

# Materi 5 Pemrograman Visual

---

Pada materi ini akan dipelajari tentang pemberian aksi pada komponen swing (listener), terutama ActionListener.

## Sesi Materi

ActionListener merupakan salah satu listener yang paling sering digunakan pada pemrograman GUI menggunakan bahasa pemrograman Java. Hal ini karena ActionListener dapat dikenai ke banyak komponen pembentuk GUI (lihat modul praktikum). Listener, khususnya ActionListener dalam materi ini merupakan interface yang memiliki method actionPerformed. Sehingga, method tersebut harus di-override oleh class yang meng-implements interface ActionListener. Isi dari method ini (untuk dituliskan kembali) berisi logika pemrograman ketika terjadi suatu hal terhadap komponen pada class tersebut. Suatu hal dapat berupa tombol yang ditekan, perubahan state, pengetikan, dan hal lainnya tergantung behavior komponen swing.

Penulisan aksi terhadap listener seharusnya bersifat private, atau dituliskan dalam bentuk inner class. Hal ini dimaksudkan agar secure implementation terpenuhi. Akan tetapi, untuk materi ini tidak menggunakan konsep tersebut. Langkah yang digunakan kali ini:

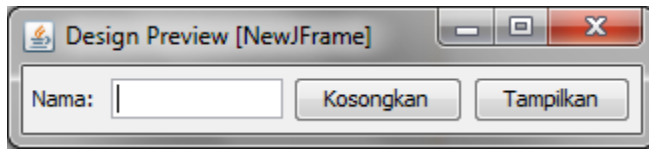
1. Melakukan implements ActionListener
2. Meng-override method actionPerformed
3. Melakukan filter asal aksi, dan memberikan proses yang bersesuaian

## Studi Kasus

Buatlah sebuah GUI dengan 2 tombol. Fungsi masing-masing tombol adalah sebagai berikut:

1. Mengosongkan text dan mengembalikan komponen ke nilai default
2. Melakukan proses pengolahan masukan menjadi keluaran (jika harus diolah) serta menampilkan keluaran tersebut

GUI tersebut ditampilkan sebagai berikut:



Tombol “Kosongkan” akan mengosongkan nama, sedangkan tombol “Tampilkan” akan menampilkan kembali hal yang sudah diketikkan oleh pengguna ke layar (melalui command prompt atau layar keluaran IDE).

Untuk membuat GUI di atas, langkah pertama yang harus dilakukan adalah mendata komponen swing yang digunakan:

1. Sebuah JFrame (layout default)
2. Sebuah JPanel (layout default)
3. Sebuah JLabel
4. Sebuah JTextField
5. Dua buah JButton

Sehingga, kode yang dapat dituliskan sebagai berikut:

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class NewClass extends JFrame{
    JPanel panel;
    JLabel labelNama;
    JTextField teksNama;
    JButton tombolReset, tombolEksekusi;

    public void aturGui(){
    }
}
```

Bentuk objeknya, dan atur kemunculan GUI pada method aturGui()

```

panel = new JPanel();
labelNama = new JLabel("Nama: ");
teksNama = new JTextField(10);
tombolReset = new JButton("Kosongkan");
tombolEksekusi = new JButton("Tampilkan");

panel.add(labelNama);
panel.add(teksNama);
panel.add(tombolReset);
panel.add(tombolEksekusi);

this.add(panel);

```

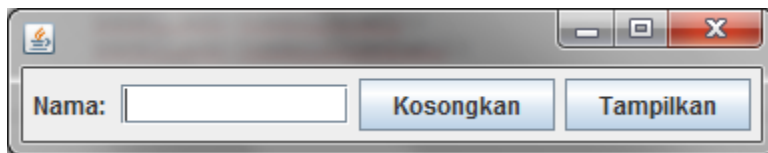
Pastikan GUI telah menampilkan hal yang seharusnya dengan membuat public static void main.

```

public static void main(String[] args) {
    NewClass fr = new NewClass();
    fr.aturGui();
    fr.setDefaultCloseOperation(3);
    fr.pack();
    fr.setLocationRelativeTo(null);
    fr.setVisible(true);
}

```

Setelah tampil seperti gambar berikut,



Tambahkan listener berupa interface ActionListener. Override method actionPerformed dari ActionListener.

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class NewClass extends JFrame implements ActionListener{

    JPanel panel;
    JLabel labelNama;
    JTextField teksNama;
    JButton tombolReset, tombolEksekusi;

    public void aturGui() {
        panel = new JPanel();

```

```

        labelNama = new JLabel("Nama: ");
        teksNama = new JTextField(10);
        tombolReset = new JButton("Kosongkan");
        tombolEksekusi = new JButton("Tampilkan");

        panel.add(labelNama);
        panel.add(teksNama);
        panel.add(tombolReset);
        panel.add(tombolEksekusi);

        this.add(panel);
    }

    public static void main(String[] args) {
        NewClass fr = new NewClass();
        fr.aturGui();
        fr.setDefaultCloseOperation(3);
        fr.pack();
        fr.setLocationRelativeTo(null);
        fr.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent ae) {

    }
}

```

Tuliskan method actionPerformed tepat seperti hal di atas. Berikan parameter masukan berupa objek dari ActionEvent. Langkah berikutnya, filter dari mana aksi berasal. Filter ini menggunakan kondisional. Ada banyak cara untuk memberi pengkondisian, dua cara yang biasa dilakukan yaitu dengan menggunakan action command, atau bisa juga dengan menggunakan getSource.

Sebelum hal tersebut dilakukan, hal penting yang dilakukan yaitu menambahkan action listener kepada dua tombol yang diberikan aksi:

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class NewClass extends JFrame implements ActionListener{

    JPanel panel;
    JLabel labelNama;
    JTextField teksNama;
    JButton tombolReset, tombolEksekusi;

    public void aturGui() {
        panel = new JPanel();
        labelNama = new JLabel("Nama: ");
        teksNama = new JTextField(10);
        tombolReset = new JButton("Kosongkan");
    }
}

```

```

        tombolEksekusi = new JButton("Tampilkan");

        tombolReset.addActionListener(this);
        tombolEksekusi.addActionListener(this);

        panel.add(labelNama);
        panel.add(teksNama);
        panel.add(tombolReset);
        panel.add(tombolEksekusi);

        this.add(panel);
    }

    public static void main(String[] args) {
        NewClass fr = new NewClass();
        fr.aturGui();
        fr.setDefaultCloseOperation(3);
        fr.pack();
        fr.setLocationRelativeTo(null);
        fr.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent ae) {

    }
}

```

Perhatikan tulisan kode yang ditebalkan, terdapat method `addActionListener(this)`. Keyword `this` menyatakan bahwa kode logika untuk aksi yang diberikan berada di tempat yang sama dengan pendeklarasian masing-masing tombol, yaitu class "NewClass". Jika aksi berada pada class lain, maka pastikan yang menjadi parameter adalah objek dari class yang memiliki logikanya.

Untuk penambahan aksi kali ini, bagi `tombolReset`, akan digunakan filter menggunakan action command, sedangkan `tombolEksekusi` dengan cara lainnya. Jika menggunakan action command, pastikan setiap tombol telah diberikan command masing-masing. Kode untuk `tombolReset` dituliskan sebagai berikut.

```

public void aturGui() {
    panel = new JPanel();
    labelNama = new JLabel("Nama: ");
    teksNama = new JTextField(10);
    tombolReset = new JButton("Kosongkan");
    tombolEksekusi = new JButton("Tampilkan");

    tombolReset.setActionCommand("Kosong");

    tombolReset.addActionListener(this);
    tombolEksekusi.addActionListener(this);

    panel.add(labelNama);
    panel.add(teksNama);
    panel.add(tombolReset);
    panel.add(tombolEksekusi);

    this.add(panel);
}

```

```
}
```

Sehingga, pada method `actionPerformed` dapat dituliskan hal berikut:

```
@Override
public void actionPerformed(ActionEvent ae) {
    if (ae.getActionCommand().equals("Kosong")) {
        System.out.println("Aksi Tombol Kosongkan");
        teksNama.setText("");
    }
}
```

Perhatikan bahwa method `equals` merupakan perbandingan antara action yang diterima dari objek `ActionEvent` sesuai dengan action command yang diberikan sebelumnya. Jika pada `setActionCommand` diberikan parameter “Hilang”, maka pada perbandingan pengkondisian juga harus dituliskan “Hilang”. Agar teks menjadi kosong, dipanggil method “`setText`” dari class `JTextField` (lihat modul praktikum).

Untuk tombol “Tampilkan”, hal lain yang dilakukan, yaitu pengambilan source asal dengan kode sebagai berikut:

```
if (ae.getSource().equals(tombolEksekusi)) {
    System.out.println("Aksi Tombol Eksekusi");
    String nama = teksNama.getText();
    System.out.println("Nama: "+nama);
}
```

Hal ini paling mungkin dilakukan jika komponen tombol berada pada class yang sama dengan logika `actionListener`. Cara ini sebaiknya tidak terlalu sering digunakan.

## Kode Lengkap:

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class NewClass extends JFrame implements ActionListener{

    JPanel panel;
    JLabel labelNama;
    JTextField teksNama;
    JButton tombolReset, tombolEksekusi;

    public void aturGui() {
        panel = new JPanel();
        labelNama = new JLabel("Nama: ");
        teksNama = new JTextField(10);
        tombolReset = new JButton("Kosongkan");
        tombolEksekusi = new JButton("Tampilkan");

        tombolReset.setActionCommand("Kosong");

        tombolReset.addActionListener(this);
        tombolEksekusi.addActionListener(this);

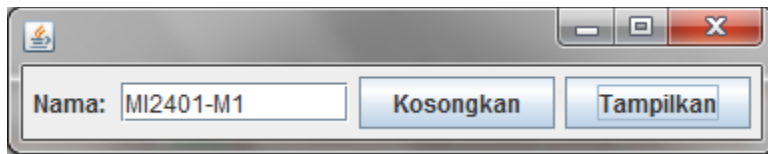
        panel.add(labelNama);
        panel.add(teksNama);
        panel.add(tombolReset);
        panel.add(tombolEksekusi);

        this.add(panel);
    }

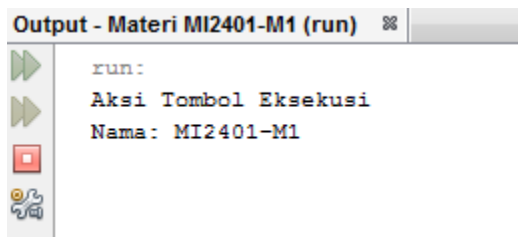
    public static void main(String[] args) {
        NewClass fr = new NewClass();
        fr.aturGui();
        fr.setDefaultCloseOperation(3);
        fr.pack();
        fr.setLocationRelativeTo(null);
        fr.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent ae) {
        if(ae.getActionCommand().equals("Kosong")){
            System.out.println("Aksi Tombol Kosongkan");
            teksNama.setText("");
        }
        if(ae.getSource().equals(tombolEksekusi)){
            System.out.println("Aksi Tombol Eksekusi");
            String nama = teksNama.getText();
            System.out.println("Nama: "+nama);
        }
    }
}
```

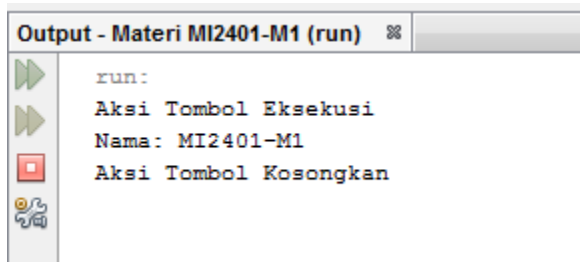
Setelah menambahkan aksi, GUI dapat dijalankan kembali. Contoh saat menekan tombol “Tampilkan” dengan state GUI



Ditampilkan sebagai berikut:



Sedangkan jika tombol “Kosongkan” ditekan setelahnya, teks akan kosong seperti pertama kali dijalankan, dan di layar keluar tambahan satu baris:



## Try-it Yourself

1. Apa yang terjadi ketika tombol “Tampilkan” ditekan ketika text field dalam keadaan kosong? Beri pengkondisian terhadap hal tersebut.
2. Tukarkan cara filter aksi untuk masing-masing tombol. Berikan action command untuk tombol “Tampilkan”, lalu cara lainnya untuk tombol “Reset”.



3. Buatlah GUI dengan tampilan berikut. Tombol “+ +” akan menambah nilai pada text field, tombol “- -” mengurangi nilai pada text field. Menu “Tampilkan” akan menampilkan angka yang tercantum di text field ke layar. Sedangkan menu “Kosongkan” akan mengubah angka menjadi 0 di text field.

