

# LAPORAN PRAKTIKUM PRAKTIK PEMROGRAMAN

OLEH: HELGA ARYA PRAYOGA (24051130022)

---



## MODUL 9

**TOPIK:**

EVENT HANDLING/ACTION PROGRAM



# TABLE OF CONTENTS

<b>Week #9 .....</b>	<b>1</b>
<b>A. Penjelasan Tugas Praktikum.....</b>	<b>3</b>
<b>B. Langkah-langkah dan Screenshot .....</b>	<b>3</b>
<b>C. Kendala yang Dialami .....</b>	<b>12</b>
<b>D. Kesimpulan.....</b>	<b>12</b>

## A. Penjelasan Tugas Praktikum

1. Program ActionListener pada mouse
2. Program ActionListener pada keyboard
3. Penjelasan delegation event model dan macam – macam delegation event model
4. Penjelasan event listener dan macam – macam event listener serta method – methodnya
5. Program kalkulator sederhana
6. Program percobaan event handling pada java

## B. Langkah-langkah dan Screenshot

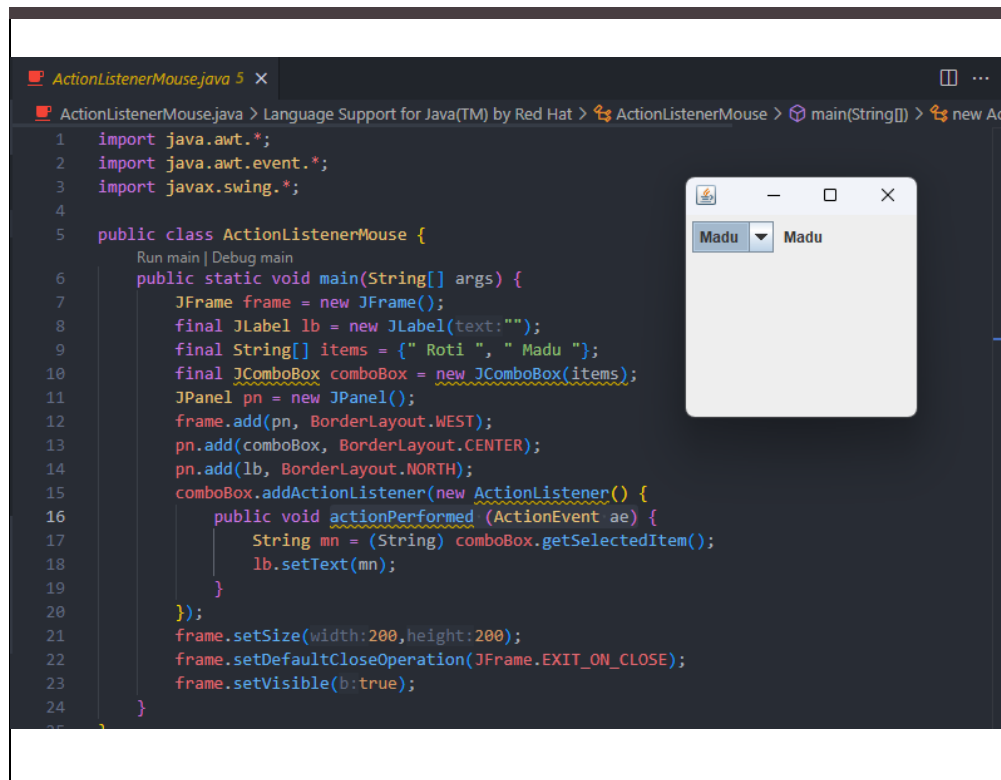
1. Program ActionListener pada mouse

Kode Program:

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

public class ActionListenerMouse {
    public static void main(String[] args) {
        JFrame frame = new JFrame();
        final JLabel lb = new JLabel("");
        final String[] items = {" Roti ", " Madu "};
        final JComboBox comboBox = new JComboBox(items);
        JPanel pn = new JPanel();
        frame.add(pn, BorderLayout.WEST);
        pn.add(comboBox, BorderLayout.CENTER);
        pn.add(lb, BorderLayout.NORTH);
        comboBox.addActionListener(new ActionListener() {
            public void actionPerformed (ActionEvent ae) {
                String mn = (String)
comboBox.getSelectedItem();
                lb.setText(mn);
            }
        });
        frame.setSize(200,200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Screenshot:



## 2. Program ActionListener pada keyboard

Kode Program:

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

class ActionListenerKeyboard extends JFrame {
    JButton m = new JButton("TEKAN KEY 1,2,3,4");
    JLabel tulisan = new JLabel ("UKURAN FRAME AKAN
    BERUBAH");

    ActionListenerKeyboard() {
        setTitle("INI EVENT PADA BUTTON");
```

```

        setSize(400,200);
        setLocation(200,100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

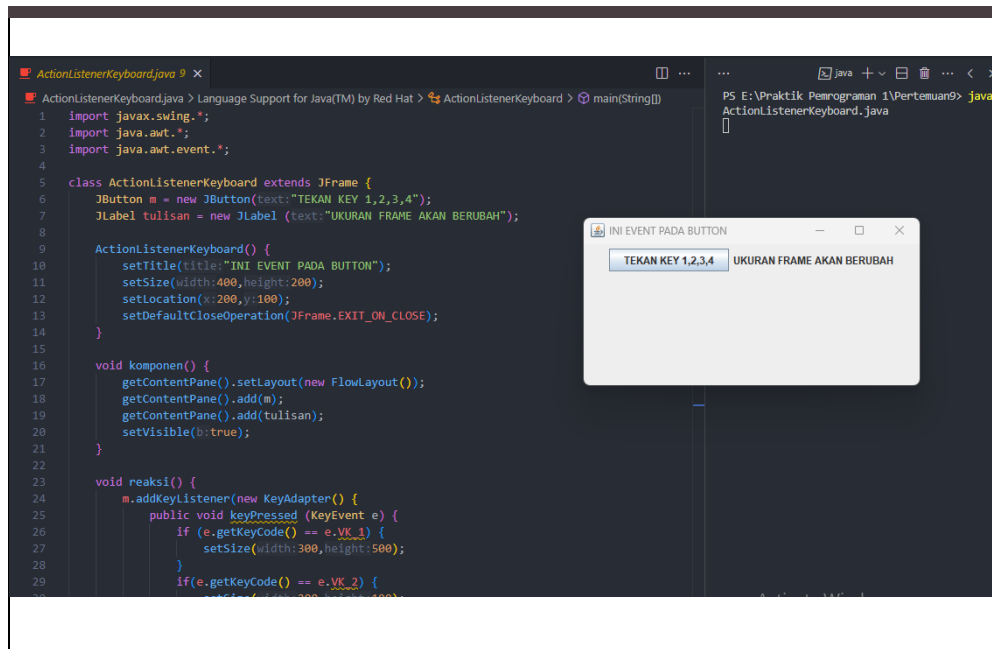
    void komponen() {
        getContentPane().setLayout(new FlowLayout());
        getContentPane().add(m);
        getContentPane().add(tulisan);
        setVisible(true);
    }

    void reaksi() {
        m.addKeyListener(new KeyAdapter() {
            public void keyPressed (KeyEvent e) {
                if (e.getKeyCode() == e.VK_1) {
                    setSize(300,500);
                }
                if(e.getKeyCode() == e.VK_2) {
                    setSize(200,100);
                }
                if(e.getKeyCode() == e.VK_3) {
                    setSize(100,10);
                }
                if(e.getKeyCode() == e.VK_4) {
                    setSize(900,200);
                }
            }
        });
    }

    public static void main (String[] args) {
        ActionListenerKeyboard eb = new
ActionListenerKeyboard();
        eb.komponen();
        eb.reaksi();
    }
}

```

Screenshot:



3. Delegation event model menguraikan bagaimana program Anda dapat merespon interaksi dari user. Terdapat 3 macam delegation event model, yaitu :
  - a. Event Source  
Objek yang memicu event. Contohnya adalah tombol (JButton), yang bisa memicu event ketika diklik oleh pengguna.
  - b. Event Listener (Pendengar Event)  
Objek yang mendengarkan event dari source. Listener ini akan menangani aksi yang dihasilkan dari event tersebut.
  - c. Event Object (Objek Event)  
Objek yang membawa informasi tentang event yang terjadi, seperti jenis event, sumber event, dan data tambahan lainnya. Contoh: ActionEvent, MouseEvent.
4. Event listeners adalah class yang mengimplementasikan interfaces <Type>Listener. Terdapat 3 macam event listeners pada java, yaitu :
  - a. Action Listener  
Digunakan untuk menangani event seperti reaksi perubahan pada mouse atau keyboard  
Method :
    - void actionPerformed(ActionEvent e)
  - b. Mouse Listener

Digunakan untuk menangani event seperti reaksi pada pergerakan mouse

Method :

- void mouseClicked(MouseEvent e)
- void mousePressed(MouseEvent e)
- void mouseReleased(MouseEvent e)
- void mouseEntered(MouseEvent e)
- void mouseExited(MouseEvent e)

c. Mouse Motion Listener

Digunakan untuk menangani event seperti drag and drop pada mouse

Method :

- void mouseDragged(MouseEvent e)
- void mouseMoved(MouseEvent e)

## 5. Program kalkulator sederhana

Kode Program:

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

class KalkulatorSederhana extends JFrame{
    JLabel angkaPertama = new JLabel ("Masukkan Angka
Pertama");
    JTextField textFieldAngkaPertama = new JTextField();
    JLabel angkaKedua = new JLabel ("Masukkan Angka Kedua");
    JTextField textFieldAngkaKedua = new JTextField();
    JLabel hasil = new JLabel ("Hasil");
    JTextField textFieldHasil = new JTextField();
    JButton btnTambah = new JButton("+");
    JButton btnKurang = new JButton("-");
    JButton btnKali = new JButton("*");
    JButton btnBagi = new JButton("/");

    KalkulatorSederhana() {
        setTitle("Kalkulator Sederhana");
        setSize(250,400);
        setLocation(200,100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    void komponen() {
        getContentPane().setLayout(null);
        angkaPertama.setBounds(40, 20, 200, 20);
        textFieldAngkaPertama.setBounds(20, 45, 200, 25);
        angkaKedua.setBounds(50, 80, 200, 20);
        textFieldAngkaKedua.setBounds(20, 105, 200, 25);
```

```

        hasil.setBounds(100, 140, 200, 20);
        textFieldHasil.setBounds(20, 165, 200, 25);
        textFieldHasil.setEditable(false);
        btnTambah.setBounds(20, 200, 50, 40);
        btnKurang.setBounds(95, 200, 50, 40);
        btnKali.setBounds(170, 200, 50, 40);
        btnBagi.setBounds(95, 265, 50, 40);
        getContentPane().add(angkaPertama);
        getContentPane().add(textFieldAngkaPertama);
        getContentPane().add(angkaKedua);
        getContentPane().add(textFieldAngkaKedua);
        getContentPane().add(hasil);
        getContentPane().add(textFieldHasil);
        getContentPane().add(btnTambah);
        getContentPane().add(btnKurang);
        getContentPane().add(btnKali);
        getContentPane().add(btnBagi);
        setVisible(true);
    }

    void reaksi() {
        textFieldAngkaPertama.addKeyListener(new
        KeyAdapter() {
            @Override
            public void keyTyped(KeyEvent e) {
                char c = e.getKeyChar();
                if (!Character.isDigit(c)) {
                    e.consume();
                }
            }
        });

        textFieldAngkaKedua.addKeyListener(new KeyAdapter()
        {
            @Override
            public void keyTyped(KeyEvent e) {
                char c = e.getKeyChar();
                if (!Character.isDigit(c)) {
                    e.consume();
                }
            }
        });

        btnTambah.addActionListener(new ActionListener() {
            public void actionPerformed (ActionEvent ae) {
                try {
                    double angkaPertama =
Double.parseDouble(textFieldAngkaPertama.getText());
                    double angkaKedua =
Double.parseDouble(textFieldAngkaKedua.getText());
                    double hasil = angkaPertama +
angkaKedua;
                    textFieldHasil.setText(String.valueOf(hasil));
                } catch (NumberFormatException ex) {

```



```

        JOptionPane.showMessageDialog(null,
"Masukkan angka yang valid!");
    }
}

});

    btnKurang.addActionListener(new ActionListener() {
        public void actionPerformed (ActionEvent ae) {
            try {
                double angkaPertama =
Double.parseDouble(textFieldAngkaPertama.getText());
                double angkaKedua =
Double.parseDouble(textFieldAngkaKedua.getText());
                double hasil = angkaPertama -
angkaKedua;

textFieldHasil.setText(String.valueOf(hasil));
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(null,
"Masukkan angka yang valid!");
            }
        }
    });

    btnKali.addActionListener(new ActionListener() {
        public void actionPerformed (ActionEvent ae) {
            try {
                double angkaPertama =
Double.parseDouble(textFieldAngkaPertama.getText());
                double angkaKedua =
Double.parseDouble(textFieldAngkaKedua.getText());
                double hasil = angkaPertama *
angkaKedua;

textFieldHasil.setText(String.valueOf(hasil));
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(null,
"Masukkan angka yang valid!");
            }
        }
    });

    btnBagi.addActionListener(new ActionListener() {
        public void actionPerformed (ActionEvent ae) {
            try {
                double angkaPertama =
Double.parseDouble(textFieldAngkaPertama.getText());
                double angkaKedua =
Double.parseDouble(textFieldAngkaKedua.getText());
                double hasil = angkaPertama /
angkaKedua;

textFieldHasil.setText(String.valueOf(hasil));
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(null,
"Masukkan angka yang valid!");
            }
        }
    });

```

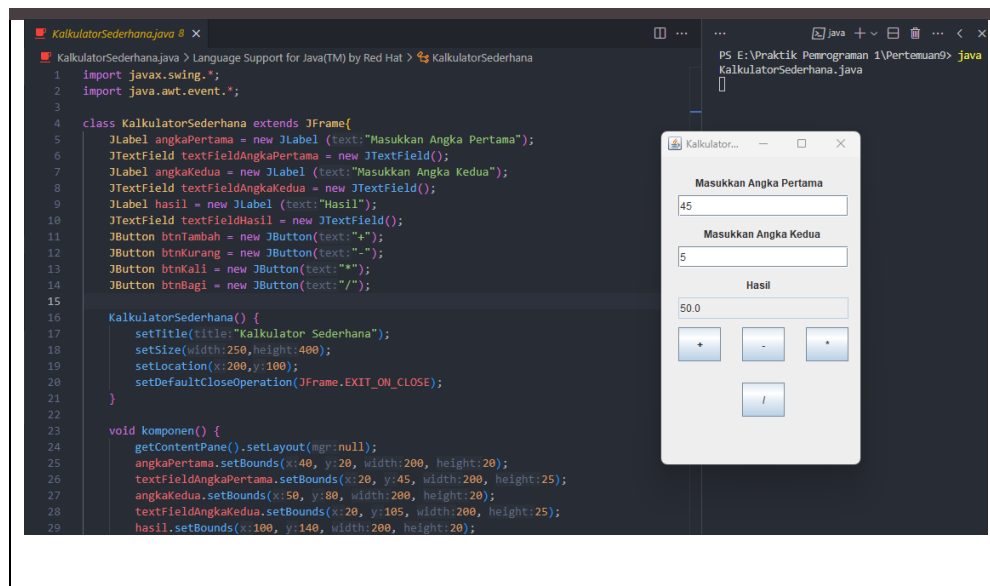
```

    }
    }
    });
}

public static void main (String[] args) {
    KalkulatorSederhana eb = new KalkulatorSederhana();
    eb.komponen();
    eb.reaksi();
}
}

```

*Screenshot:*



## 6. Program percobaan event handling

Kode Program:

```

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

public class CobaAksiReaksi extends JFrame {
    JTextArea nama = new JTextArea(10,10);
    JButton bt = new JButton("Copy");
    JTextArea txnama = new JTextArea(10,10);

    CobaAksiReaksi() {
        super("Coba Event Handling");
        setLocation(200,300);
        setSize(400,300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
}

```

```

    }

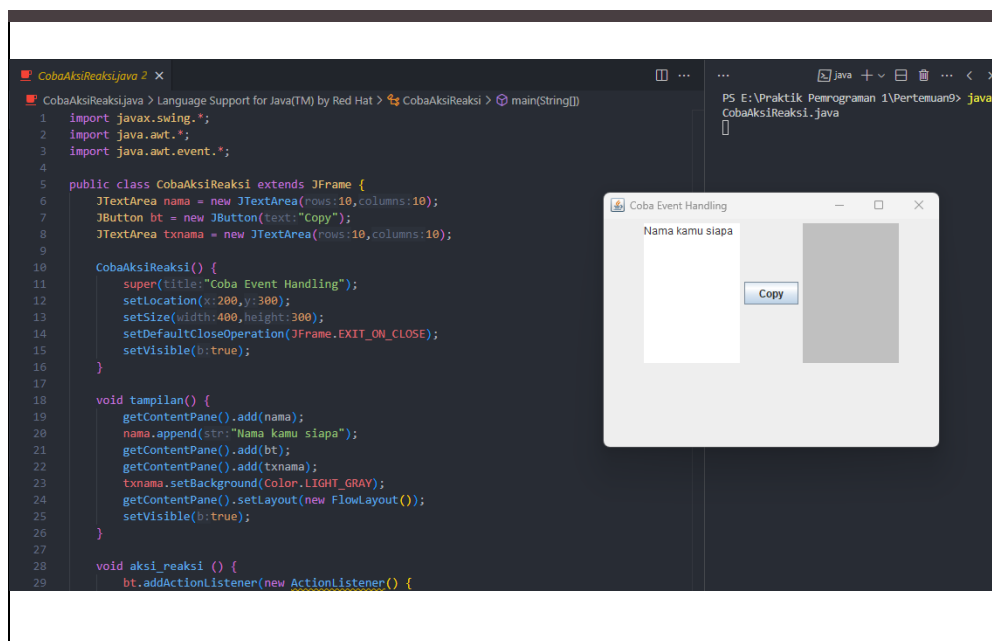
    void tampilan() {
        getContentPane().add(nama);
        nama.append("Nama kamu siapa");
        getContentPane().add(bt);
        getContentPane().add(txnama);
        txnama.setBackground(Color.LIGHT_GRAY);
        getContentPane().setLayout(new FlowLayout());
        setVisible(true);
    }

    void aksi_reaksi () {
        bt.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                txnama.append(nama.getSelectedText());
            }
        });
    }

    public static void main(String[] args) {
        CobaAksiReaksi f = new CobaAksiReaksi();
        f.tampilan();
        f.aksi_reaksi();
    }
}

```

*Screenshot:*



### **C. Kendala yang Dialami**

Tidak ada kendala dalam membuat program

### **D. Kesimpulan**

Delegation event model menguraikan bagaimana program Anda dapat merespon interaksi dari user. Terdapat 3 macam delegation event model, yaitu Event Source, Event Listener, dan Event Object. Event listeners adalah class yang mengimplementasikan interfaces <Type>Listener. Terdapat 3 macam event listeners pada java, yaitu Action Listener, Mouse Listener, Mouse Motion Listener.