

Nama : Putri Norchasana

Kelas / No : TI-2H / 24

## JOBSHEET 11

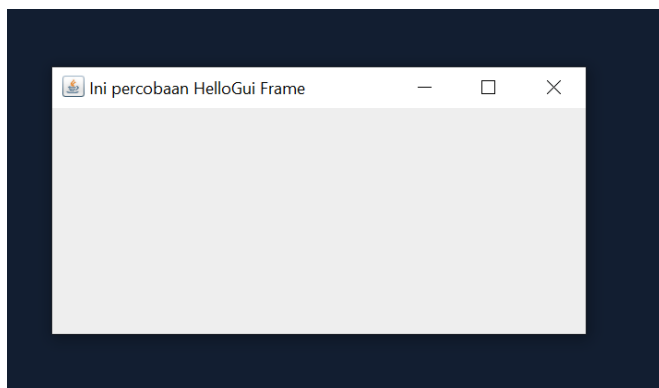
### Percobaan 1

```
package hellogui;

/**
 *
 * @author Putri Chasanah
 */
import javax.swing.*;
public class HelloGui {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        JFrame frame;
        frame = new JFrame( string:"Ini percobaan HelloGui Frame");
        //Program akan berhenti jika ditutup
        frame.setDefaultCloseOperation( operation:JFrame.EXIT_ON_CLOSE);
        frame.setSize( width:600, height:300);
        frame.setLocation( x:200, y:200); // x,y tampilan pada windows
        frame.setLocationRelativeTo( c:null);
        frame.setVisible( b:true); //untuk menampilkan frame
    }
}
```

Hasil :



## Percobaan 2

```
* @author Putri Chasanah
*/
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class MyInputForm extends JFrame{
    private static final int FRAME_WIDTH = 600;
    private static final int FRAME_HEIGHT = 200;
    private JLabel aLabel;
    private JLabel bLabel;
    private JLabel cLabel;
    private JTextField aField;
    private JTextField bField;
    private JButton button;
    private JPanel panel;

    public MyInputForm(){
        createTextField();
        createButton();
        createPanel();
        setSize( width:FRAME_WIDTH, height:FRAME_HEIGHT);
    }
}
```

```
private void createTextField(){
    aLabel = new JLabel( string:"Nilai A :");
    bLabel = new JLabel( string:"Nilai C :");
    cLabel= new JLabel( string:"Hasil : ");

    final int FIELD_WIDTH = 10;
    aField = new JTextField(1:FIELD_WIDTH);
    aField.setText( t:"0");
    bField = new JTextField(1:FIELD_WIDTH);
    bField.setText( t:"0");
}

private void createButton(){
    button = new JButton( string:"Calculate");
    class AddInterestListener implements ActionListener {
        //untuk event click
        @Override
        public void actionPerformed(ActionEvent event){
            int a = Integer.valueOf( s:aField.getText());
            int b = Integer.valueOf( s:bField.getText());
            int c = a * b;
            cLabel.setText("Hasil : " +c);
        }
    }
    ActionListener listener = new AddInterestListener();
    button.addActionListener(1:listener);
}
```

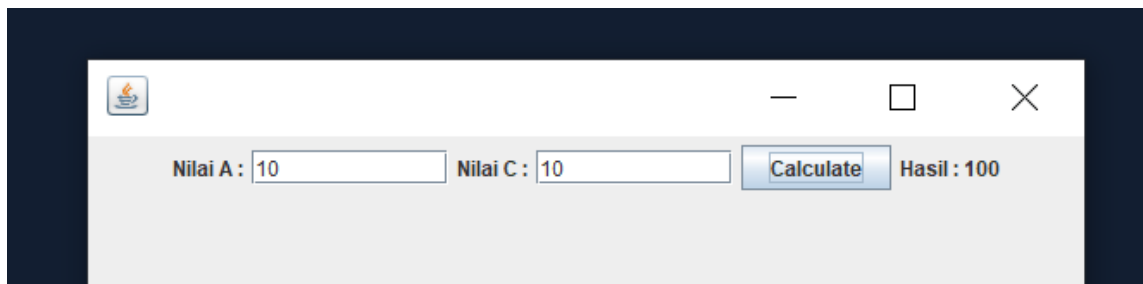
```

private void createPanel(){
    panel = new JPanel();
    panel.add( comp:aLabel);
    panel.add( comp:aField);
    panel.add( comp:bLabel);
    panel.add( comp:bField);
    panel.add( comp:button);
    panel.add( comp:cLabel);
    add( comp:panel);
}

public static void main(String[] args) {
    //TODO code application logic here
    JFrame frame = new MyInputForm();
    frame.setDefaultCloseOperation( operation:JFrame.EXIT_ON_CLOSE);
    frame.setVisible( b:true);
    frame.setLocationRelativeTo( c:null);
}
}

```

Hasil :



### Pertanyaan

- Modifikasi kode program dengan menambahkan JButton baru untuk melakukan fungsi perhitungan penambahan, sehingga ketika button di klik (event click) maka akan menampilkan hasil penambahan dari nilai A dan B

```

private void createTextField(){
    aLabel = new JLabel( string:"Nilai A :");
    bLabel = new JLabel( string:"Nilai C :");
    cLabel= new JLabel( string:"Hasil perkalian : ");
    dLabel= new JLabel( string:"Hasil penjumlahan : ");
    final int FIELD_WIDTH = 10;
    aField = new JTextField( i:FIELD_WIDTH);
    aField.setText( t:"0");
    bField = new JTextField( i:FIELD_WIDTH);
    bField.setText( t:"0");
}

```

```

        public void actionPerformed(ActionEvent event){
            int a = Integer.valueOf(aField.getText());
            int b = Integer.valueOf(bField.getText());
            int c = a * b;
            int d = a + b;
            cLabel.setText("Hasil perkalian : " + c);
            dLabel.setText("Hasil penjumlahan : " + d);
        }
    }

    ActionListener listener = new AddInterestListener();
    button.addActionListener(l:listener);
}

private void createPanel(){
    panel = new JPanel();
    panel.add(comp:aLabel);
    panel.add(comp:aField);
    panel.add(comp:bLabel);
    panel.add(comp:bField);
    panel.add(comp:button);
    panel.add(comp:cLabel);
    panel.add(comp:dLabel);
    add(comp:panel);
}
}

```



Nilai A :  Nilai C :

Hasil perkalian : 100 Hasil penjumlahan : 20

### Percobaan 3

```
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Border extends JFrame{
    private static final int FRAME_WIDTH = 600;
    private static final int FRAME_HEIGHT = 200;
    private JPanel panel;

    public Border(){
        panel = new JPanel();
        panel.setLayout(new BorderLayout());

        panel.add(new JButton( string:"1"), constraints:BorderLayout.NORTH);
        panel.add(new JButton( string:"2"), constraints:BorderLayout.WEST);
        panel.add(new JButton( string:"3"), constraints:BorderLayout.CENTER);
        panel.add(new JButton( string:"4"), constraints:BorderLayout.EAST);
        panel.add(new JButton( string:"5"), constraints:BorderLayout.SOUTH);
        add( comp:panel);
        setSize( width:FRAME_WIDTH, height:FRAME_HEIGHT);
        setTitle( title:"Border Layout Demo");
    }
}
```

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

public class Box extends JFrame {
    private static final int FRAME_WIDTH = 600;
    private static final int FRAME_HEIGHT = 200;
    private JPanel panel;

    public Box(){
        panel= new JPanel();
        panel.setLayout(new BoxLayout( cntnr:panel, i:BoxLayout.Y_AXIS));
        panel.add(new JButton( string:"1"));
        panel.add(new JButton( string:"2"));
        panel.add(new JButton( string:"3"));
        panel.add(new JButton( string:"4"));
        panel.add(new JButton( string:"5"));
        add( comp:panel);
        setSize( width:FRAME_WIDTH, height:FRAME_HEIGHT);
        setTitle( title:"Box Layout Demo");
    }
}
```

```

import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
public class Grid extends JFrame {

    private static final int FRAME_WIDTH = 600;
    private static final int FRAME_HEIGHT = 200;
    private JPanel panel;

    public Grid(){
        panel = new JPanel();
        panel.setLayout (new GridLayout (1:4, 1:2));
        panel.add(new JButton ( string: "1"));
        panel.add(new JButton ( string: "2"));
        panel.add(new JButton ( string: "3"));
        panel.add(new JButton ( string: "4"));
        panel.add(new JButton ( string: "5"));
        panel.add(new JButton ( string: "6"));
        panel.add(new JButton ( string: "7"));
        panel.add(new JButton ( string: "8"));

        add ( comp: panel);
        setSize ( width: FRAME_WIDTH, height: FRAME_HEIGHT);
        setTitle ( title: "Grid Layout Demo");
    }
}

```

```

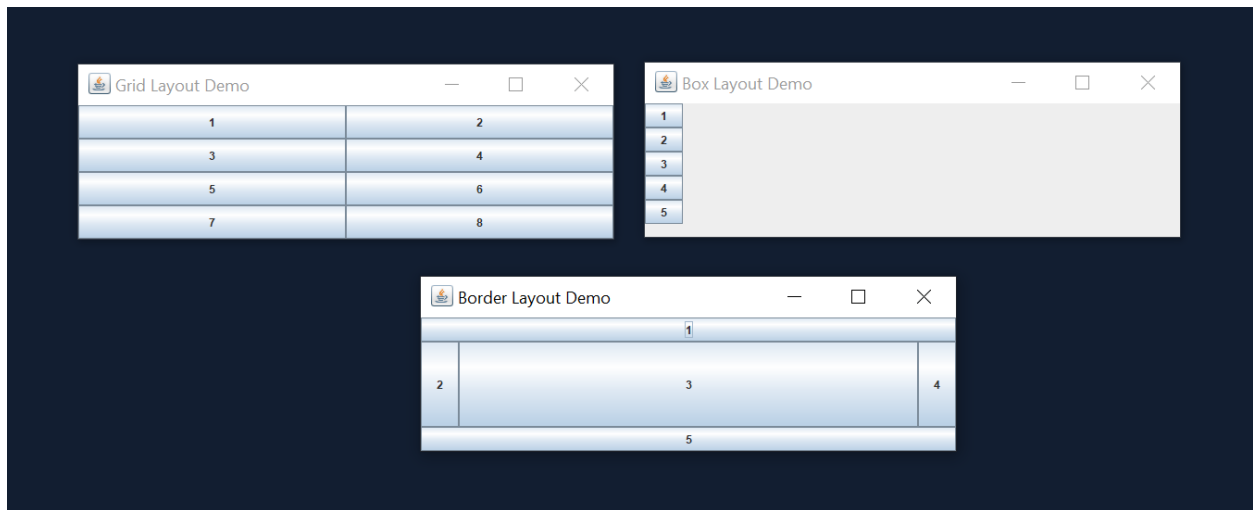
import javax.swing.*;
public class LayoutGUI {
    public static void main(String[] args) {
        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation ( operation: JFrame.EXIT_ON_CLOSE);
        frame.setVisible ( b: true);

        JFrame frame2 = new JFrame();
        frame2.setDefaultCloseOperation ( operation: JFrame.EXIT_ON_CLOSE);
        frame2.setVisible ( b: true);

        JFrame frame3 = new JFrame();
        frame3.setDefaultCloseOperation ( operation: JFrame.EXIT_ON_CLOSE);
        frame3.setVisible ( b: true);
    }
}

```

Hasil :



Pertanyaan :

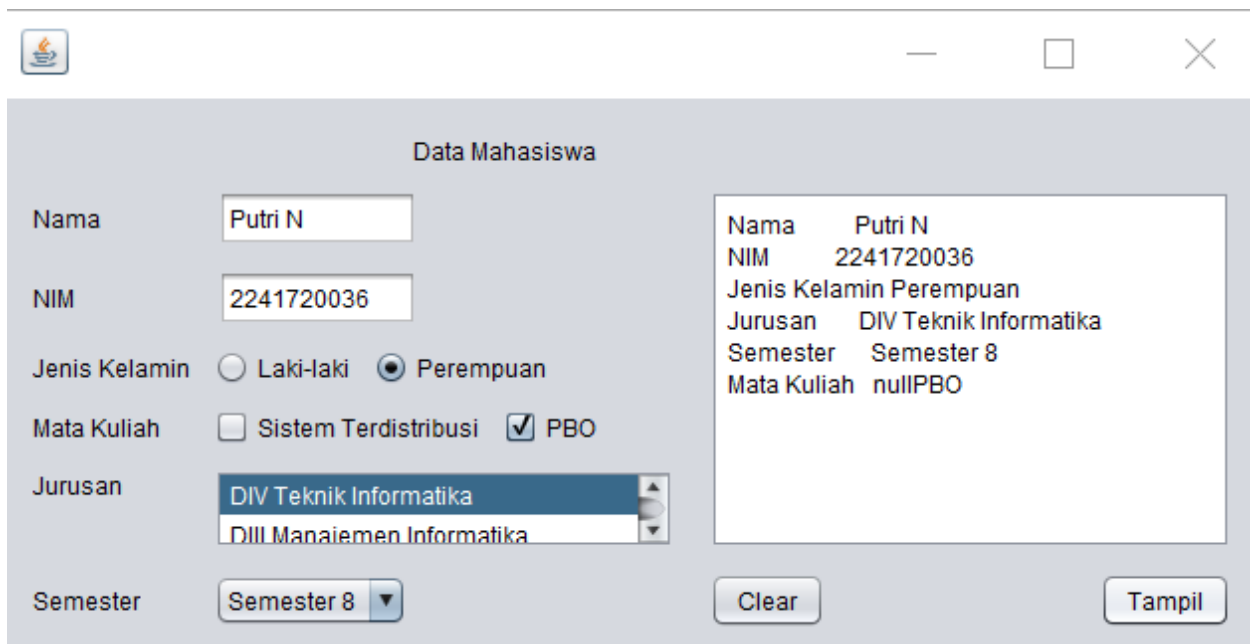
**a )Apa perbedaan dari Grid Layout, Box Layout dan Border Layout?**

- Grid Layout mengatur komponen GUI dalam bentuk grid atau tabel.
- Box Layout mengatur komponen dalam satu baris atau satu kolom.
- Border Layout mengatur komponen dalam lima area utama :NORTH,SOUTH,EAST,WEST dan CENTER.

**b) Apakah fungsi dari masing-masing kode berikut?**

- JFrame frame = new Border(); : Kode ini membuat sebuah objek dari kelas JFrame, yang merepresentasikan jendela (frame) dari aplikasi GUI. Objek frame adalah jendela utama yang akan menampung berbagai komponen GUI seperti tombol, panel, atau teks area.
- frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); : Perintah ini mengatur jendela saat tombol close (tutup) di jendela diklik. JFrame.EXIT\_ON\_CLOSE adalah perintah yang memberi tahu jendela untuk menutup aplikasi sepenuhnya saat tombol close diklik.
- frame.setVisible(true); : Perintah ini membuat jendela (frame) menjadi terlihat. Ketika nilai true diberikan sebagai argumen ke metode setVisible, jendela akan ditampilkan ke pengguna. Ini menampilkan jendela yang sebelumnya hanya dibuat sebagai objek, sehingga dapat dilihat dan diakses oleh pengguna.

#### Percobaan 4



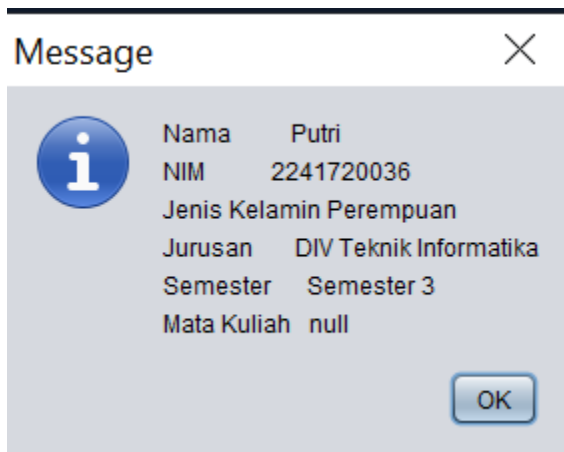
The screenshot shows a Java Swing window titled "Data Mahasiswa". It contains a form with the following fields and controls:

- Nama:** Text input field containing "Putri N".
- NIM:** Text input field containing "2241720036".
- Jenis Kelamin:** Radio buttons for "Laki-laki" and "Perempuan" (selected).
- Mata Kuliah:** Checkboxes for "Sistem Terdistribusi" and "PBO" (checked).
- Jurusan:** A list box showing "DIV Teknik Informatika" (selected) and "DIII Manaiemen Informatika".
- Semester:** A dropdown menu showing "Semester 8".

On the right side of the form, there is a preview window displaying the entered data:

Nama Putri N  
NIM 2241720036  
Jenis Kelamin Perempuan  
Jurusan DIV Teknik Informatika  
Semester Semester 8  
Mata Kuliah nullPBO

At the bottom right of the form are two buttons: "Clear" and "Tampil".



#### Pertanyaan

a) Apakah fungsi dari kode berikut?

```
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new Swing().setVisible(true);  
    }  
});
```

- `Java.awt.EventQueue.invokeLater` adalah cara untuk menjalankan sebuah blok kode dalam Event Dispatch Thread (EDT). EDT adalah thread khusus yang digunakan untuk mengelola peristiwa (events) dalam aplikasi GUI Java Swing. Menggunakan `invokeLater`



memastikan bahwa blok kode tersebut dijalankan dalam EDT, yang diperlukan untuk menjaga konsistensi dan keamanan saat berinteraksi dengan antarmuka pengguna.

- `new Runnable() {...}`: Membuat objek Runnable yang mengimplementasikan metode `run()`. Runnable ini akan dijadwalkan untuk dijalankan dalam EDT.
- `public void run() {...}`: Metode `run()` dari Runnable, yang akan dijalankan di dalam EDT.
- `new Swing().setVisible(true)`: Membuat objek dari kelas Swing dan menampilkan jendela dengan menjalankan metode `setVisible(true)`. Ini diasumsikan bahwa kelas Swing adalah kelas yang menggambarkan antarmuka pengguna (GUI) yang ingin ditampilkan.

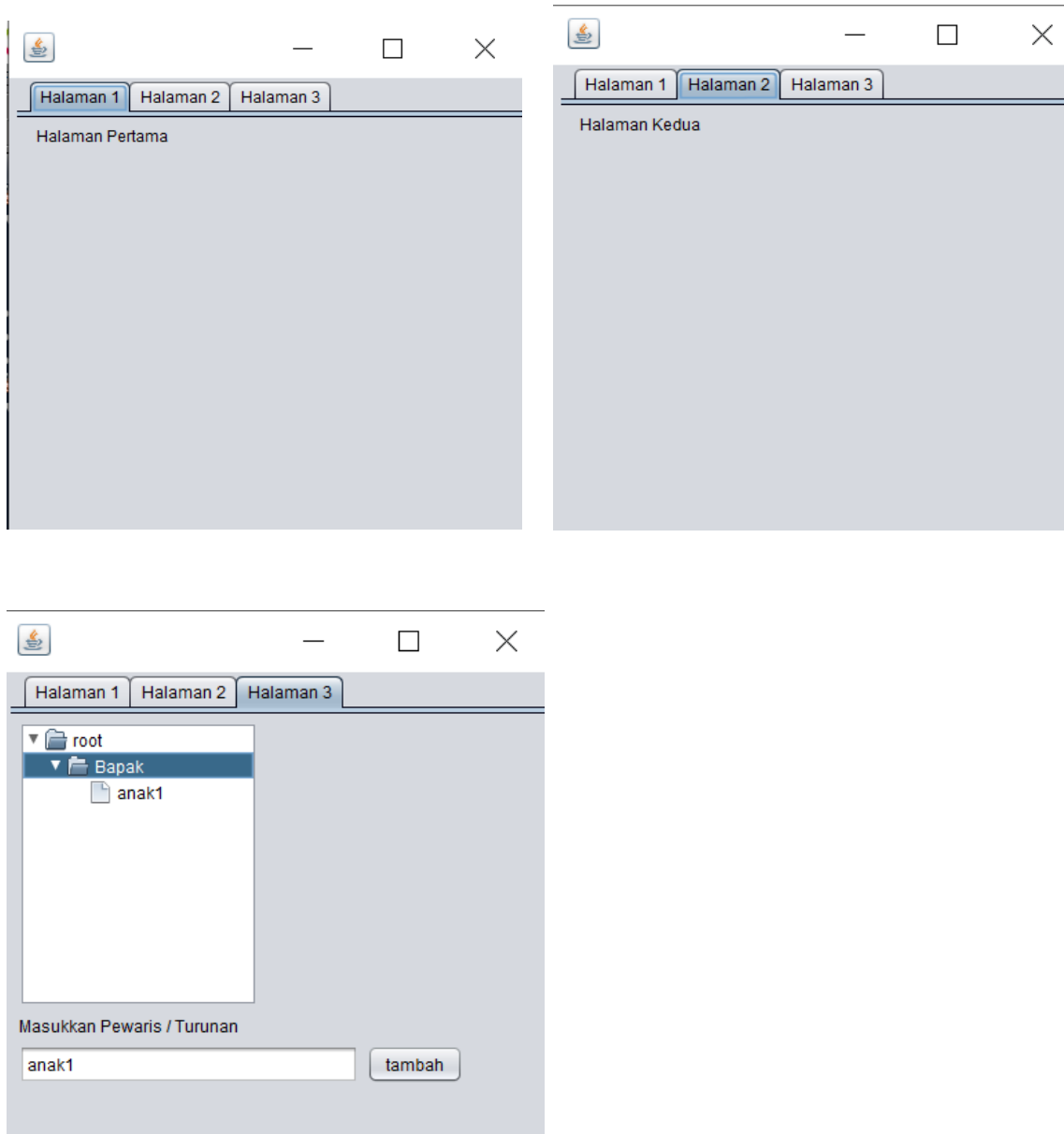
**b) Mengapa pada bagian logika checkbox dan radio button digunakan multiple if ?**

- Penanganan Kasus Khusus: Setiap kondisi if dapat mewakili satu kasus atau kondisi khusus yang perlu ditangani. Misalnya, jika checkbox diaktifkan, satu blok kode dapat dijalankan. Jika radioButton1 dipilih, blok kode lain dapat dijalankan, dan seterusnya.
- Logika Kondisional: Logika kondisional dapat berubah tergantung pada pilihan pengguna. Dengan menggunakan if, kita dapat menentukan tindakan yang berbeda sesuai dengan kombinasi atau kondisi spesifik yang mungkin terjadi.
- Pilihan Ganda: Ketika ada beberapa pilihan atau kondisi yang mungkin, if-else atau if berturut-turut dapat digunakan untuk menangani berbagai kemungkinan.

**c) Lakukan modifikasi pada program untuk melakukan menambahkan inputan berupa alamat dan berikan fungsi pemeriksaan pada nilai Alamat tersebut jika belum diisi dengan menampilkan pesan peringatan**

The screenshot shows a Java Swing window titled "Data Mahasiswa". It contains several input fields: "Nama" (filled with "Putri"), "NIM" (filled with "2241720036"), "Alamat" (empty), "Jenis Kelamin" (radio buttons for "Laki-laki" and "Perempuan"), "Mata Kuliah" (checkbox for "Sistem T"), "Jurusan" (a list box with "DIV Teknik Informatika" selected and "DIII Manajemen Informatika" below it), and "Semester" (a dropdown menu showing "Semester 3"). At the bottom are "Clear" and "Tampil" buttons. A modal dialog box titled "Peringatan" is overlaid on the "Alamat" field. It features a yellow warning triangle icon and the text "Alamat belum diisi!". An "OK" button is at the bottom right of the dialog.

## Percobaan 5



## Pertanyaan

### a) Apa kegunaan komponen swing JTabbedPane, JTree, pada percobaan 5?

- JTabbedPane digunakan untuk membuat antarmuka pengguna dengan beberapa tab, di mana setiap tab dapat berisi komponen atau panel yang berbeda. Ini membantu dalam mengatur dan menyajikan informasi yang terkait dalam satu frame.
- JTree digunakan untuk menampilkan dan menavigasi data dalam bentuk pohon. Setiap node dalam pohon dapat memiliki anak-anak, dan ini cocok untuk menampilkan data yang memiliki struktur hierarki

**b) Modifikasi program untuk menambahkan komponen JTable pada tab Halaman 1 dan tab Halaman 2**

Halaman 1 | Halaman 2 | Halaman 3

Nama	Kelas
Putri	2H
Dilla	2H

Nama  
Dilla

Kelas  
2H

tambah

Halaman 1 | Halaman 2 | Halaman 3 | Halaman 4

Nama	NIM
Putri	2H
Nadilla	2H
Daffa	2H
Dony	2H
Dilla	2H

Nama  
Dony

Kelas  
2H

tambah

## Tugas

```
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;
class calculator extends JFrame implements ActionListener {
    // create a frame
    static JFrame f;

    // create a textfield
    static JTextField l;

    // store operator and operands
    String s0, s1, s2;

    // default constructor
    calculator()
    {
        s0 = s1 = s2 = "";
    }

    // main function
    public static void main(String args[])
    {
        // create a frame
        f = new JFrame("calculator");

        try {
            // set look and feel
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        }
        catch (Exception e) {
            System.err.println(e.getMessage());
        }

        // create a object of class
        calculator c = new calculator();

        // create a textfield
        l = new JTextField(16);

        // set the textfield to non editable
        l.setEditable(false);

        // create number buttons and some operators
        JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, plus, minus, bagi, kali, be, beq1;

        // create number buttons
        b0 = new JButton("0");
        b1 = new JButton("1");
        b2 = new JButton("2");
        b3 = new JButton("3");
        b4 = new JButton("4");
        b5 = new JButton("5");
        b6 = new JButton("6");
        b7 = new JButton("7");
        b8 = new JButton("8");
        b9 = new JButton("9");

        // equals button
        beq1 = new JButton("=");

        // create operator buttons
        plus = new JButton("+");
        minus = new JButton("-");
        bagi = new JButton("/");
        kali = new JButton("*");

        // create . button
        be = new JButton(".");
```

```
// create a panel
JPanel p = new JPanel();

// add action listeners
plus.addActionListener(c);

minus.addActionListener(c);
bagi.addActionListener(c);
b9.addActionListener(c);
b8.addActionListener(c);
b7.addActionListener(c);
b6.addActionListener(c);
b5.addActionListener(c);
b4.addActionListener(c);
b3.addActionListener(c);
b2.addActionListener(c);
b1.addActionListener(c);
b0.addActionListener(c);
be.addActionListener(c);
beq1.addActionListener(c);

// add elements to panel
p.add(l);
p.add(b7);
p.add(b8);
p.add(b2);
p.add(b9);
p.add(bagi);
p.add(b4);
p.add(b5);
p.add(b6);
p.add(kali);
p.add(b1);
p.add(b2);
p.add(b3);
p.add(minus);
p.add(b0);
p.add(be);
p.add(plus);
p.add(beq1);

// add panel to frame
f.add(p);

f.setSize(200, 220);
f.show();
}
```

```

public void actionPerformed(ActionEvent e)
{
    String s = e.getActionCommand();

    // if the value is a number
    if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') || s.charAt(0) == '.') {
        // if operand is present then add to second no
        if (!s1.equals(""))
            s2 = s2 + s;
        else
            s0 = s0 + s;

        // set the value of text
        l.setText(s0 + s1 + s2);
    }
    else if (s.charAt(0) == 'C') {
        // clear the one letter
        s0 = s1 = s2 = "";

        // set the value of text
        l.setText(s0 + s1 + s2);
    }
    else if (s.charAt(0) == '=') {
        double te;

        // store the value in 1st
        if (s1.equals("+"))
            te = (Double.parseDouble(s0) + Double.parseDouble(s2));
        else if (s1.equals("-"))
            te = (Double.parseDouble(s0) - Double.parseDouble(s2));
        else if (s1.equals("/"))
            te = (Double.parseDouble(s0) / Double.parseDouble(s2));
        else
            te = (Double.parseDouble(s0) * Double.parseDouble(s2));

        // set the value of text
        l.setText(s0 + s1 + s2 + "=" + te);

        // convert it to string
        s0 = Double.toString(te);

        s1 = s2 = "";
    }
    else {
        // if there was no operand
        if (s1.equals("") || s2.equals(""))
            s1 = s;
        // else evaluate
        else {
            double te;

            // store the value in 1st
            if (s1.equals("+"))
                te = (Double.parseDouble(s0) + Double.parseDouble(s2));
            else if (s1.equals("-"))
                te = (Double.parseDouble(s0) - Double.parseDouble(s2));
            else if (s1.equals("/"))
                te = (Double.parseDouble(s0) / Double.parseDouble(s2));
            else
                te = (Double.parseDouble(s0) * Double.parseDouble(s2));

            // convert it to string
            s0 = Double.toString(te);

            // place the operator
            s1 = s;

            // make the operand blank
            s2 = "";
        }

        // set the value of text
        l.setText(s0 + s1 + s2);
    }
}
}

```

