

# MOBILE PROGRAMMING



Oleh :

CHALIFA CHAZAR



## MATERI 6

### PENGGUNAAN ACTIVITY & INTENT

#### 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**.





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
```

## 4. Pengertian Intent

Pada modul-modul sebelumnya, Anda hanya diperlihatkan contoh membuat aplikasi android yang hanya melibatkan satu **Activity**. Namun, dalam membuat sebuah aplikasi sering kali melibatkan penggunaan lebih dari satu **Activity**. Maka kita diperlukan **Intent**. **Intent** adalah suatu mekanisme yang digunakan untuk menjabarkan suatu tindakan tertentu.

Beberapa contoh aplikasi yang melibatkan Intent, antara lain adalah sebagai berikut:

- ★ Menjalankan web browser;
- ★ Merekam suara;
- ★ Menjalankan aplikasi kamera;
- ★ Dan tindakan lainnya.

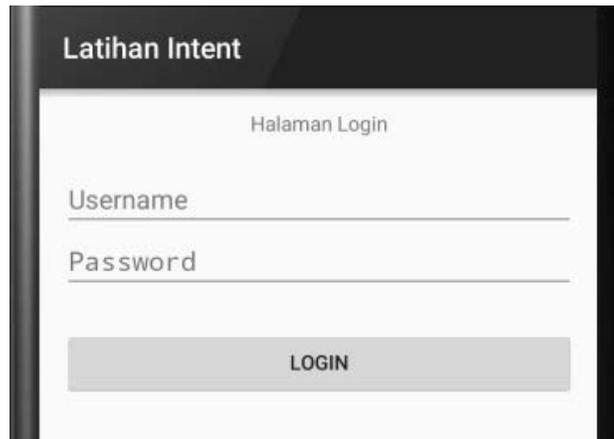
## 5. Penerapan Intent: Pemanggilan Activity Lain

Salah satu bentuk penerapan **Intent** adalah untuk memanggil **Activity** lain dari suatu **Activity**. Misalnya suatu halaman Login, ketika tombol Login di klik maka akan masuk ke halaman profil atau halaman utama.



Berikut ini langkah-langkah untuk membuat sebuah **Intent**.

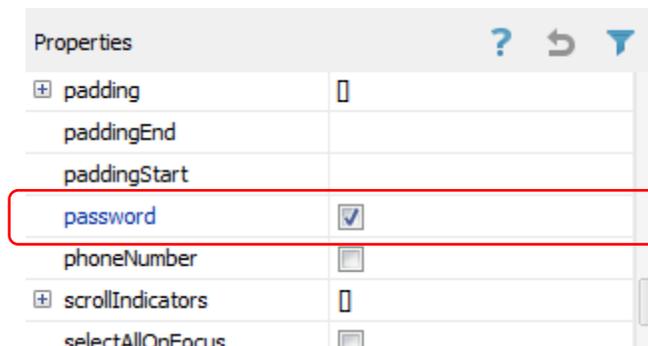
1. Buka Android Studio kemudian pilih menu **File** dan klik **Close Project** untuk menutup semua file yang terbuka.
2. Buat sebuah project Android baru.
3. Buatlah tampilan seperti berikut ini.



4. Melalui panel Properties, lakukan perubahan pada sejumlah properti milik beberapa komponen sebagai berikut.

Komponen	Properti	Nilai Baru
textView	Id	textViewHalSatu
	Text	Halaman Login
editText1	Id	editTextUsername
	Text	Username
editText2	Id	editTextPassword
	Text	Password
Button	Id	buttonLogin
	Text	Login

5. Pada komponen yang akan diisi password ubah Properties **Password** menjadi **True**.

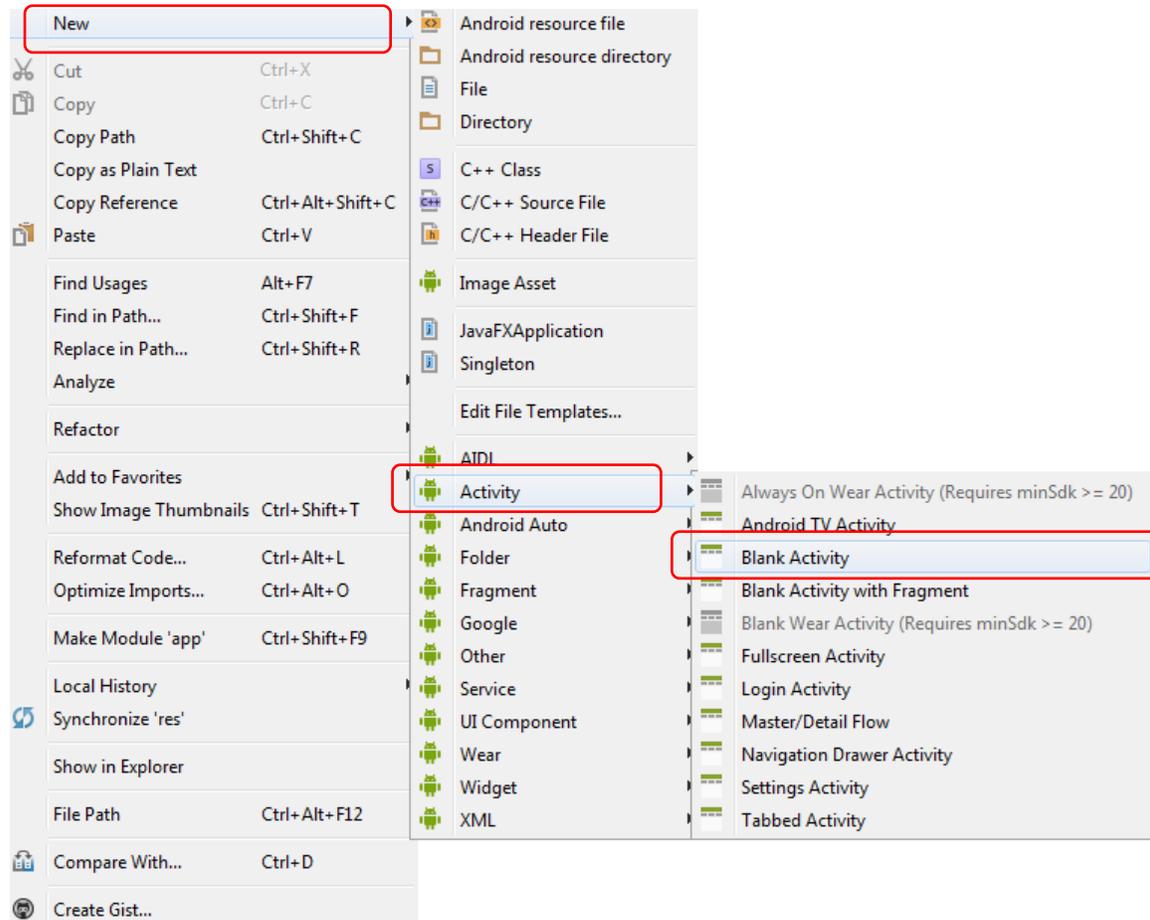


6. Tekan **Ctrl+S** untuk menyimpan project.

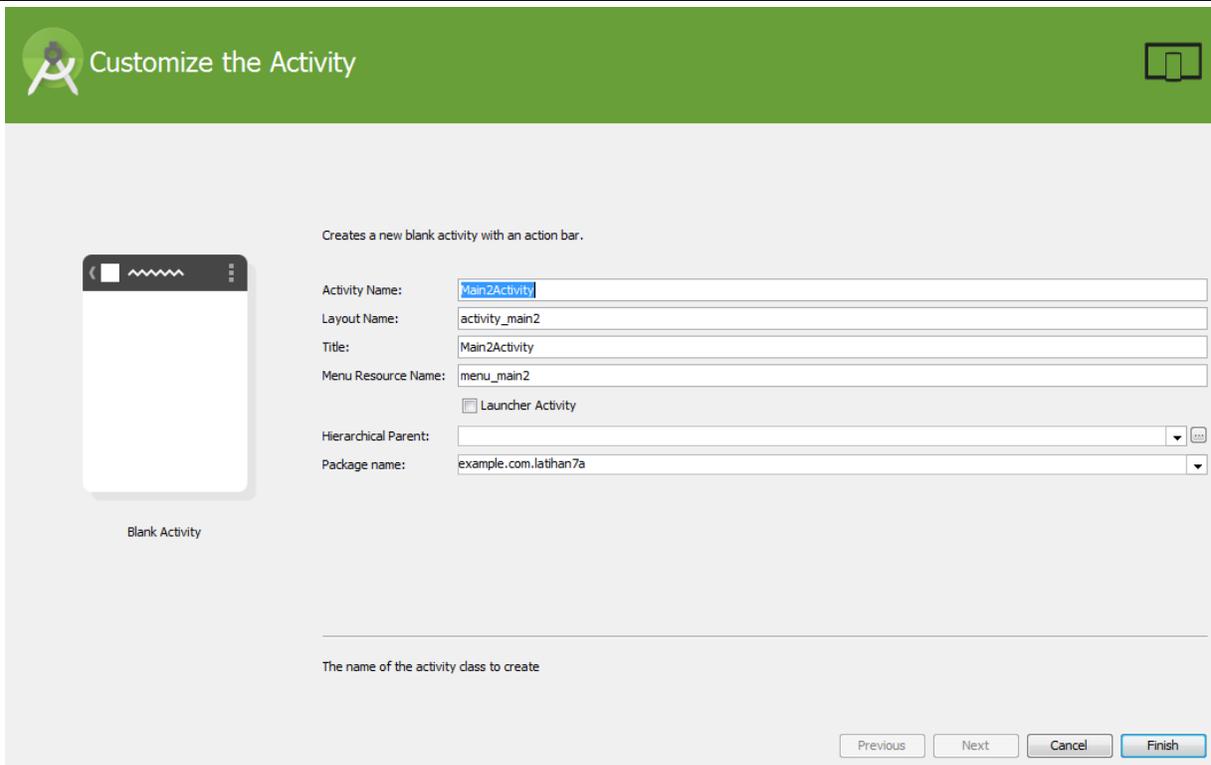


Untuk membuat Activity kedua, lakukan langkah-langkah sebagai berikut.

1. Letakan kursor pada panel **Project** kemudian klik kanan.
2. Pilih **New >> Activity >> Blank Activity**.



3. Selanjutnya akan muncul sebuah form dialog untuk memberi nama pada **Activity** yang akan dibuat. Silahkan isi **Activity Name** tersebut dengan nama **MainActivityKedua**. Kemudian tekan tombol **Finish**.



4. Setelah melakukan langkah-langkah diatas maka Android Studio akan menambahkan 2 buah file baru ke dalam project yang Anda buat yaitu MainActivityKedua.java dan activity\_main\_activity\_kedua.xml.
5. Ubahlah tampilan pada file activity\_main\_activity\_kedua.xml menjadi sebagai berikut.



6. Melalui panel Properties, lakukan perubahan pada sejumlah properti milik beberapa komponen sebagai berikut.



Komponen	Properti	Nilai Baru
textView	Id	textViewHalDua
	Text	Halaman Setelah Login
Button	Id	buttonSignout
	Text	Kembali Ke Halaman Login

7. Kemudian tekan Ctrl+S untuk menyimpan project.

Kita akan membuat sebuah aplikasi yang apabila tombol Login di klik akan membuat activity halaman kedua dijalankan, untuk itu tulislah beberapa kode berikut ini pada MainActivity.java.

```
package example.com.latihan7a;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button tombolKeHalDua = (Button) findViewById(R.id.buttonLogin);
        tombolKeHalDua.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent1 = new
                Intent(getApplicationContext(), MainActivityKedua.class);
                intent1.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent1);
                finish();
            }
        });
    }

    @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);
}
```

Bagian yang berada di kotak merah adalah bagian yang perlu ditambahkan. Berikut ini adalah penjelasan dari beberapa kode yang ditambahkan.

Kode:

```
Button tombolKeHalDua = (Button) findViewById(R.id.buttonLogin);
```

digunakan untuk mendeklarasikan komponen Button.

Kode:

```
tombolKeHalDua.setOnClickListener(new View.OnClickListener() {
    ...
});
```

digunakan untuk menangani event klik. Dimana ketika kita menuliskan kode tersebut, maka secara otomatis akan muncul sebuah kode berikut ini.

```
@Override
public void onClick(View v) {
    ...
}
```

Kode:

```
Intent intent1 = new
Intent(getApplicationContext(), MainActivityKedua.class);
```

digunakan untuk menciptakan object berkelas **Intent** yang dikaitkan dengan halaman.class (yang disimpan di MainActivityKedua.java, menyatakan activity yang dihubungkan dengan intent1). Argument pertama Intent() menyatakan konteks.



Kode:

```
intent1.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
```

digunakan untuk meng-handle Action saat back button di klik. Fungsi setSeperti yang diketahui, android tidak memiliki tombol khusus untuk keluar dari aplikasi, semua sistem android hanya memiliki tombol **Back** yang digunakan untuk kembali pada activity sebelumnya atau keluar dari aplikasi.

Kode:

```
startActivity(intent1);
```

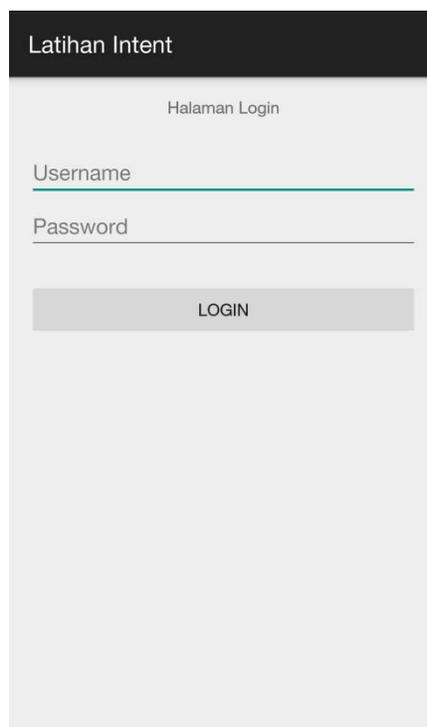
digunakan untuk menjalankan sebuah **Activity**.

Kode:

```
finish();
```

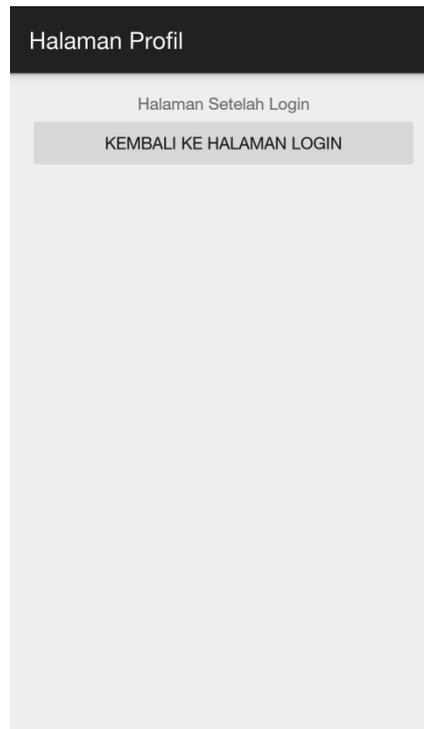
digunakan untuk mengakhiri aplikasi, sistem Android akan melakukan *method* `onDestroy`.

Ketika aplikasi dijalankan, berikut ini adalah tampilan yang akan terlihat pada device atau emulator.





Ketika tombol **LOGIN** di klik maka tampilan activity halaman kedua akan ditampilkan. Berikut ini adalah tampilan setelah tombol LOGIN di klik.



Dan jika tombol **KEMBALI KE HALAMAN LOGIN** di klik, maka aplikasi akan mengembalikan tampilan ke activity halaman pertama. Pada halaman kedua, jika tombol **BACK** di klik, maka aplikasi akan langsung menutup, karena penggunaan fungsi `addFlags`.

Pada aplikasi diatas, fungsi dari username dan password belum ditentukan. Dan bagaimana cara untuk melewati data ke activity lain? Bagaimana cara membuka activity lain seperti browser, kamera, atau dial?