



# GUI Event Handling



# ***Pengantar Event Handling***

- Aplikasi GUI bersifat **event driven**
- Saat user berinteraksi dengan komponen GUI, interaksi tsb dikenal sbg **event**, akan memicu program untuk melaksanakan sebuah proses (*task*).
- Contoh interaksi yang memicu event :
  - ☐ Klik tombol
  - ☐ Menulis di textfield
  - ☐ Memilih dari check box
  - ☐ Menggerakkan mouse di frame
  - ☐ Memilih menu, dsb



## ***Pengantar Event Handling***

- Kode program yang akan dijalankan saat terjadi event disebut dengan ***event handler***.
- Dan keseluruhan proses penanganan event disebut dengan ***event handling***.



# ***Delegation Event Model***

Delegasi event model menguraikan bagaimana program Anda dapat merespon interaksi dari user. Untuk memahami model, kita pelajari pertama-tama dengan tiga komponen utamanya.

## **1. Event Source**

Event source mengacu pada komponen GUI yang men-generate event. Sebagai contoh, jika user menekan tombol, event source dalam hal ini adalah tombol.

## **2. Event Listener/Handler**

Event listener menerima berita dari event-event dan memprosesnya.

## **3. Event Object**

Objek yang terbentuk saat terjadi event



## ***Class Event***

- Sebuah event object mempunyai sebuah class event sebagai tipe data acuannya. Akar dari hirarki class event adalah class *EventObject*, yang dapat ditemukan pada paket *java.util*.



# Class Event

<i>Class Event</i>	<i>Deskripsi</i>
ComponentEvent	Extends <i>AWTEvent</i> . Dijalankan ketika sebuah komponen dipindahkan, di-resize, dibuat <i>visible</i> atau <i>hidden</i> .
InputEvent	Extends <i>ComponentEvent</i> . Abstrak root class event untuk semua komponen-level input class-class event.
ActionEvent	Extends <i>AWTEvent</i> . Dijalankan ketika sebuah tombol ditekan, melakukan double-klik daftar item, atau memilih sebuah menu.
ItemEvent	Extends <i>AWTEvent</i> . Dijalankan ketika sebuah item dipilih atau di- <i>deselect</i> oleh user, seperti sebuah list atau checkbox.



# Class Event

<i>Class Event</i>	<i>Deskripsi</i>
KeyEvent	Extends <i>InputEvent</i> . Dijalankan ketika sebuah <i>key</i> ditekan, dilepas atau diketikkan.
MouseEvent	Extends <i>InputEvent</i> . Dijalankan ketika sebuah tombol mouse ditekan, dilepas, atau di-klik (tekan dan lepas), atau ketika sebuah kursor mouse masuk atau keluar dari bagian visible dari komponen.
TextEvent	Extends <i>AWTEvent</i> . Dijalankan ketika nilai dari text field atau text area dirubah.
WindowEvent	Extends <i>ComponentEvent</i> . Dijalankan sebuah objek <i>Window</i> dibuka, ditutup, diaktifkan, nonaktifkan, <i>iconified</i> , <i>deiconified</i> , atau ketika <i>focus</i> ditransfer kedalam atau keluar window.



# Event Listener

- Event listeners adalah class yang mengimplementasikan interfaces *<Type>Listener*.

Tabel di bawah menunjukkan beberapa listener interfaces yang biasanya digunakan.

<i>Class Event</i>	<i>Deskripsi</i>
ActionListener	Bereaksi atas perubahan mouse atau keyboard.
MouseListener	Bereaksi atas pergerakan mouse.
MouseMotionListener	Interface MouseMotionListener mendukung MouseListener. Menyediakan method-method yang akan memantau pergerakan mouse, seperti drag dan pemindahan mouse.
WindowListener	Bereaksi atas perubahan window.





# ***Method ActionListener***

- Interface ActionListener hanya terdiri dari satu method.

## ***ActionListener Method***

```
public void actionPerformed(ActionEvent e)
```

Mengendalikan *ActionEvent e* yang terjadi.



# ***Method MouseListener***

## ***MouseListener Method***

**public void mouseClicked(MouseEvent e)**

Dipanggil pada saat tombol mouse di click (seperti tekan dan lepas).

**public void mouseEntered(MouseEvent e)**

Dipanggil pada saat kursor mouse memasuki area komponen.

**public void mouseExited(MouseEvent e)**

Dipanggil pada saat kursor mouse meninggalkan area komponen.

**public void mousePressed(MouseEvent e)**

Dipanggil pada saat tombol mouse ditekan di atas komponen

**public void mouseReleased(MouseEvent e)**

Dipanggil pada saat tombol mouse dilepas di atas komponen



# ***Method MouseMotionListener***

## ***MouseMotionListener Method***

### **public void mouseDragged(MouseEvent e)**

Digunakan untuk memantau pergerakan mouse yang melintasi objek pada saat tombol mouse ditekan. Tindakan ini persis sama dengan tindakan pada saat memindahkan sebuah window.

### **public void mouseMoved(MouseEvent e)**

Digunakan untuk memantau pergerakan mouse pada saat mouse melintasi area suatu objek.



# Method WindowListener

## WindowListener Method

### **public void windowOpened(WindowEvent e)**

Dipanggil pada saat objek window dibuka (pertama kali window dibuat tampil).

### **public void windowClosing(WindowEvent e)**

Dipanggil pada saat user mencoba untuk menutup objek *Window* dari menu *sistem* objek.

### **public void windowClosed(WindowEvent e)**

Dipanggil pada saat objek *Window* ditutup setelah memanggil penempatan (*misal*, release dari resource-resource yang digunakan oleh source) pada objek.

### **public void windowActivated(WindowEvent e)**

Dilibatkan ketika objek *Window* adalah window yang aktif (window masih dipakai).

### **public void windowDeactivated(WindowEvent e)**

Dilibatkan ketika objek *Window* tidak lagi merupakan window yang aktif.

### **public void windowIconified(WindowEvent e)**

Dipanggil ketika objek *Window* di-minimize.

### **public void windowDeiconified(WindowEvent e)**

Dipanggil ketika objek *Window* kembali setelah di-minimize ke keadaan normal.



# ***Kelas Adapter***

- Merupakan kelas yang digunakan untuk membuat event listener
- Tidak harus mengimplementasikan seluruh method
- Digunakan dengan cara meng-extends dari kelas adapter

## **Macam-macam Kelas Adapter**

- **ComponentAdapter**
- **ContainerAdapter**
- **FocusAdapter**
- **KeyAdapter**
- **MouseAdapter**
- **MouseMotionAdapter**
- **WindowAdapter**