

LAPORAN PRAKTIKUM PEMOGRAMAN II



Novan Ramdan (223040031)

**FAKULTAS TEKNIK
UNIVERSITAS PASUNDAN
BANDUNG 2024**

1. ButtonExample.java

```
Untitled-1

import javax.swing.*;/**
 * ButtonExample
 */
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(50, 50, 150, 30);
        frame.add(button);
        frame.setSize(300, 200);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

- ```
Untitled-1

JFrame frame = new JFrame("Button Example");
JButton button = new JButton("Click me");
```

  - Membuat frame dengan judul “Button example”
  - Membuat tombol yang isinya “Click me”

- ```
Untitled-1

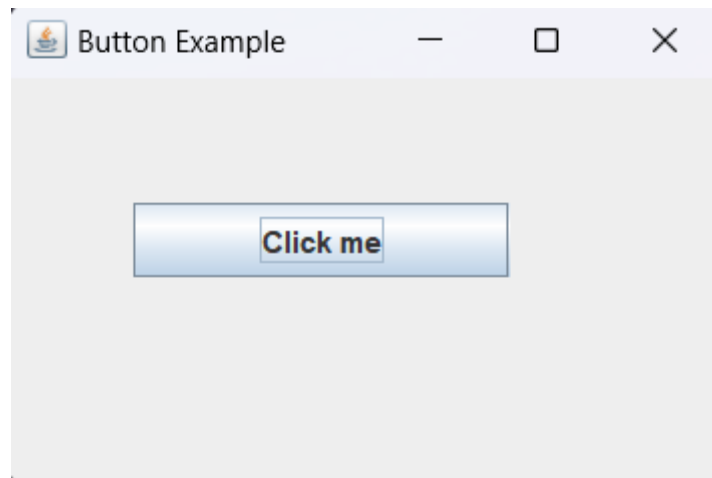
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("Button clicked!");
    }
});
```

 - Membuat method actionPerformed Dimana method ini akan dipanggil Ketika button di klik.

```
Untitled-1  
button.setBounds(50, 50, 150, 30);  
frame.add(button);  
frame.setSize(300, 200);  
frame.setLayout(null);  
frame.setVisible(true);  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

- Mengatur ukuran button,
- mengatur ukuran jendela menjadi 300, 200,
- mengatur jendela menjadi terlihat(true)
- setDefaultCloseOperation(): Menutup aplikasi saat jendela ditutup.

Output:



2. MouseListener.java

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

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

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

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

            // Dijalankan ketika mouse ditekan tanpa dilepaskan
            public void mousePressed(MouseEvent e) {
                label.setText("Mouse Pressed at: (" + e.getX() + ", " + e.getY() + ")");
            }

            // Dijalankan ketika mouse dilepaskan setelah ditekan
            public void mouseReleased(MouseEvent e) {
                label.setText("Mouse Released at: (" + e.getX() + ", " + e.getY() + ")");
            }

            // Dijalankan ketika mouse masuk ke area komponen
            public void mouseEntered(MouseEvent e) {
                label.setText("Mouse Entered the area.");
            }

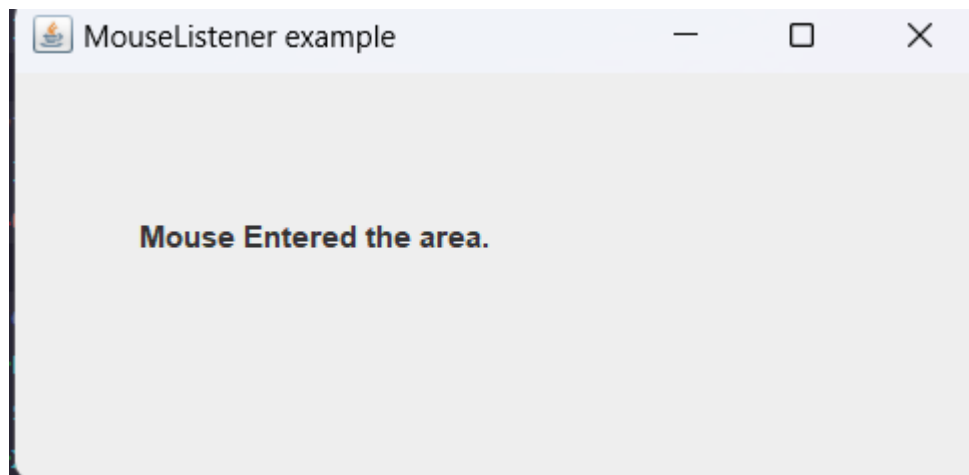
            // Dijalankan ketika mouse keluar dari area komponen
            public void mouseExited(MouseEvent e){
                label.setText("Mouse exited the area.");
            }
        });

        // menambahkan label ke frame
        frame.add(label);

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

- `addMouseListener()` menambahkan *listener* yang akan menangkap dan merespon *event* dari interaksi mouse dengan label.
1. **`mouseClicked()`**: Dijalankan saat mouse diklik (ditekan dan dilepaskan) pada area label. Koordinat klik akan ditampilkan di label.
 2. **`mousePressed()`**: Dijalankan saat mouse ditekan tanpa dilepaskan.
 3. **`mouseReleased()`**: Dijalankan saat mouse dilepaskan setelah ditekan.
 4. **`mouseEntered()`**: Dijalankan saat kursor mouse masuk ke area label.
 5. **`mouseExited()`**: Dijalankan saat kursor keluar dari area label.

Output:



3. KeylistenerExample.java

```
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.");
        label.setBounds(50,50,300,30);

        // 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
            public void keyPressed(KeyEvent e){
                label.setText("Key Pressed: " + KeyEvent.getKeyText(e.getKeyCode()));
            }

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

            // Dijalankan ketika tombol ditekan dan dilepaskan (sama dengan mengetik karakter)
            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);
    }
}
```

- **addKeyListener():** Digunakan untuk menerima inputan dari user

1. **keyPressed(KeyEvent e):**

- Dipanggil saat pengguna menekan sebuah tombol di keyboard.
- `getKeyText(e.getKeyCode())` akan mengembalikan nama tombol yang ditekan.
- Contoh: Jika menekan tombol **A**, teks pada label menjadi **"Key Pressed: A"**.

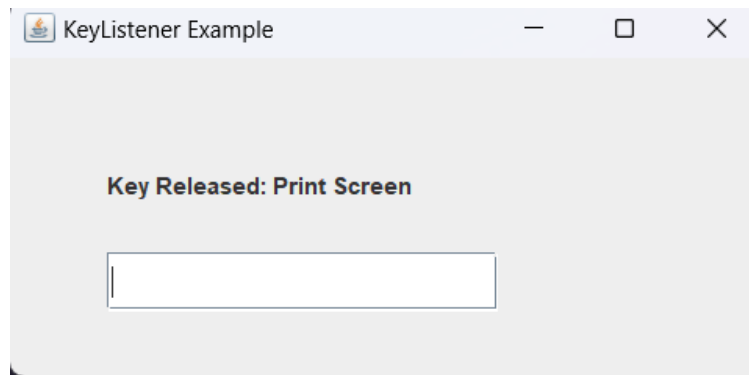
2. **keyReleased(KeyEvent e):**

- Dipanggil saat tombol yang ditekan dilepaskan.
- Sama seperti sebelumnya, nama tombol akan ditampilkan.

3. **keyTyped(KeyEvent e):**

- Dipanggil saat sebuah karakter diketik (tekan dan lepaskan).
- Menggunakan `e.getKeyChar()` untuk mendapatkan karakter yang diketik.
- Contoh: Jika mengetik **A**, teks menjadi **"Key Typed: A"**.

Output:



4. WindowListenerExample.java

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

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

        // Membuat label untuk menampilkan pesan
        JLabel label = new JLabel("Lakukan operasi pada jendela.");
        label.setBounds(50, 50, 300, 30);

        // Menambahkan WindowListener ke frame
        frame.addWindowListener(new WindowListener() {
            // Dijalankan ketika jendela dibuka
            public void windowOpened(WindowEvent e) {
                label.setText("Window Opened.");
            }

            // Dijalankan ketika jendela sedang dalam proses penutupan
            public void windowClosing(WindowEvent e) {
                label.setText("Window Closing.");
                // Bisa menambahkan System.exit(0); jika ingin menghentikan program
            }

            // Dijalankan ketika jendela benar-benar ditutup
            public void windowClosed(WindowEvent e) {
                System.out.println("Window Closed.");
            }

            // Dijalankan ketika jendela diminimalkan
            public void windowIconified(WindowEvent e) {
                label.setText("Window minimized.");
            }

            // Dijalankan ketika jendela dipulihkan dari minimalkan
            public void windowDeiconified(WindowEvent e) {
                label.setText("Window Restored.");
            }

            // Dijalankan ketika jendela menjadi aktif (berfokus)
            public void windowActivated(WindowEvent e) {
                label.setText("Window Activated.");
            }

            // Dijalankan ketika jendela kehilangan fokus
            public void windowDeactivated(WindowEvent e) {
                label.setText("Window Deactivated.");
            }
        });

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

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

addWindowListener(): Menambahkan WindowListener untuk mendeteksi dan menangani peristiwa jendela.

1. windowOpened(WindowEvent e)

- Dipanggil saat jendela pertama kali dibuka.
- Contoh: Label menampilkan **"Window Opened."**.

2. windowClosing(WindowEvent e)

- Dipanggil ketika jendela sedang dalam proses ditutup.
- Dalam kode ini, hanya mengubah teks label.
- Bisa ditambahkan `System.exit(0);` untuk menghentikan program saat jendela ditutup.

3. **windowClosed(WindowEvent e)**

- Dipanggil setelah jendela benar-benar ditutup.
- Di sini, teks ditampilkan ke konsol dengan `System.out.println()`.

4. **windowIconified(WindowEvent e)**

- Dipanggil saat jendela diminimalkan.
- Contoh: Label menampilkan "**Window minimized.**".

5. **windowDeiconified(WindowEvent e)**

- Dipanggil saat jendela dipulihkan dari mode minimalkan.
- Contoh: Label menampilkan "**Window Restored.**".

6. **windowActivated(WindowEvent e)**

- Dipanggil saat jendela menjadi aktif atau berfokus.
- Contoh: Label menampilkan "**Window Activated.**".

7. **windowDeactivated(WindowEvent e)**

- Dipanggil saat jendela kehilangan fokus.
- Contoh: Label menampilkan "**Window Deactivated.**".

Output:

