

MOBILE PROGRAMMING



Oleh :

CHALIFA CHAZAR



MATERI 6

PENGGUNAAN ACTIVITY

Tujuan:

Mahasiswa memahami siklus hidup suatu activity dan mampu menerapkan penggunaan intent dalam pembuatan suatu aplikasi Android.

Pustaka:

Abdul Kadir, 2013: Pemograman Aplikasi Android. Andi Yogyakarta.

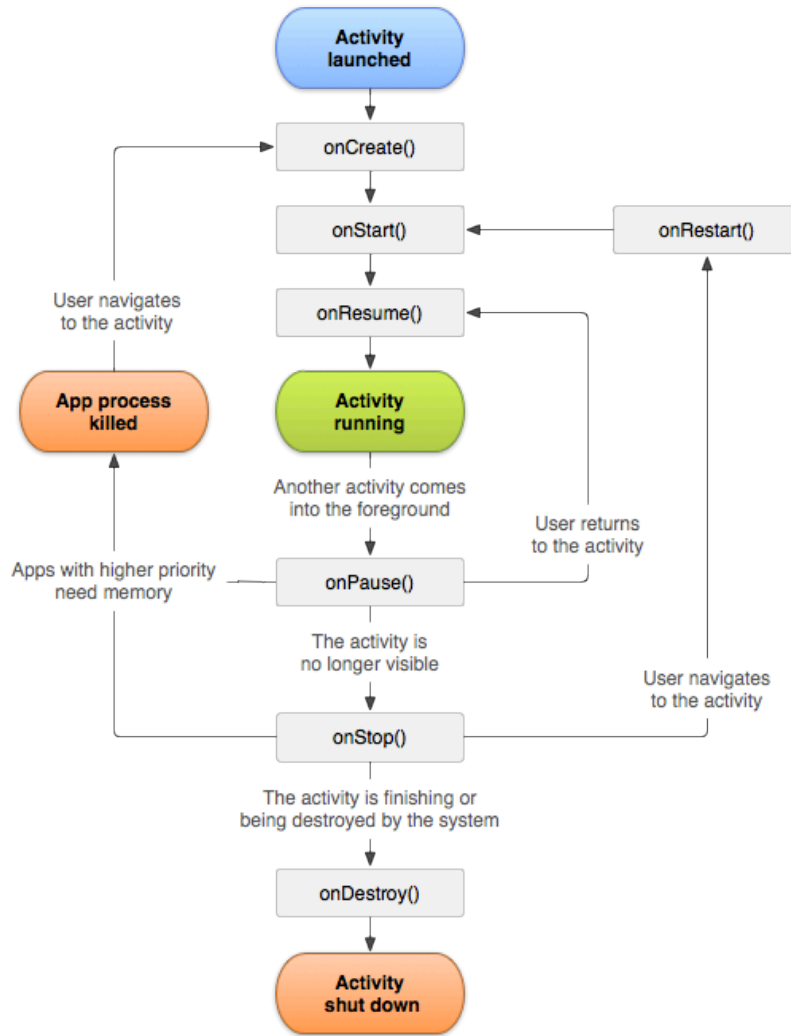
[Http://www.gits.co.id](http://www.gits.co.id)

1. Pengertian Activity

Activity merupakan *public class* pada aplikasi android, dimana **Activity** merupakan bagian terpenting dari sebuah siklus aplikasi. **Activity** merepresentasikan satu layar tampilan UI (*user interface*) yang mengatur semua hal yang dilakukan *user*. Misalnya suatu aplikasi *messenger*, memiliki **Activity** yang menunjukkan daftar teman, dan **Activity** lainnya seperti menulis pesan (*message*), menambah teman, dan lain-lain. Apabila sebuah aplikasi android memiliki beberapa halaman UI yang saling berinteraksi, berarti aplikasi tersebut memiliki beberapa **Activity** yang saling berinteraksi.

Kebanyakan paradigma dalam pemograman lain yang bermula dari *method main()*. Pada pemograman android, program dimulai dengan *method callback onCreate()*. Berikut ini adalah urutan *method callback* dari mulai **Activity** sampai berakhirnya **Activity** dapat dilihat pada diagram **Activity lifecycle**.





Method callback mendefinisikan suatu **event**. Tetapi kita tidak perlu mengimplementasikan semua *method callback*. Namun, penting untuk memahami masing-masing *method callback*, sehingga aplikasi kita dapat memenuhi harapan dari pengguna.

Method	Deskripsi
<code>onCreate()</code>	Method ini pertama kali dipanggil ketika Activity pertama dimulai
<code>onStart()</code>	Method ini dipanggil ketika Activity mulai terlihat oleh user
<code>onResume()</code>	Method ini dipanggil ketika Activity mulai berinteraksi dengan user
<code>onPause()</code>	Method ini dipanggil ketika Activity berhenti sementara tidak menerima input dari user dan tidak mengeksekusi kode apapun
<code>onStop()</code>	Method ini dipanggil ketika Activity sudah tidak terlihat pada user
<code>onDestroy()</code>	Method ini dipanggil ketika sebuah Activity dimatikan
<code>onRestart()</code>	Method ini dipanggil setelah Activity berhenti dan ditampilkan ulang oleh user





Beberapa poin dasar yang diharapkan dari aplikasi android, dalam mengimplementasikan **Activity lifecycle** yang benar, antara lain:

- ★ Tidak *crash* ketika menerima panggilan masuk atau membuka aplikasi lain ketika sedang menggunakan sebuah aplikasi
- ★ Tidak mengonsumsi power berlebihan (baterai), saat user tidak menggunakan atau mengaktifkan sebuah aplikasi
- ★ Tidak kehilangan data user, ketika user menutup aplikasi dan membukanya pada lain waktu
- ★ Tidak *crash* atau kehilangan apa yang telah di input, apabila layar diganti orientasinya (*landscape* atau *portrait*)

2. Membuat Activity

Activity merupakan sebuah *class* yang mengelola sebuah UI dari sebuah aplikasi Android. Untuk membuat sebuah **Activity**, dilakukan dengan membuat sebuah *class* yang meng-*extends class Activity*. Pada saat membuat sebuah project baru pada Android Studio, maka akan ada sebuah *class* bernama MainActivity.java yang ter-*generate* secara otomatis. *Class* ini meng-*ekstends class activity* (MainActivity *extends Activity*) sehingga class ini merupakan sebuah **Activity**. **Activity** ini menampilkan sebuah UI dengan memanggil fungsi onCreate(). Di dalam sebuah fungsi, kita dapat menentukan xml yang akan dijadikan UI dengan memanggil fungsi setContentView() dengan parameter berupa halaman xml dari UI, misalnya setContentView (R.layout.activity_main).

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

3. Contoh Penggunaan Activity-Activity Lainnya

Berikut ini kita akan mempelajari sebuah siklus **Activity** lainnya. Ikuti langkah-langkah sebagai berikut.

1. Buka Android Studio, dan buatlah sebuah project baru.





2. Lakukan perubahan pada file MainActivity.java sebagai berikut. Pada file MainActivity.java akan dibuat beberapa *method callback* yang dapat menjelaskan siklus hidup suatu **Activity**. *Method* Log.d() digunakan untuk menghasilkan suatu pesan Log.

```
package example.com.latihan7activity;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    String status = "Android :";
    //method ini dipanggil ketika activity pertama dimulai
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(status, "onCreate() event");
    }

    //method ini dipanggil ketika activity sudah terlihat oleh user
    @Override
    protected void onStart(){
        super.onStart();
        Log.d(status, "onStart() event");
    }

    //method ini dipanggil ketika activity mulai berinteraksi dengan
    user
    @Override
    protected void onResume() {
        super.onResume();
        Log.d(status, "onResume() event");
    }

    //method ini dipanggil ketika activity berhenti sementara
    @Override
    protected void onPause() {
        super.onPause();
        Log.d(status, "onPause() event");
    }
}
```





```
//method ini dipanggil ketika activity sudah tidak terlihat pada user  
@Override  
protected void onStop(){  
    super.onStop();  
    Log.d(status, "onStop() event");  
}
```

```
//method ini dipanggil sebelum activity dimatikan  
@Override  
protected void onDestroy(){  
    super.onDestroy();  
    Log.d(status, "onDestroy() event");  
}
```

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    // Handle action bar item clicks here. The action bar will automatically handle clicks on the Home/Up button, so long as you specify a parent activity in AndroidManifest.xml.  
    int id = item.getItemId();  
  
    //noinspection SimplifiableIfStatement  
    if (id == R.id.action_settings) {  
        return true;  
    }  
  
    return super.onOptionsItemSelected(item);  
}  
}
```

Sekarang coba jalankan aplikasi untuk melihat siklus hidup suatu activity. Perhatikan pesan log dari LogCat di Android Studio.

Berikut ini adalah tampilan pesan log dari LogCat di Android Studio ketika aplikasi dijalankan.





```
logcat ADB logs → Memory → CPU →
09-11 20:21:11.755 19746-19746/example.com.latihan7activity D/Android :: onCreate() event
09-11 20:21:11.755 19746-19746/example.com.latihan7activity D/Android :: onStart() event
09-11 20:21:11.755 19746-19746/example.com.latihan7activity D/Android :: onResume() event
```

Coba klik menu home pada device. Maka LogCat akan menampilkan pesan sebagai berikut.

```
logcat ADB logs → Memory → CPU → Log level: Verbose
09-11 20:21:33.705 19746-19746/example.com.latihan7activity D/Android :: onPause() event
09-11 20:21:34.375 19746-19746/example.com.latihan7activity W/PGA: [19746] egl: eglDestroySurface (0x557b17a0, 0x78a0ac70)
09-11 20:21:34.375 19746-19746/example.com.latihan7activity W/PGA: [19746] egl: eglDestroySurface (0x557b17a0, 0x78a0ac70) ret
```

Jika kita membuka kembali aplikasi, maka LogCat akan menampilkan pesan sebagai berikut.

```
logcat ADB logs → Memory → CPU →
09-11 20:21:35.305 19746-19746/example.com.latihan7activity D/Android :: onStop() event
09-11 20:21:57.785 19746-19746/example.com.latihan7activity D/Android :: onStart() event
09-11 20:21:57.785 19746-19746/example.com.latihan7activity D/Android :: onResume() event
09-11 20:21:57.805 19746-19746/example.com.latihan7activity W/PGA: [19746] egl: eglCreateWindowSurf
09-11 20:21:57.815 19746-19746/example.com.latihan7activity W/PGA: [19746] egl: eglCreateWindowSurf
```

