

IF34348 - PEMROGRAMAN LANJUT

EVENT

13

Oleh : Andri Heryandi, M.T.

EVENT

IF34348 - Pemrograman Lanjut

- Event adalah suatu kejadian dimana kondisi dari suatu objek berubah.
- Event dibuat sebagai hasil interaksi user dengan komponen GUI. Sebagai contoh adanya klik pada tombol, adanya pergerakan mouse, atau pun ketika adanya pemasukan karakter melalui keyboard.
- Event Handling (penanganan event) adalah suatu mekanisme yang mengendalikan event dan menentukan apa yang seharusnya dilakukan ketika event terjadi.
- Di Java, semua event yang terjadi akan menghasilkan sebuah class Event.
- Dan class yang digunakan untuk menangani event adalah Listener.



JENIS-JENIS EVENT

IF34348 - Pemrograman Lanjut

- **Class-Class event yang tersedia ada berbagai macam, diantaranya :**
 - **Class ActionEvent** : class yang terjadi ketika terjadi penekanan tombol, atau ada double-klik.
 - **Class KeyEvent** : class yang terjadi ketika memasukan karakter.
 - **Class MouseEvent** : class yang terjadi ketika sebuah aksi mouse terjadi.
 - **Class WindowEvent** : class yang terjadi ketika ada perubahan status di window.
- **Setiap jenis event memiliki method dan atribut masing-masing.**



JENIS-JENIS LISTENER

IF34348 - Pemrograman Lanjut

- Event Listener merepresentasikan interface yang bertanggungjawab untuk menangani event.
- Jenis-jenis listener
 - ActionListener : Listener untuk(ActionEvent).
 - KeyListener : Listener untuk(KeyEvent).
 - MouseListener : Listener untuk(MouseEvent).
 - WindowListener : Listener untuk(WindowEvent).
- Setiap Listener dapat diimplementasikan di class anda.
- Setiap Listener mempunyai method yang harus diimplementasikan.



Oleh : Andri Heryandi, M.T.

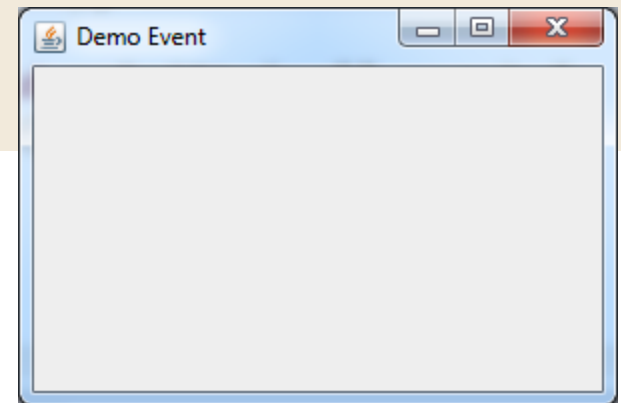
CONTOH MENANGKAP EVENT

IF34348 - Pemrograman Lanjut

■ Contoh aplikasi yang belum mempunyai event.

```
import javax.swing.*;

public class DemoEvent extends JFrame{
    public DemoEvent(){
        super("Demo Event");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300,200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new DemoEvent();
    }
}
```



Oleh : Andri Heryandi, M.T.

CONTOH MENANGKAP EVENT

IF34348 - Pemrograman Lanjut

■ Tambahkan implement ke MouseListener

```
import javax.swing.*;
```

```
public class DemoEvent extends JFrame implements MouseListener{
```

■ Tambahkan import ke import java.awt.event.MouseListener;

```
import java.awt.event.MouseListener;
```

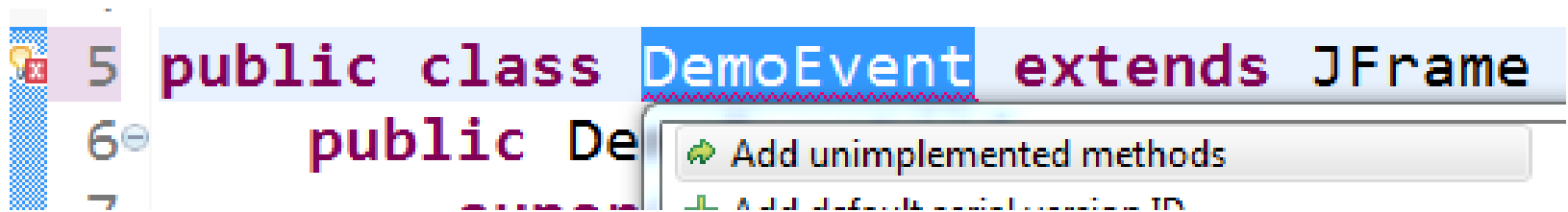
```
import javax.swing.*;
```

```
public class DemoEvent extends JFrame implements MouseListener{
```

CONTOH MENANGKAP EVENT

IF34348 - Pemrograman Lanjut

- Pasti ada error di baris pembuatan class, ini dikarenakan kita belum memiliki method-method yang ada di interface `MouseListener`. Klik di [x] errornya, pilih “Add unimplemented methods”.



CONTOH MENANGKAP EVENT

IF34348 - Pemrograman Lanjut

- Secara otomatis eclipse akan membuat method-method turunan dari interface `MouseListener`

```
@Override
public void mouseClicked(MouseEvent e) {
    // TODO Auto-generated method stub
}
@Override
public void mouseEntered(MouseEvent e) {
    // TODO Auto-generated method stub
}
@Override
public void mouseExited(MouseEvent e) {
    // TODO Auto-generated method stub
}
@Override
public void mousePressed(MouseEvent e) {
    // TODO Auto-generated method stub
}
@Override
public void mouseReleased(MouseEvent e) {
    // TODO Auto-generated method stub
}
```



CONTOH MENANGKAP EVENT

IF34348 - Pemrograman Lanjut

- Ubahlah setiap method tersebut sesuai keperluannya.

```
public void mouseClicked(MouseEvent e) {
    if(e.getButton()==MouseEvent.BUTTON1)
        System.out.println("Tombol Kiri");
    else
        if(e.getButton()==MouseEvent.BUTTON2)
            System.out.println("Tombol Tengah");
        else
            if(e.getButton()==MouseEvent.BUTTON3)
                System.out.println("Tombol Kanan");

    System.out.println("Mouse diklik di posisi : "+e.getX()+","+e.getY()); }
public void mouseEntered(MouseEvent e) {
    System.out.println("Mouse masuk");
}
public void mouseExited(MouseEvent e) {
    System.out.println("Mouse Keluar");
}
public void mousePressed(MouseEvent e) {
    System.out.println("Mouse ditekan");
}
public void mouseReleased(MouseEvent e) {
    System.out.println("Mouse dilepas");
}
```



CONTOH MENANGKAP EVENT

IF34348 - Pemrograman Lanjut

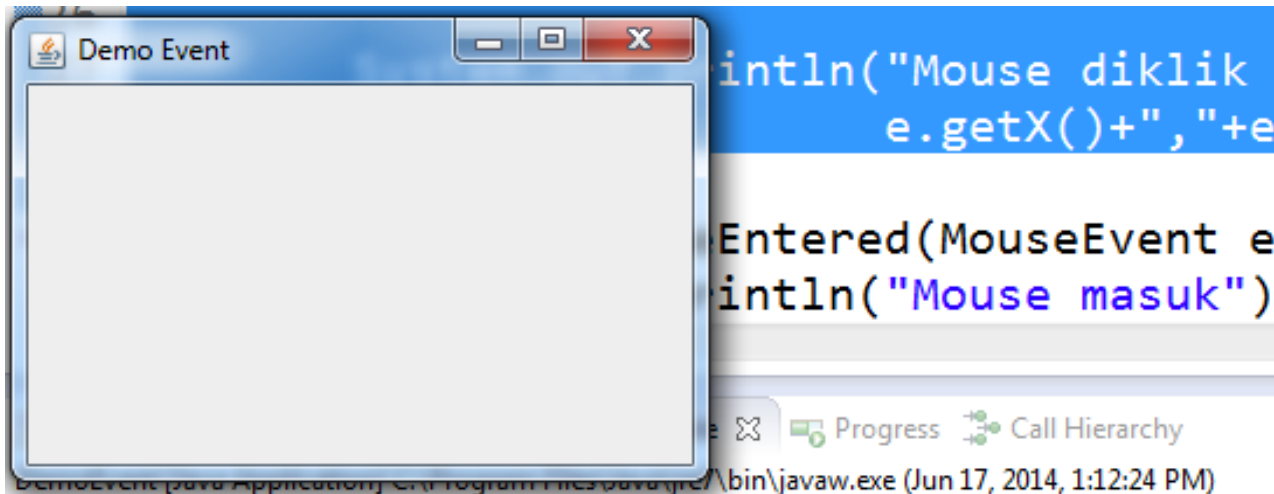
■ Tambahkan addMouseListener.

```
public DemoEvent() {  
    super("Demo Event");  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setSize(300,200);  
  
    addMouseListener(this);  
  
    setVisible(true);  
}
```

CONTOH MENANGKAP EVENT

IF34348 - Pemrograman Lanjut

- Run. Gerakan aktivitas yang dilakukan oleh Mouse.



Tombol Kanan
Mouse diklik di posisi : 121,54
Mouse Keluar
Mouse masuk

LATIHAN

IF34348 - Pemrograman Lanjut

- Dengan cara yang seperti tadi, silahkan pelajari Event-Event listener yang lain :
 - ActionListener
 - KeyListener
 - WindowListener



Oleh : Andri Heryandi, M.T.

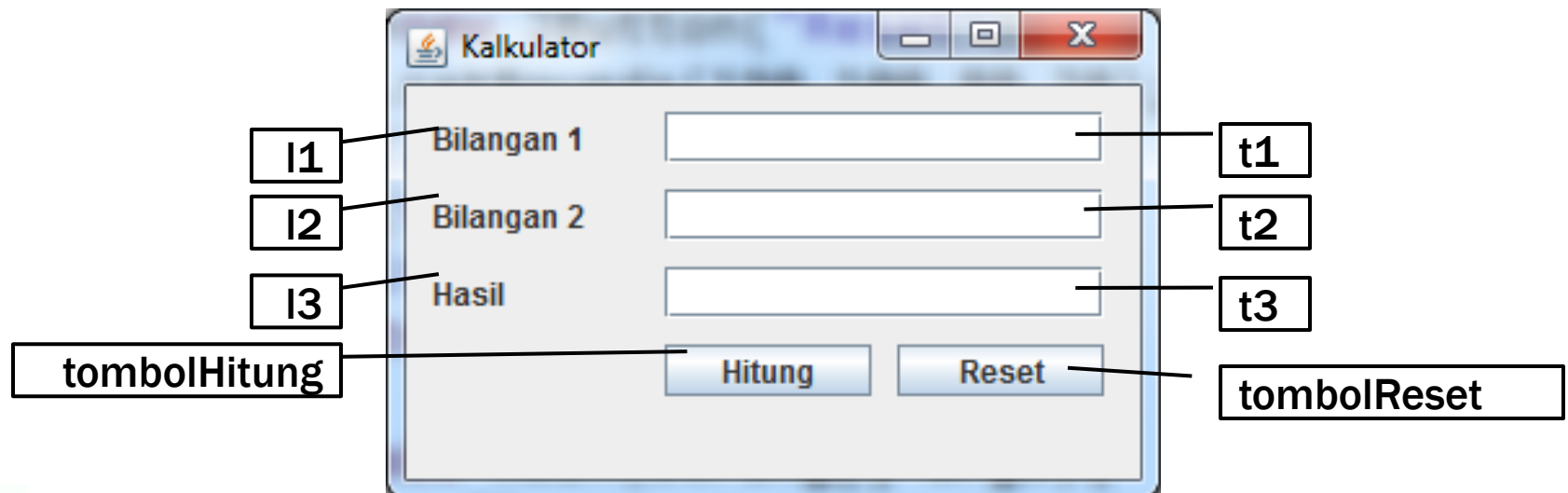
MERESPON AKSI KLIK

IF34348 - Pemrograman Lanjut

■ Kasus:

Buatlah sebuah aplikasi untuk menghitung hasil perkalian antara 2 buah bilangan dan **menampilkan hasilnya ketika tombol Hitung diklik**. Sedangkan **tombol Reset akan mendefaultkan bilangan 1 dan bilangan 2 dengan nilai 0**.

Contoh tampilan :



MERESPON AKSI KLIK

IF34348 - Pemrograman Lanjut

```
import javax.swing.*;
public class Kalkulator extends JFrame{
    private JButton tombolHitung,tombolReset;
    private JLabel l1,l2,l3;
    private JTextField t1,t2,t3;
    public Kalkulator(){
        super("Kalkulator");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300,190);
        setLayout(null);
        l1=new JLabel("Bilangan 1");l1.setBounds(10,10,90,20);
        l2=new JLabel("Bilangan 2");l2.setBounds(10,40,90,20);
        l3=new JLabel("Hasil ");l3.setBounds(10,70,90,20);
        t1=new JTextField();t1.setBounds(100,10,170,20);
        t2=new JTextField();t2.setBounds(100,40,170,20);
        t3=new JTextField();t3.setBounds(100,70,170,20);
        tombolHitung=new JButton("Hitung");
        tombolHitung.setBounds(100,100,80,20);
        tombolReset=new JButton("Reset");
        tombolReset.setBounds(190,100,80,20);
        add(l1);add(l2);add(l3);
        add(t1);add(t2);add(t3);
        add(tombolHitung);add(tombolReset);
        setVisible(true);
    }
    public static void main(String[] args){
        new Kalkulator();
    }
}
```



MERESPON AKSI KLIK

IF34348 - Pemrograman Lanjut

■ Respons ketika tombolHitung diklik.

```
tombolHitung.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        double b1,b2,b3;  
        b1=Double.parseDouble(t1.getText());  
        b2=Double.parseDouble(t2.getText());  
        b3=b1*b2;  
        t3.setText(Double.toString(b3));  
    }  
});
```

- Tambahkan potongan source code di atas ke dalam aplikasi. Boleh ditempatkan sebelum setVisible.



MERESPON AKSI KLIK

IF34348 - Pemrograman Lanjut

■ Respons ketika tombolReset diklik.

```
tombolReset.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        t1.setText("0");  
        t2.setText("0");  
        t3.setText("0");  
        t1.requestFocus();  
    }  
});
```

- Tambahkan potongan source code di atas didalam aplikasi. Boleh ditempatkan sebelum setVisible.

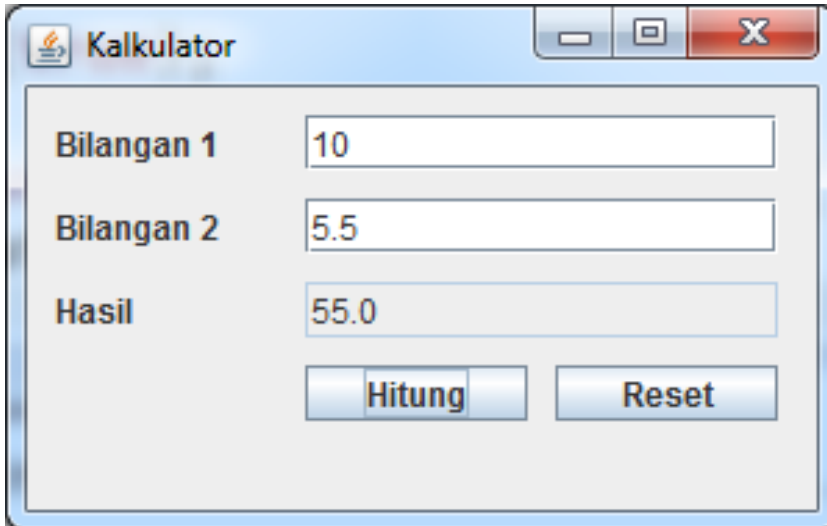


MERESPON AKSI KLIK

IF34348 - Pemrograman Lanjut

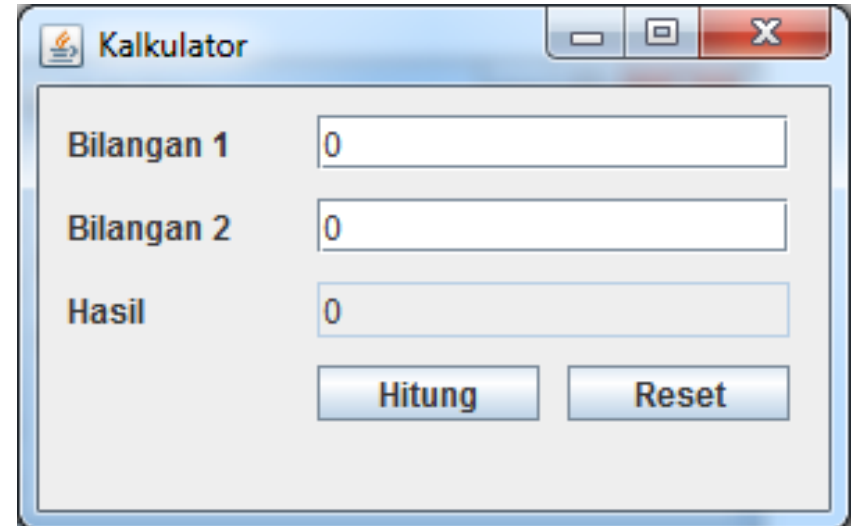
■ Hasil Run.

TombolHitung diklik



Bilangan 1	10
Bilangan 2	5.5
Hasil	55.0
<input type="button" value="Hitung"/> <input type="button" value="Reset"/>	

TombolReset diklik



Bilangan 1	0
Bilangan 2	0
Hasil	0
<input type="button" value="Hitung"/> <input type="button" value="Reset"/>	