

**TUGAS PRAKTIKUM
PEMOGRAMAN II**



DISUSUN OLEH :

Bintang Arya

223040032

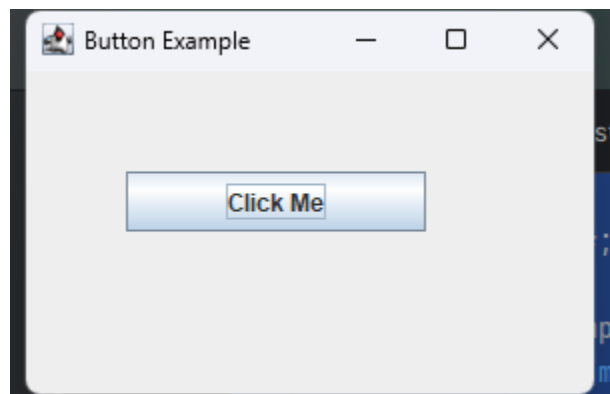
***TEKNIK INFORMATIKA
UNIVERSITAS
PASUNDAN BANDUNG
2024***

1. Buatlah sebuah kelas bernama ButtonExample dengan kode berikut.

Code :

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

Hasil :



Penjelasan :

Kode di atas membuat antarmuka grafis dengan Java Swing yang menampilkan tombol "Click Me". Ketika tombol diklik, pesan "Button clicked!" ditampilkan di konsol. Program ini menggunakan JFrame sebagai window utama dan JButton sebagai tombol, serta menambahkan *ActionListener* untuk menangani klik tombol. Ukuran dan posisi tombol diatur secara manual, dan window akan ditutup saat pengguna menutupnya.

2. Buat kelas bernama `MouseListenerExample` dengan kode sebagai berikut.

Code :

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

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

        // Membuat label untuk menampilkan pesan
        JLabel label = new JLabel( text: "Arahkan dan klik mouse pada area ini.");
        label.setBounds( x: 50, y: 50, width: 300, height: 30);

        // Menambahkan MouseListener ke label
        label.addMouseListener(new MouseListener() {
            // Dijalankan ketika mouse diklik (klik kiri, kanan atau tengah)
            public void mouseClicked(MouseEvent e) {
                label.setText("Mouse Clicked at: (" + e.getX() + ", " + e.getY() + ")");
            }

            // Dijalankan ketika mouse ditekan (tombol belum 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() + ")");
            }

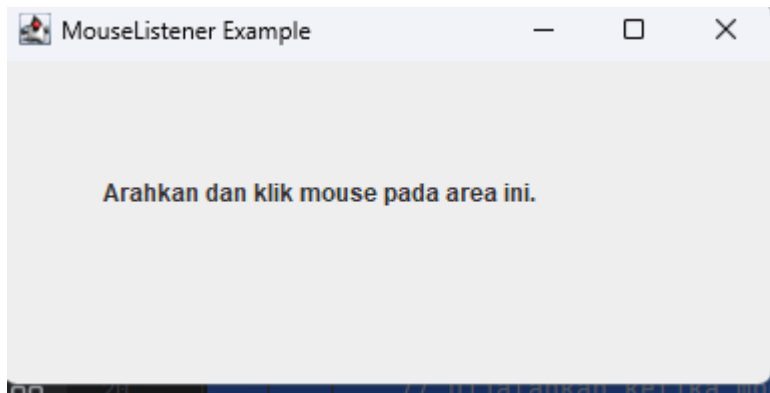
            // Dijalankan ketika mouse masuk ke area komponen
            public void mouseEntered(MouseEvent e) {
                label.setText("Mouse Entered the area.");
            }

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

Hasil :



Penjelasan :

Kode di atas membuat program Java yang menggunakan *MouseListener* untuk menangani event mouse. Ketika mouse berinteraksi dengan label, event seperti klik, tekan, rilis, masuk, dan keluar dari area label akan memicu perubahan teks pada label. Program ini menampilkan posisi koordinat mouse atau pesan sesuai dengan event yang terjadi. *JFrame* digunakan sebagai window, dan *JLabel* menampilkan teks di dalamnya.

3. Buat kelas bernama KeyListenerExample dengan kode sebagai berikut.

Code :

```
ButtonExample.java  KeyListenerExample.java  MouseListenerExample.java  WindowListenerExample.java

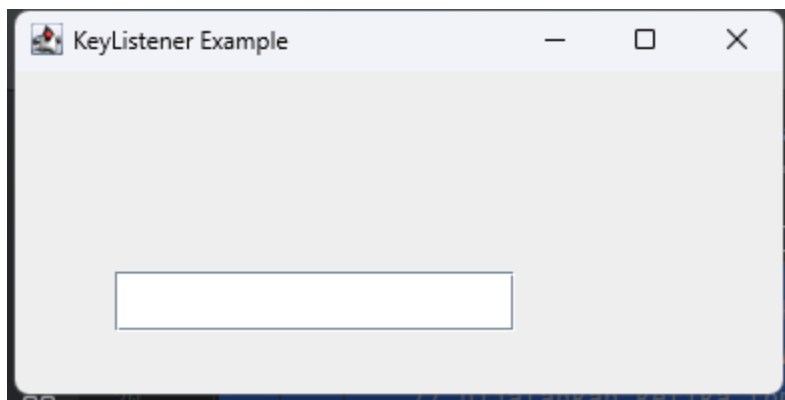
1  import javax.swing.*;
2  import java.awt.event.*;
3
4
5  ▶ public class KeyListenerExample {
6
7  ▶      public static void main(String[] args){
8          // Membuat frame
9          JFrame frame = new JFrame( title: "KeyListener Example");
10
11          // Membuat label untuk menampilkan pesan
12          JLabel label = new JLabel( text: "Tekan tombol pada keyboard");
13
14          // Membuat text field untuk fokus keyboard
15          JTextField textField = new JTextField();
16          textField.setBounds( x: 50, y: 100, width: 200, height: 30);
17
18          // Menambahkan KeyListener ke text field
19          textField.addKeyListener(new KeyListener() {
20              // Dijalankan ketika tombol ditekan
21              @Override
22              ▶ public void keyPressed(KeyEvent e) {
23                  label.setText("Key Pressed: " + KeyEvent.getKeyText(e.getKeyCode()));
24              }
25
26              // Dijalankan ketika tombol dilepaskan
27              @Override
28              ▶ public void keyReleased(KeyEvent e) {
29                  label.setText("Key Released: " + KeyEvent.getKeyText(e.getKeyCode()));
30              }
31
32              // Dijalankan ketika tombol ditekan dan dilepaskan (sama dengan mengetik karakter)
33              @Override
34              ▶ public void keyTyped(KeyEvent e) {
35                  label.setText("Key Typed: " + e.getKeyChar());
36              }
37          });
38
39          // Menambahkan Komponen ke frame
40          frame.add(label);
41          frame.add(textField);
```

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

}

}
```

Hasil :



Penjelasan :

Kode di atas membuat program Java yang menggunakan *KeyListener* untuk menangani event keyboard. Saat pengguna menekan, melepaskan, atau mengetik karakter di text field, pesan yang menunjukkan tombol yang ditekan atau karakter yang diketik akan ditampilkan di label. *JFrame* digunakan sebagai window utama, *JLabel* untuk menampilkan pesan, dan *TextField* untuk menangkap input keyboard.

4. Buat kelas bernama WindowListenerExample dengan kode sebagai berikut.

Code :

```
ButtonExample.java  KeyListenerExample.java  MouseListenerExample.java  WindowListenerExample.java x
1  import javax.swing.*;
2  import java.awt.event.*;
3
4  public class WindowListenerExample {
5      public static void main(String[] args) {
6          // Membuat frame
7          JFrame frame = new JFrame( title: "WindowListener Example");
8
9          // Membuat label untuk menampilkan pesan
10         JLabel label = new JLabel( text: "Lakukan operasi pada jendela.");
11         label.setBounds( x: 50, y: 50, width: 300, height: 30);
12
13         // Menambahkan window listener ke frame
14         frame.addWindowListener(new WindowListener() {
15
16             public void windowOpened(WindowEvent e) {
17                 label.setText("Window Opened");
18             }
19
20             public void windowClosing(WindowEvent e) {
21                 label.setText("Window Closing");
22             }
23
24             public void windowClosed(WindowEvent e) {
25                 label.setText("Window Closed");
26             }
27
28             public void windowIconified(WindowEvent e) {
29                 label.setText("Window Minimized");
30             }
31
32             public void windowDeiconified(WindowEvent e) {
33                 label.setText("Window Restored. ");
34             }
35
36             public void windowActivated(WindowEvent e) {
37                 label.setText("Window Activated");
38             }
39
40             public void windowDeactivated(WindowEvent e) {
41                 label.setText("Window Deactivated");
42             }
43         });
44     }
45 }
```

```

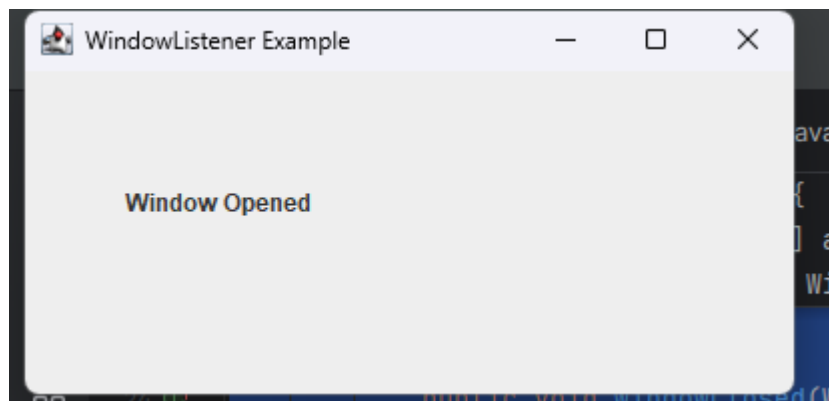
});

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

// Setting frame
frame.setSize( width: 400, height: 200);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Agar jendela benar-benar tertutup
}

```

Hasil :



Penjelasan :

Kode di atas membuat program Java menggunakan *WindowListener* untuk menangani event pada jendela. Saat jendela dibuka, ditutup, diminimalkan, dipulihkan, diaktifkan, atau dinonaktifkan, pesan yang sesuai akan ditampilkan di label. *JFrame* digunakan sebagai window utama, dan *WindowListener* menangani event seperti "Window Opened", "Window Closing", dan lainnya.