

# PRAKTIKUM 28

---

## PENANGANAN EVENT

---

### A. TUJUAN PEMBELAJARAN

1. Memahami konsep Graphical User Interface (GUI).
2. Mampu membuat program yang menerapkan konsep GUI.
3. Mampu membuat program berbasis GUI yang melibatkan penggunaan komponen GUI.
4. Mampu membuat program berbasis GUI yang bisa menerima respon user, baik melalui mouse maupun keyboard.

### B. DASAR TEORI

User bisa berinteraksi dengan program aplikasi melalui sejumlah komponen. Sebagai contoh, setelah memasukkan data-data yang diperlukan pada proses login, kita menekan sebuah button untuk memasukkan data-data tersebut ke dalam program. Dalam hal ini objek button tersebut menerima *event* click. Tidak hanya objek button yang bisa menerima *event*. Objek lain seperti textfield juga bisa menerima *event*, hanya saja jenis *event* yang terjadi pada objek textfield berbeda dengan objek button. Jadi kita harus memahami jenis *event* apa yang bisa dikenakan terhadap suatu jenis objek tertentu.

Java menerima event melalui sebuah interface. Setiap komponen yang akan menerima suatu event harus disetting terlebih dahulu agar terasosiasi dengan event tersebut. Tanpa didahului dengan proses asosiasi ini, sebuah komponen tidak akan bisa "menangkap" (Java memberi istilah "mendengar" / listening) suatu event walaupun sesungguhnya bisa. Sebagai contoh, jika kita memasang sebuah objek button untuk membatalkan suatu proses, sebelum kita mengasosiasi objek tersebut dengan event yang menangani *click* mouse maka button tersebut tidak akan

memberikan efek pembatalan proses pada saat kita melakukan proses click tombol mouse di atasnya.

Bagian ini membahas tentang bagaimana proses interaksi antara user dan program melalui sejumlah komponen dapat terjadi. Dari sejumlah interface untuk menangani event yang disediakan oleh Java, kita akan membahas 4 (empat) saja yang kami rasa mudah dimengerti sebagai pengantar untuk memahami interface-interface yang lain. Keempat interface tersebut adalah:

1. Interface ActionListener
2. Interface MouseListener
3. Interface MouseMotionListener
4. Interface KeyListener

Interface pertama akan menangkap event yang ditimbulkan oleh *mouse* dan *keyboard*, interface kedua dan ketiga akan menangkap event yang ditimbulkan oleh *mouse* dan interface keempat akan menangkap event yang ditimbulkan oleh *keyboard*. Mouse dan keyboard adalah dua piranti yang sudah sangat umum digunakan pada komputer masa kini.

Bentuk umum penggunaan interface adalah:

```
[public|private] class NamaClass
    [extends BaseClass]
    [implements NamaInterface]
{
    ...isi class...
}
```

Implementasi sebuah interface mengharuskan kita memasukkan sejumlah method yang akan digunakan untuk melakukan suatu proses dalam interface tersebut. Pada bagian selanjutnya anda akan menjumpai method apa saja yang harus disediakan jika kita mengimplementasikan suatu interface.

### **Interface ActionListener**

Interface ActionListener berguna untuk menerima event yang ditimbulkan oleh mouse atau keyboard. Pada prakteknya interface ini akan memeriksa objek yang terkena event, tidak peduli alat mana yang menimbulkan event tersebut. Sebuah objek button misalnya, mungkin akan

mengalami event *click* dari mouse atau event *enter* dari keyboard, yang jelas keduanya merupakan event dari button. Button inilah yang diperiksa oleh ActionListener.

Class yang mengimplementasikan interface ActionPerformed harus mengandung method bernama actionPerformed(ActionEvent) sekalipun di dalamnya tidak ada instruksi apa pun. Kita bisa memeriksa objek yang menimbulkan event di dalam method ini.

Seperti telah dijelaskan di depan, agar sebuah objek bisa "mendengar" terjadinya event, objek tersebut harus diasosiasikan dengan interface yang berhubungan dengan event tersebut. Java menyediakan method addActionListener(ActionListener) untuk mengasosiasikan sebuah objek dengan interface ActionListener. Method ini membutuhkan parameter berupa class yang mengimplementasikan interface ActionListener. Jika class tersebut merupakan class yang sama dengan tempat objek dimaksud, maka parameter ini cukup diberi nilai this.

### **Interface MouseListener**

Interface MouseListener digunakan untuk menangkap event yang diakibatkan oleh mouse. Event yang dibangkitkan oleh keyboard tidak akan memberi efek, kecuali kita menggunakan interface ActionListener bersamaan dengan interface MouseListener. Dan sebenarnya anda boleh menggunakan sejumlah interface sesuai keperluan, tidak terbatas harus satu interface saja.

Objek yang akan mendengar event dari MouseListener diasosiasikan menggunakan method addMouseListener(MouseListener). Class yang mengimplementasikan interface MouseListener harus menyediakan 5 (lima) method, walaupun tidak ada isinya. Kelima method tersebut adalah:

1. Method mouseClicked(MouseEvent), dipanggil pada saat tombol mouse ditekan lalu dilepas.
2. Method mouseEntered(MouseEvent), dipanggil pada saat kursor mouse memasuki area komponen.
3. Method mouseExited(MouseEvent), dipanggil pada saat kursor mouse meninggalkan area komponen.
4. Method mousePressed(MouseEvent), dipanggil pada saat tombol mouse ditekan.
5. Method mouseReleased(MouseEvent), dipanggil pada saat tombol mouse dilepas.

Kita bisa katakan bahwa method mouseClicked(MouseEvent) merupakan perpaduan antara mousePressed(MouseEvent) dan mouseReleased(MouseEvent). Berbagai method pada class

MouseEvent bisa kita gunakan dalam proses event mouse ini, misalnya mendeteksi posisi kursor mouse saat ini.

### **Interface MouseMotionListener**

Java menyediakan interface MouseMotionListener untuk mendukung MouseListener. Interface MouseMotionListener menyediakan dua method yang akan memantau pergerakan mouse yaitu mouseDragged(MouseEvent) dan mouseMoved(MouseEvent).

- Method mouseDragged(MouseEvent) digunakan untuk memantau pergerakan mouse yang melintasi objek pada saat tombol mouse ditekan. Tindakan ini persis sama dengan tindakan kita pada saat memindahkan sebuah window.
- Method mouseMoved(MouseEvent) digunakan untuk memantau pergerakan mouse pada saat mouse melintasi area suatu objek. Pada saat ini tidak ada tombol mouse yang ditekan; kita hanya memindahkan pointer mouse melalui objek.

Berikut ini urutan method yang dieksekusi setelah event mouse ditangkap, diasumsikan kita melibatkan interface MouseListener dan MouseMotionListener.

1. Method mouseEntered(MouseEvent) : dijalankan pada saat pointer mouse digerakkan memasuki area objek.
2. Method mouseMoved(MouseEvent) : dijalankan pada saat pointer mouse digerakkan melintasi objek dalam kondisi tidak ada tombol mouse yang ditekan.
3. Method mousePressed(MouseEvent) : dijalankan pada saat tombol mouse ditekan.
4. Method mouseDragged(MouseEvent) : dijalankan pada saat pointer mouse digeser melintasi objek sambil tombol mouse ditekan.
5. Method mouseClicked(MouseEvent) : dijalankan pada saat tombol mouse dilepas setelah ditekan dengan rentang waktu yang tidak terlalu lama.
6. Method mouseReleased(MouseEvent) : dijalankan pada saat tombol mouse dilepas.
7. Method mouseExited(MouseEvent) : dijalankan pada saat pointer mouse meninggalkan area objek. Setelah pointer mouse berada di luar objek, segala macam event mouse tidak diperhatikan lagi, sampai pointer mouse memasuki area objek.

## Interface KeyListener

Setelah mengetahui cara *handle* event yang dibangkitkan oleh piranti mouse, sekarang kita akan membahas bagaimana caranya *handle* event yang dibangkitkan oleh piranti keyboard. Interface `KeyListener` disediakan untuk menangani event yang terjadi akibat tindakan dari keyboard, misalnya jika terjadi penekanan tombol ENTER atau ESCAPE.

Interface `KeyListener` mengharuskan kehadiran tiga method:

- Method `keyPressed(KeyEvent)` : akan dijalankan pada saat sebuah tombol keyboard ditekan.
- Method `keyReleased(KeyEvent)` : akan dijalankan pada saat sebuah tombol keyboard dilepas.
- Method `keyTyped(KeyEvent)` : akan dijalankan pada saat sebuah tombol keyboard ditekan dan dilepas dengan selisih waktu yang singkat.

## C. TUGAS PENDAHULUAN

1. Pelajari event – event handling yang berhubungan dengan User Interface, kemudian buatlah ringkasan kelas/interface event handling tersebut beserta kegunaannya.

## D. PERCOBAAN

### Percobaan 1: Membuat program menebak angka

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  class MyFrame extends JFrame implements ActionListener
6  {
7      private Container container = new Container();
8      private Button tombolCek = new Button ("Cek");
9      private TextField inputan = new TextField ("", 8);
10     private int angkaAcak = (int)(Math.random()*10);
11
12     MyFrame()
13     {
14         super ("Tebak angka");
15
16         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
17         setSize (200,65);
```

```

18     setLocation (100,100);
19
20     tombolCek.addActionListener (this);
21
22     container = getContentPane();
23     container.setLayout (new BorderLayout());
24     container.add (new JLabel("Tebak angka antara 0-9"),
25                     BorderLayout.NORTH);
26     container.add (new JLabel ("Tebakan anda :"),
27                     BorderLayout.WEST);
28
29     container.add (inputan, BorderLayout.CENTER);
30     container.add (tombolCek, BorderLayout.EAST);
31
32     show();
33 }
34
35 public void actionPerformed (ActionEvent e)
36 {
37     Object objek = e.getSource();
38
39     if (objek == tombolCek)
40     {
41         int jawaban = Integer.parseInt(inputan.getText());
42
43         if (jawaban == angkaAcak)
44             JOptionPane.showMessageDialog (null,
45                 "Benar.. Anda memang peramal jitu");
46         else
47             JOptionPane.showMessageDialog (null,
48                 "Jawaban anda salah..");
49     }
50 }
51
52 // -----
53 // Class utama
54 // -----
55 public class ActionListener_1
56 {
57     public static void main (String[] args)
58     {
59         new MyFrame();
60     }
61 }

```

## Percobaan 2: Membuat program dengan event mouselistener

```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4

```

```

5  class MyFrame extends JFrame implements MouseListener
6  {
7      private Container container = new Container();
8      private JLabel labelClicked = new JLabel ("");
9      private JLabel labelEntered = new JLabel ("");
10     private JLabel labelExited = new JLabel ("");
11     private JLabel labelPressed = new JLabel ("");
12     private JLabel labelReleased = new JLabel ("");
13
14     MyFrame()
15     {
16         super ("Test MouseListener");
17
18         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
19         setSize (220,150);
20         setLocation (100,100);
21
22         // -----
23         //  mengasosiasi seluruh objek dg event-handler
24         // -----
25         labelClicked.addMouseListener (this);
26         labelEntered.addMouseListener (this);
27         labelExited.addMouseListener (this);
28         labelPressed.addMouseListener (this);
29         labelReleased.addMouseListener (this);
30
31         // -----
32         //  penambahan komponen ke frame
33         // -----
34         container = getContentPane();
35         container.setLayout (new GridLayout (5,2));
36         container.add (new JLabel ("mouseClicked :"));
37         container.add (labelClicked);
38         container.add (new JLabel ("mouseEntered :"));
39         container.add (labelEntered);
40         container.add (new JLabel ("mouseExited :"));
41         container.add (labelExited);
42         container.add (new JLabel ("mousePressed :"));
43         container.add (labelPressed);
44         container.add (new JLabel ("mouseReleased :"));
45         container.add (labelReleased);
46
47         show();
48     }
49
50     public void mouseClicked (MouseEvent e)
51     {
52         labelClicked.setText ("AKTIF");
53         labelEntered.setText ("");
54         labelExited.setText ("");
55         labelPressed.setText ("");
56         labelReleased.setText ("");
57     }
58

```

```

59     public void mouseEntered (MouseEvent e)
60     {
61         labelClicked.setText ("");
62         labelEntered.setText ("AKTIF");
63         labelExited.setText ("");
64         labelPressed.setText ("");
65         labelReleased.setText ("");
66     }
67
68     public void mouseExited (MouseEvent e)
69     {
70         labelClicked.setText ("");
71         labelEntered.setText ("");
72         labelExited.setText ("AKTIF");
73         labelPressed.setText ("");
74         labelReleased.setText ("");
75     }
76
77     public void mousePressed (MouseEvent e)
78     {
79         labelClicked.setText ("");
80         labelEntered.setText ("");
81         labelExited.setText ("");
82         labelPressed.setText ("AKTIF");
83         labelReleased.setText ("");
84     }
85
86     public void mouseReleased (MouseEvent e)
87     {
88         labelClicked.setText ("");
89         labelEntered.setText ("");
90         labelExited.setText ("");
91         labelPressed.setText ("");
92         labelReleased.setText ("AKTIF");
93     }
94 }
95
96 // -----
97 //  Class utama
98 // -----
99 public class MouseListener_1
100 {
101     public static void main (String[] args)
102     {
103         new MyFrame();
104     }
105 }

```

### Percobaan 3: Memantau kondisi mouse dan menerima event pada saat mouse digerakkan

```

1  import javax.swing.*;

```



```

2  import java.awt.*;
3  import java.awt.event.*;
4
5  class MyFrame extends JFrame
        implements MouseMotionListener
6  {
7      private Container container = new Container();
8      private JLabel labelDragged = new JLabel ("");
9      private JLabel labelMoved = new JLabel ("");
10
11     MyFrame()
12     {
13         super ("Test MouseListener");
14
15         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
16         setSize (250,80);
17         setLocation (100,100);
18
19         // -----
20         //  mengasosiasi seluruh objek dg event-handler
21         // -----
22         labelDragged.addMouseListener (this);
23         labelMoved.addMouseListener (this);
24
25         // -----
26         //  penambahan komponen ke frame
27         // -----
28         container = getContentPane();
29         container.setLayout (new GridLayout (2,2));
30         container.add (new JLabel ("mouseDragged :"));
31         container.add (labelDragged);
32         container.add (new JLabel ("mouseMoved :"));
33         container.add (labelMoved);
34
35         show();
36     }
37
38     public void mouseDragged (MouseEvent e)
39     {
40         labelDragged.setText ("AKTIF");
41         labelMoved.setText ("");
42     }
43
44     public void mouseMoved (MouseEvent e)
45     {
46         labelDragged.setText ("");
47         labelMoved.setText ("AKTIF");
48     }
49 }
50
51 // -----
52 //  Class utama
53 // -----
54 public class MouseMotionListener_1

```

```

55  {
56      public static void main (String[] args)
57      {
58          new MyFrame();
59      }
60  }

```

#### Percobaan 4: Mendeteksi penekanan tombol ENTER pada sebuah objek textfield

```

1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  class MyFrame extends JFrame implements KeyListener
6  {
7      private Container container = new Container();
8      private JTextField textField = new JTextField ("", 8);
9
10     MyFrame()
11     {
12         super ("Tebak angka");
13
14         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
15         setSize (250,150);
16         setLocation (100,100);
17
18         // -----
19         // mengasosiasi seluruh objek dg event-handler
20         // -----
21         textField.addKeyListener (this);
22
23         // -----
24         // penambahan komponen ke frame
25         // -----
26         container = getContentPane();
27         container.setLayout (new GridLayout (3,1));
28         container.add (new JPanel());
29         container.add (textField);
30         container.add (new JPanel());
31
32         show();
33     }
34
35     public void keyTyped (KeyEvent e)
36     {
37         if (e.getKeyCode() == e.VK_ENTER)
38             textField.setText("ENTER ->> ditekan + dilepas");
39     }
40
41     public void keyPressed (KeyEvent e)
42     {

```

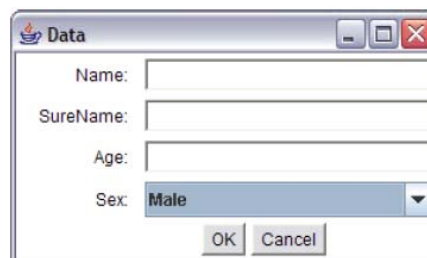
```

43     if (e.getKeyCode() == e.VK_ENTER)
44         textField.setText ("Tombol ENTER ->> ditekan");
45     }
46
47     public void keyReleased (KeyEvent e)
48     {
49         if (e.getKeyCode() == e.VK_ENTER)
50             textField.setText ("Tombol ENTER ->> dilepas");
51     }
52 }
53
54 // -----
55 // Class utama
56 // -----
57 public class KeyListener_1
58 {
59     public static void main (String[] args)
60     {
61         new MyFrame();
62     }
63 }

```

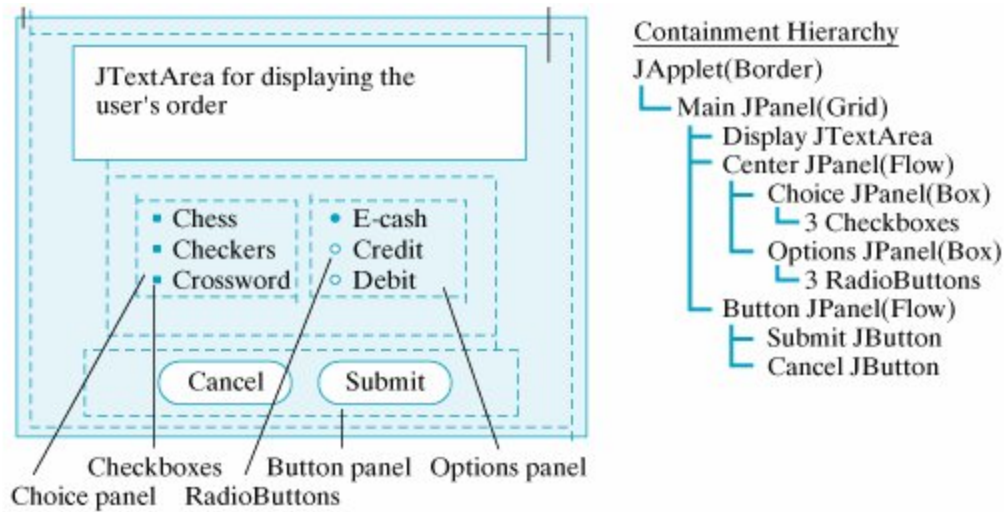
## E. LATIHAN

**Latihan 1: Membuat tampilan input data dan menampilkan hasil input ketika ditekan OK**



The image shows a Java Swing window titled "Data". It contains four input fields: "Name:", "SureName:", "Age:", and "Sex:". The "Sex:" field is a dropdown menu with "Male" selected. At the bottom of the window are "OK" and "Cancel" buttons.

**Latihan 2 : Buat Tampilan dan event handling yang sesuai, seperti pada gambar**



**Latihan 3: Modifikasi latihan 2 untuk pembelian tiket online pada konser music**

**F. LAPORAN RESMI**

Kumpulkan hasil percobaan di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat.