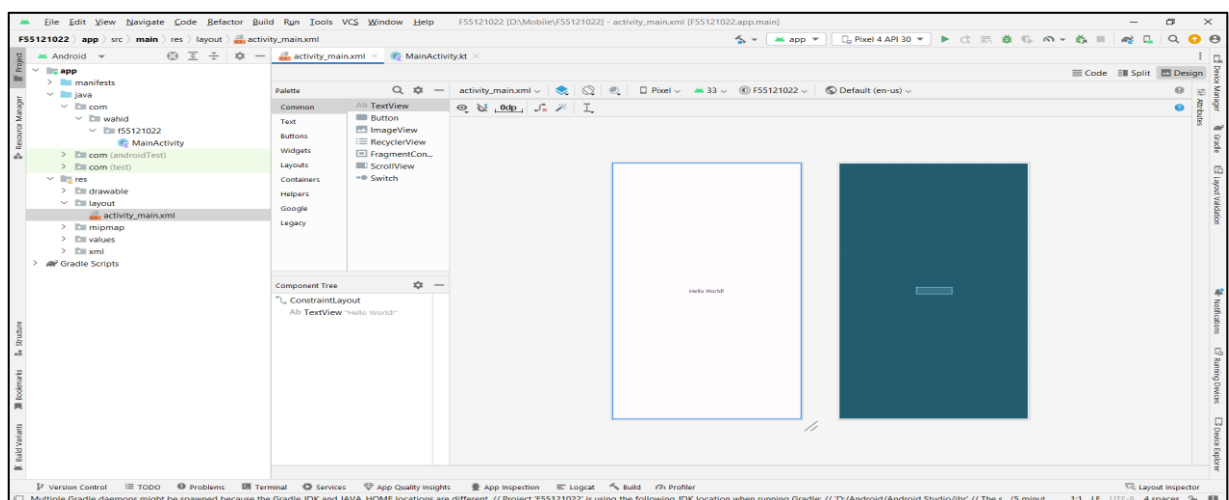


5. Tampilan layar Anda akan seperti contoh di bawah ini:



B. Menyusun Tampilan Aplikasi (Layouting)

1. Carilah berkas **activity_main.xml** pada *workspace* Anda yang terletak di folder **res/layout/activity_main.xml**. Jika sudah menemukan, klik dua kali untuk membuka file tersebut. Pastikan **project window** pada pilihan **Android** seperti di bawah ini.



2. Fokus pada bagian pojok kanan atas, di sana terdapat beberapa pilihan tampilan untuk mengatur layouting. Anda bisa menggunakan **Code** untuk menuliskan kode layouting, **Design** untuk menyusun tampilan dengan *drag & drop*, atau **Split** untuk membuka keduanya.

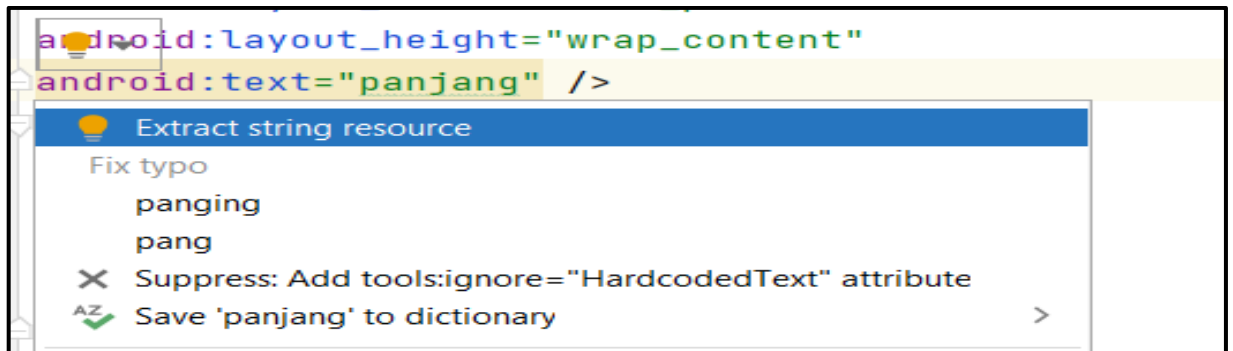


3. Ubah *layout* dasar dari *ConstraintLayout* menjadi *LinearLayout* seperti berikut dan menambahkan baris *TextView* dan *ListView* seperti kode di bawah ini :

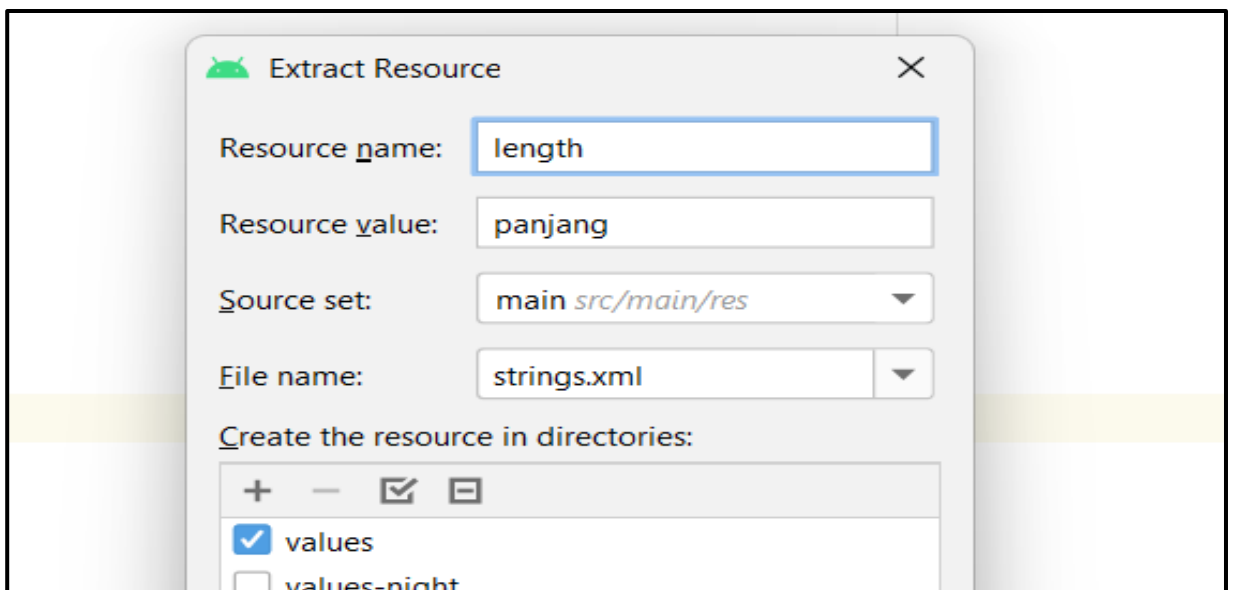
```
Pro
Resource Manager
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:padding="16dp"
6     android:orientation="vertical">
7
8     <TextView
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:text="Panjang" />
12
13    <EditText
14        android:id="@+id/edt_length"
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content"
17        android:inputType="numberDecimal"
18        android:lines="1" />
19
20    <TextView
21        android:layout_width="match_parent"
22        android:layout_height="wrap_content"
23        android:text="Lebar" />
24
25    <EditText
26        android:id="@+id/edt_width"
27        android:layout_width="match_parent"
28        android:layout_height="wrap_content"
29        android:inputType="numberDecimal"
30        android:lines="1" />
31
32    <TextView
33        android:layout_width="match_parent"
34        android:layout_height="wrap_content"
35        android:text="Tinggi" />
36
37    <EditText
38        android:id="@+id/edt_height"
39        android:layout_width="match_parent"
40        android:layout_height="wrap_content"
41        android:inputType="numberDecimal"
42        android:lines="1" />
43
44    <Button
45        android:id="@+id/btn_calculate"
46        android:layout_width="match_parent"
47        android:layout_height="wrap_content"
48        android:text="Hitung" />
49
50    <TextView
51        android:id="@+id/tv_result"
52        android:layout_width="match_parent"
53        android:layout_height="wrap_content"
54        android:gravity="center"
55        android:text="Hasil"
56        android:textSize="24sp"
57        android:textStyle="bold" />
58 </LinearLayout>
```

Device Manager
Gradle
Layout Validation
Notifications
Run

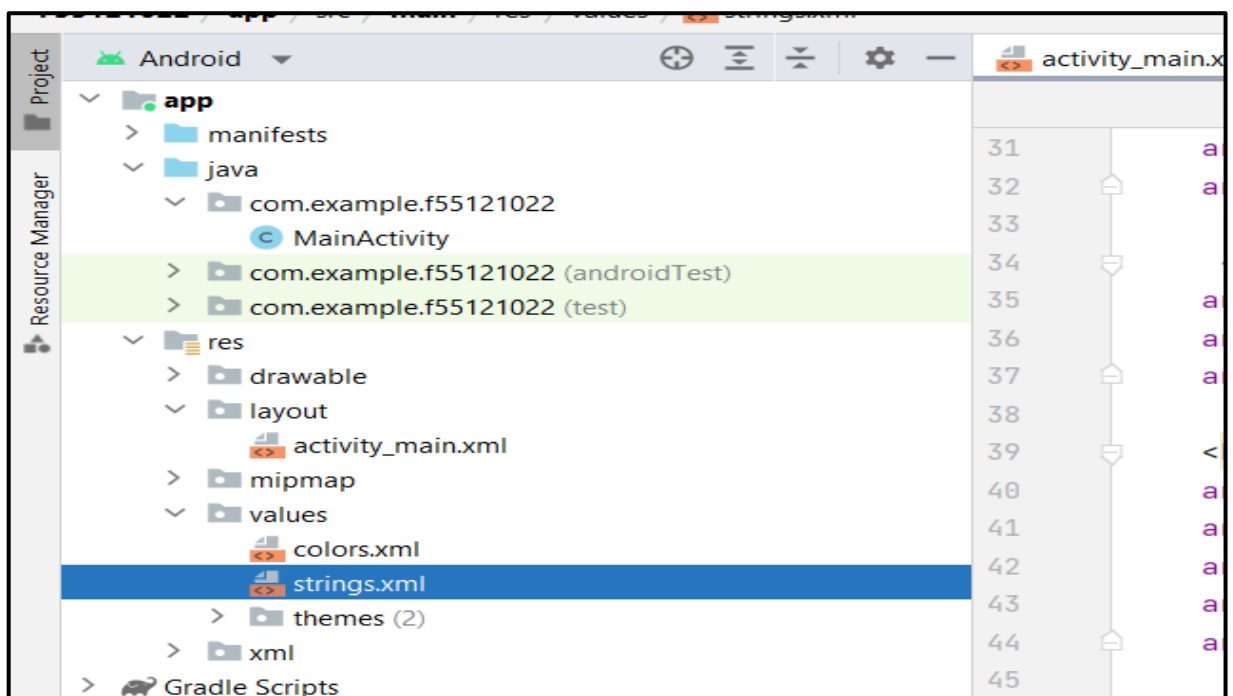
4. Kemudian akan muncul warning pada atribut `android:text` pada `layout` tersebut. Ini karena kita melakukan *hardcoding* pada nilai `string`-nya. Mari kita hilangkan `code warning` tersebut dengan menekan `Alt+Enter` (*option + return* pada *Mac*) atau menekan lampu kuning yang muncul pada `attribute android:text`. Akan muncul dialog seperti ini, pilih *extract string resource*.



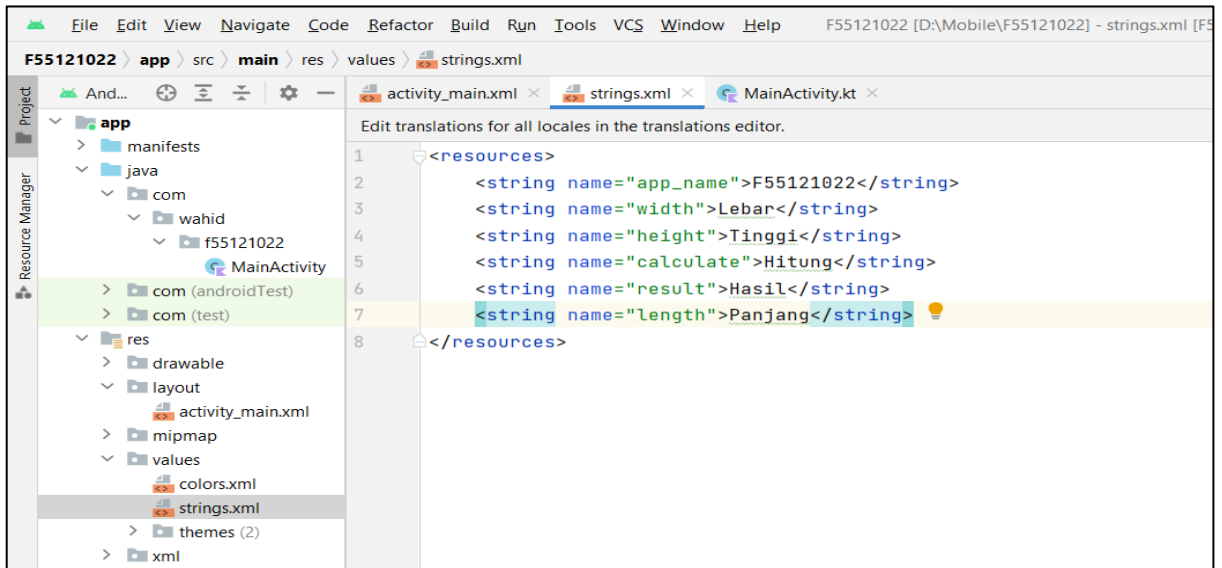
5. Kemudian akan muncul dialog seperti di bawah ini. Sesuaikan dengan nama yang ada.



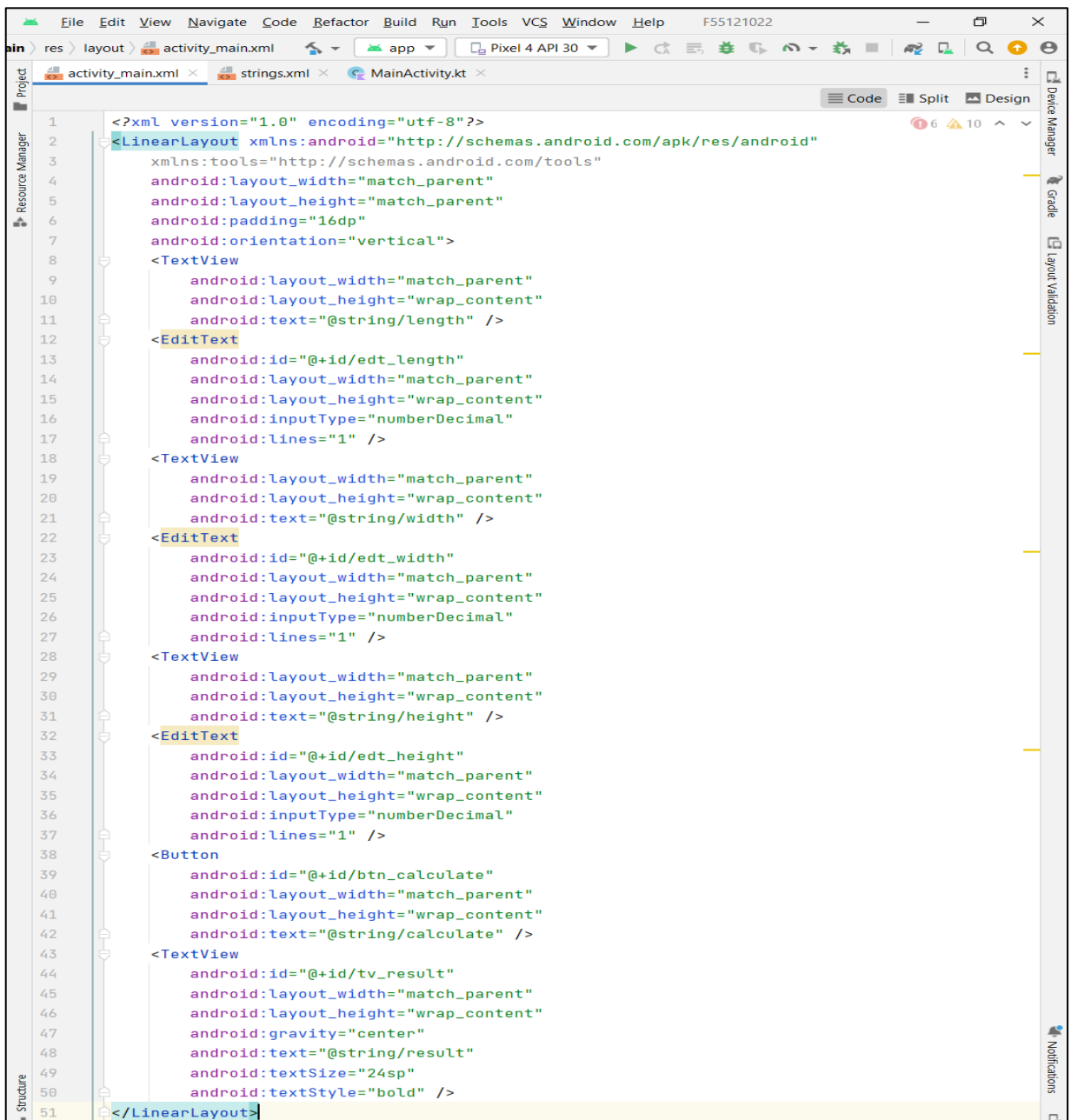
6. Fungsi *extract string resource* akan secara otomatis menambahkan nilai dari `android:text` ke dalam berkas `res` → `values` → `strings.xml`.



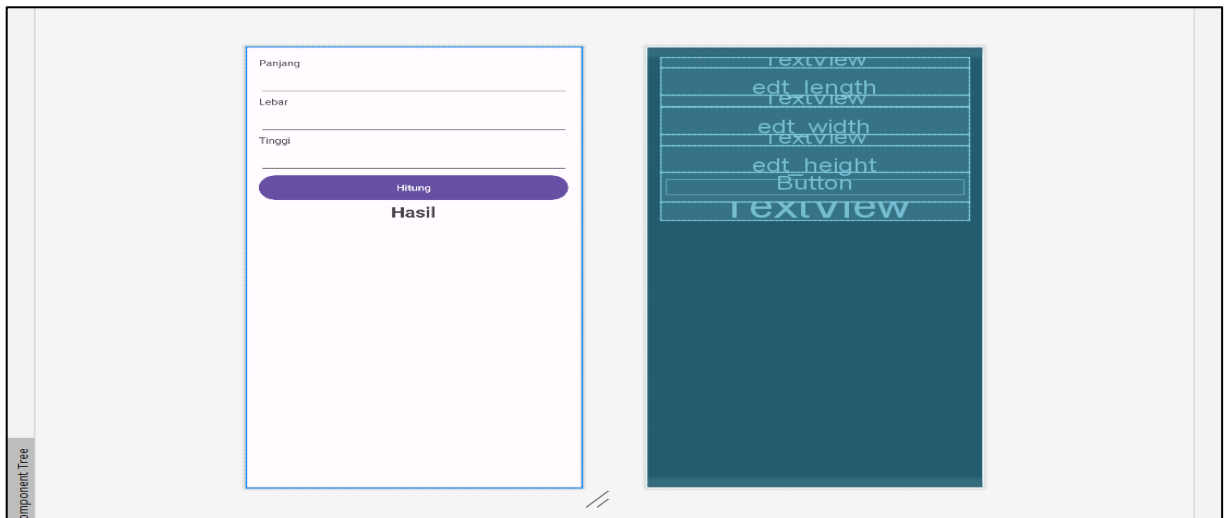
7. Lakukan hal yang sama pada komponen *view* lainnya hingga tidak ada *warning* lagi. Jika kita buka berkas *strings.xml*, maka isinya akan menjadi seperti ini:



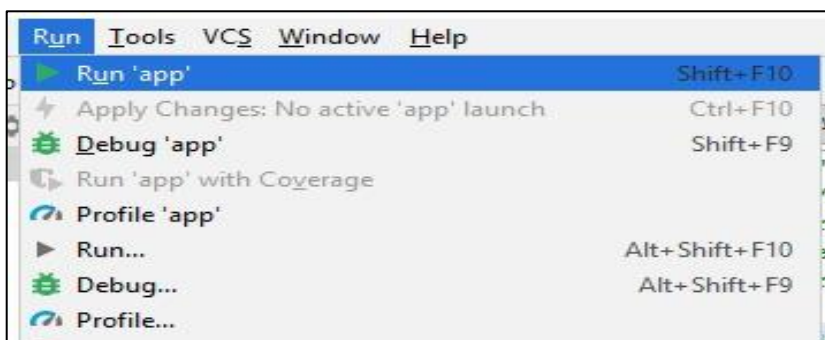
8. Maka kode di dalam *activity_main.xml* akan menjadi seperti ini:



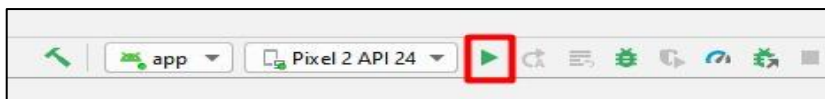
9. Jika Anda perhatikan, hasil *layout* sementara akan menjadi seperti ini.



10. Setelah selesai menyusun tampilan aplikasi, jalankan aplikasi dengan memilih menu **Run** → **Run 'app'** dari *menu bar*.

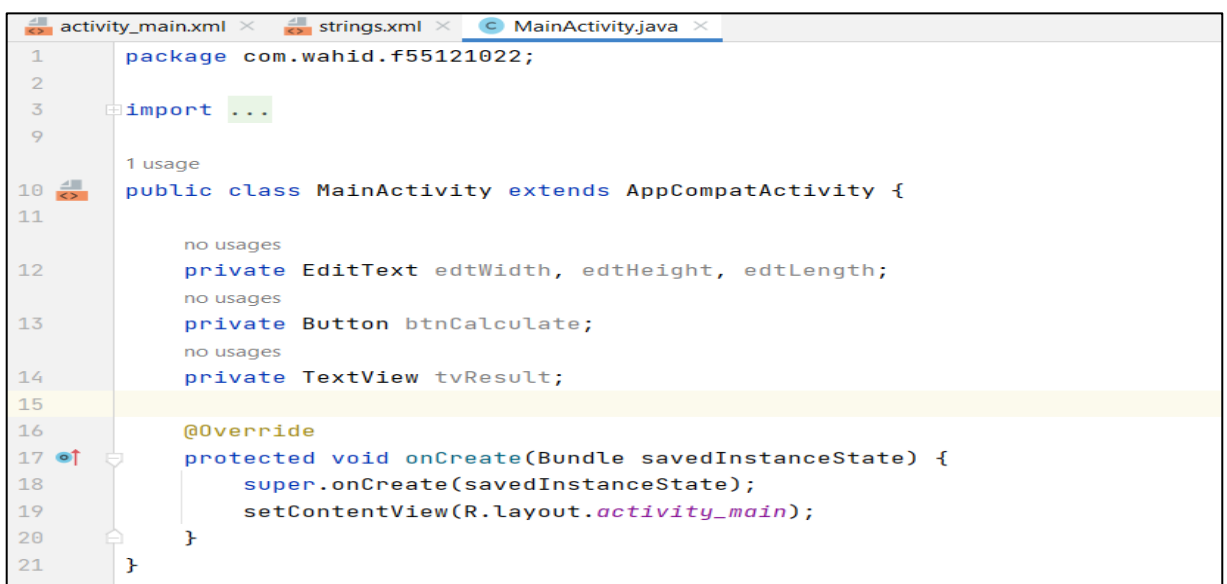


Selain cara di atas, Anda juga dapat menekan icon berikut di toolbar:



C. Codelab Kode Logika pada *MainActivity*.

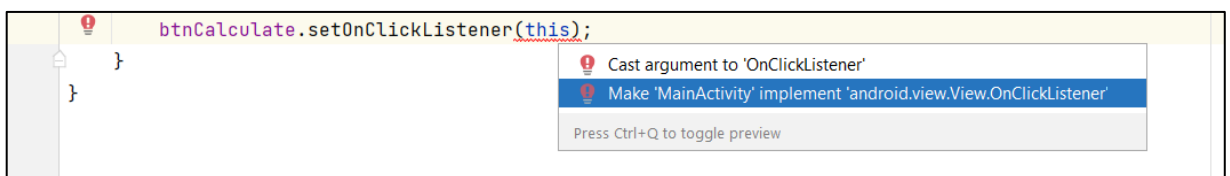
1. Selanjutnya setelah selesai, lanjutkan dengan membuka berkas *MainActivity* dan lanjutkan ketik baris-baris di bawah ini. Tambahkan beberapa variabel yang akan digunakan untuk menampung *view*.



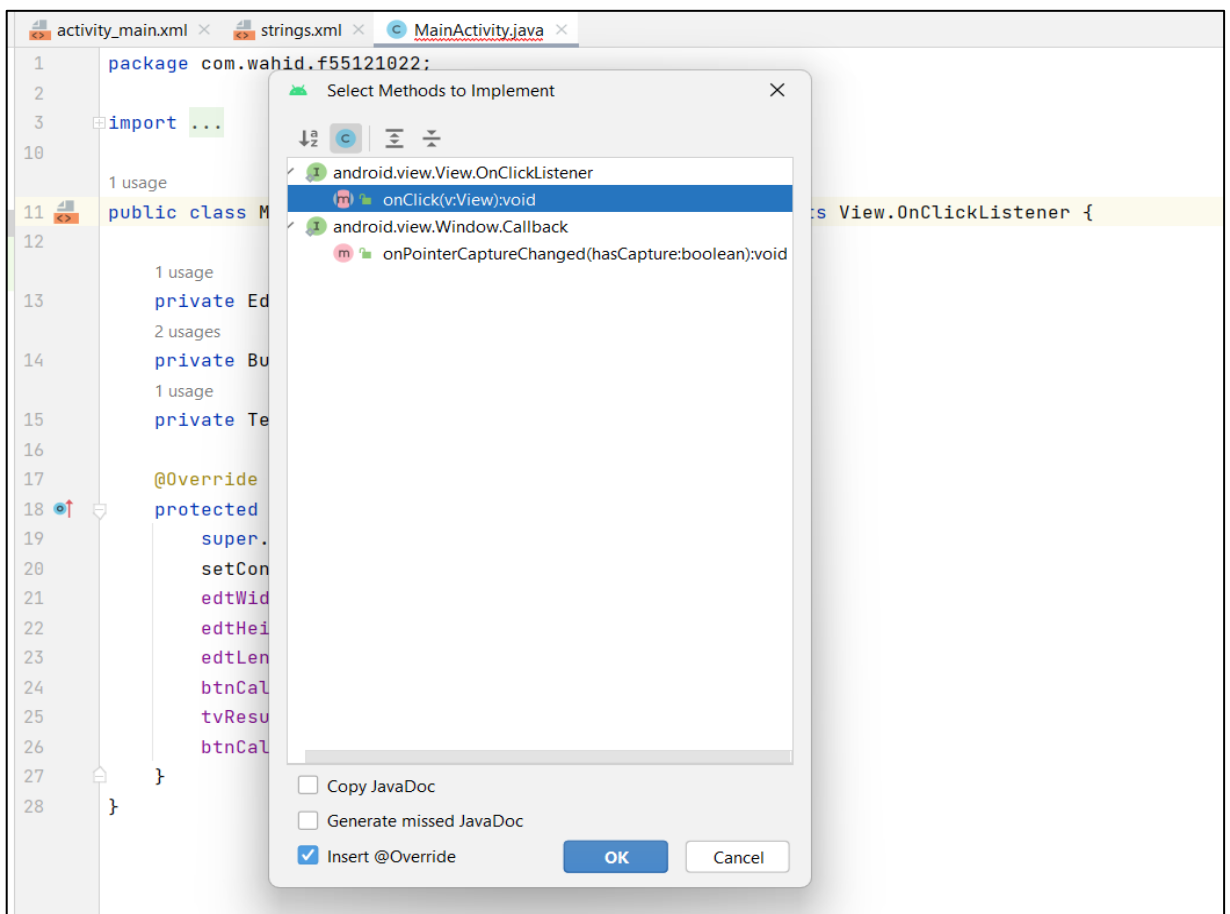
2. Kemudian inisiasi variabel yang telah kita buat dengan menambahkan kode berikut di dalam metode *onCreate*.

```
15
16      @Override
17      protected void onCreate(Bundle savedInstanceState) {
18          super.onCreate(savedInstanceState);
19          setContentView(R.layout.activity_main);
20          edtWidth = findViewById(R.id.edt_width);
21          edtHeight = findViewById(R.id.edt_height);
22          edtLength = findViewById(R.id.edt_length);
23          btnCalculate = findViewById(R.id.btn_calculate);
24          tvResult = findViewById(R.id.tv_result);
25          btnCalculate.setOnClickListener(this);
26      }
27  }
```

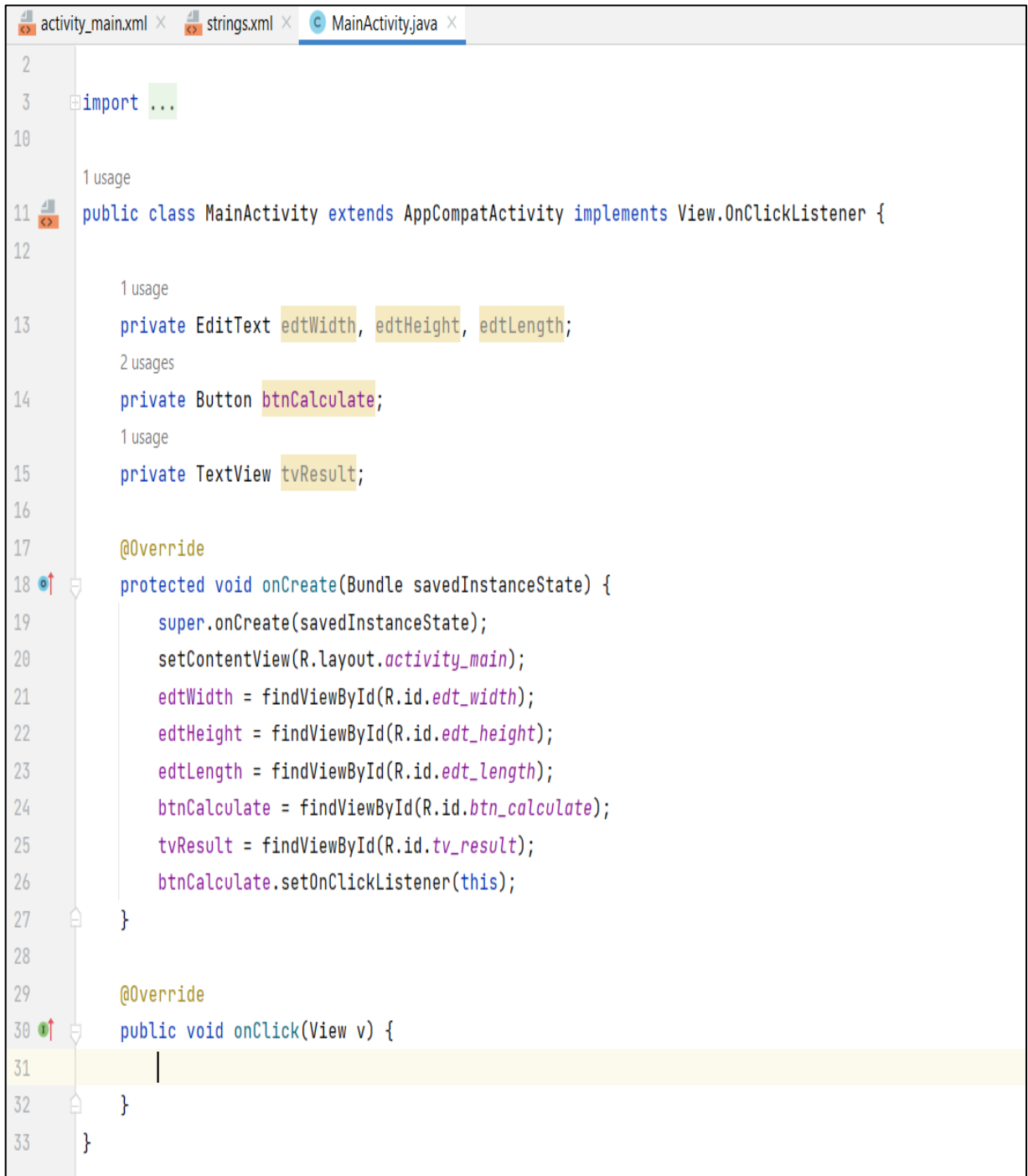
3. Akan muncul baris merah pada kata *this*. Hal ini karena kita belum menambahkan *interface* di kelas *MainActivity*. Silakan klik di atas baris merah tersebut, kemudian tekan tombol *Alt + Enter* (*option + return* pada *Mac*) atau menekan lampu merah yang muncul lalu pilih aksi berikut untuk *implement interface*.



4. Android Studio akan menawarkan Anda kode yang akan diimplementasikan. Anda bisa langsung klik tombol **OK** untuk melanjutkan.




5. Dengan begitu, akan ada tambahan kode implementasi `View.OnClickListener()` secara otomatis kelas **MainActivity**.



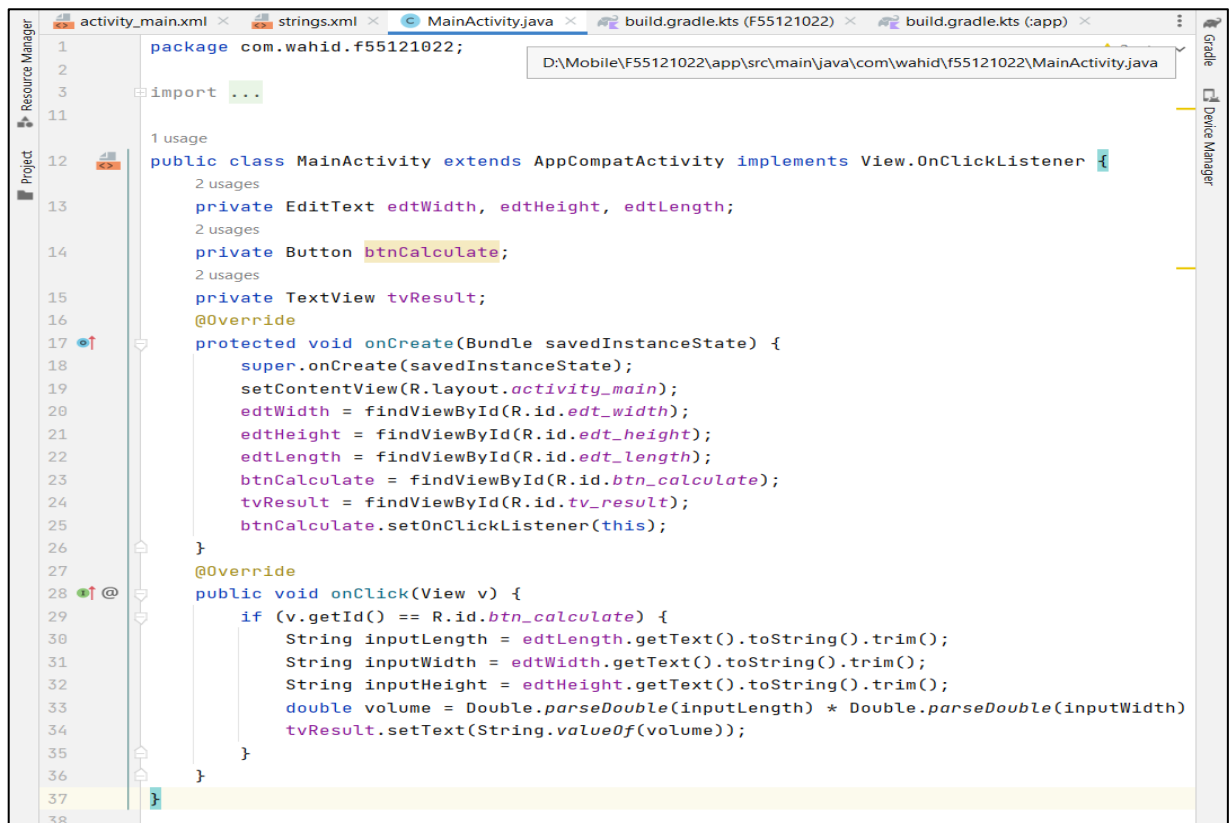
```
2
3 import ...
10
11 1 usage
12 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
13
14     1 usage
15     private EditText edtWidth, edtHeight, edtLength;
16     2 usages
17     private Button btnCalculate;
18     1 usage
19     private TextView tvResult;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25         edtWidth = findViewById(R.id.edt_width);
26         edtHeight = findViewById(R.id.edt_height);
27         edtLength = findViewById(R.id.edt_length);
28         btnCalculate = findViewById(R.id.btn_calculate);
29         tvResult = findViewById(R.id.tv_result);
30         btnCalculate.setOnClickListener(this);
31     }
32
33     @Override
34     public void onClick(View v) {
35
36     }
37
38 }
```

6. Maka secara otomatis akan ada penambahan metode `onClick` di kelas *MainActivity*. Setelah itu, tambahkan kode berikut ke dalam metode `onClick`:



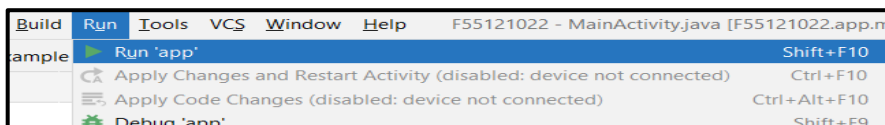
```
28
29 @Override
30 public void onClick(View v) {
31     if (v.getId() == R.id.btn_calculate) {
32         String inputLength = edtLength.getText().toString().trim();
33         String inputWidth = edtWidth.getText().toString().trim();
34         String inputHeight = edtHeight.getText().toString().trim();
35         boolean isEmptyFields = false;
36     }
37 }
38 }
```

7. Akhirnya kelas *MainActivity* akan memiliki kode seperti berikut ini:

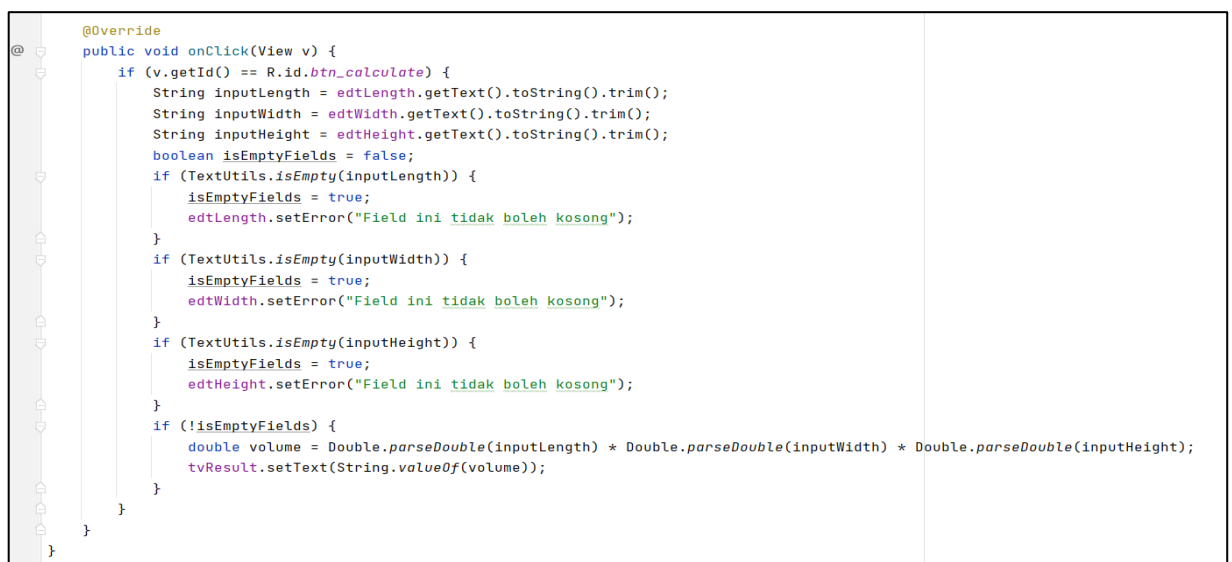


```
1 package com.wahid.f55121022;
2
3 import ...
4
5 1 usage
6
7 12 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
8     2 usages
9     13 private EditText edtWidth, edtHeight, edtLength;
10    2 usages
11    14 private Button btnCalculate;
12    2 usages
13    15 private TextView tvResult;
14    16 @Override
15    17 protected void onCreate(Bundle savedInstanceState) {
16    18     super.onCreate(savedInstanceState);
17    19     setContentView(R.layout.activity_main);
18    20     edtWidth = findViewById(R.id.edt_width);
19    21     edtHeight = findViewById(R.id.edt_height);
20    22     edtLength = findViewById(R.id.edt_length);
21    23     btnCalculate = findViewById(R.id.btn_calculate);
22    24     tvResult = findViewById(R.id.tv_result);
23    25     btnCalculate.setOnClickListener(this);
24    26 }
25    27 @Override
26    28 public void onClick(View v) {
27    29     if (v.getId() == R.id.btn_calculate) {
28    30         String inputLength = edtLength.getText().toString().trim();
29    31         String inputWidth = edtWidth.getText().toString().trim();
30    32         String inputHeight = edtHeight.getText().toString().trim();
31    33         double volume = Double.parseDouble(inputLength) * Double.parseDouble(inputWidth)
32    34         tvResult.setText(String.valueOf(volume));
33    35     }
34    36 }
35    37
36    38
```

8. Setelah selesai, silakan jalankan aplikasi dengan memilih menu *Run* → *Run 'app'* dari *menu bar*.

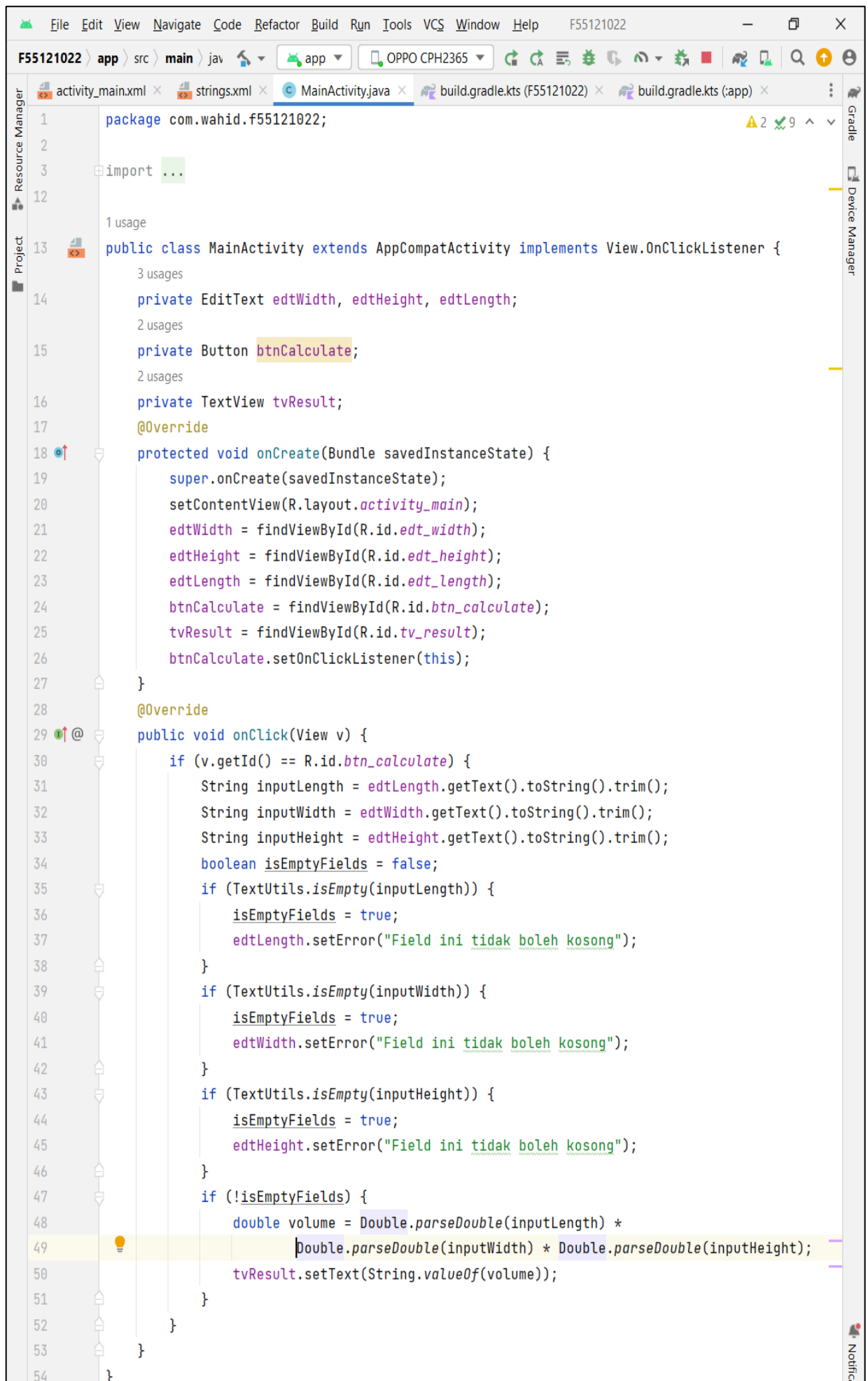


9. Silakan masukkan nilai “panjang”, “lebar”, dan “tinggi” kemudian tekan tombol “Hitung” dan hasilnya akan ditampilkan di objek *textview tvResult*. Namun masih ada sedikit masalah di sini, yaitu Anda tetap melakukan proses perhitungan walaupun salah satu nilainya kosong. Hal ini akan menyebabkan aplikasi *force close* karena perhitungan tidak dapat diproses. Maka untuk mengatasinya Anda akan menggunakan percabangan untuk mengecek apakah masing-masing *EditText* kosong atau tidak. Silakan buka kembali kelas *MainActivity*. Tambahkan kode berikut ke dalam metode *onClick* sebelum melakukan perhitungan.



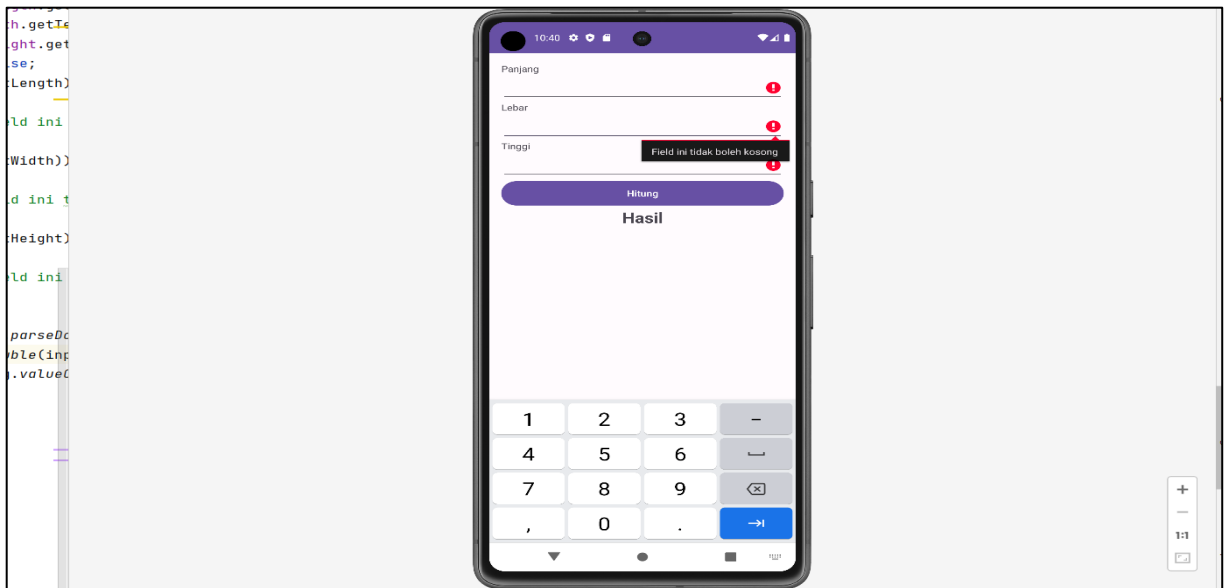
```
@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_calculate) {
        String inputLength = edtLength.getText().toString().trim();
        String inputWidth = edtWidth.getText().toString().trim();
        String inputHeight = edtHeight.getText().toString().trim();
        boolean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

10. Akhirnya kelas *MainActivity* akan memiliki kode seperti berikut ini:

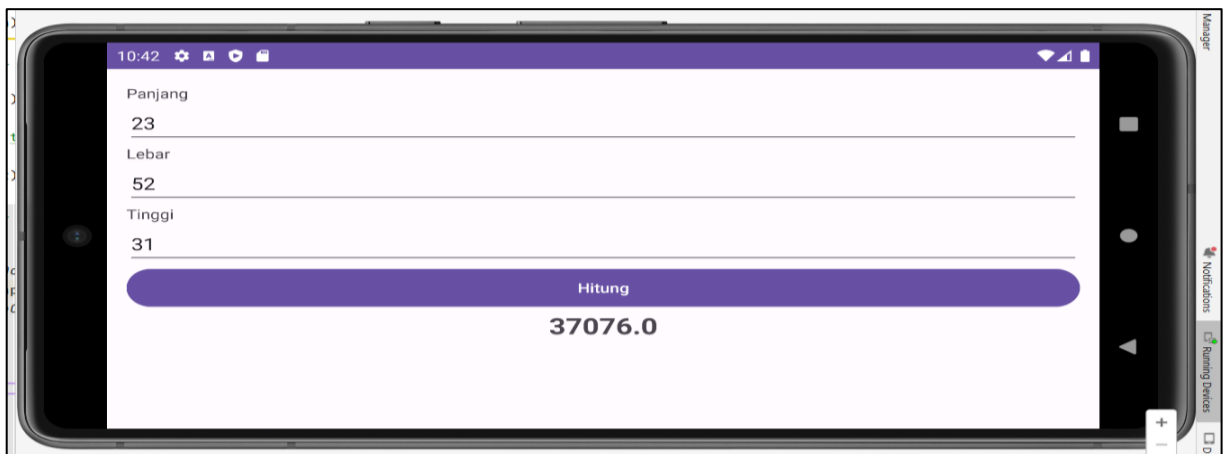


```
1 package com.wahid.f55121022;
2
3 import ...
4
5 1 usage
6
7 13 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
8     3 usages
9     private EditText edtWidth, edtHeight, edtLength;
10    2 usages
11    private Button btnCalculate;
12    2 usages
13    private TextView tvResult;
14    @Override
15    protected void onCreate(Bundle savedInstanceState) {
16        super.onCreate(savedInstanceState);
17        setContentView(R.layout.activity_main);
18        edtWidth = findViewById(R.id.edt_width);
19        edtHeight = findViewById(R.id.edt_height);
20        edtLength = findViewById(R.id.edt_length);
21        btnCalculate = findViewById(R.id.btn_calculate);
22        tvResult = findViewById(R.id.tv_result);
23        btnCalculate.setOnClickListener(this);
24    }
25    @Override
26    public void onClick(View v) {
27        if (v.getId() == R.id.btn_calculate) {
28            String inputLength = edtLength.getText().toString().trim();
29            String inputWidth = edtWidth.getText().toString().trim();
30            String inputHeight = edtHeight.getText().toString().trim();
31            boolean isEmptyFields = false;
32            if (TextUtils.isEmpty(inputLength)) {
33                isEmptyFields = true;
34                edtLength.setError("Field ini tidak boleh kosong");
35            }
36            if (TextUtils.isEmpty(inputWidth)) {
37                isEmptyFields = true;
38                edtWidth.setError("Field ini tidak boleh kosong");
39            }
40            if (TextUtils.isEmpty(inputHeight)) {
41                isEmptyFields = true;
42                edtHeight.setError("Field ini tidak boleh kosong");
43            }
44            if (!isEmptyFields) {
45                double volume = Double.parseDouble(inputLength) *
46                    Double.parseDouble(inputWidth) * Double.parseDouble(inputHeight);
47                tvResult.setText(String.valueOf(volume));
48            }
49        }
50    }
51 }
52
53 }
54 }
```

11. Jalan kembali aplikasi Anda dengan memilih menu “Run → Run app” atau *shortcut Shift + F10*. Cobalah langsung menekan tombol “HITUNG” tanpa mengisi *EditText*, maka aplikasi Anda tidak akan *force close* dan akan muncul peringatan bahwa “Field ini tidak boleh kosong”.



12. Ketika nilai *volume* sudah dihitung dan kemudian terjadi pergantian orientasi (*portrait-landscape*) pada peranti, maka hasil perhitungan tadi akan hilang. Hal ini karena di dalam *Android*, jika melakukan pergantian orientasi, *Android* akan memanggil fungsi *onCreate* kembali, sehingga data akan kembali menjadi seperti semula.



13. Untuk mengatasinya, tambahkan metode *onSaveInstanceState* pada *MainActivity* dan sesuaikan seperti berikut:

```
17 1 usage
18 private static final String STATE_RESULT = "state_result";
19 @Override
20 protected void onCreate(Bundle savedInstanceState) {
21     super.onCreate(savedInstanceState);
22     setContentView(R.layout.activity_main);
23     edtWidth = findViewById(R.id.edt_width);
24     edtHeight = findViewById(R.id.edt_height);
25     edtLength = findViewById(R.id.edt_length);
26     btnCalculate = findViewById(R.id.btn_calculate);
27     tvResult = findViewById(R.id.tv_result);
28     btnCalculate.setOnClickListener(this);
29 }
30
31 @Override
32 protected void onSaveInstanceState(@NonNull Bundle outState) {
33     super.onSaveInstanceState(outState);
34     outState.putString(STATE_RESULT, tvResult.getText().toString());
35 }
```

14. Kemudian tambahkan juga beberapa baris berikut pada baris terakhir metode *onCreate*.

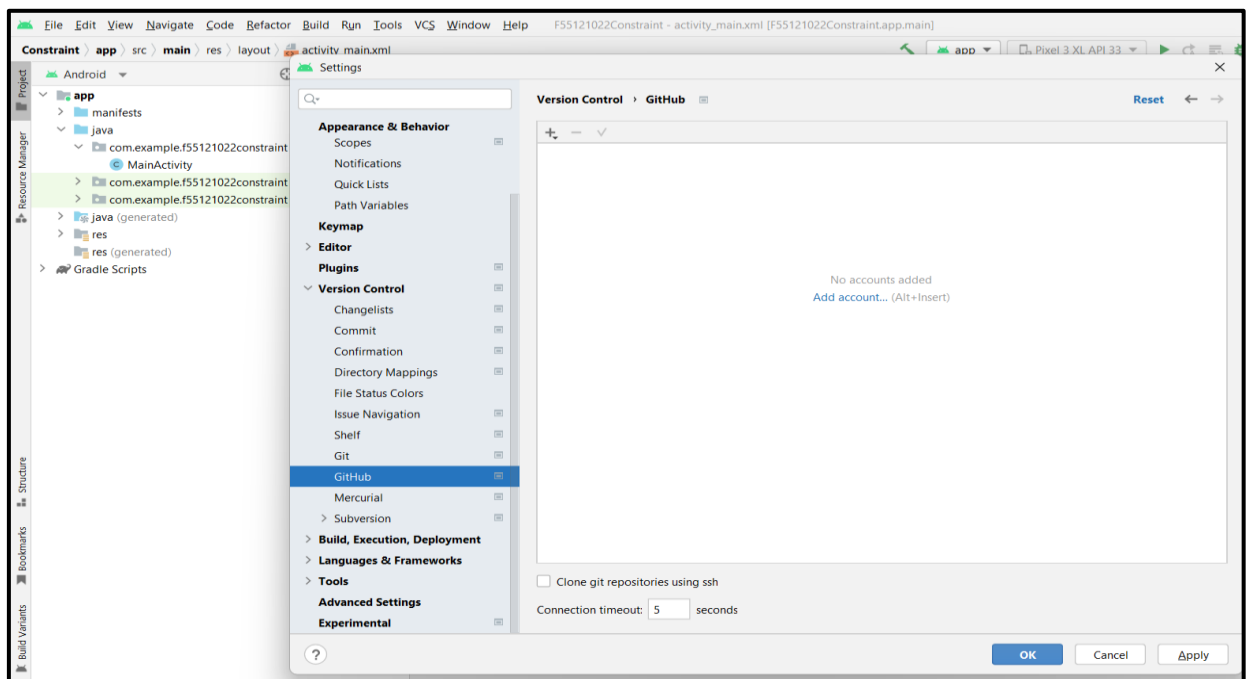
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    edtWidth = findViewById(R.id.edt_width);
    edtHeight = findViewById(R.id.edt_height);
    edtLength = findViewById(R.id.edt_length);
    btnCalculate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

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

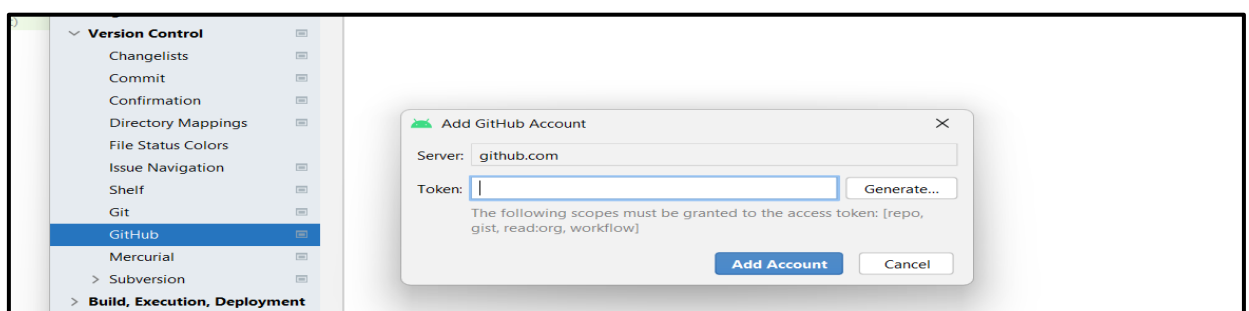
15. Silakan jalankan kembali aplikasinya. Ulangi proses perhitungan seperti sebelumnya. Kemudian ganti orientasi peranti Anda. Jika sudah benar maka hasil perhitungan tidak akan hilang.

D. Menghubungkan antara *android studio* ke *Github*

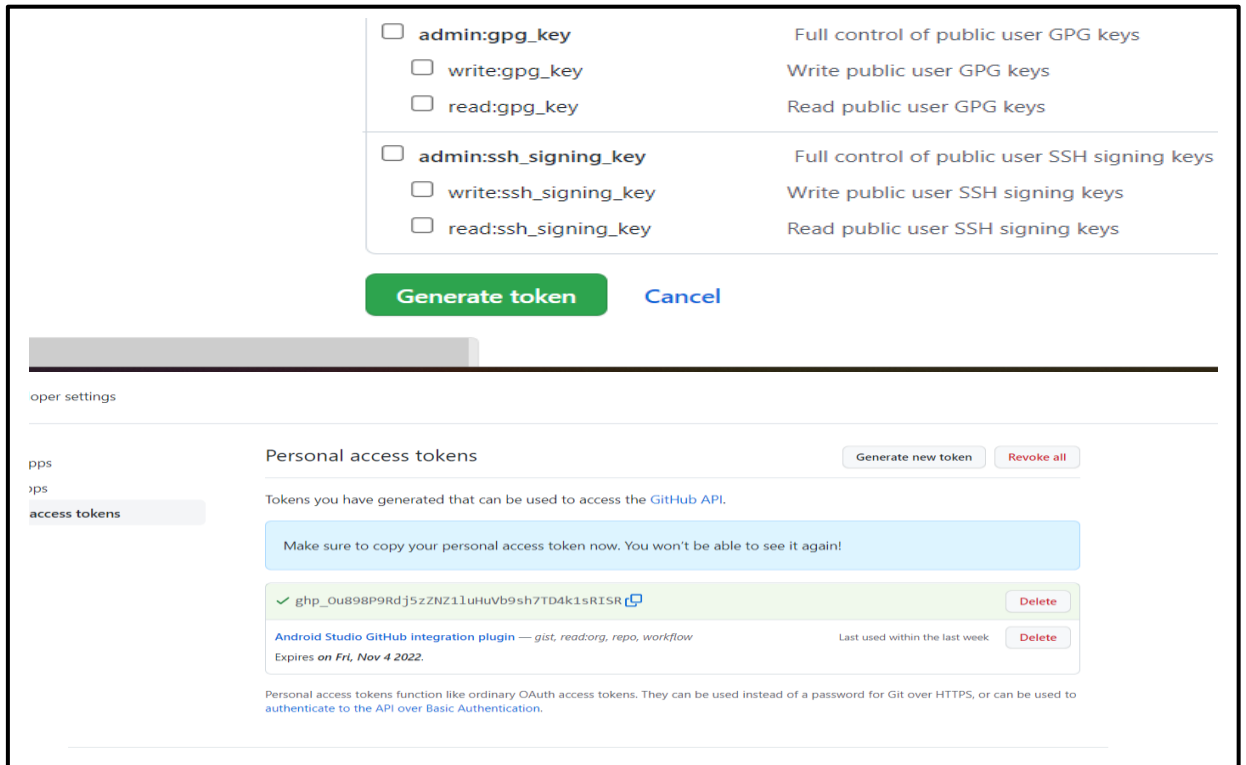
1. Pada jendela “Setting”, pilih bagian *version control* lalu pilih *github*.



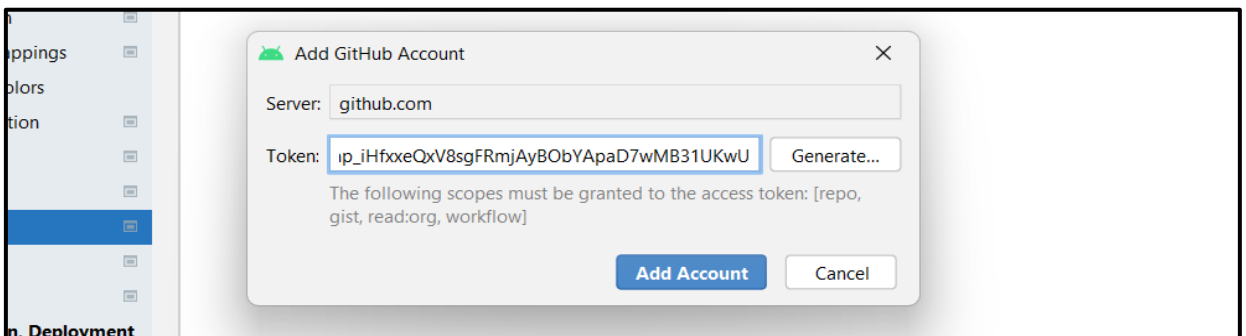
2. Klik *add account* lalu klik *generate* untuk mengambil token dari *github* anda.



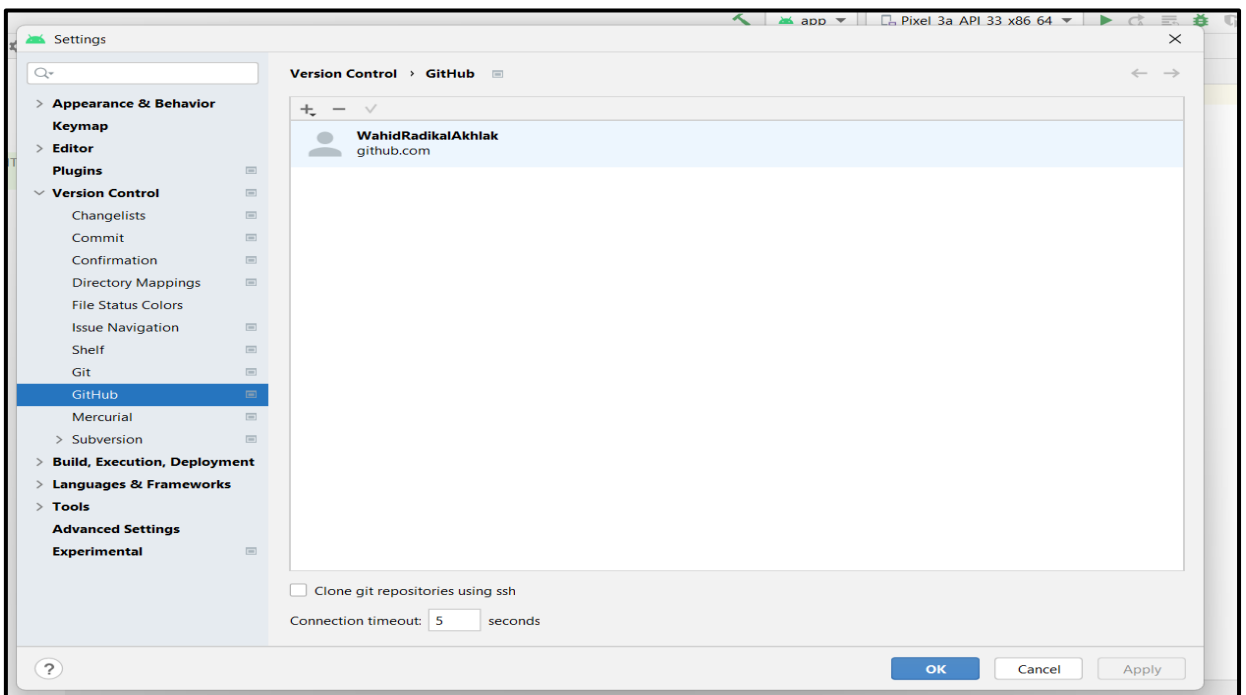
3. Setelah tersambungkan dengan *github* di *web* klik “*generate*” token untuk membuat token. Kemudian salinlah token yang tersedia di *github*.



4. Kemudian tempel token yang sudah disalin, dan klik *add account*.

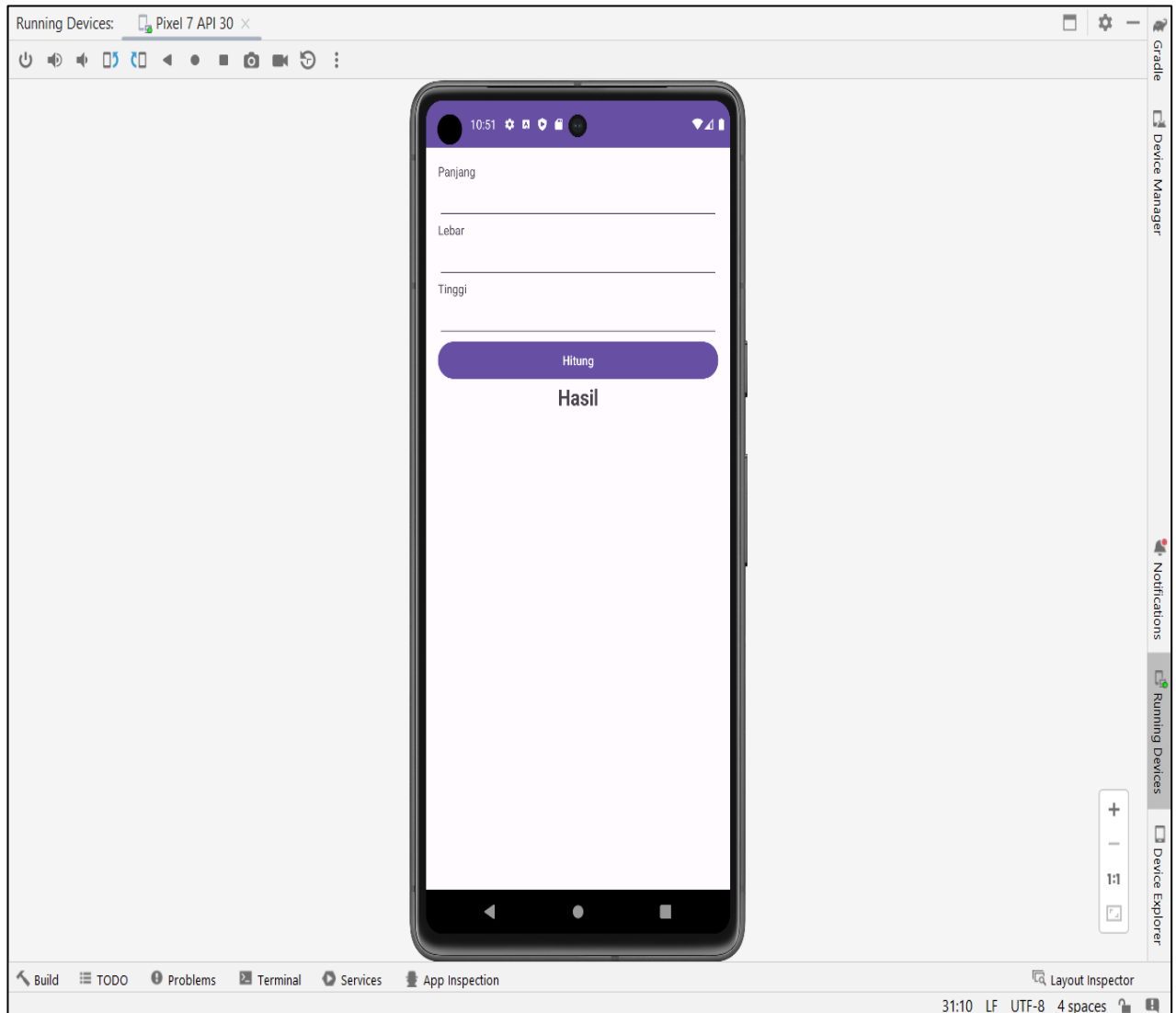


5. Jika akun sudah ada klik *apply* dan ok.

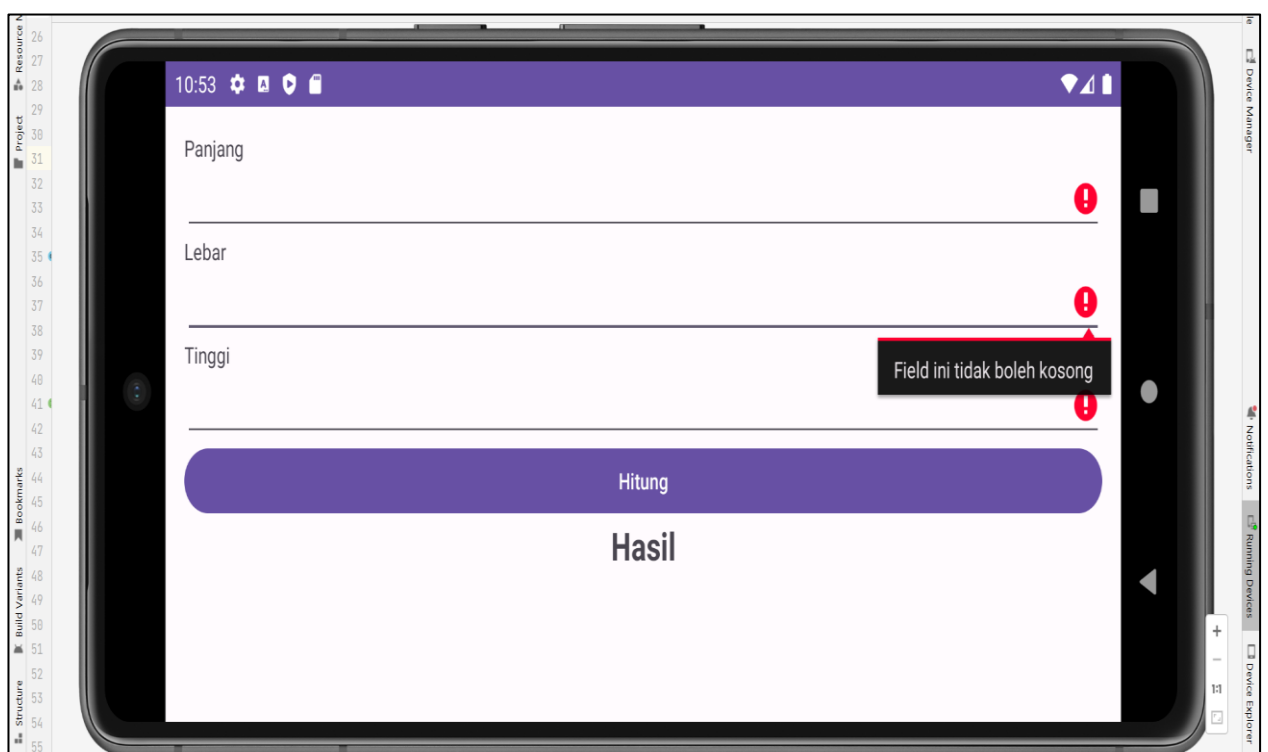


V. HASIL PERCOBAAN

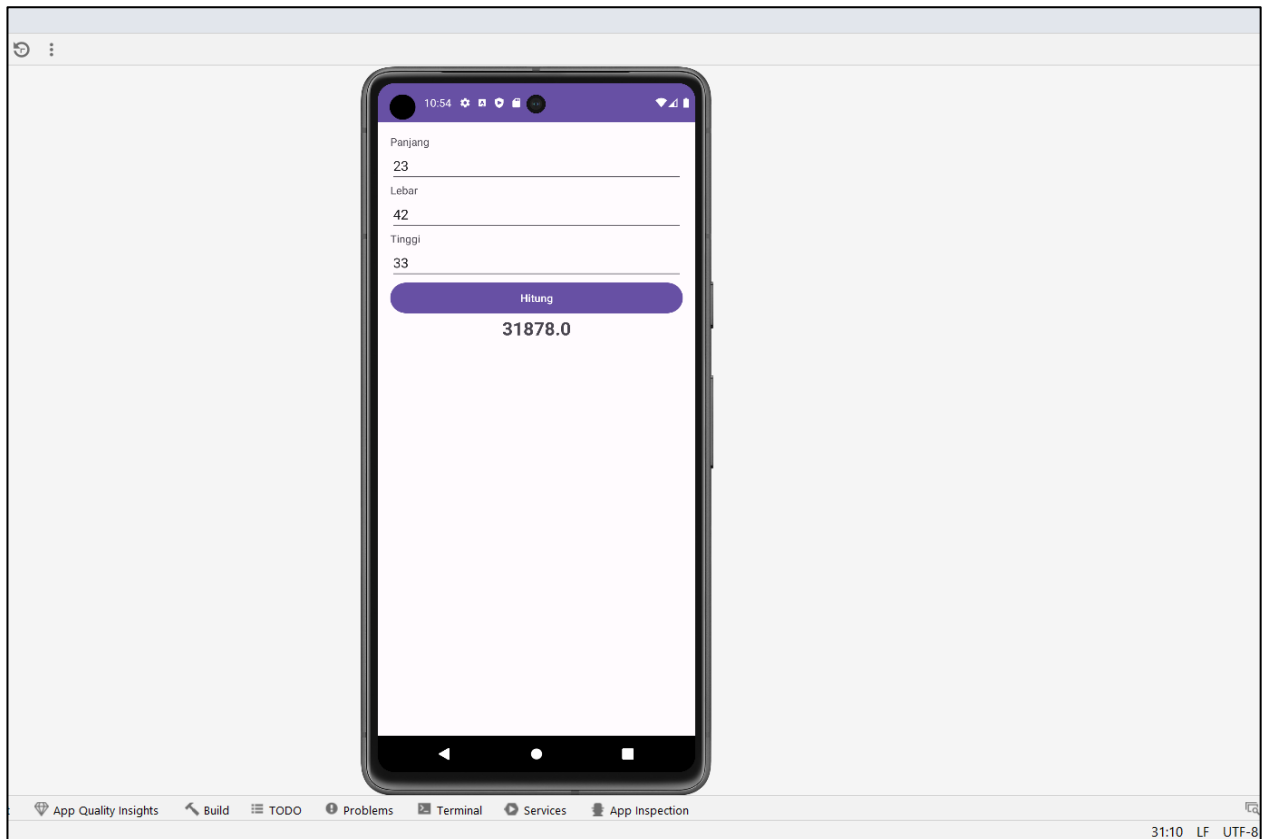
A. Tampilan Kosong



B. Tampilan *Landscape* dan juga peringatan.



C. Tampilan data dengan hasil



D. Tampilan *Repository* pada *Github*.

