

MOBILE PROGRAMMING



Oleh :

CHALIFA CHAZAR



MATERI 3

MENULISKAN KODE

Tujuan:

Mahasiswa memahami dan mampu menggunakan komponen dan penulisan kode untuk pembuatan suatu aksi pada aplikasi Android.

Pustaka:

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

1. Memahami Sebuah Event

Kejadian (*event*) adalah sebuah aksi yang melekat dalam suatu kontrol atau *activity*, misalnya ketika tombol di-klik. Dengan menangani suatu *event*, kita bisa memprogram ketika *event* tersebut berlangsung. Contoh, kita bisa mengatur ketika sebuah tombol di-klik, maka warna latar belakang diubah.

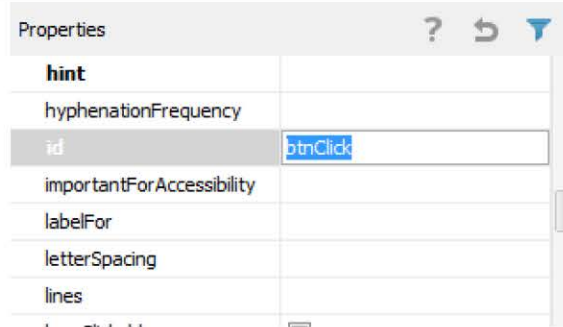
Sebuah *event* diimplementasikan dalam bentuk metode. Metode adalah sekumpulan perintah yang dikemas dengan sebuah nama. Itu sebabnya, kejadian seperti `onClick` biasa ditulis menjadi `onCreate()`, yang menyatakan bahwa `onCreate()` adalah sebuah metode.

2. Mengganti Nama Komponen

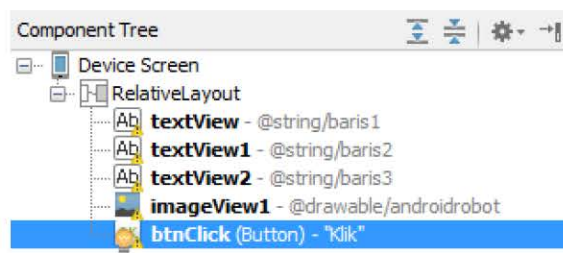
Untuk kepentingan penulisan kode yang berhubungan dengan *event*, sebaiknya komponen yang akan dilibatkan diberi nama yang memudahkan kita untuk mengingatnya.

Pada materi sebelumnya, Anda diminta untuk membuat sebuah aplikasi yang menampilkan Nama, NIM, Jurusan, sebuah gambar dan Button “klik”. Berikut ini adalah cara untuk memberi nama pada komponen **Button**.

1. Klik tombol button yang ada pada layar design
2. Pada panel **Properties**, ubah ID contohnya, menjadi `btnClick`



- Perhatikan nama Button, pada panel **Structure** juga akan ikut berubah sesuai dengan nama ID yang diberikan.



3. Memahami Activity

Untuk memprogram sebuah *event* pada tombol “klik”, Anda perlu memahami bagaimana dan dimana kode ditempatkan. Kode untuk membuat sebuah aplikasi Android ditulis di file `MainActivity.java` yang berada di folder `src`.

Untuk melihat isi dari file `MainActivity.java` bisa dilakukan dengan cara mengklik ganda file tersebut. Isi dari file `MainActivity.java` adalah sebagai berikut.

```
package example.com.latihan2;

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

public class MainActivity extends AppCompatActivity {

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



```
@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);
}
}
```

Penjelasan lebih lanjut dari kode di atas adalah sebagai berikut.

```
package example.com.latihan2;
```

Menyatakan nama paket. Pada contoh diatas nama paket adalah latihan2. Adapun com dan example secara berturut-turut menyatakan nama domain dan subdomain.

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

Import menyatakan pustaka yang diperlukan oleh kode. Jumlah import bergantung pada komponen yang digunakan dan juga kelas-kelas yang dipakai di kode.

```
public class MainActivity extends AppCompatActivity {
    ...
}
```



Digunakan untuk menciptakan kelas sebuah *activity*. Sebuah *activity* diciptakan melalui kelas baru yang memperluas kelas dasar yang bernama *Activity*. Pada contoh diatas, kelas yang diciptakan bernama *AppCompatActivity*.

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}
```

Digunakan untuk mendefinisikan metode `onCreate ()` milik *activity*.

```
super.onCreate(savedInstanceState);
```

Pernyataan ini berguna untuk menjalankan metode `onCreate ()` milik kelas dasar. Kata `super` menyatakan kelas dasar.

```
setContentView(R.layout.activity_main);
```

Pernyataan ini digunakan untuk menampilkan antarmuka di layar. Dalam hal ini, `R` menyatakan *resource* (folder *res*), *layout* menyatakan subfolder *layout*, dan *activity_main* merujuk file *activity_main.xml*.

Kelas *activity* memuat komponen-komponen yang menyusun antarmuka yang definisinya terletak di file XML yang berada di folder *res/layout*, yang bernama *activity_main.xml*.

Kode dalam *activity_main.xml* dapat ditulis secara langsung atau dimodifikasi melalui layar design (dengan melibatkan komponen-komponen di panel **Structure**).

4. Menuliskan Kode

Untuk membuat sebuah event `onClick ()` pada **Button**, Anda perlu menuliskan kode diantara

```
setContentView(R.layout.activity_main);
```

Dan

```
}
```

Tuliskan kode (hanya yang diberi tanda merah), sehingga tampilan kode secara keseluruhan menjadi seperti dibawah ini.

```
package example.com.latihan2;  
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 btnClick = (Button) findViewById(R.id.btnClick);
        btnClick.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                //kode aksi yang akan dijalankan
            }
        });

    }

    @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();

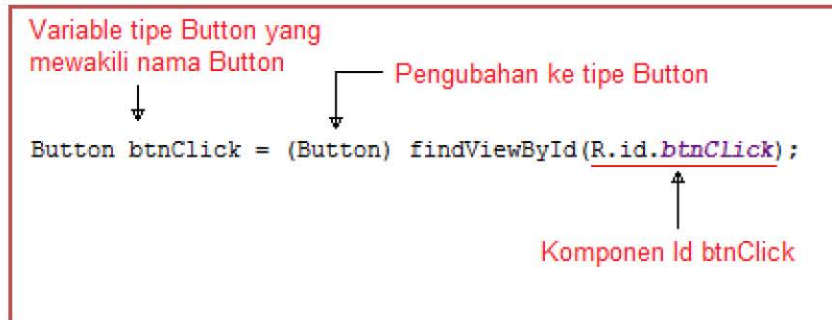
        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```



Penjelasan untuk beberapa tambahan kode adalah sebagai berikut.

- Kode `import android.view.View;`
Diperlukan kalau ada komponen View yang dilibatkan dalam kode
- Kode `import android.widget.Button;`
Diperlukan karena komponen Button dilibatkan dalam kode
- Kode `Button btnClick = (Button) findViewById(R.id.btnClick);`
Digunakan untuk memperoleh referensi terhadap komponen bernama `btnClick` (yang dinyatakan dengan `R.id.btnClick`). Variable `btnClick` yang dideklarasikan sebagai Button akan merujuk ke komponen `btnClick`

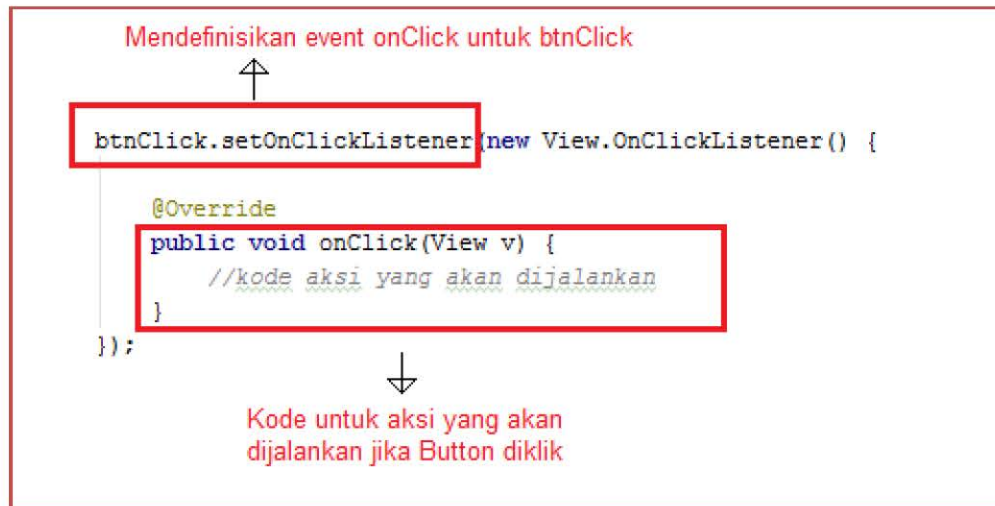


- Kode
`btnClick.setOnClickListener(new View.OnClickListener() {`

```
    @Override  
    public void onClick(View v) {  
        //kode aksi yang akan dijalankan  
    }  
});
```

Digunakan untuk menentukan pemantauan event `onClick` pada `btnClick`. Untuk menuliskan sebuah aksi pada tombol `btnClick`, dituliskan pada bagian

```
    @Override  
    public void onClick(View v) {  
        //kode aksi yang akan dijalankan  
    }
```



Selanjutnya kita akan membuat sebuah aplikasi yang menampilkan pesan berisi Nama Anda ketika tombol diklik.

Tuliskan baris kode di bawah ini.

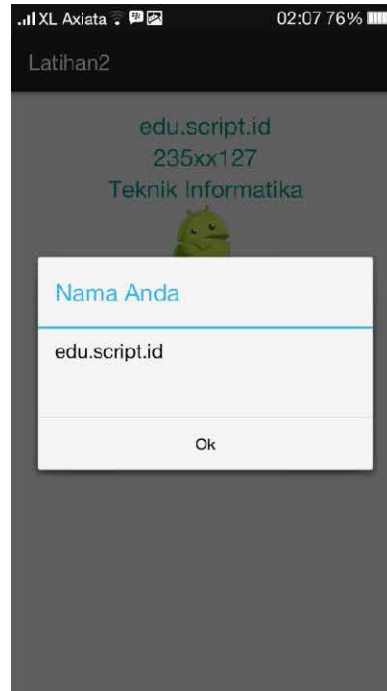
```
@Override  
public void onClick(View v) {  
    //kode aksi yang akan dijalankan  
  
    AlertDialog msg=new  
AlertDialog.Builder(MainActivity.this).create();  
    msg.setTitle("Nama Anda");  
    TextView test1=(TextView) findViewById(R.id.textView);  
    String isitest1=test1.getText().toString();  
    msg.setMessage(isitest1);  
  
    msg.setButton(AlertDialog.BUTTON_NEUTRAL,"Ok",  
        new DialogInterface.OnClickListener(){  
            public void onClick(DialogInterface dialog, int  
which){  
                dialog.dismiss();  
            }  
        });  
    msg.show();  
}
```

Tambahkan pula pustaka untuk menampilkan sebuah dialog dengan sebuah pilihan button. Berikut ini adalah kode untuk mengambil pustaka yang dapat menampilkan dialog.



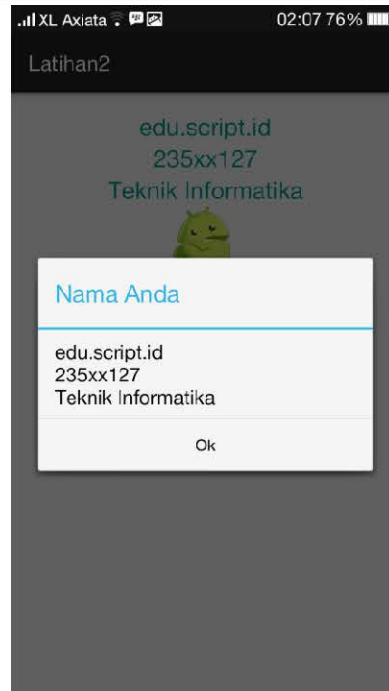
```
import android.app.AlertDialog;  
import android.content.DialogInterface;
```

Selanjutnya simpan *project* dan jalankan. Berikut ini adalah tampilan ketika program dijalankan pada mobile device.



Latihan 5

Ketika tombol diklik maka akan muncul sebuah dialog yang menampilkan Nama Anda. Tetapi seperti yang dilihat bahwa pada program terdapat juga NIM dan Jurusan Anda. Silahkan ubah program sehingga dapat menampilkan Nama, NIM, dan Jurusan Anda, seperti yang terlihat pada tampilan di bawah ini.



Latihan 6

Sekarang coba buat sebuah program dimana terdapat input berupa Username dan Password, dan ketika sebuah tombol diklik maka akan muncul message box yang menampilkan kata yang di input.

