BAB 8 GUI- JAVA EVENT HANDLING

8.1 Tujuan Pembelajaran

Mahasiswa mampu mengimplementasikan Java Event Handling pada GUI.

8.2 Dasar Teori

8.2.1 Event Handling

Perubahan keadaan suatu objek dikenal sebagai aksi (event). Misalnya, klik tombol, seret mouse, dll. Paket `java.awt.event` menyediakan banyak kelas *event* dan antarmuka *Listener* untuk penanganan event.

Event Class	Listener Interfaces
ActionEvent	ActionListener
MouseEvent	MouseListener and MouseMotionListener
MouseWheel Event	MouseWheelListener
KeyEvent	KeyListener
ItemEvent	ItemListener
TextEvent	TextListener
AdjustmentEvent	AdjustmentListener
WindowEvent	WindowListener
Component Event	ComponentListener
ContainerEvent	ContainerListener
FocusEvent	FocusListener

8.2.1.1 Registration Methods

Untuk mendaftarkan komponen dengan Listener, banyak kelas menyediakan metode pendaftaran. Sebagai contoh:

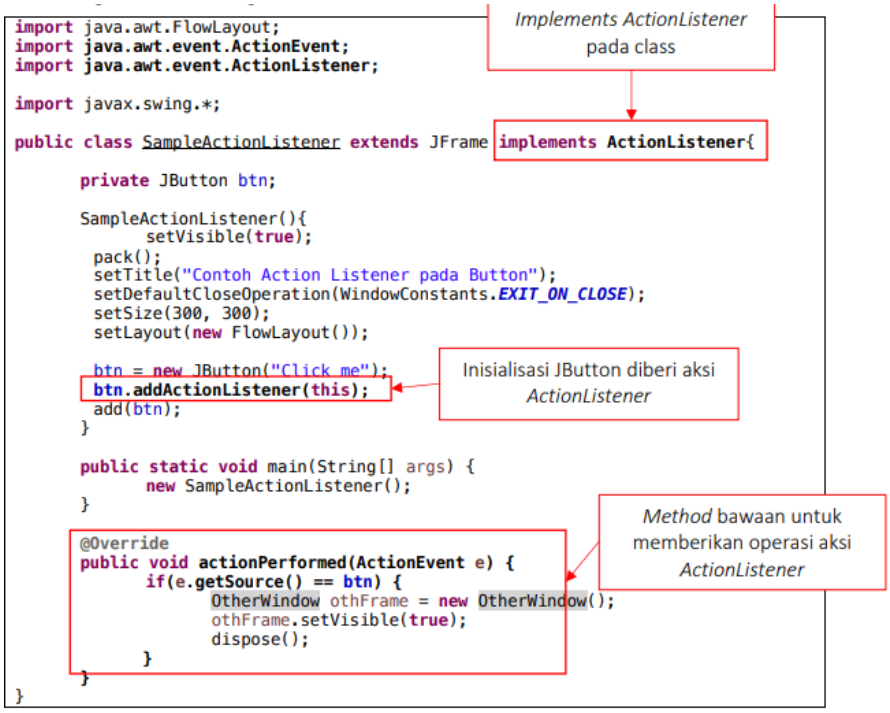
- Button
  - public void addActionListener(ActionListener a){}
- MenuItem
  - public void addActionListener(ActionListener a){}

- TextField
  - public void addActionListener(ActionListener a){}
  - public void addTextListener(TextListener a){}
- TextArea
  - public void addTextListener(TextListener a){}
- Checkbox
  - public void addItemListener(ItemListener a){}
- Choice
  - public void addItemListener(ItemListener a){}
- List
  - public void addActionListener(ActionListener a){}
  - public void addItemListener(ItemListener a){}

8.2.1.2 Java Event Handling Code ActionListener

Berikut adalah cara untuk menempatkan kode penanganan *event ActionListener* (melakukan aksi klik pada komponen) ke salah satu komponen *swing* berikut:

1. Penempatan kode *event* pada kelas:



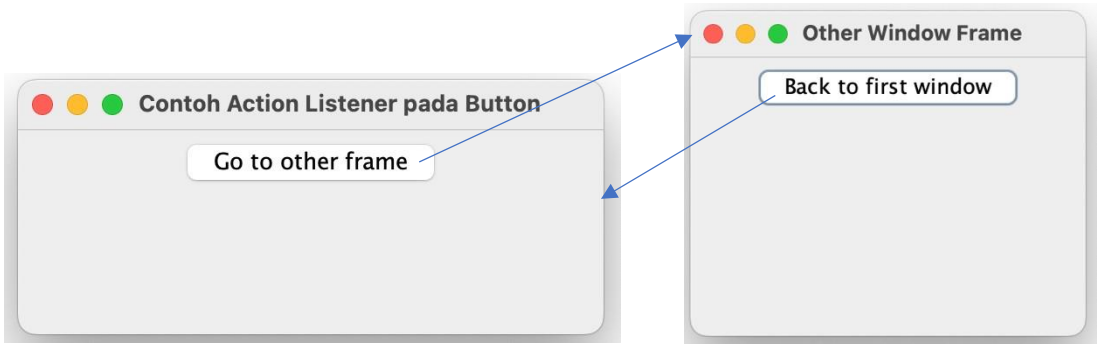

```
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

public class SampleActionListener extends JFrame implements ActionListener{

	private JButton btn;

	SampleActionListener(){
		setVisible(true);
	 pack();
	 setTitle("Contoh Action Listener pada Button");
	 setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
	 setSize(300, 300);
	 setLayout(new FlowLayout());

	 btn = new JButton("Click me");
	 btn.addActionListener(this);
	 add(btn);
	}

	public static void main(String[] args) {
		new SampleActionListener();
	}

	@Override
	public void actionPerformed(ActionEvent e) {
		if(e.getSource() == btn) {
			OtherWindow othFrame = new OtherWindow();
			othFrame.setVisible(true);
			dispose();
		}
	}
}
```

```
        btn = new JButton("Click me");
btn.addActionListener(new ActionListener()
                        {

                        @Override                    e  {
                      public void                    )
            actionPerformed(ActionEvent
          OtherWind   othFrame = new OtherWindow();
              ow
          othFrame.setVisible(true
               ); dispose();
    }
```

2. Penempatan kode *event* didalam kelas:

```
            public class SampleActionListener2 extends JFrame{
    private JButton btn;

  SampleActionListener2(

            ){

                                setVisible(true);
                                    pack();
                    setTitle("Contoh Action Listener pada Button");
                  setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                                setSize(300, 300);
                             setLayout(new FlowLayout());









                                                  });

                                               add(btn);
                                }

    public static void main(String[] args) {

            new SampleActionListener2();
}
```

Memberikan aksi JButton *ActionListener*


Contoh Action Listener pada Button
Go to other frame
Other Window Frame
Back to first window

8.2.2 Event Handler MouseListener

Java *MouseListener* memberi tahukan setiap kali ada perubahan aksi terhadap status *mouse*. *MouseEvent* merupakan library yang dapat menangkap aksi pada mouse. Antarmuka *MouseListener* ditemukan dalam paket java.awt.event. Library ini memiliki lima metode:

Method	Deskripsi
mouseClicked(MouseEvent e)	Memberitahukan aksi pada mouse berupa klik
mouseEntered(MouseEvent e)	Memberitahukan aksi pada mouse berupa kursor mouse memasuki wilayah komponen
mouseExited(MouseEvent e)	Memberitahukan aksi pada mouse berupa kursor mouse keluar dari wilayah komponen
mousePressed(MouseEvent e)	Memberitahukan aksi bahwa mouse ditekan
mouseReleased(MouseEvent e)	Memberitahukan aksi bahwa mouse sudah tidak ditekan

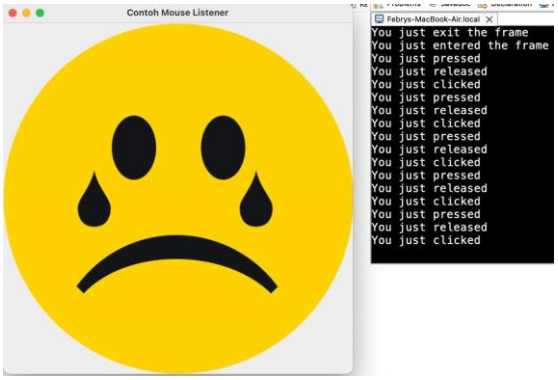
Berikut ini adalah contoh penerapan MouseListener:

```
 import java.awt.GridLayout;
import java.awt.event.MouseEvent;
import java.net.URL;

 import javax.swing.*;

 import javax.swing.event.MouseInputListener;      implements MouseInputListener{



 public class SampleMouseListener extends JFrame

         private JLabel lblImage;

         ImageIcon smile, angry, cry, surprised;



             SampleMouseLi
        stener(){
        setVisible(true);
        pack();

                 setTitle("Contoh Mouse Listener");
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
         lblImage.addMouseListener(frame);
                 setSize(500, 500);
        setLayout(new GridLayout());
        initialize(this);

         }



         public static void main(String[] args) {

                 new SampleMouseListener();
```

Inisialisasi JLabel diberikan aksi

```
        URL angry_path =
SampleMouseListener.class.getClassLoader().getResource(packageName+"angry.png");
        angry = new ImageIcon(angry_path);

        lblImage.setIcon(smile);

        frame.add(lblImage);
    }

    @Override
    public void mouseClicked(MouseEvent e) {
        System.out.println("You just clicked");
    }


    @Override
    public void mousePressed(MouseEvent e) {
        System.out.println("You just pressed");
        lblImage.setIcon(angry);
    }


    @Override
    public void mouseReleased(MouseEvent e) {
        System.out.println("You just released");
        lblImage.setIcon(cry);
    }


    @Override
    public void mouseEntered(MouseEvent e) {
        System.out.println("You just entered the frame");
        lblImage.setIcon(surprised);
    }


    @Override
    public void mouseExited(MouseEvent e) {
        System.out.println("You just exit the frame");
        lblImage.setIcon(smile);
    }


    @Override
    public void mouseDragged(MouseEvent e) {
        System.out.println("You just dragged the mouse");
    }


    @Override
    public void mouseMoved(MouseEvent e) {
        System.out.println("You just moved the mouse");
    }
}
```

Method bawaan MouseListener



Pada program diatas jika dijalankan maka akan memberikan perintah jika kursor mouse memasuki wilayah JLabel lbl Image maka akan mengganti gambar icon menjadi gambar *surprise*, perintah ini akan menjalan method mouseEntered(MouseEvent e). Jika kursor mouse keluar dari wilayah lbl Image maka gambar icon menjadi *smile*, perintah tersebut akan menjalankan method mouseExited(MouseEvent e). Jika lbl Image ditekan dengan mouse maka akan menjalankan method mousePressed(MouseEvent e) yaitu mengganti gambar dengan icon *angry*. Sedangkan jika lblImage sudah tidak diklik atau

ditekan maka akan menjalankan method mouseReleased(MouseEvent e) dengan mengganti gambar icon menjadi *sad*.

8.2.3 Event Handler KeyListener

Java *KeyListener* memberitahukan setiap kali ada aksi terhadap *keyboard*. *Library* yang dapat digunakan ialah KeyEvent. Antarmuka *KeyListener* ditemukan dalam paket java.awt.event, dan memiliki tiga metode:

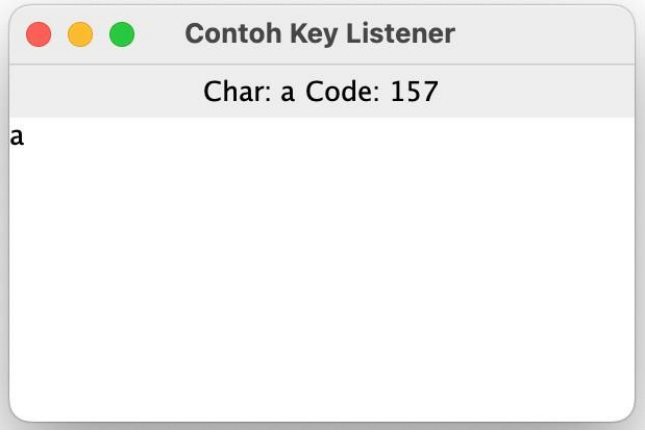
Method	**Deskripsi**
keyPressed (KeyEvent e)	Memberikan aksi jika keyboard ditekan
keyReleased (KeyEvent e)	Memberikan aksi jika keyboard dilepas
keyTyped (KeyEvent e)	Memberikan aksi Ketika keyboard dalam keadaan mengetik

Berikut adalah contoh penggunaan KeyListener:

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import javax.swing.*;

public class SampleKeyListener extends JFrame implements KeyListener {
	private JLabel labelChar,labelCode;

	SampleKeyListener(){
		setVisible(true);
		pack();
		setTitle("Contoh Key Listener");
		setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
		setLocationRelativeTo(null);
		setSize(300, 200);
		setLayout(new GridLayout());
		initialize(this);
	}

	public static void main(String[] args) {
		new SampleKeyListener();
	}

	private void initialize(SampleKeyListener frame) {
		BorderLayout bl = new BorderLayout();
		frame.setLayout(bl);
		JPanel panel = new JPanel();
		panel.setLayout(new FlowLayout());
		labelChar = new JLabel("Char:");
		panel.add(labelChar);
		labelCode = new JLabel("Code:");
		panel.add(labelCode);
		frame.add(panel, bl.NORTH);
		JTextArea textarea = new JTextArea();
		textarea.addKeyListener(this);
		frame.add(textarea, bl.CENTER);
	}

	@Override
	public void keyTyped(KeyEvent e) {
		char letter = e.getKeyChar();
		String word = "Char: "+ letter;
		labelChar.setText(word);
	}

	@Override
	public void keyPressed(KeyEvent e) {
		int letter_code = e.getKeyCode();
		String word = "Code: "+ letter_code;
		labelCode.setText(word);
	}

	@Override
	public void keyReleased(KeyEvent e) {}
}
```

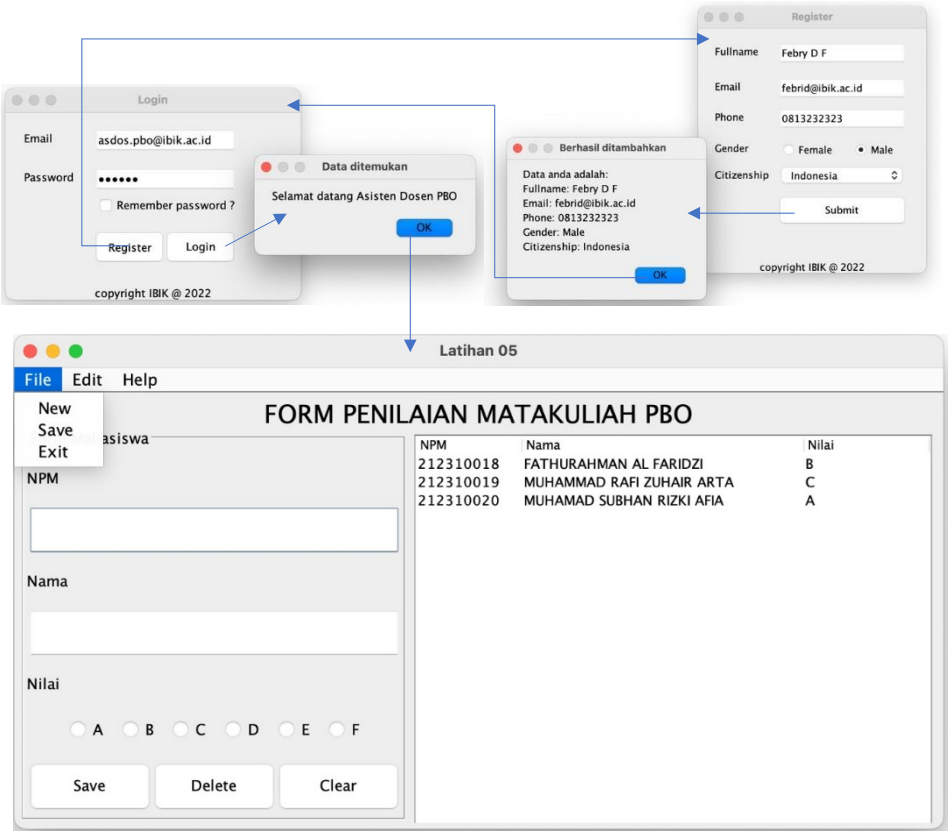
Inisialisasi *JTextArea* untuk menambahkan aksi *KeyListener*

Method bawaan *KeyListener*



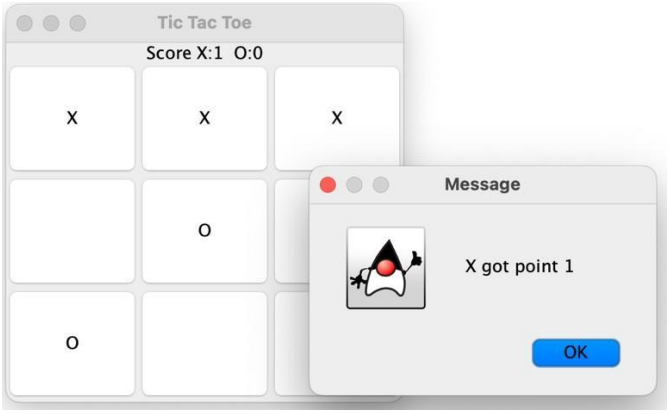
8.3 Latihan Pembelajaran

1. Buatlah project maven dengan nama PBO-NPM-Pembelajaran-8 dan *package groupid* bernama com.ibik.pbo.Pembelajaran.
2. Berdasarkan UI pada soal Pembelajaran-7 nomor 3, 4 dan 5 buatlah flow aplikasi sebagai berikut:

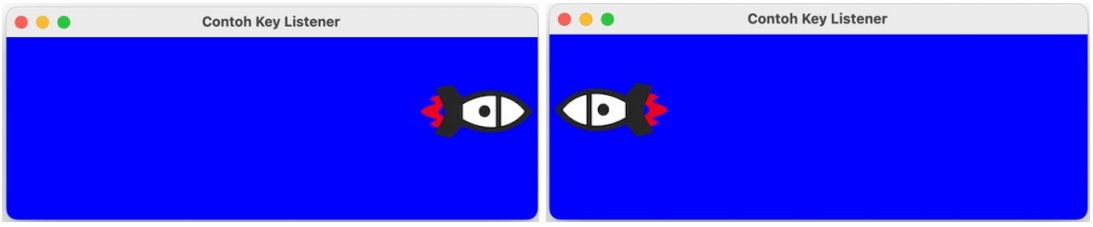


   Anda dapat menggunakan *ActionListener* untuk membuat alur aplikasi seperti gambar diatas.

3. Berdasarkan soal nomor 2 buatlah *Activity Diagram*.
4. Buatlah program sederhana dengan menggunakan *MouseListener* sebagai berikut:



5. Buatlah program sederhana seperti dibawah ini dengan menggunakan KeyListener:



Pada program diatas, jika memencet tombol pada keyboard akan memindahkan posisi objek gambar dengan kondisi sebagai berikut:

- Tombol [a] atau panah kiri [□] memindahkan posisi gambar ke arah kiri
- Tombol [d] atau panah kiri [□] memindahkan posisi gambar ke arah kanan
- Tombol [w] atau panah atas [□] memindahkan posisi gambar ke arah atas
- Tombol [s] atau panah bawah [□] memindahkan posisi gambar ke arah bawah

Jika objek gambar berada di ujung layer frame maka berganti objek gambar