

LAPORAN PRAKTIKUM  
PEMOGRAMAN II  
PERTEMUAN 4



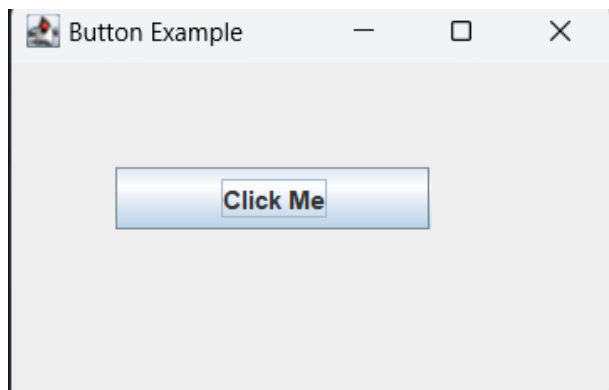
Muhammad Rackha Fauziansyah  
223040036

Kelas A  
TEKNIK INFORMATIKA  
UNIVERSITAS PASUNDAN  
2024

- Code & output

## ❖ Latihan 1

```
Praktikum4 > Latihan1 > ButtonExample.java > ButtonExample > main(String[])
1 package Latihan1;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 public class ButtonExample {
7
8     Run | Debug
9     public static void main(String[] args) {
10         JFrame frame = new JFrame(title:"Button Example");
11         JButton button = new JButton(text:"Click Me");
12
13         // Menambahkan ActionListener ke JButton
14         button.addActionListener(new ActionListener() {
15             public void actionPerformed(ActionEvent e) {
16                 System.out.println(x:"Button Clicked!");
17             }
18         });
19
20         button.setBounds(x:50, y:50, width:150, height:30);
21         frame.add(button);
22         frame.setSize(width:300, height:200);
23         frame.setLayout(manager:null);
24         frame.setVisible(b:true);
25         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26     }
27 }
```



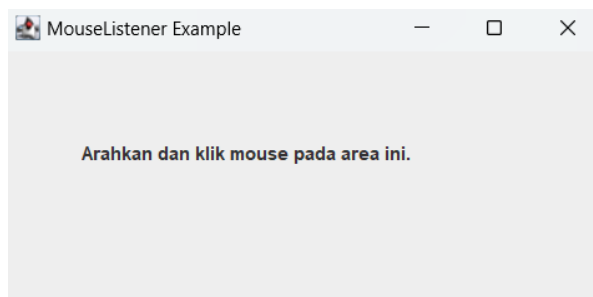
- JFrame dan JButton digunakan untuk membuat jendela dengan judul “Button Example” dan untuk membuat tombol dengan label click me
- ActionListener digunakan untuk sebuah aksi Ketika tombol diklik, dan pada saat diklik akan menampilkan button clicked pada konsol

## ❖ Latihan 2

```
1 package Latihan2;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 public class MouseListenerExample {
7     // Run | Debug
8     // Membuat Frame
9     JFrame frame = new JFrame(title:"MouseListener Example");
10
11     // Membuat label untuk menampilkan pesan
12     JLabel label = new JLabel(text:"Arahkan dan klik mouse pada area ini.");
13     label.setBounds(x:50, y:50, width:300, height:30);
14
15     // Menambahkan MouseListener ke Label
16     label.addMouseListener(new MouseListener() {
17         // dijalankan ketika mouse diklik (klik kiri, kanan atau tengah)
18         public void mouseClicked(MouseEvent e) {
19             label.setText("Mouse Clicked at: (" + e.getX() + ", " + e.getY() + ")");
20         }
21
22         // dijalankan ketika mouse ditekan (tanpa dilepaskan)
23         public void mousePressed(MouseEvent e) {
24             label.setText("Mouse Pressed at: (" + e.getX() + ", " + e.getY() + ")");
25         }
26
27         // dijalankan ketika mouse dilepaskan setelah ditekan
28         public void mouseReleased(MouseEvent e) {
29             label.setText("Mouse Released at: (" + e.getX() + ", " + e.getY() + ")");
30         }
31
32         // dijalankan ketika mouse masuk area komponen
33         public void mouseEntered(MouseEvent e) {
34             label.setText("Mouse Entered at: (" + e.getX() + ", " + e.getY() + ")");
35         }
36
37         // dijalankan ketika mouse keluar dari area komponen
38         public void mouseExited(MouseEvent e) {
39             label.setText("Mouse Exited at: (" + e.getX() + ", " + e.getY() + ")");
40         }
41     });
42 }
```

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

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



- JLabel digunakan untuk teks awal dengan tulisan “Arahkan dan klik mouse pada area ini “
- mouseClicked(): Dijalankan ketika mouse diklik
- mousePressed(): Dijalankan ketika mouse ditekan, tapi belum dilepaskan.
- mouseReleased(): Dijalankan ketika mouse dilepaskan setelah ditekan. Teks akan memperlihatkan saat mouse dilepaskan.

- `mouseEntered()`: Dijalankan ketika mouse masuk ke area.
- `mouseExited()`: Dijalankan ketika mouse keluar dari area.

### ❖ Latihan 3

```

1 package Latihans;
2
3 import javax.swing.*;
4 import java.awt.event.*;
5
6 public class KeyListenerExample {
7     public static void main(String[] args) {
8         // membuat frame
9         JFrame frame = new JFrame(title:"KeyListener Example");
10
11         // membuat label untuk menampilkan pesan
12         JLabel label = new JLabel(text:"Tekan tombol pada keyboard");
13         label.setBounds(x:50, y:50, width:300, height:30);
14
15         // membuat text field untuk fokus keyboard
16         JTextField textField = new JTextField();
17         textField.setBounds(x:50, y:100, width:200, height:30);
18
19         // menambahkan keylistener ke textField
20         textField.addKeyListener(new KeyListener() {
21             // dijalankan ketika tombol ditekan
22             public void keyPressed(KeyEvent e) {
23                 label.setText("Key Pressed : " + KeyEvent.getKeyText(e.getKeyCode()));
24             }
25
26             // dijalankan ketika tombol dilepaskan
27             public void keyReleased(KeyEvent e) {
28                 label.setText("Key Released : " + KeyEvent.getKeyText(e.getKeyCode()));
29             }
30
31             // dijalankan ketika tombol ditekan dan dilepaskan
32             public void keyTyped(KeyEvent e) {
33                 label.setText("Key Typed : " + e.getKeyChar());
34             }
35         });
36     }
37 }

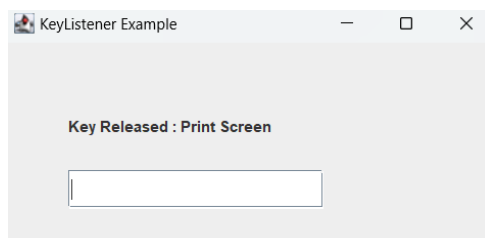
```

```

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

// setting Frame
frame.setSize(width:400, height:200);
frame.setLayout(manager:null);
frame.setVisible(b:true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

```



- `JLabel` digunakan untuk teks awal “Tekan tombol pada keyboard”
- `JTextField` digunakan untuk input keyboard
- `keyPressed()`: Dijalankan ketika sebuah tombol pada keyboard ditekan. Teks akan berubah menampilkan tombol yang ditekan
- `keyReleased()`: Dijalankan ketika tombol pada keyboard dilepaskan.
- `keyTyped()`: Dijalankan ketika sebuah tombol diketik

## ❖ Latihan 4

```
package Latihan4;

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

public class WindowListenerExample {
    Run | Debug
    public static void main(String[] args) {
        JFrame frame = new JFrame(title:"WindowListener Example");

        // membuat label untuk menampilkan pesan
        JLabel label = new JLabel(text:"lakukan operasi pada jendela");
        label.setBounds(x:50, y:50, width:300, height:30);

        // menambahkan windowlistener ke frame
        frame.addWindowListener(new WindowListener() {
            // dijalankan ketika jendela dibuka
            public void windowOpened(WindowEvent e) {
                label.setText(text:"Window Opened.");
            }

            // dijalankan ketika jendela sedang dalam proses penutupan
            public void windowClosing(WindowEvent e) {
                label.setText(text:"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(x:"Window Closed.");
            }

            // dijalankan ketika jendela diminimalkan
            public void windowIconified(WindowEvent e) {
                label.setText(text:"Window Minimized.");
            }

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

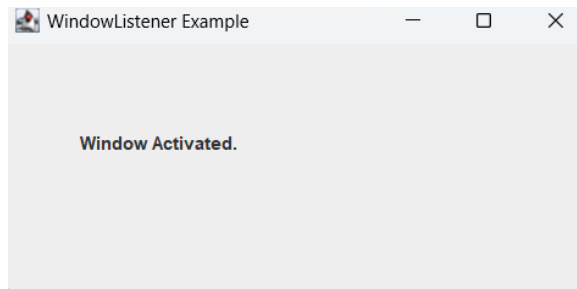
```
        // dijalankan ketika jendela dipulihkan dari minimalkan
        public void windowDeiconified(WindowEvent e) {
            label.setText(text:"Window Restored.");
        }

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

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

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

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



- JLabel digunakan untuk teks “lakukan operasi pada jendela”
- `windowOpened()`: Dijalankan ketika jendela dibuka.
- `windowClosing()`: Dijalankan ketika jendela sedang dalam proses ditutup.
- `windowClosed()`: Dijalankan ketika jendela benar-benar telah ditutup. Dalam contoh ini, hanya pesan ke konsol yang ditampilkan, "Window Closed."
- `windowIconified()`: Dijalankan ketika jendela diminimalkan.
- `windowDeiconified()`: Dijalankan ketika jendela dipulihkan dari keadaan minimalkan.
- `windowActivated()`: Dijalankan ketika jendela menjadi aktif
- `windowDeactivated()`: Dijalankan ketika jendela kehilangan fokus.
- Semua `windowlistener` akan menampilkan pesan sesuai dengan event yang dijalankan
- Jendela ditampilkan menggunakan `setVisible(true)` dan Ketika ditutup menggunakan `setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)`.