

# Modul 8 - Java Swing

**Durasi Modul = 2-5 menit**

## Capaian pembelajaran:

Setelah mempelajari modul ini, mahasiswa diharapkan mampu:

- Memahami konsep dasar Java Swing dan komponen-komponennya
- Mengimplementasikan antarmuka grafis sederhana menggunakan Java Swing
- Mengelola event handling pada komponen Java Swing
- Merancang tata letak (layout) aplikasi menggunakan berbagai layout manager
- Membuat aplikasi desktop interaktif dengan memanfaatkan fitur-fitur Java Swing

## Deskripsi Pembelajaran

Java Swing adalah toolkit GUI (Graphical User Interface) yang kuat dan fleksibel untuk membangun aplikasi desktop dalam bahasa pemrograman Java. Modul ini akan memperkenalkan Anda pada konsep dasar Java Swing dan komponen-komponennya yang penting.

Anda akan belajar cara membuat jendela, menambahkan berbagai komponen seperti tombol, label, dan kotak teks, serta mengatur tata letak mereka menggunakan layout manager. Selain itu, Anda akan mempelajari cara menangani event untuk membuat aplikasi Anda interaktif.

Melalui contoh-contoh praktis dan latihan, Anda akan mengembangkan keterampilan dalam merancang dan mengimplementasikan antarmuka pengguna yang menarik dan fungsional. Pada akhir modul ini, Anda akan

memiliki pemahaman yang solid tentang Java Swing dan kemampuan untuk membuat aplikasi desktop sederhana namun efektif.

## 8.1. Java Swing



**Graphical user interface (GUI)** membantu kita berinteraksi dengan suatu aplikasi. GUI membuat suatu aplikasi menjadi lebih user friendly. Dalam Java komponen UI disediakan dalam paket Swing. **Komponen Swing** merupakan elemen UI seperti kotak dialog dan tombol dapat dikenali melalui Namanya yang **dimulai dengan "j"**. Komponen GUI yang disediakan Java pada dasarnya adalah sebuah objek.

### Inisialisasi GUI Pada Java

Setiap komponen swing merupakan turunan dari JComponent yang kemudian diturunkan dari **class** **java.awt.Container**.

Untuk menggunakan komponen swing dilakukan dengan menambahkan perintah **import javax.swing.\*;** pada bagian awal program. Salah contoh pemanfaatan komponen Swing adalah penggunaan **JOptionPane untuk input/output**, seperti pada contoh program berikut.

```
import javax.swing.JOptionPane;
public class DemoSwing {
    public static void main(String[] args) {
        String nama;
        int usia;
        nama= JOptionPane.showInputDialog ("Nama anda: ");
        usia= Integer.parseInt(JOptionPane.showInputDialog("Usia anda: "));
        JOptionPane.showMessageDialog(null, "Hallo "+nama+","+"\n usia anda:
    }
}
```

## GUI Component

Selain JOptionPane kita dapat menggunakan komponen- komponen lain yang disediakan oleh **Swing untuk membuat user interface** dari program atau aplikasi kita. Tabel berikut merupakan daftar beberapa komponen dasar Swing yang dapat digunakan untuk membuat GUI.

Komponen	Deskripsi
JLabel	Menampilkan text yang tidak dapat diedit/icon
JTextField	Menerima input dari user
JButton	Mentrigger sebuah even ketika mouse di klik
JCheckBox	Menampilkan opsi yang dapat dipilih
JComboBox	Drop down list item yang dapat dipilih
JList	List item yang dapat dipilih oleh pengguna dengan mengklik salah satu pilihan
JPanel	Area untuk meletakkan komponen

Berikut contoh pemanfaatan beberapa komponen GUI tersebut.

### A. Pembuatan Window

Komponen GUI yang dapat digunakan untuk **membuat window adalah JFrame**. **JFrame** merupakan sebuah objek yang dapat dibuat dengan menggunakan class JFrame. Berikut merupakan contoh pemanfaatan class JFrame untuk pembuatan window.

```
import javax.swing.*;  
public class JendelaWindow extends JFrame {  
    private static final int WIDTH = 400;  
    private static final int HEIGHT = 300;  
    public RectangleProgramOne() {  
        setTitle("Luas dan Keliling Kotak");  
        setSize(WIDTH, HEIGHT);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        setVisible(true);  
    }  
    public static void main(String[] args) {  
        JendelaWindow jw = new JendelaWindow();  
    }  
}
```

### B. Menampilkan Text

Berikut contoh untuk menampilkan

**text menggunakan label.**

```
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import javax.swing.Icon;
import javax.swing.ImageIcon;
public class LabelFrame extends JFrame {
    private JLabel label1;
    private JLabel label2;
    private JLabel label3;
    public LabelFrame() {
        super("Testing JLabel");
        setLayout(new FlowLayout());
        label1 = new JLabel("Label with text");
        label1.setToolTipText("This is label1");
        add(label1);
        Icon bug = new ImageIcon(getClass().getResource("bug1.png"));
        label2 = new JLabel("Label with text and icon", bug, SwingConstants.LEFT)
        label2.setToolTipText("This is label2");
        add(label2);
        label3 = new JLabel();
        label3.setText("Label with icon and text at bottom");
        label3.setIcon(bug);
        label3.setHorizontalTextPosition(SwingConstants.CENTER);
        label3.setVerticalTextPosition(SwingConstants.BOTTOM);
        label3.setToolTipText("This is label3");
        add(label3);
    }
}
```

```
import javax.swing.JFrame;
public class LabelTest {
    public static void main( String[] args ) {
        LabelFrame labelFrame = new LabelFrame();labelFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        labelFrame.setVisible(true);
    }
}
```

```
    labelFrame.setSize( 260, 180 );
    labelFrame.setVisible( true );
}
}
```

### C. Penggunaan TextField

Berikut contoh penggunaan **text field** untuk menerima masukan dari user.

```
import java.awt.FlowLayout;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.JOptionPane;
public class TextFieldFrame extends JFrame {
    private JTextField textField1;
    private JTextField textField2;
    private JTextField textField3;
    private JPasswordField passwordField;
    public TextFieldFrame() {
        super("Testing JTextField and JPasswordField");
        setLayout(new FlowLayout());
        textField1 = new JTextField(10);
        add(textField1);
        textField2 = new JTextField("Enter text here");
        add(textField2);
        textField3 = new JTextField("Uneditable text field", 21);
        textField3.setEditable(false);
        add(textField3);
        passwordField = new JPasswordField("Hidden text");
        add(passwordField);
        TextFieldHandler handler = new TextFieldHandler();
        textField1.addActionListener(handler);
        textField2.addActionListener(handler);
        textField3.addActionListener(handler);
    }
}
```

```
        passwordField.addActionListener(handler);
    }
private class TextFieldHandler implements ActionListener {
    public void actionPerformed(ActionEvent event) {
        String string = "";
        if (event.getSource() == textField1) {
            string = String.format("textField1: %s", event.getActionCommand());
        }
        else if (event.getSource() == textField2) {
            string = String.format("textField2: %s", event.getActionCommand());
        }
        else if (event.getSource() == textField3) {
            string = String.format("textField3: %s", event.getActionCommand());
        }
        else if (event.getSource() == passwordField) {
            string = String.format("passwordField: %s", event.getActionCommand());
        }
        JOptionPane.showMessageDialog(null, string);
    }
}
```



## REFERENSI

Aziefah, & Nurmasari, J. (2013). Pemrograman Berorientasi Objek. Pekanbaru: Politeknik Caltex Riau.

Barclay, K., & Savage, J. (2004). Object Oriented Design with UML and Java. Elsevier.

Deitel, P., & Deitel, H. (2012). Java: How to Program (9 ed.). US: Prentice Hall.