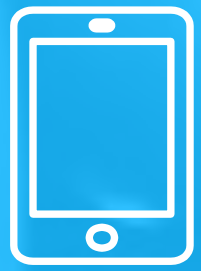
PENGEMBANGAN APLIKASI PERANGKAT BERGERAK (MOBILE)

Event Handling

K Candra Brata

andra.course@gmail.com



Event Handling

http://developer.android.com/guide/topics/ui/ui-events.html

Definition

Event: Action that occurs when user interacts with widgets.

An external stimulus your program can respond to.

e.g. clicks, typing, scrolling, etc.

Common kinds of events include:

– Mouse motion / tapping, Keys pressed,

– Timers expiring, Network data available

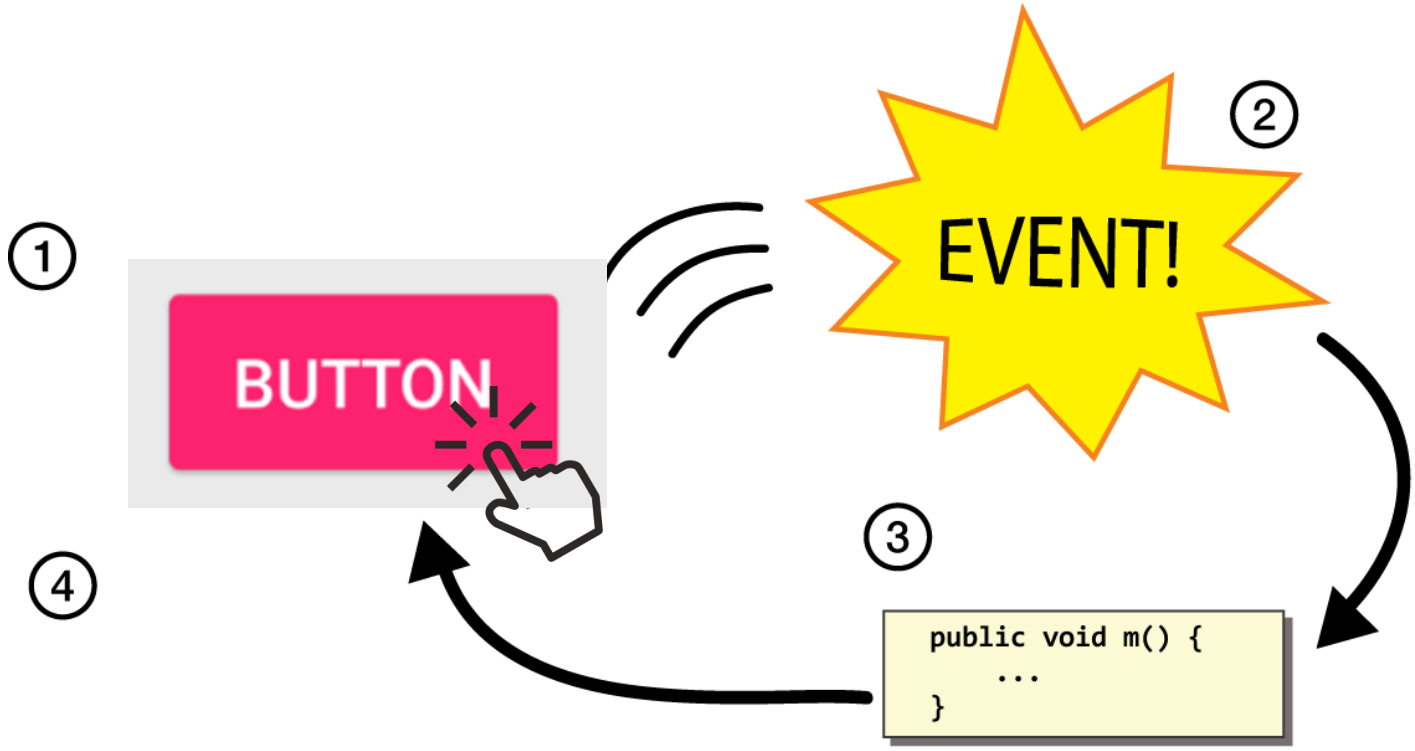
Event-driven programming:

- Overall execution of your program is largely dictated by user events.
- Commonly used in graphical interface programs.

To respond to events in a program, you must:

1. Write methods to handle each kind of event ("listener" methods).
2. Attach those methods to particular GUI widgets.

Event Scenario

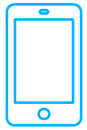


Note:

One listener is not limited to be attached to one widget.
*It can be attached to **any widgets** for the same events.*

Widget : Views That Have Events

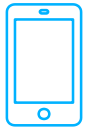
- For a list of the widgets provided by Android, see the android.widget package.
- Some Examples
  - Button
  - CheckBox
  - DatePicker
  - EditText
  - ImageView
  - Spinner (ComboBox)
  - Or even ViewGroups may also have events



Setting an event listener (Step By Step)

1. Interacting with widgets.

- in layout, give view an unique ID property value.
- in Java code, call **findViewById()** to access its View object pass it a parameter of ***R.id.yourUnique_ID.***
- Cast the returned value to the appropriate type (Button, TextView, etc.).



Setting an event listener

1. Interacting with widgets.

- In Layout XML file:

```
<TextView android:id="@+id/textViewName" />
```

- In Activity onCreate() method:

```
TextView nameView = (TextView) findViewById(R.id.textViewName);
```

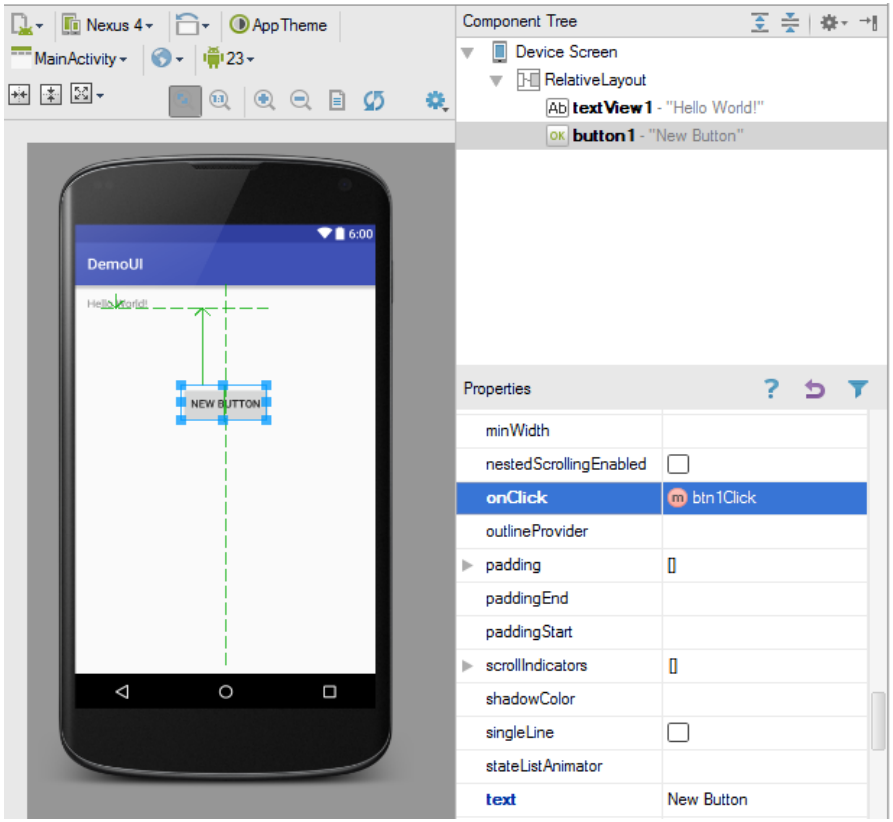
Setting an event listener

2. modify onClick property in layout file.

- Scroll down its Properties until you find onClick.

  or edit .XML file manually via text.

- Type the **name of a method** you'll write to handle the click event




```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="100dp"
    android:text="New Button"
    android:onClick="btn1Click" />
```

Setting an event listener

3. Add method in onclick Event in Java Code

- In your Activity, create method that have been registered in layout.
- Add **View** object in method parameter.

MainActivity.java × activity_main.xml ×

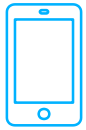
```
1  package com.example.stepp.numbergame;
2
3  import ...
8
9  public class MainActivity extends ActionBarActivity {
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         setContentView(R.layout.activity_main);
13         super.onCreate(savedInstanceState);
14     }
15
16     public void btn1Click (View view) {
17         // your code goes here
18
19     }
```

activity_main.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.andra.demoui.MainActivity">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:onClick="btn1Click"
        android:text="New Button" />

</RelativeLayout>
```

MainActivity.Java

```
public class MainActivity extends AppCompatActivity {
    private Button btn1;
    private TextView txt1;

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

        txt1 = (TextView) findViewById (R.id.textView1);
        btn1 = (Button) findViewById (R.id.button1);

    }


    public void btn1Click(View view) {

        txt1.setText("Clicked Button !!");   }

}
```



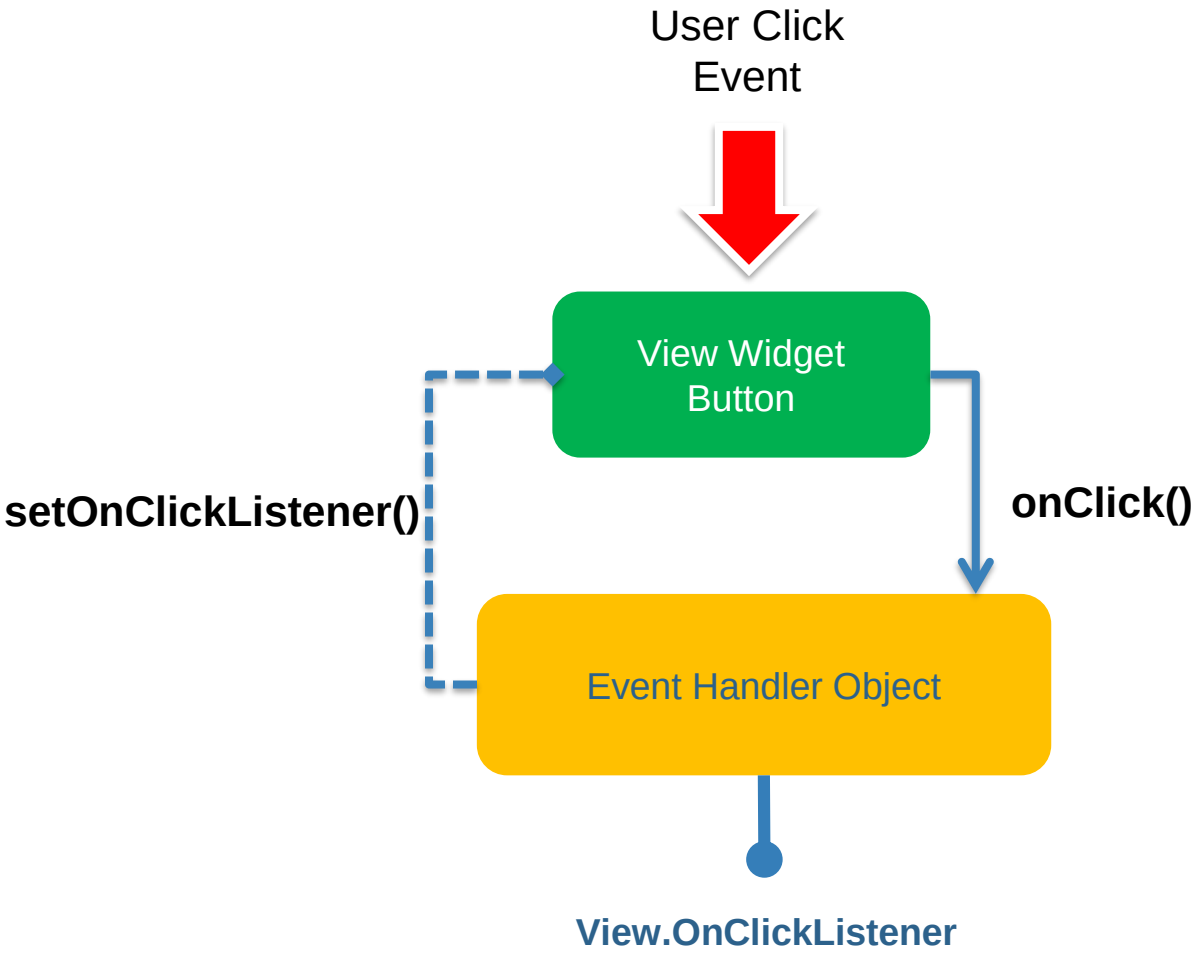
Setting an event listener (Hard Code)

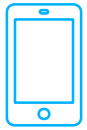
An event is handled by an ***event listener*** object.

Define an object that implements event listener interface and register it with the designated View.

- View.OnClickListener (for handling "clicks" on a View),
- View.OnTouchListener (for handling screen touch events),
- View.OnKeyListener (for handling device key presses).
- etc..

Setting an event listener (Hard Code)





Setting an event listener (Hard Code)

Step 1: Add View to Activity (Button)

(Button) findViewById (ButtonViewID)

Step 2: Implement Event Handler

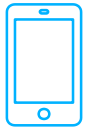
for a View **click** event means the handler object should implements the **View.OnClickListener** interface.

TWO OPTIONS :

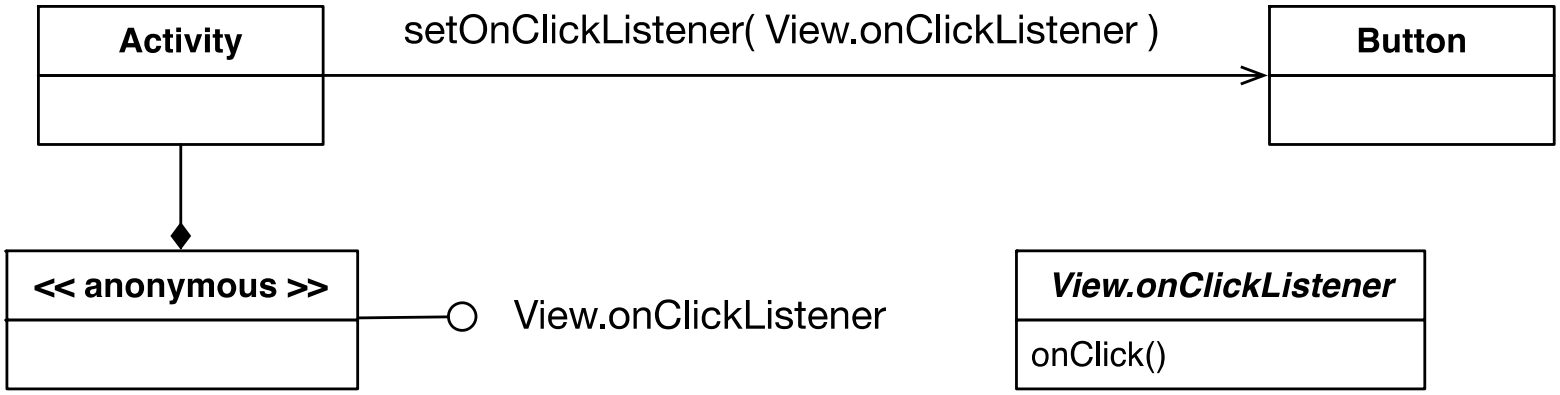
- using a separate (specific) object to handle event(s) that implements the interface.
- have the Activity containing the button do the event handling and letting the Activity implements the interface.

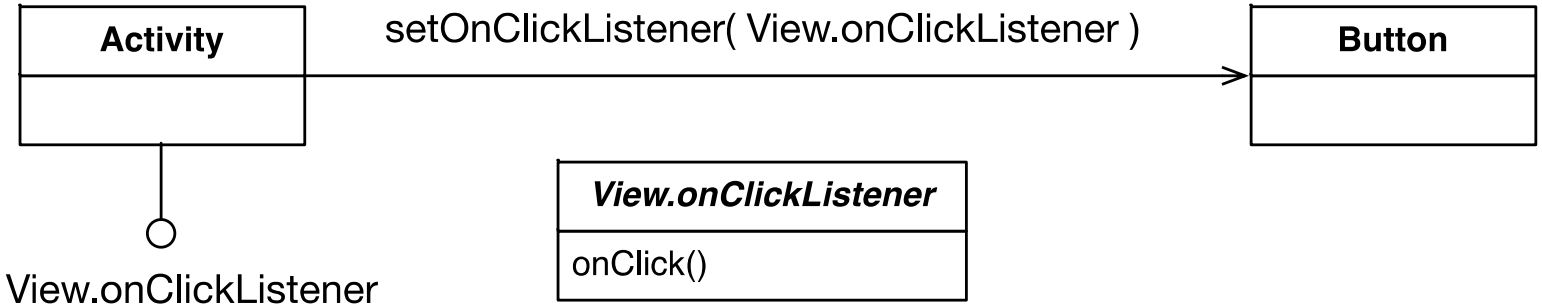
Step 3: Register Event Handler to the button

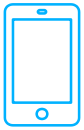
button.setOnClickListener(clickEventHandlerObject)



Setting an event listener (Hard Code)



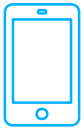




Setting an event listener (Hard Code)

EVENT HANDLING CODE in separate object named mCorkyListener

```
private Button button;
private View.OnClickListener mCorkyListener;
//Now inside your Activity class onCreate event method
protected void onCreate(Bundle savedValues) {
   ...

 // STEP 1:   Capture our button from layout
    Button button = (Button)findViewById(R.id.corky);
 // STEP 2 : Create an anonymous implementation of OnClickListener
    mCorkyListener = new View.OnClickListener() {
    public void onClick(View v) {
                // do something when the button is clicked
      }
    };

   // STEP 3: Register the onClick listener  to the view
    button.setOnClickListener(mCorkyListener);
    ...
}
```



Setting an event listener (Hard Code)

Here's the code to handle Button's click event using the Activity itself

```
public class ExampleActivity extends Activity implements View.OnClickListener {

    protected void onCreate(Bundle savedValues) {

        ...

        Button button = (Button)findViewById(R.id.corky);   // STEP 1
        button.setOnClickListener( this);   //STEP 3 – registration

    }

    // Implement the View.OnClickListener callback method
    // STEP 2 – event handler
    public void onClick(View v) {

      // do something when the button is clicked

    }
    ...
}
```

activity2.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout  xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text=" Luas PERSEGI !!" />

    <EditText
        android:id="@+id/txtSisi"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberSigned|numberDecimal"
        android:hint="Masukkan nilai sisi!!"></EditText>

    <Button
        android:id="@+id/btnHitung"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Hitung Luas"></Button>

    <EditText
        android:id="@+id/txtLuas"
        android:layout_width=“match_parent"
        android:layout_height="wrap_content"
        android:editable="false"
        android:hint="Nilai Luas "></EditText>


</LinearLayout>
```

Activity2.Java

```
public class Activity2 extends AppCompatActivity {

    private EditText txtsisi;
    private EditText txtLuas;
    private Button btnhitung;
    private View.OnClickListener click;

@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity2);

        txtsisi = (EditText) findViewById (R.id.txtSisi);
        txtLuas = (EditText) findViewById(R.id.txtLuas);
        btnhitung = (Button) findViewById (R.id.btnHitung);

        click = new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                try {
                    int sisi = Integer.parseInt (txtsisi.getText().toString());
                    int luas = sisi * sisi;
                    txtLuas.setText(String.valueOf(luas));

                } catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        };

    btnhitung.setOnClickListener(click);
    }
```

Tugas 2.

Buat Applikasi Android Potato Head.

Write an app that displays a **"Mr. Potato Head"** toy on the screen as an **ImageView.**

You can download all resource files of each body part and accessory, such as body.png, ears.png, hat.png etc in our course group.

Clue :

Modify android:visibility property.

You can customize the application behaviour (use Button or Switch panel or radio button etc as Views to handle user event).

Tugas 2.

Buat Applikasi Android Potato Head.

If you are not following this rule, I will assume that you are not complete the assignment

Deadline : 2 Oct 2017, 09:59 AM (WIB)

Thanks!

JOIN !!

http://bit.do/papb_si_b