

MODUL PBO
**“PENGENALAN GUI (*GRAPHICAL USER*
INTERFACE)”**



Disusun Oleh
NOVI DYAH PUSPITASARI

MODUL PBO

“PENGENALAN GUI (GRAPHICAL USER INTERFACE)”

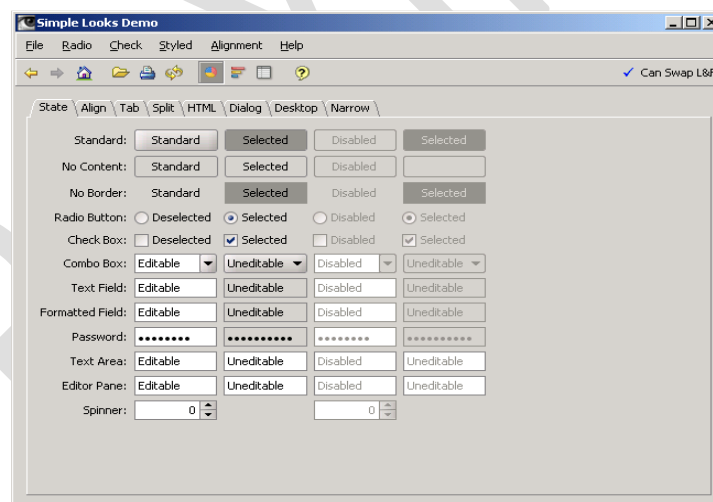
A. TUJUAN

- Memahami komponen dasar swing
- Memahami implementasi swing dalam program
- Memahami implementasi tentang event/kejadian

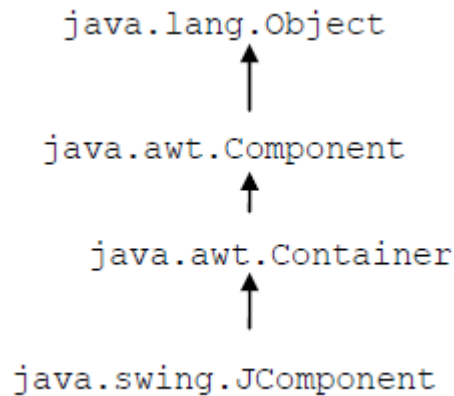
B. DASAR TEORI

1. Pemahaman Dasar Graphical User Interface

Pada modul-modul sebelumnya, program yang kita buat menggunakan *console* sebagai antarmuka. Pada modul kali ini kita akan mempelajari antarmuka grafis yang dikenal dengan nama *Graphical User Interface* (GUI). GUI merupakan antar muka grafis yang memfasilitasi interaksi antara pengguna dengan program aplikasi. Berikut adalah contoh GUI.



Salah satu komponen dalam bahasa pemrograman Java untuk membangun GUI adalah **Swing**. Komponen ini didefinisikan paket **javax** di dalam **swing**. **Swing** diturunkan dari **Abstract Windowing Toolkit** dalam paket **java.awt**. Hirarki dari komponen **Swing** adalah sebagai berikut



JComponent adalah superclass dari semua komponen Swing. Sebagian besar fungsionalitas komponen diturunkan dari superclass ini. Beberapa komponen utama dalam GUI adalah:

- Containers:** merupakan wadah yang berfungsi untuk menempatkan komponen-komponen lain di dalamnya.
- Canvas:** merupakan komponen GUI yang berfungsi untuk menampilkan gambar atau untuk membuat program grafis. Dengan canvas, kita bisa menggambar berbagai bentuk seperti lingkaran, segitiga, dll.
- User Interface (UI) components:** contohnya adalah buttons, list, simple popup menus, check boxes, text fields, dan elemen lain
- Komponen pembentuk window:** seperti frames, menu bars, windows, dan dialog boxes.

2. Komponen-komponen Dasar User Interface dengan Swing

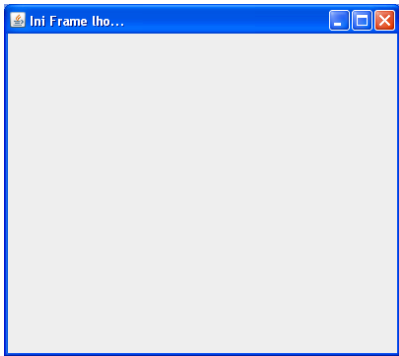
Berikut adalah beberapa komponen dasar antarmuka pengguna (user interface) yang disediakan oleh Swing:

- Top level containers:** adalah container dasar untuk meletakkan komponen-komponen lain. Contohnya adalah `JFrame`, `JDialog`, dll.
- Intermediate level containers:** merupakan container perantara. Umumnya digunakan hanya untuk meletakkan atau mengelompokkan komponen-komponen yang digunakan, baik itu container atau berupa atomic component. Contoh dari Intermediate level container adalah `JPanel` (panel).
- Atomic component:** merupakan komponen yang memiliki fungsi khusus. Umumnya pengguna langsung berinteraksi dengan komponen ini. Contohnya adalah `JButton` (tombol), `JLabel` (label), `TextField`, `TextArea` (area untuk menulis teks), dll.

Contoh:

```
import javax.swing.*;
public class FrameSederhana{
    public static void main(String[] args){
        JFrame f1;
        f1 = new JFrame("Ini Frame lho...");
        f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f1.setSize(500,500);
        f1.setVisible(true);
    }
}
```

Keluaran:



Konstruktor untuk membentuk JFrame adalah JFrame() atau JFrame(String NamaFrame). Salah satu method yang sering digunakan di kelas JFrame adalah setDefaultCloseOperation(int ops). Method ini berfungsi untuk mengatur apa yang akan dilakukan oleh program ketika tombol close di klik. Parameter method tersebut di antaranya adalah:

- JFrame.DO_NOTHING_ON_CLOSE: tidak ada satu aktivitas pun yang dilakukan secara otomatis apabila frame ditutup.
- JFrame.HIDE_ON_CLOSE: frame hanya disembunyikan, namun secara fisik frame masih ada di memori sehingga dapat dimunculkan kembali. Merupakan aktivitas default.
- JFrame.DISPOSE_ON_CLOSE: mengilangkan tampilan frame dari layar, menghilangkannya dari memori, dan membebaskan resource yang dipakai.
- JFrame.EXIT_ON_CLOSE: menghentikan eksekusi program.

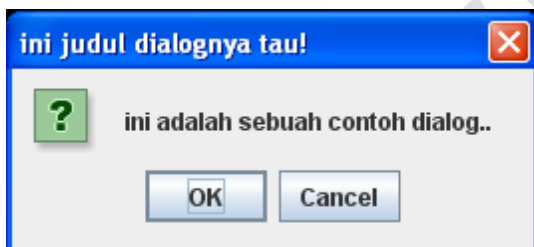
Selain JFrame, top level container yang lain adalah JDialog. Berbeda dengan JFrame, JDialog tidak dibuat berdiri sendiri, melainkan dibuat bersama-sama dengan frame sebagai parent-nya. Jika frame parent-nya ditutup, maka dialog akan dihapus dari memori. Kemunculan dialog akan

membuat semua input terhadap frame parent-nya akan terblokir sampai dialog tersebut ditutup. Berikut contoh JDialog.

```
import javax.swing.*;
public class Dialog{
    public static void main (String[] args){
        JFrame f2;
        f2 = new JFrame("Ini sebuah frame lho...");
        f2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f2.setVisible(true);

        JOptionPane.showConfirmDialog(f2,"ini adalah sebuah contoh
        dialog..", "ini judul dialognya tau!",
        JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE);
    }
}
```

Hasil:



a) *Layout Manager*

Layout manager berfungsi untuk menyusun komponen-komponen GUI di atas container. Penggunaan layout manager memudahkan programmer untuk menyusun komponen-komponen GUI dibandingkan dengan menentukan ukuran eksak dan posisi setiap komponen, sehingga para programmer akan lebih berkonsentrasi terhadap urusan "look and feel" saja.

a. **Border Layout**

BorderLayout merupakan default manager untuk ContentPane. Layout ini menempatkan komponen dengan pendekatan arah mata angin. Komponen-komponen pada layout ini ditempatkan pada posisi north, south, west, center, east. Coba ketik dan jalankan kode program berikut, yang akan menampilkan tampilan border layout berikut:

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class DemoBorderLayout extends JFrame implements ActionListener{
    private JButton tombol[];
    private String names[]={"Hilangkan North","Hilangkan South",
        "Hilangkan East","Hilangkan West",
        "Hilangkan Center"};

    private BorderLayout lout;

    public DemoBorderLayout(){
        super ("Ini Adalah Contoh Border Layout");
        Container c = getContentPane();
        lout = new BorderLayout (10,10);
        c.setLayout(lout);
        tombol = new JButton[names.length];

        for (int i=0 ; i < names.length; i++){
            tombol[i] = new JButton(names[i]);
            tombol[i].addActionListener(this);
        }
        c.add(tombol[0], BorderLayout.NORTH);
        c.add(tombol[1], BorderLayout.SOUTH);
        c.add(tombol[2], BorderLayout.EAST);
        c.add(tombol[3], BorderLayout.WEST);
        c.add(tombol[4], BorderLayout.CENTER);
        setSize (500,300);
    }

    public void actionPerformed(ActionEvent e){
        for(int i=0; i<tombol.length;i++){
            if(e.getSource()==tombol[i]){
                tombol[i].setVisible(false);
            } else tombol[i].setVisible(true);
            layout.layoutContainer (getContentPane());
        }
    }

    public static void main(String[] args) {
        DemoBorderLayout dbl = new DemoBorderLayout();
        dbl.setVisible(true);
        dbl.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Keluaran:



b. Flow Layout

FlowLayout adalah layout yang menyusun komponen dari kiri ke kanan, selanjutnya ke baris berikutnya. Jika ukuran windows diperbesar ukuran komponen pada FlowLayout tidak berubah. Method-method penting dari FlowLayout adalah sebagai berikut :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class FlowLayoutTest extends JFrame {
    public FlowLayoutTest() {
        super("Contoh Flow Layout");
        FlowLayout lay = new FlowLayout();
        lay.setAlignment(FlowLayout.RIGHT);
        lay.setVgap(25); //mengatur jarak vertikal antar komponen
        lay.setHgap(25); //mengatur jarak horizontal antar komponen
        Container c = getContentPane();
        c.setLayout(lay);
        JButton t1 = new JButton("Tombol 1");
        JButton t2 = new JButton("Tombol 2");
        JButton t3 = new JButton("Tombol 3");
        JButton t4 = new JButton("Tombol terpanjang no 4");
        JButton t5 = new JButton("Tombol 5");
        JButton t6 = new JButton("Tombol 6");
        c.add(t1);
        c.add(t2);
        c.add(t3);
        c.add(t4);
        c.add(t5);
        c.add(t6);
    }
    public static void main(String[] args) {
        FlowLayoutTest flt = new FlowLayoutTest();
        flt.setSize(500, 200);
        flt.setVisible(true);
        flt.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }
}
```

Output program di atas adalah sebagai berikut:



c. Box Layout

Layout jenis ini meletakkan komponen-komponen dalam satu baris atau satu kolom saja.

Pelajari contoh berikut.

```
import javax.swing.*;
import java.awt.*;

public class ContohBoxLayout extends JFrame{

    public ContohBoxLayout(){
        super("Contoh Box Layout Dalam Satu Baris");
        Container c = getContentPane();
        Box box = new Box(BoxLayout.X_AXIS);
        JTextArea t1 = new JTextArea("Praktikum java",10,15);
        JButton b1 = new JButton("Tombol 1");
        JButton b2 = new JButton("Tombol 2");
        JTextArea t2 = new JTextArea("Ilmu komputasi",10,15);
        box.add(new JScrollPane(t1));
        box.add(b1);
        box.add(b2);
        box.add(new JScrollPane(t2));
        c.add(box);
    }

    public static void main(String[] args){
        ContohBoxLayout cbl = new ContohBoxLayout();
        cbl.setSize(300, 300);
        cbl.setVisible(true);
        cbl.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Keluaran:



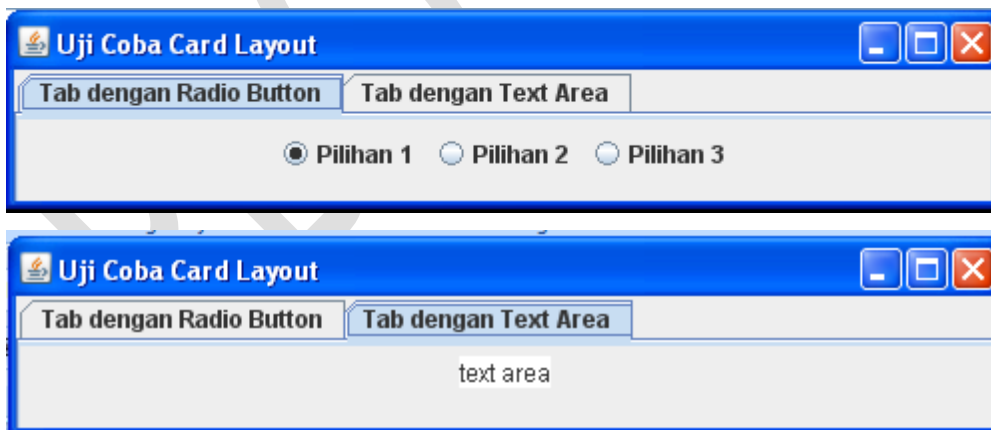
d. Card Layout

Card Layout menampilkan container-container/panel-panel seperti tumpukan kartu. Hanya satu container yang akan tampil untuk satu waktu. Pelajari kode berikut:


```
import javax.swing.*;
import java.awt.*;

public class TestCardLayout extends JFrame{
    JPanel p1;
    JPanel p2;
    JRadioButton pil1, pil2, pil3;
    ButtonGroup radioGroup;
    JTextArea g;
    public TestCardLayout(){
        super("Uji Coba Card Layout");
        p1 = new JPanel();
        p2 = new JPanel();
        pil1 = new JRadioButton("Pilihan 1",true);
        pil2 = new JRadioButton("Pilihan 2",false);
        pil3 = new JRadioButton("Pilihan 3",false);
        p1.add(pil1);
        p1.add(pil2);
        p1.add(pil3);
        radioGroup = new ButtonGroup();
        radioGroup.add(pil1);
        radioGroup.add(pil2);
        radioGroup.add(pil3);
        g = new JTextArea("text area");
        p2.add(g);
        JTabbedPane tab = new JTabbedPane();
        tab.add(p1,"Tab dengan Radio Button");
        tab.add(p2,"Tab dengan Text Area");
        Container c = getContentPane();
        c.add(tab, BorderLayout.NORTH);
    }
    public static void main (String[] args){
        TestCardLayout tcl = new TestCardLayout();
        tcl.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        tcl.setSize(500,100);
        tcl.setVisible(true);
    }
}
```

Keluaran:



e. Grid Layout

Grid layout merupakan layout manager berbentuk grid. Semua komponen dalam layout ini memiliki ukuran yang sama. Konstruktor GridLayout adalah sebagai berikut.

- 1) `GridLayout(rows, columns, hGap, vGap)`. Konstruktor ini mendefinisikan jumlah baris, kolom, dan ukuran gap horisontal/vertikal antar elemen dalam pixel.
- 2) `GridLayout(rows, columns)`. Sama halnya dengan konstruktor pertama, namun dengan nilai default `hGap` dan `vGap` sama dengan 0.

Pelajari kode program berikut:

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class GridLayoutTest extends JFrame{
    private JButton tombol[];
    private String m[] = {"Satu", "Dua", "Tiga", "Empat", "Lima", "Enam"};
    private Container c;
    private GridLayout g;
    public GridLayoutTest(){
        super("Demonstrasi GridLayout");
        g = new GridLayout(2,3,5,5);
        c = getContentPane();
        c.setLayout(g);
        tombol = new JButton[names.length];
        for(int i = 0; i<names.length;i++){
            tombol[i] = new JButton(m[i]);
            c.add(button[i]);
        }
        setSize(300,300);
        show();
    }
    public static void main(String[] args) {
        GridLayoutTest glt = new GridLayoutTest();
        glt.addWindowListener(
            new WindowAdapter(){
                public void windowsClosing(WindowEvent e){
                    System.exit(0);
                }
            }
        );
    }
}
```

Output:



f. Grid Bag Layout

Grid bag layout adalah pengaturan layout yang dapat digunakan untuk meletakkan komponen secara bebas. Setiap komponen dapat menempati ukuran yang berbeda dengan komnponen lainnya. Setiap komponen dapat menempati lebih dari satu grid.

Berikut ini langkah-langkah untuk menggunakan GridBagLayout :

- a) Gambarkan terlebih dahulu desain GUI di kertas.
- b) Bagi GUI tersebut ke dalam beberapa grid mulai dari baris dan kolom 0. Hal ini dilakukan untuk menempatkan komponen di posisi yang benar.
- c) Buat sebuah objek GridBagConstraints, untuk menentukan bagaimana komponen-komponen ditempatkan.
- d) Kemudian, definisikan variabel instance seperti berikut :
 - gridx - kolom.
 - gridy - baris.
 - gridwidth - jumlah kolom yang dialokasikan.
 - gridheight - jumlah baris yang dialokasikan.
 - weightx - ukuran ruang horisontal.
 - weighty - ukuran ruang vertikal.
- e) Inisialisasi nilai weight ke nilai positif (nilai default adalah 0).
- f) Definisikan instance variable fill dari GridBagConstraints ke NONE (default), VERTICAL, HORIZONTAL, atau BOTH.
- g) Definisikan Instance variable anchor ke NORTH, NORTHEAST, EAST, SOUTHEAST, SOUTH, SOUTHWEST, WEST, NORTHWEST, atau CENTER (nilai default).
- h) Masukkan komponen dan GridBagConstraints ke method setConstraints milik kelas GridBagLayout.
`gbLayout.setConstraints(c, gbConstraints);`
- i) Setelah constraints ditentukan, tambahkan komponen ke ContentPane.
`container.add(c);`

Untuk lebih memahami penggunaan GridBagLayout, pelajari contoh kode program berikut ini.

```
import javax.swing.*;
import java.awt.*;

public class UjiGbl{
    JFrame f;
    GridBagLayout gbl;
    GridBagConstraints c;

    public UjiGbl(){
        f = new JFrame("Contoh GridBagLayout");
        gbl = new GridBagLayout();
        c = new GridBagConstraints();
        f.getContentPane().setLayout(gbl);
        c.fill = GridBagConstraints.HORIZONTAL;

        JButton j1 = new JButton(" 1 ");
        c.weightx = 2; //ukuran ruang horizontal
        c.gridx = 0; // menempati kolom 0
        c.gridy = 0; // menempati baris 0
        gbl.setConstraints(j1,c);
        f.getContentPane().add(j1);

        JButton j2 = new JButton(" 2 ");
        c.gridx = 1; // menempati kolom 1
        c.gridy = 0; // menempati baris 0
        gbl.setConstraints(j2,c);
        f.getContentPane().add(j2);

        JButton j3 = new JButton(" 3 ");
        c.ipady = 30; //tingginya ditambah
        c.gridwidth = 2; // tombol menempati dua kolom
        c.gridx = 0; // menempati kolom 0
        c.gridy = 1; // menempati baris 1
        gbl.setConstraints(j3,c);
        f.getContentPane().add(j3);

        JButton j4 = new JButton(" 4 ");
        c.ipady = 50; //tingginya ditambah
        c.gridwidth = 1; // tombol menempati satu kolom
        c.gridx = 0; // menempati kolom 0
        c.gridy = 2; // menempati baris 2
        gbl.setConstraints(j4,c);
        f.getContentPane().add(j4);

        JButton j5 = new JButton(" 5 ");
        c.ipady = 0; //tingginya normal
        c.gridwidth = 1; // tombol menempati satu kolom
        c.gridx = 1; // menempati kolom 1
        c.gridy = 2; // menempati baris 2
        c.anchor = GridBagConstraints.SOUTH;

        gbl.setConstraints(j5,c);
        f.getContentPane().add(j5);

        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.pack();
        f.setVisible(true);
    }

    public static void main(String[] args){
        UjiGbl ugbl = new UjiGbl();
    }
}
```

Keluaran:



g. *Spring Layout*

Ini adalah pengaturan tata letak yang sangat fleksibel, dan dapat meniru fitur-fitur layout manager yang lain. Pada layout ini, ukuran komponen dapat berubah-ubah secara otomatis jika ukuran window diubah. Pelajari contoh kode program berikut:

```
import javax.swing.JTextField;
import javax.swing.JLabel;
import java.awt.Container;

public class ContohSpring{
    JFrame f;
    Container c;
    SpringLayout l;
    JLabel label;
    JTextField t;

    public ContohSpring(){
        f = new JFrame("Demo SpringLayout");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        c = f.getContentPane();
        l = new SpringLayout();
        c.setLayout(l);
        label = new JLabel("Ini label");
        t = new JTextField("Tuliskan Teks", 20);
        c.add(label);
        c.add(t);

        //mengatur batasan (constraint) untuk label sehingga berada
        //di posisi (20,10)
        l.putConstraint(SpringLayout.WEST,label,20,SpringLayout.WEST, c);
        l.putConstraint(SpringLayout.NORTH,label,10,SpringLayout.NORTH, c);

        //mengatur batasan untuk text field supaya posisinya di
        //(sisi_kanan_label+10,10)
        l.putConstraint(SpringLayout.WEST,t,10,SpringLayout.EAST, label);
        l.putConstraint(SpringLayout.NORTH,t,10,SpringLayout.NORTH, c);

        //mengatur posisi content pane: sisi kanannya 5 pixel di luar sisi
        //kanan text field dan sisi kirinya 5 pixel di bawah sisi bawah
        //komponen terpanjang
        l.putConstraint(SpringLayout.EAST,c,5,SpringLayout.EAST, t);
        l.putConstraint(SpringLayout.SOUTH,c,5,SpringLayout.SOUTH, t);
        f.pack();
        f.show();
    }

    public static void main(String[] args){
        ContohSpring cs = new ContohSpring();
    }
}
```

4. Event Handler

Event adalah peristiwa atau kejadian yang dibangkitkan atau distimulasi oleh pengguna terhadap GUI. Untuk mendeteksi dan menangani apa yang dilakukan oleh pengguna terhadap GUI, diperlukan suatu mekanisme yang disebut Event Handling. Mekanisme event handling pada bahasa Java ditangani oleh event handling component, yang terbagi menjadi dua bagian, yaitu event listener dan event handler.

Ilustrasi dari mekanisme event handling adalah sebagai berikut:

Misalkan suatu button di tekan. Maka terciptalah suatu objek event, yaitu kejadian ditekannya button. Kejadian atau event itu ditangkap oleh event listener. Selanjutnya event tersebut harus ditangani oleh program dengan memanggil method event handler. Method tersebut berisi blok kode program yang melakukan suatu proses sebagai respon atas terjadinya event tersebut.

Sebuah objek event mempunyai sebuah kelas event sebagai acuan bagi tipe datanya. Kelas event memiliki akar hirarki yaitu kelas EventObject. Kelas ini berada di package java.util. Kelas AWTEvent merupakan turunan dari kelas EventObject. Kelas AWTEvent didefinisikan dalam package java.awt.

Berikut ini adalah daftar kelas yang merupakan turunan dari kelas AWTEvent.

- a) **ComponentEvent**: turunan dari AWTEvent, dijalankan ketika sebuah komponen dijalankan, di resize, dibuat terlihat, atau disembunyikan.
- b) **InputEvent**: turunan dari kelas ComponentEvent dan Abstract root class Event untuk semua komponen input.
- c) **KeyEvent**: turunan dari kelas InputEvent, dijalankan saat sebuah key pada keyboard ditekan, dilepas, atau diketikkan.
- d) **ItemEvent**: turunan dari kelas AWTEvent, dijalankan ketika sebuah item dipilih seperti pada combo box dan list.
- e) **ActionEvent**: turunan dari kelas AWTEvent, dijalankan ketika sebuah tombol ditekan, melakukan double click pada daftar item, atau memilih menu.
- f) **MouseEvent**: turunan dari kelas InputEvent, dijalankan ketika pengguna melakukan sesuatu terhadap mouse, seperti, ditekan, dilepas, diklik, masuk atau keluar window, di drag, dll.
- g) **TextEvent**: turunan dari kelas AWTEvent, dijalankan ketika sebuah nilai text berubah
- h) **WindowEvent**: turunan dari kelas ComponentEvent, dijalankan ketika pengguna melakukan sesuatu terhadap window, seperti dibuka, ditutup, diaktifkan, atau dinon-aktifkan.

Event Listener adalah kelas yang mengimplemetasi interface *Listener*. Beberapa event listener yang sering digunakan adalah:

- **ActionListener** : listener yang bereaksi jika terjadi event pada mouse atau keyboard.
- **MouseListener** : listener yang bereaksi atas event yang terjadi pada mouse.
- **MouseMotionListener** : menyediakan beberapa method untuk memantau perubahan mouse seperti drag atau perpindahan mouse.
- **WindowListener** : listener yang bereaksi atas event yang terjadi pada window.

Untuk lebih jelasnya, pelajari dan jalankan kode program berikut:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class EventHandling{
    private JButton b1, b2;
    private JTextField t1, t2;
    private JPanel panel1, panel2;
    private JLabel l1, l2;
    private JFrame f;
    private double a, b, c;
    private String konv1 = "Suhu dalam Celcius adalah...";
    private String konv2 = "Suhu dalam Fahrenheit adalah...";

    public EventHandling(){
        f = new JFrame("Konversi Suhu");
        b1 = new JButton("Konversi ke Fahrenheit");

        b2 = new JButton("Konversi ke Celcius");
        l1 = new JLabel("Suhu dalam Celcius");
        l2 = new JLabel("Suhu dalam Fahrenheit");
        t1 = new JTextField(10);
        t2 = new JTextField(10);
    }
```

```

public void hitungSuhu() {
    panel1 = new JPanel();
    panel2 = new JPanel();
    b1.addActionListener(new FahrenheitHandler());
    b2.addActionListener(new CelciusHandler());
    panel1.add(l1);
    panel1.add(t1);
    panel1.add(b1);
    panel2.add(l2);
    panel2.add(t2);
    panel2.add(b2);
    JTabbedPane tab = new JTabbedPane();
    tab.add(panel1, "Celcius ke Fahrenheit");
    tab.add(panel2, "Fahrenheit ke Celcius");
    f.getContentPane().add(tab, BorderLayout.NORTH);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setSize(600,100);
    f.setVisible(true);
}

private class FahrenheitHandler implements ActionListener{
    public void actionPerformed(ActionEvent e) {
        try{
            a = Double.parseDouble(t1.getText());
            b = a*1.8+32;
            JOptionPane.showMessageDialog(null, konv2+b, "Hasil",
            JOptionPane.OK_CANCEL_OPTION, JOptionPane.INFORMATION_MESSAGE);
        }catch(Exception j){
            JOptionPane.showMessageDialog(null, "Masukkan Angka!", "Error",
            JOptionPane.OK_CANCEL_OPTION, JOptionPane.INFORMATION_MESSAGE);
        }
    }
}

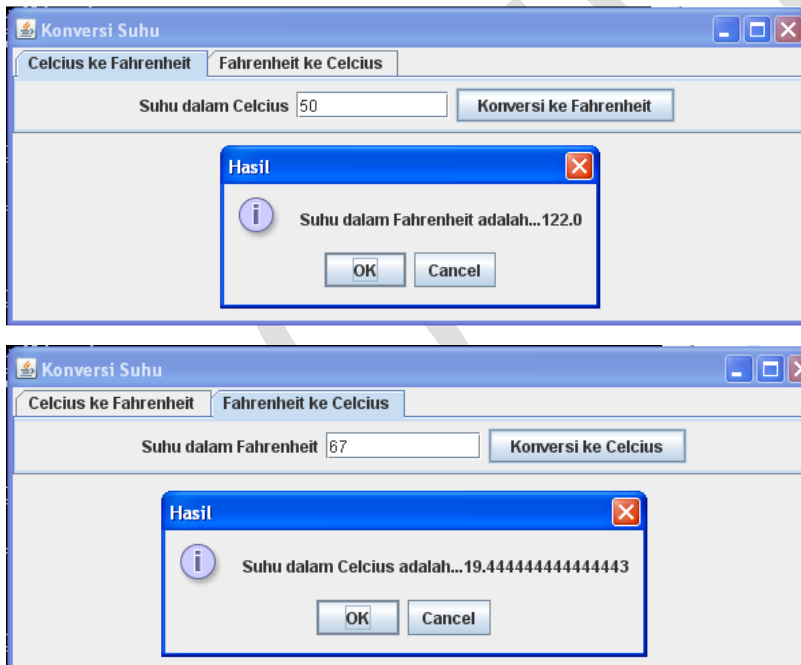
```




```
private class CelciusHandler implements ActionListener{
    public void actionPerformed(ActionEvent e){
        try{
            a = Double.parseDouble(t2.getText());
            b = (a-32)/1.8;
            JOptionPane.showMessageDialog(null,konv1+b,"Hasil",
            JOptionPane.OK_CANCEL_OPTION,JOptionPane.INFORMATION_MESSAGE);
        }catch(Exception j){
            JOptionPane.showMessageDialog(null,"Masukkan Angka!","Error",
            JOptionPane.OK_CANCEL_OPTION,JOptionPane.INFORMATION_MESSAGE);
        }
    }
}

public static void main(String[] args){
    EventHandlering eh = new EventHandlering();
    eh.hitungSuhu();
}
}
```

Output:



C. LATIHAN

Cobalah semua contoh yang terdapat dalam materi di modul!

D. TUGAS PRAKTIKUM

1. Buatlah program berbasis GUI yang tentang pendaftaran siswa baru di SMKN 2 Trenggalek dengan inputan: nama, alamat, no. Hp, asal smp, nilai UN, Jurusan yang dipilih, lalu menampilkan hasilnya!

😊😊😊😊😊 SELAMAT MENGERJAKAN 😊😊😊😊😊