

# Pemrograman Aplikasi Mobile berbasis Android

## Modul 1 : Activity

### Referensi

- Android Application Development Guide (<http://developer.android.com/guide/index.html>)
- Android Application Development Training (<http://developer.android.com/training/index.html>)
- Buku : Professional Android 4 Application Development by Reto Meier (PA4AD)

### Pembuatan Aplikasi

Apapun platformnya konsep dasar dalam mengembangkan aplikasi adalah bagaimana menampilkan informasi di layar, melakukan proses yang dibutuhkan dan berpindah dari satu halaman ke halaman lain dengan atau tanpa data. Apabila kita bandingkan antara Web App Dev dengan Android App Dev dengan menggunakan Android Studio, adalah sebagai berikut.

Komponen	Web	Android
Halaman	Page	Activity dan Fragment
UI komponen	HTML, CSS	XML
Logic	PHP, ASP, JSP, RUBY etc	Java
Berpindah halaman	Tag (Hyperlink)	Intent

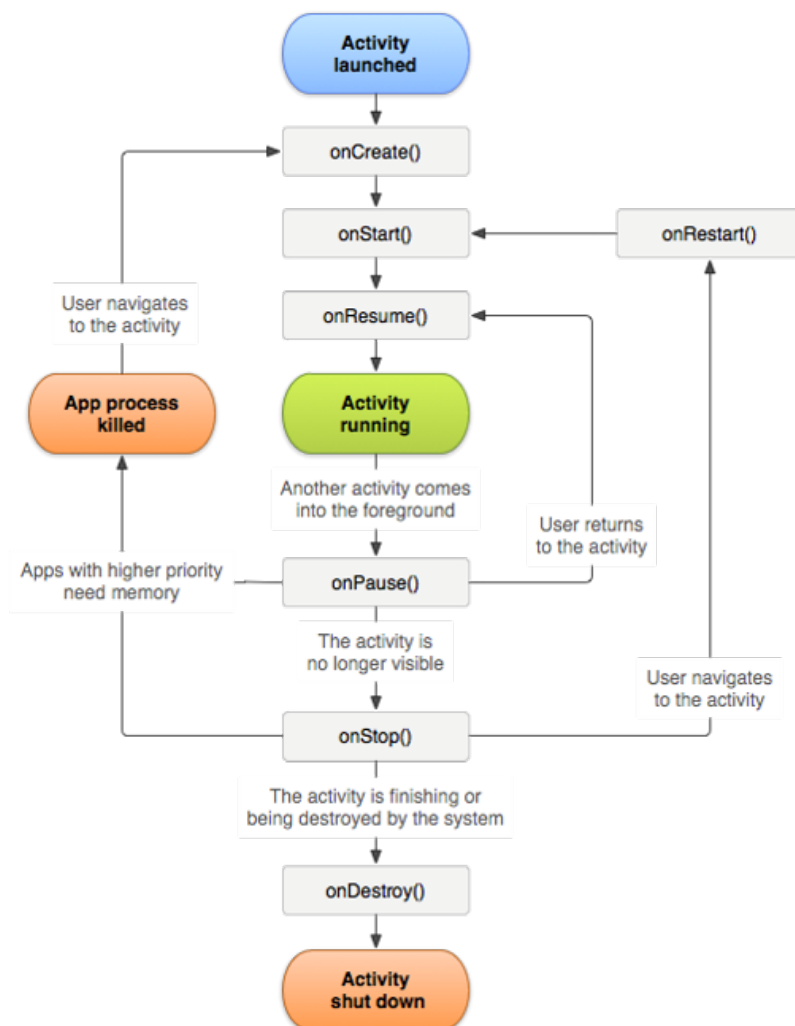
### AndroidManifest.xml dan Gradle (Module app)

- File yang akan mengontrol semua komponen, level SDK dan permission yang ada di aplikasi
- File yang akan mengontrol semua dependency library yang digunakan oleh Aplikasi

### Activity

- Activity adalah Komponen android yang menyediakan tampilan didalam sebuah aplikasi dimana pengguna akan berinteraksi dengannya. Misal layar list chat pada aplikasi whatsapp android, user timeline yang ada di aplikasi twitter dan facebook anda, layar inbox pada aplikasi gmail anda dan lain sebagainya.
- Setiap class Java dinyatakan sebagai sebuah Activity jika class tersebut meng-extends pada class Activity
- Activity umumnya memiliki sebuah tampilan dalam format layout .xml
- Activity memiliki lifecycle di memory dimana para developer android harus peduli dan tau kapan sebuah Activity diciptakan (create) hingga dimatikan (destroy) dari memory

- Setiap Activity yang dimiliki dan akan digunakan di aplikasi harus di registrasikan di file AndroidManifest.xml
- Activity menganut konsep Stack : Last in First Out (LIFO) yang mana setiap activity yang terakhir diciptakan atau ditampilkan dilayar akan keluar pertama kali dari stack jika Pengguna menekan tombol:
  - Physical back button, atau
  - Back button pada Action Bar Aplikasi



Contoh Activity Lifecycle dari sebuah Aplikasi

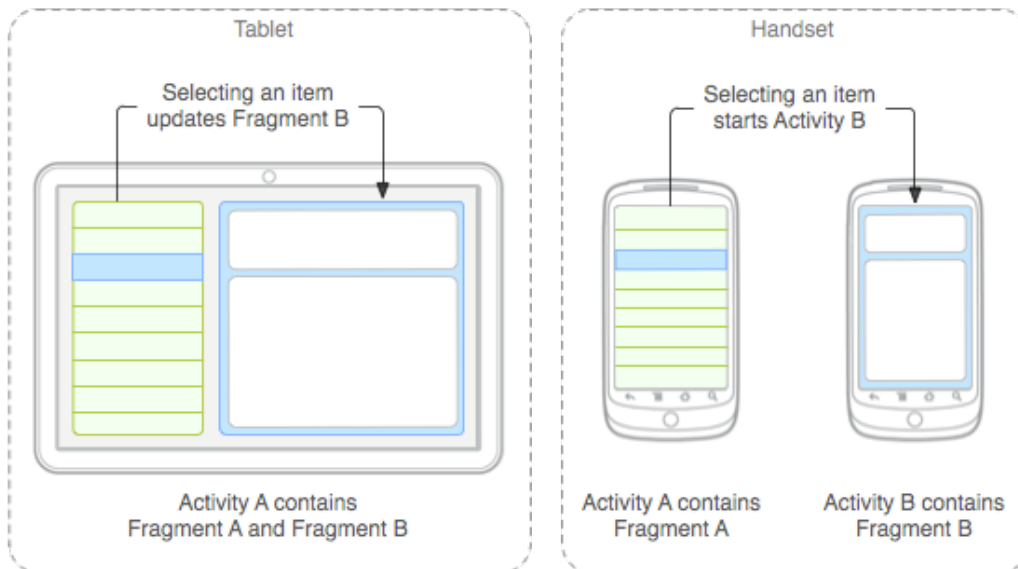
Baca Lebih Lanjut

<http://developer.android.com/guide/components/activities.html> atau PA4AD page 86 s/d 93

## Fragments

- Fragment adalah bagian user interface dalam sebuah Activity.
- Fragment menjadi sebuah tampilan UI dalam Android dipecah-pecah menjadi kecil-kecil dan bisa mempermudah untuk digunakan ulang (reusable)
- Satu Activity dalam sebuah aplikasi Android dapat menampung lebih dari 1 fragment.

- Tidak seperti Activity, fragment tidak perlu diregistrasikan di file AndroidManifest.xml cukup dengan menggunakan fragment transaction
- Satu class Java dinyatakan sebagai sebuah fragment jika class tersebut meng-extends dari class Fragment
- Sama dengan Activity, Fragment juga memiliki Lifecycle
- Analogi yang paling mendekati fragment adalah komponen Frame pada pengembangan aplikasi berbasis Web



Penggunaan fragment banyak diimplementasikan pada aplikasi-aplikasi yang membutuhkan fleksibilitas pada tampilan misal pada aplikasi berita. Dengan menggunakan Fragment proses perubahan pada tampilan landscape di layar tablet yang besar akan lebih proporsional dan akan menyesuaikan jika berubah ke tampilan portrait pada layar smartphone umum.

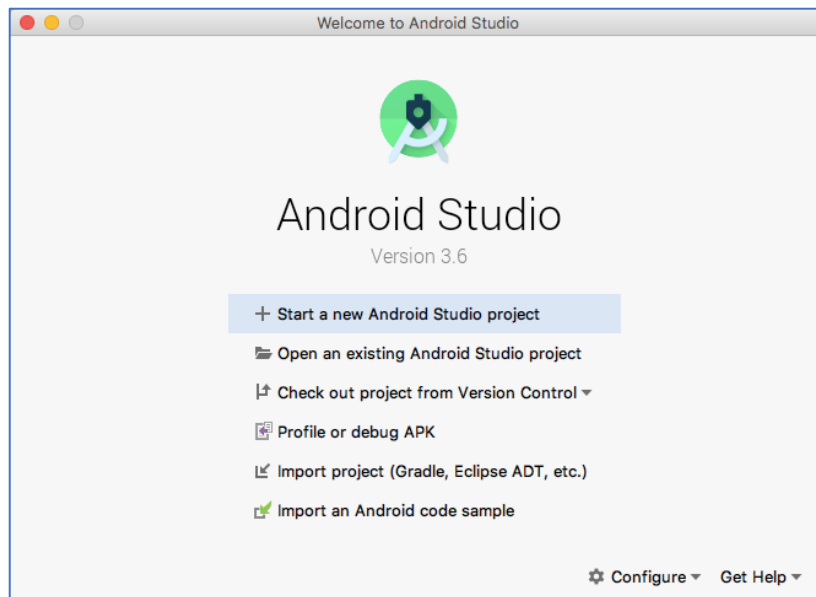
Baca lebih lanjut

<http://developer.android.com/guide/components/fragments.html> atau PA4AD : 114 s/d 126

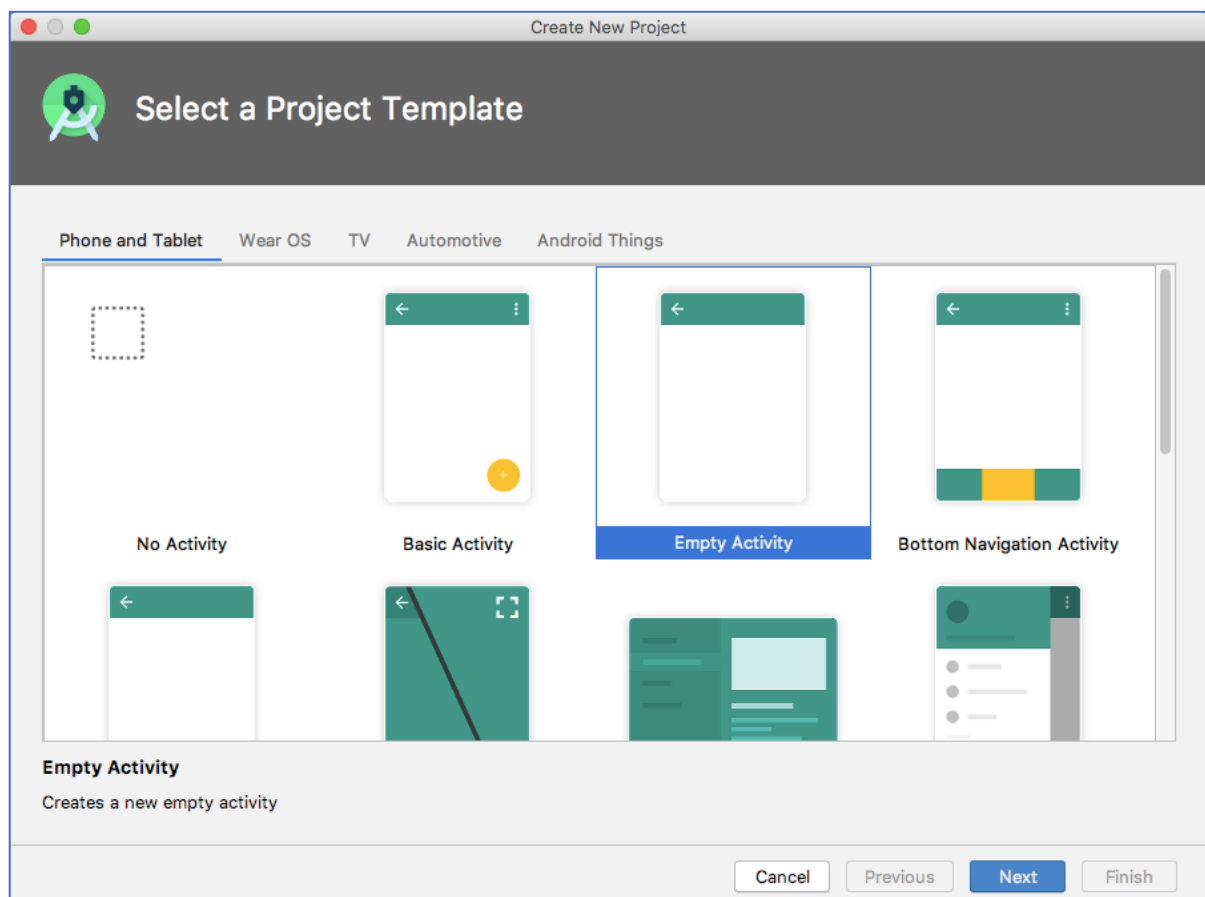
### Practice Session #1 :

*Pengembangan Aplikasi Hitung Luas untuk Implementasi Activity pada Pengembangan Aplikasi Android*

1. Buat Project dengan **Start a new Android Studio Project**



2. Pilih **Empty Activity** dari template yang disediakan oleh Android Studio, kemudian klik Next.



3. Lalu isikan data seperti pada gambar berikut. Anda dapat melakukan penyesuaian jika perlu dengan ketentuan:

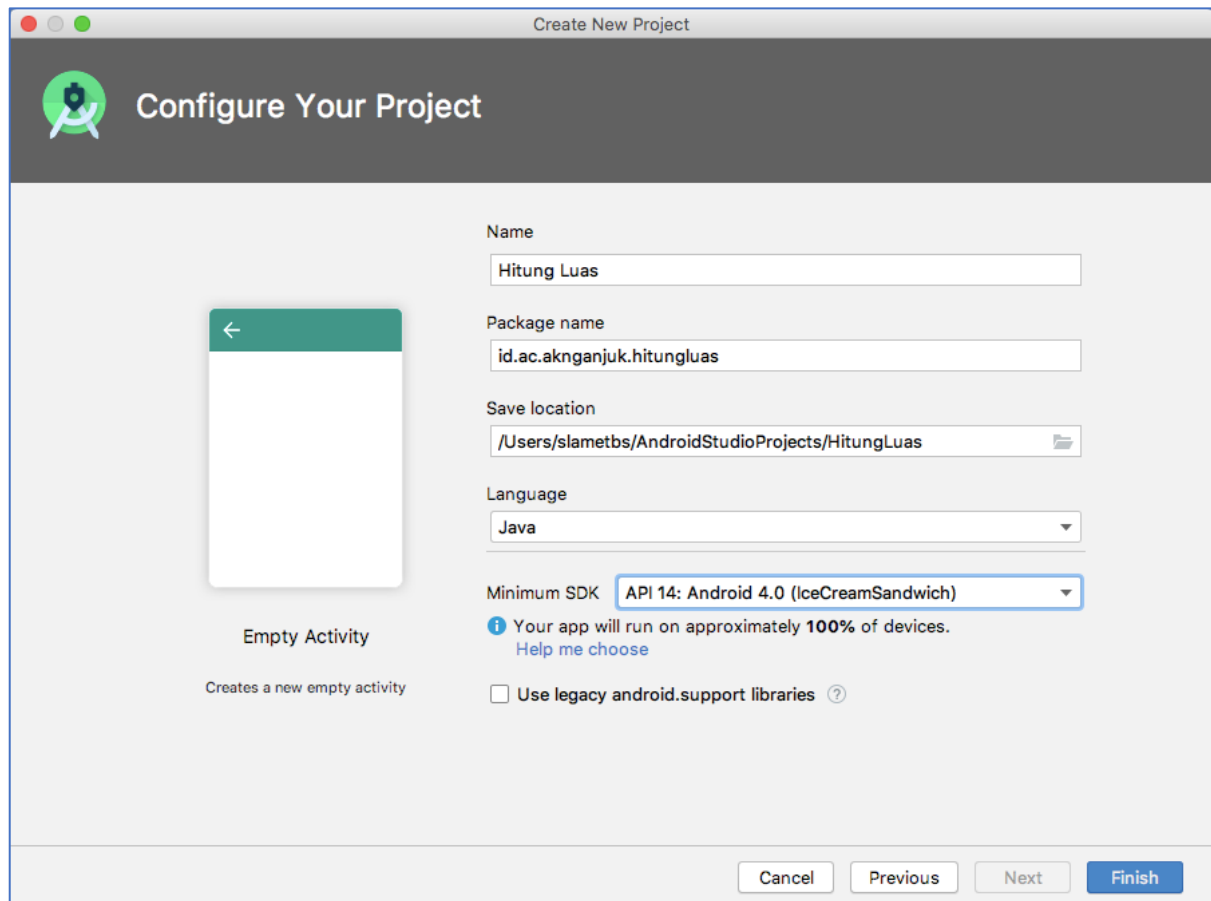
**Nama**, merupakan nama dari aplikasi yang akan dibuat

**Package name**, merupakan nama aplikasi (tanpa spasi) ditambah nama domain yang ditulis terbalik (pada contoh ini: *aknganjuk.ac.id* ditulis menjadi *id.ac.aknganjuk*).

**Save location**, adalah lokasi penyimpanan file-file project.

**Language**, *default* Bahasa pemrograman saat ini adalah Kotlin, tapi kita masih bias menggunakan Java.

**Minimum SDK**, adalah versi Android minimum agar aplikasi yang dibuat dapat berjalan dengan baik. Pada contoh ini digunakan versi Android 4.0 dengan cakupan 100% perangkat Android yang beredar saat ini.



Kemudian Klik Finish. Biarkan gradle menggenerate project kita.

4. Pada **activity\_main.xml** (folder res/layout) silakan ketikkan kode seperti di bawah ini, untuk menghasilkan layout. Anda bisa membuat layout dengan sistem *drag-n-drop* apabila dikehendaki.

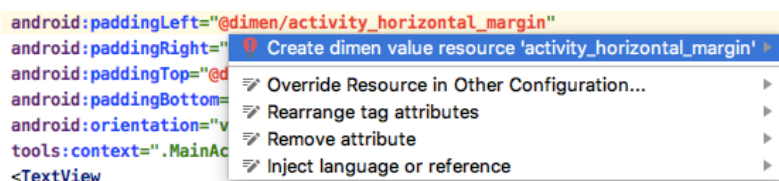
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:text="Panjang"
```

```

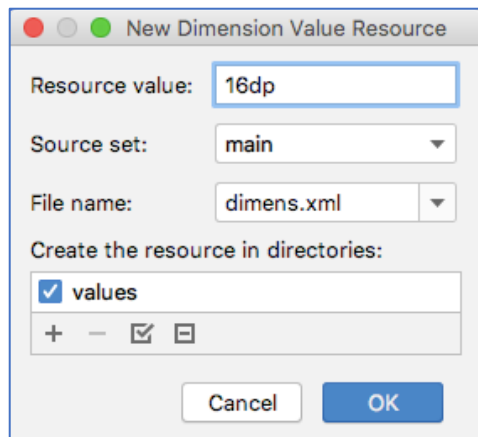
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"/>
<EditText
    android:id="@+id/edt_panjang"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="0"
    android:inputType="numberDecimal"
    android:layout_marginBottom="5dp"/>
<TextView
    android:text="Lebar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"/>
<EditText
    android:id="@+id/edt_lebar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="0"
    android:inputType="numberDecimal"
    android:layout_marginBottom="5dp"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/btn_hitung"
    android:text="Hitung"
    android:layout_marginBottom="8dp"/>
<TextView
    android:text="Hasil"
    android:id="@+id/txt_luas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:textSize="18sp"
    android:textStyle="bold"/>
</LinearLayout>

```

Jika menjumpai *error* yang ditandai dengan teks berwarna merah atau digarisbawahi, Anda dapat memposisikan kursor pada teks tersebut kemudian menekan kombinasi tombol **Alt+Enter** untuk menampilkan alternatif koreksi yang dapat dilakukan. Contoh pada teks @dimen tersebut akan diberi saran seperti berikut ini:



Yang akan dilanjutkan dengan form berikut, isikan nilai **16dp** pada kolom **Resource value**:



Lakukan hal yang sama hingga semua error diperbaiki.

5. Pada **MainActivity.java** silakan lengkapi kode seperti di bawah ini untuk melakukan Activity pada layout yang telah kita buat pada langkah sebelumnya.

```
public class MainActivity extends AppCompatActivity {
    private EditText edtPanjang, edtLebar;
    private Button btnHitung;
    private TextView txtLuas;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setTitle("Hitung Luas Persegi Panjang");

        edtPanjang = (EditText)findViewById(R.id.edt_panjang);
        edtLebar = (EditText)findViewById(R.id.edt_lebar);
        btnHitung = (Button)findViewById(R.id.btn_hitung);
        txtLuas = (TextView)findViewById(R.id.txt_luas);

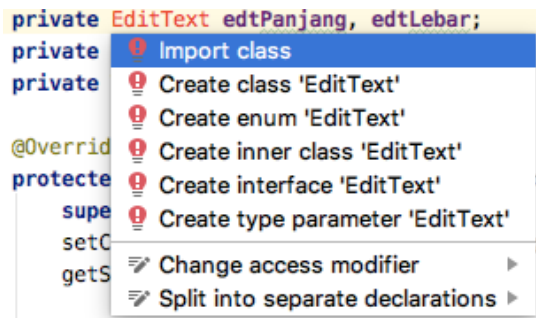
        btnHitung.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String panjang = edtPanjang.getText().toString().trim();
                String lebar = edtLebar.getText().toString().trim();

                double p = Double.parseDouble(panjang);
                double l = Double.parseDouble(lebar);

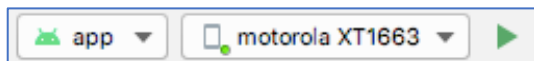
                double luas = p * l;

                txtLuas.setText("Luas : "+luas);
            }
        });
    }
}
```

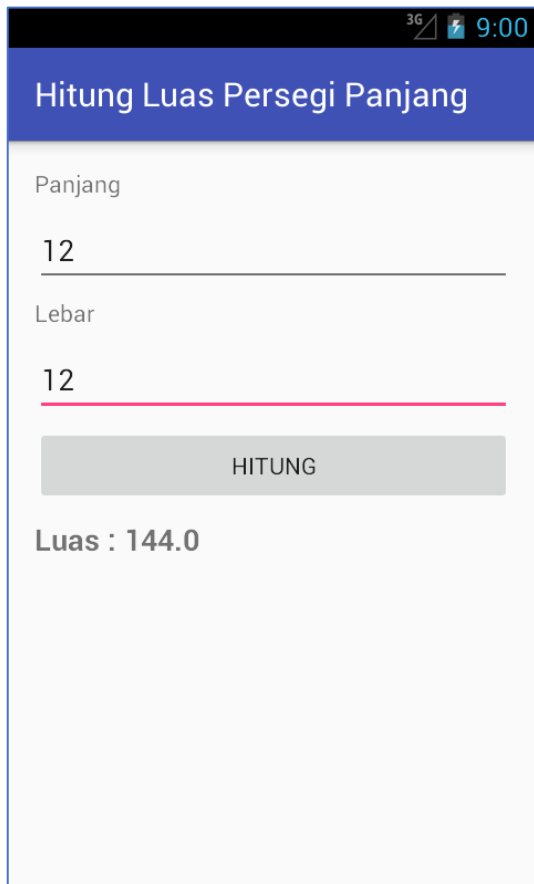
Sama halnya pada Layout, jika dijumpai error (teks berwarna merah), lakukan hal yang sama hingga semua error diperbaiki.



6. Silakan jalankan aplikasi dengan tombol **Run** (segitiga hijau) ke *Android Virtual Device* atau perangkat Android yang dimiliki atau export ke apk untuk instal secara manual di device.



Tampilannya akan tampak seperti berikut.





**Tambahan** tentang validasi. Ada beberapa cara yang dapat digunakan untuk memvalidasi sebuah inputan yang berupa nilai.

- if - else

```
if(TextUtils.isEmpty(edtPanjang.getText()) || TextUtils.isEmpty(edtLebar.getText())){
    Toast.makeText(this,"Nilai yang dimasukkan tidak boleh kosong",Toast.LENGTH_SHORT).show();
}else {
    //Do Something
}
```

- try-catch

```
try{
    double panjang = Double.valueOf(edtPanjang.getText().toString());
    double lebar = Double.valueOf(edtLebar.getText().toString());
}catch (NumberFormatException e){
    Toast.makeText(this,e.getMessage(),Toast.LENGTH_SHORT).show();
}
```

## Latihan

Modifikasi aplikasi Anda dengan menambahkan validasi agar inputan hanya diisi data yang sesuai saja.