

**LAPORAN**  
**PRAKTIKUM PEMROGRAMAN II**  
(Pertemuan 4)



Ahmad Mulia Huda (223040029)

***JURUSAN TEKNIK INFORMATIKA***  
**UNIVERSITAS PASUNDAN**  
**BANDUNG**  
**2024**

## 1. ButtonExample

Kode ini membuat GUI sederhana dengan JFrame dan JButton. Button ditambahkan ke frame dan diberi ActionListener. Ketika button di klik, akan mencetak pesan "Button diklik" ke konsol.

```
package Latihan4;

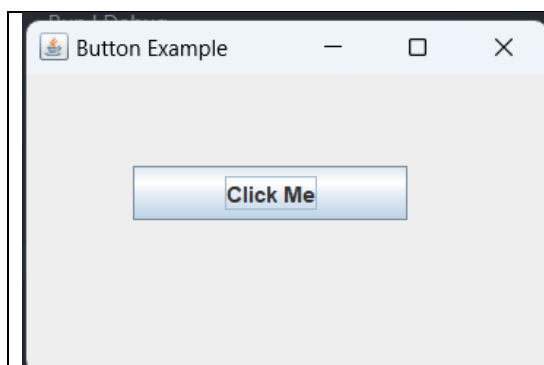
import javax.swing.*;
import java.awt.event.*;

public class ButtonExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Button Example");
        JButton button = new JButton("Click Me");

        // menambahkan ActionListener ke JButton
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.out.println("Button clicked");
            }
        });

        button.setBounds(59, 50, 150, 30);
        frame.add(button);
        frame.setSize(300, 200);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Output



## 2. MouseListenerExample

Program ini membuat JFrame dengan JPanel. MouseListener ditambahkan ke panel untuk mendeteksi berbagai event mouse seperti klik, press, release, enter, dan exit. Setiap event akan mencetak pesan yang sesuai console.

Kode:

```
package Latihan4;

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

public class MouseListenerExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("MouseListener Example");

        JLabel label = new JLabel("Arahkan dan klik mouse pada area ini");
        label.setBounds(50, 50, 300, 30);

        label.addMouseListener(new MouseListener() {
            public void mouseClicked(MouseEvent e) {
                label.setText("Mouse Clicked at: (" + e.getX() + ", " + e.getY() + ")");
            }

            public void mousePressed(MouseEvent e) {
                label.setText("Mouse Pressed at: (" + e.getX() + ", " + e.getY() + ")");
            }

            public void mouseReleased(MouseEvent e) {
                label.setText("Mouse Released at: (" + e.getX() + ", " + e.getY() + ")");
            }

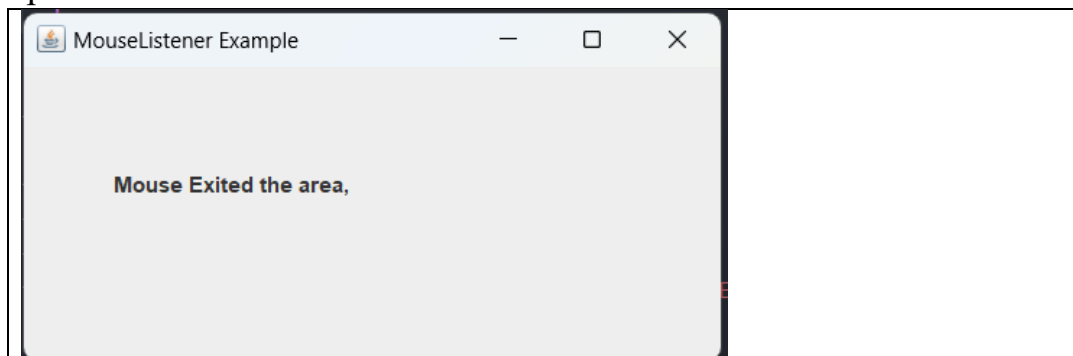
            public void mouseEntered(MouseEvent e) {
                label.setText("Mouse Entered the area.");
            }

            public void mouseExited(MouseEvent e) {
                label.setText("Mouse Exited the area.");
            }
        });

        frame.add(label);

        frame.setSize(400, 200);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

## Output



### 3. WindowListenerExample

Kode ini membuat JFrame dengan WindowListener. Listener ini menangani berbagai event jendela seperti opening, closing, activated, deactivated dll.

Setiap event akan mencetak pesan yang sesuai ke console.

Kode:

```
package Latihan4;

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

public class WindowListenerExample {
    public static void main(String[] args) {
        JFrame f = new JFrame("Window Listener Example");

        JLabel label = new JLabel("lakukan operasi pada jendela");
        label.setBounds(50, 50, 300, 30);

        f.addWindowListener(new WindowListener() {

            public void windowOpened(WindowEvent e) {
                label.setText("Window Opened");
            }

            public void windowClosing(WindowEvent e) {
                label.setText("Window Closing");
            }

            public void windowClosed(WindowEvent e) {
                System.out.println("Window Closed");
            }

            public void windowIconified(WindowEvent e) {
                label.setText("Window Minimized");
            }

            public void windowDeiconified(WindowEvent e) {
                label.setText("Window Restored");
            }

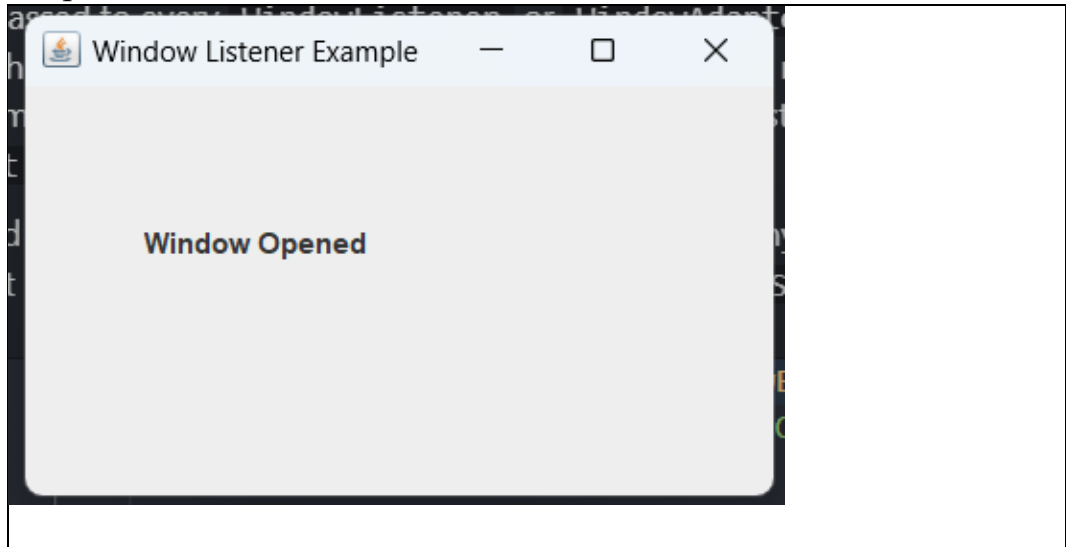
            public void windowActivated(WindowEvent e) {
                label.setText("Window Activated");
            }

            public void windowDeactivated(WindowEvent e) {
                label.setText("Window Deactivated");
            }

        });

        f.add(label);
        f.setSize(400, 400);
        f.setLayout(null);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

## Output



#### 4. KeyListenenrExample

Program ini membuat JFrame dengan JTextField. KeyListener ditambahkan ke text field untuk mendeteksi event keyboard. Saat tombol ditekan , dilepas, atau di ketik, akan mencetak informasi.

```
package Latihan4;

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

public class keyListenerExample {

    public static void main(String[] args){
        // Membuat frame
        JFrame frame = new JFrame("KeyListener Example");

        // Membuat label untuk menampilkan pesan
        JLabel label = new JLabel("Tekan tombol pada keyboard");

        // Membuat text field untuk fokus keyboard
        JTextField textField = new JTextField();
        textField.setBounds(50, 100, 200, 30);

        // Menambahkan KeyListener ke text field
        textField.addKeyListener(new KeyListener() {
            // Dijalankan ketika tombol ditekan
            @Override
            public void keyPressed(KeyEvent e) {
                label.setText("Key Pressed: " + KeyEvent.getKeyText(e.getKeyCode()));
            }

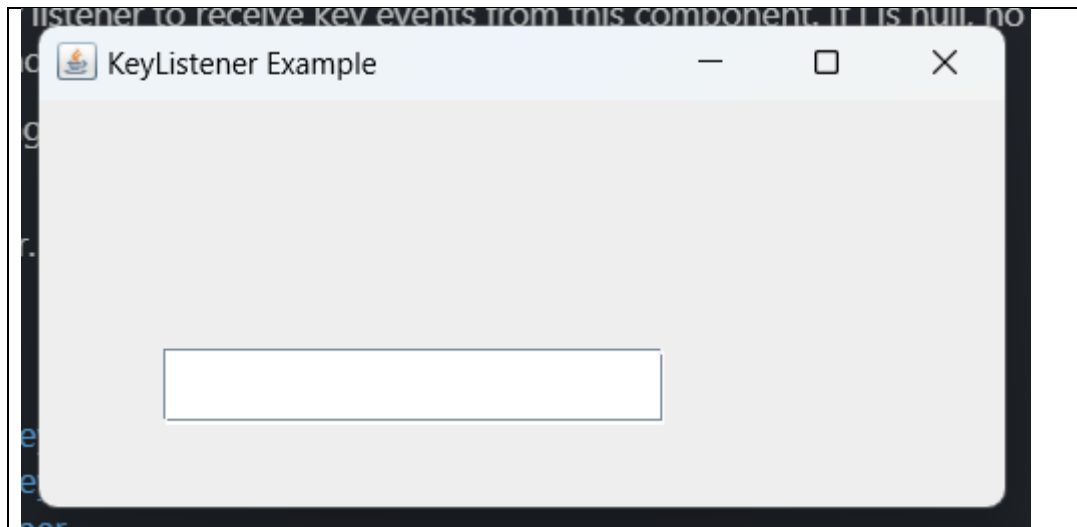
            // Dijalankan ketika tombol dilepaskan
            @Override
            public void keyReleased(KeyEvent e) {
                label.setText("Key Released: " + KeyEvent.getKeyText(e.getKeyCode()));
            }

            // Dijalankan ketika tombol ditekan dan dilepaskan (sama dengan mengetik
            karakter)
            @Override
            public void keyTyped(KeyEvent e) {
                label.setText("Key Typed: " + e.getKeyChar());
            }
        });

        // Menambahkan komponen ke frame
        frame.add(label);
        frame.add(textField);

        // Setting frame
        frame.setSize(400, 200);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Output:



Semua kode di atas mendemonstrasikan oenggunaan berbagai jenis listener dalam pemrograman GUI java untuk menangani interaksi pengguna seoerti klik mouse, penekanan tombol, dan event jendela.