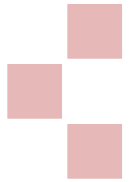
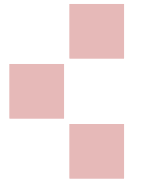


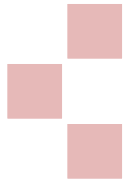
GUI, Event Handling, Exception Handling in Java

Nurochman



Apa itu GUI?

- GUI > Graphical User Interface
- Tanpa GUI program kita kurang menarik
- Program GUI terdiri dari Container dan Component
- Container sebagai wadah
- Component adalah sesuatu yg bs ditempelkan pd container
- Program GUI minimal memiliki satu container
- Container bisa berlapis (level paling atas disebut top level container)



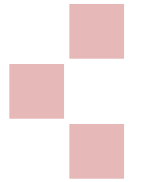
Top-level Container

- **JFrame**

  untuk GUI Desktop Application
- **JDialog**

  untuk menampilkan kotak dialog
- **JApplet**

  untuk aplikasi applet pada halaman web site



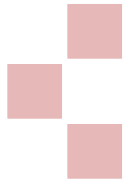
GUI

- Java GUI > AWT dan Swing
- AWT > Abstract Window Toolkit
- Beberapa komponen AWT ditulis dengan *native code*
- Swing ditulis dengan bahasa Java
- Swing > platform independent, artinya platform berbeda mempunyai tampilan sama
- AWT dan Swing dpt digunakan bersama
- Disarankan menggunakan Swing

Komponen AWT

Class AWT	*Deskripsi*
Komponen	Abstract Class untuk object yang dapat ditampilkan pada console dan berinteraksi dengang user. Bagian utama dari semua class AWT.
Kontainer	Abstract Subclass dari Component Class. Sebuah komponen yang dapat menampung komponen yang lainnya.
Panel	Turunan dari Container Class. Sebuah frame atau window tanpa titlebar, menubar tidak termasuk border. Superclass dari applet class.
Window	Turunan dari Container class. Top level window, dimana berarti tidak bisa dimasukkan dalam object yang lainnya.Tidak memiliki border dan menubar.

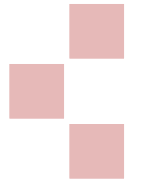
Frame	Turunan dari window class. Window dengan judul, menubar, border dan pengatur ukuran di pojok. Memiliki empat constructor , dua diantaranya memiliki penulisan seperti dibawah ini :
	`Frame()`
	`Frame(String title)`



Contoh GUI AWT

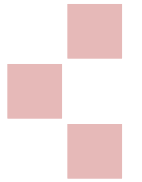
```
package gui;

import java.awt.Frame;

public class SampleFrame extends Frame {

    public SampleFrame() { }


    public static void main(String args[]) {

        SampleFrame sf = new SampleFrame();

        sf.setSize(300, 300); //Coba hilangkan baris ini

        sf.setVisible(true); //Coba hilangkan baris ini } }

    }

}
```

Frame di atas belum bisa ditutup dengan menekan tombol close pada pojok kanan atas, Karena belum ada mekanisme Event Handling



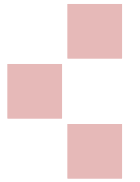
Method dalam class Graphic

drawLine()	drawPolyline()	setColor()
fillRect()	drawPolygon()	getFont()
drawRect()	fillPolygon()	setFont()
clearRect()	getColor()	drawString()



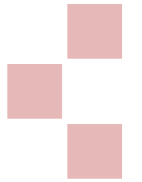
Constructor pada class Color

Format Constructor	*Deskripsi*
Color(int r, int g, int b)	Nilai integer 0 - 255.
Color(float r, float g, float b)	Nilai float 0.0 - 1.0.
Color(int rgbValue)	Panjang nilai : 0 ke 2^{24}-1 (hitam ke putih). Red: bits 16-23 Green: bits 8-15 Blue: bits 0-7

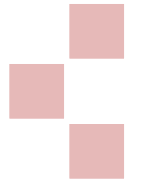


Contoh Graphic

```
package gui;
import java.awt.Color;
import java.awt.Font;
import java.awt.Frame;
import java.awt.Graphics;
import java.awt.Panel;
public class GraphicPanel extends Panel {
    public GraphicPanel() {
        setBackground(Color.black);
    }

    public void paint(Graphics g) {
        g.setColor(new Color(0,255,0)); //green
        g.setFont(new Font("Helvetica",Font.PLAIN,16));
        g.drawString("Hello GUI World!", 30, 100);
        g.setColor(new Color(1.0f,0,0)); //red
        g.fillRect(30, 100, 150, 10);
```



```
    }


    public static void main(String args[]) {
        Frame f = new Frame("Testing Graphics Panel");
        GraphicPanel gp = new GraphicPanel();
        f.add(gp);
        f.setSize(600, 300);
        f.setVisible(true);
    }
}
```

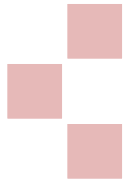


Komponen AWT

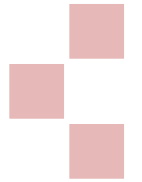
Label	Button	Choice
TextField	Checkbox	List
TextArea	CheckboxGroup	Scrollbar

Contoh GUI dg Komponen

```
import java.awt.*;
public class FrameControls extends Frame {
    public FrameControls() {
        FrameControls fwc = new FrameControls();
        fwc.setLayout(new FlowLayout()); //more on this later
        fwc.setSize(600, 600);
        fwc.add(new Button("Test Me!"));
        fwc.add(new Label("Label"));
        fwc.add(new TextField());
        CheckboxGroup cbg = new CheckboxGroup();
        fwc.add(new Checkbox("chk1", cbg, true));
        fwc.add(new Checkbox("chk2", cbg, false));
        fwc.add(new Checkbox("chk3", cbg, false));
        List list = new List(3, false);
        list.add("MTV");
        list.add("SCTV");
```

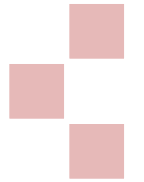


```
        fwc.add(list);
        Choice chooser = new Choice();
        chooser.add("Avril");
        chooser.add("Monica");
        chooser.add("Britney");
        fwc.add(chooser);
        fwc.add(new Scrollbar());
        fwc.setVisible(true);
    }
}
```



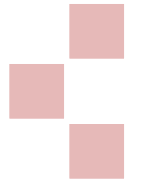
Layout

- **Where does a component go?**
- **Container uses a layout to decide**
- **Different layouts**
  - Flow Layout
  - Border Layout
  - Grid Layout
  - Card Layout

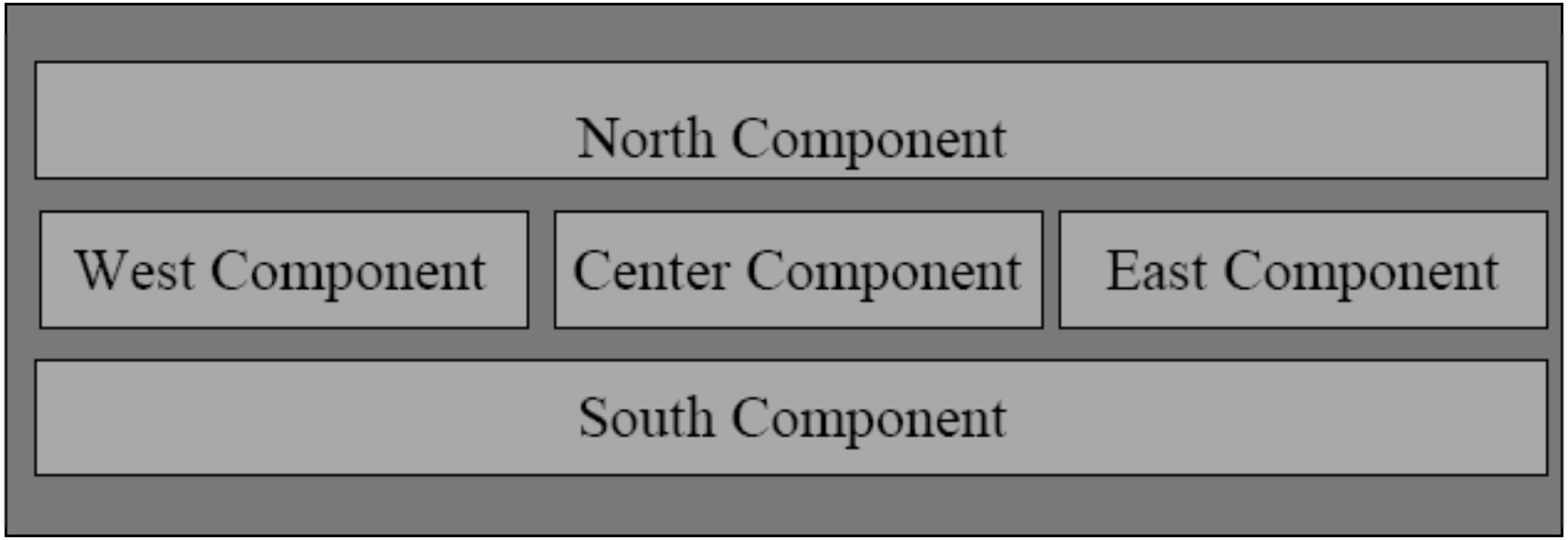


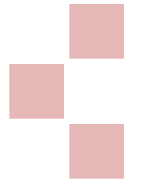
Advantages of layouts

- Co-ordinate positioning might result in component overlaps
- You need take no special action when the container is resized
- Most user interfaces consist of several containers which may employ different layouts

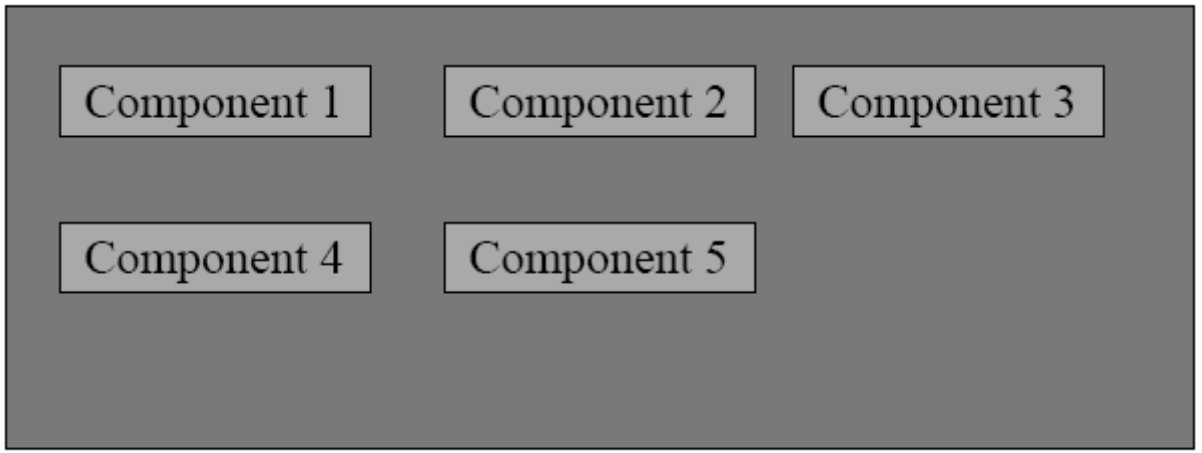


Border Layout

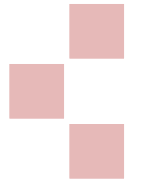




Flow Layout

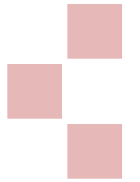


Can be left, centre or right aligned.
Picture shows left alignment

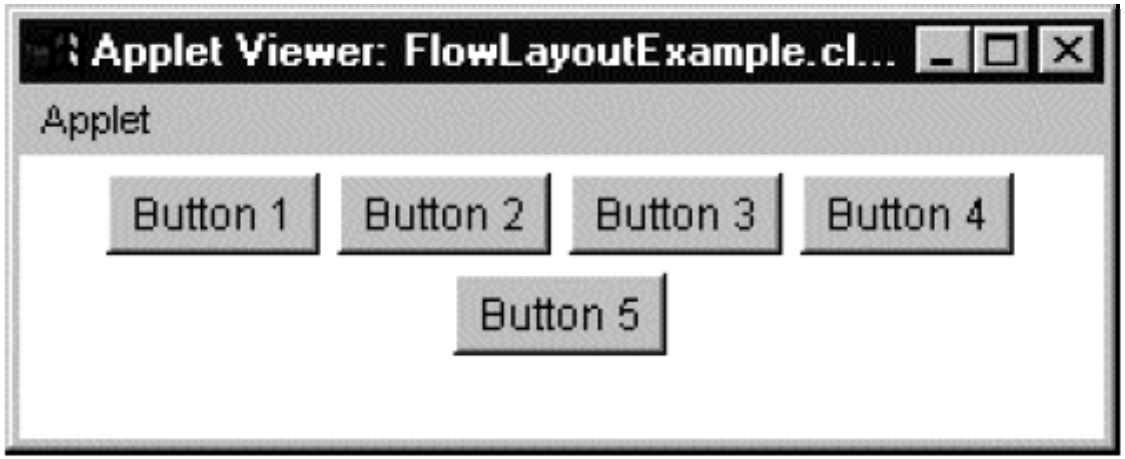


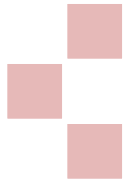
Example of Flow Layout

```
import java.applet.*;
import java.awt.*;
public class FlowLayoutExample extends Applet{
    public void init(){
        setLayout(new FlowLayout());
        add(new Button("Button 1"));
        add(new Button("Button 2"));
        add(new Button("Button 3"));
        add(new Button("Button 4"));
        add(new Button("Button 5"));
    }
}
```

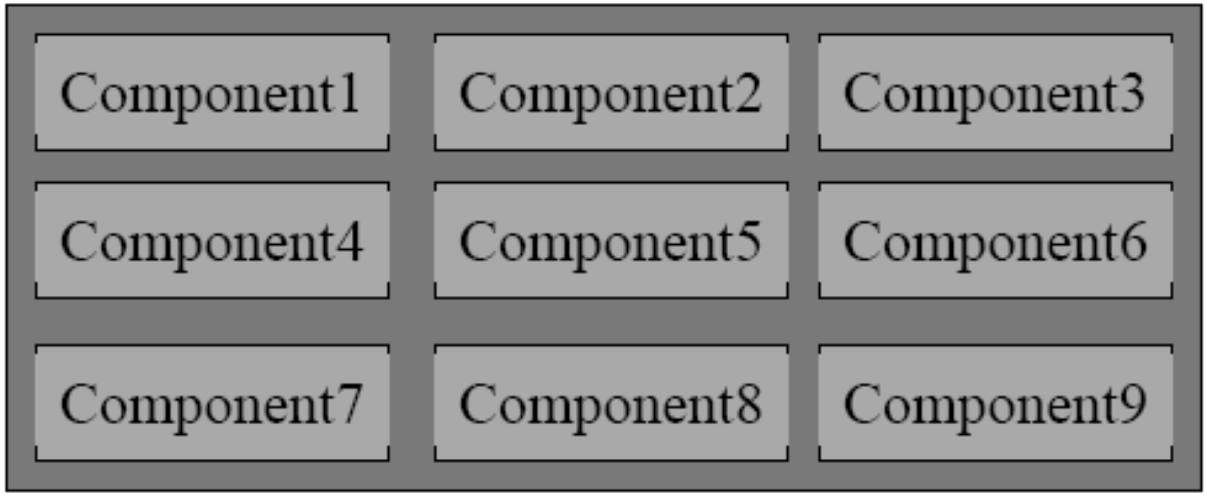


Flow Layout


Applet Viewer: FlowLayoutExample.cl...
Applet
Button 1
Button 2
Button 3
Button 4
Button 5

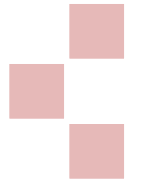


Grid Layout



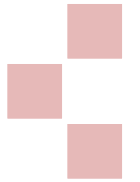
An M*N matrix of components

```
public GridLayout(int rows, int cols)
```

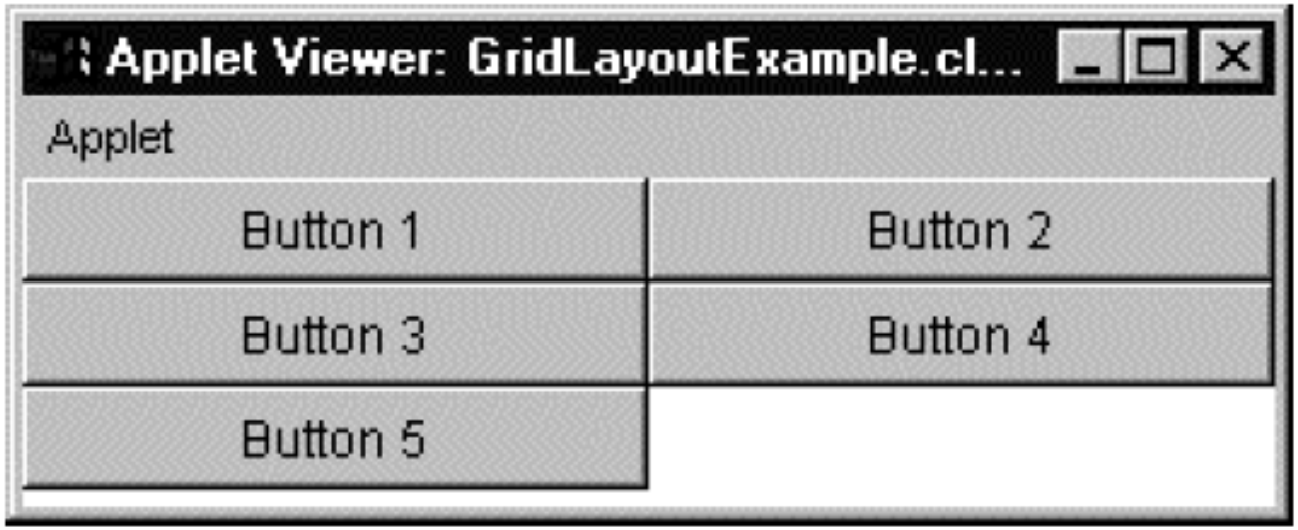


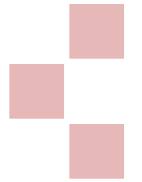
Grid Layout Example

```
import java.applet.*;
import java.awt.*;
public class GridLayoutExample extends Applet{
   public void init(){
        setLayout(new GridLayout(3,2));
        add(new Button("Button 1"));
        add(new Button("Button 2"));
        add(new Button("Button 3"));
        add(new Button("Button 4"));
        add(new Button("Button 5"));
   }
}
```



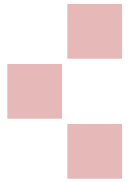
Grid Layout


Applet Viewer: GridLayoutExample.cl...
Applet
Button 1
Button 2
Button 3
Button 4
Button 5



Card Layout

- Not designed to lay components out
- Displays one component at a time
- Every component is a "card“
- Components are ordered in a stack
- Use for implementing HyperCard style applications and applets

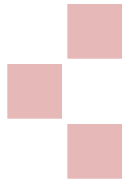


Swing

- Bagian dari JFC (Java Foundation Class)
- Package javax.swing.*
- Penamaan class diawali huruf ‘J’
- Misal JFrame, JButton, JLabel, JTextField,

Komponen Swing

Komponen Swing	*Penjelasan*
JComponent	class induk untuk semua komponen Swing, tidak termasuk top-level kontainer
JButton	Tombol "push". Berhubungan dengan class button dalam package AWT
JCheckBox	Item yang dapat dipilih atau tidak oleh pengguna. Berhubungan dengan class checkbox dalam package AWT
JFileChooser	Mengijinkan pengguna untuk memilih sebuah file. Berhubungan dengan class filechooser dalam package AWT

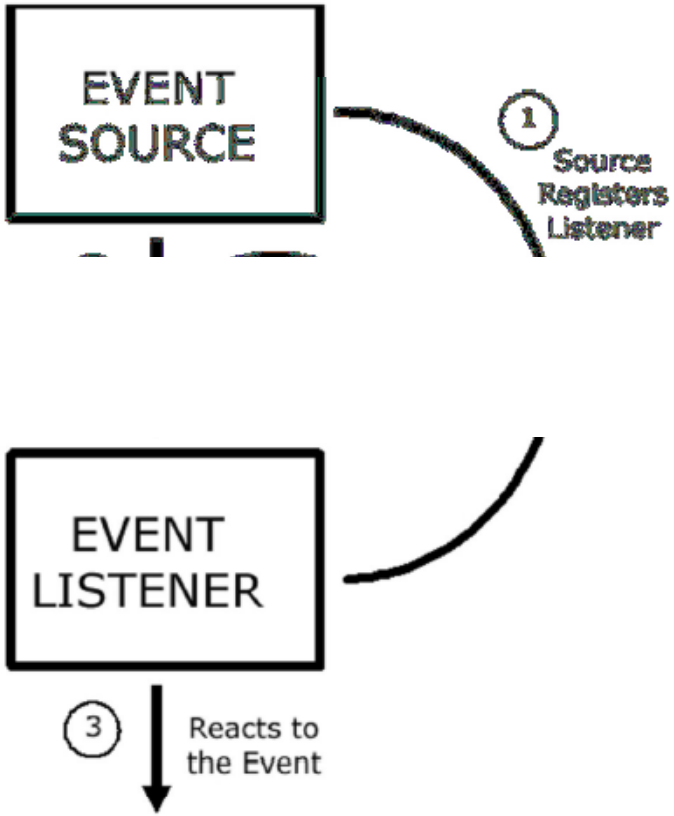


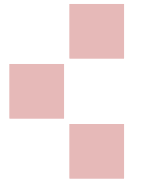
Komponen Swing

Komponen Swing	*Penjelasan*
JTextField	Mengijinkan untuk mengedit text satu baris. Berhubungan dengan class textfield dalam package AWT.
JFrame	Turunan dan Berhubungan dengan class frame dalam package AWT tetapi keduanya sedikit tidak cocok dalam kaitannya dengan menambahkan komponen pada kontainer. Perlu mendapatkan content pane yang terbaru sebelum menambah sebuah komponen.
JPanel	Turunan Jcomponent. Class Container sederhana tetapi bukan top-level. Berhubungan dengan class panel dalam package AWT.
JApplet	Turunan dan Berhubungan dengan class Applet dalam package AWT. Juga sedikit tidak cocok dengan class applet dalam kaitannya dengan menambahkan komponen pada container
JOptionPane	Turunan Jcomponent. Disediakan untuk mempermudah menampilkan pop-up kotak dialog.
JDialog	Turunan dan Berhubungan dengan class dialog dalam package AWT. Biasanya digunakan untuk menginformasikan sesuatu kepada pengguna atau prompt pengguna untuk input.
JColorChooser	Turunan Jcomponent. Memungkinkan pengguna untuk memilih warna yang diinginkan.

Event Handling

- Bagaimana GUI dapat merespons aksi yang dilakukan user?
- Delegation event model merupakan model bagaimana program dapat merespon interaksi dari user.



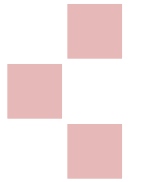


Delegation Event Model

- **Event Source**, Event source mengacu pada komponen GUI yang meng generate event. Sebagai contoh, jika user menekan tombol, event source dalam hal ini adalah tombol.
- **Event Listener/Handler**, Event listener menerima berita dari event event dan proses proses interaksi user. Ketika tombol ditekan, listener akan mengendalikan dengan menampilkan sebuah informasi yang berguna untuk user.
- **Event Object**, Ketika sebuah event terjadi (misal, ketika user berinteraksi dengan komponen GUI), sebuah object event diciptakan. Object berisi semua informasi yang perlu tentang event yang telah terjadi. Informasi meliputi tipe dari event yang telah terjadi, seperti ketika mouse telah di klik. Ada beberapa class event untuk kategori yang berbeda dari user action. Sebuah event object mempunyai tipe data mengenai salah satu dari class ini.

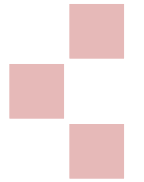
Class-class Event

Class Event	*Deskripsi*
ComponentEvent	Extends *AWTEvent*. Dijalankan ketika sebuah komponen dipindahkan, di-*resize*, dibuat *visible* atau *hidden*.
InputEvent	Extends *ComponentEvent*. Abstrak root class event untuk semua komponen-level input class-class event.
ActionEvent	Extends *AWTEvent*. Dijalankan ketika sebuah tombol ditekan, melakukan double-klik daftar item, atau memilih sebuah menu.
ItemEvent	Extends *AWTEvent*. Dijalankan ketika sebuah item dipilih atau di-*deselect* oleh user, seperti sebuah list atau checkbox.
KeyEvent	Extends *InputEvent*. Dijalankan ketika sebuah *key* ditekan, dilepas atau diketikkan.
MouseEvent	Extends *InputEvent*. 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 *AWTEvent*. Dijalankan ketika nilai dari text field atau text area dirubah.
WindowEvent	Extends *ComponentEvent*. Dijalankan sebuah object *Window* dibuka, ditutup, diaktifkan, nonaktifkan, *iconified*, *deiconified*, atau ketika *focus* ditransfer kedalam atau keluar window.



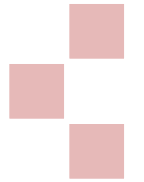
Event Listeners

Event Listeners	*Deskripsi*
ActionListener	Bereaksi atas perubahan mouse atau 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.



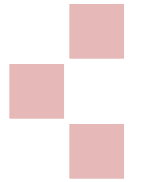
Exception

- Bugs dan Error sering muncul meski programmer hebat
- Mekanisme exception handling akan menghemat waktu error checking
- Exception > Exceptional Events
- Runtime exception mengganggu aliran program



Contoh runtime error

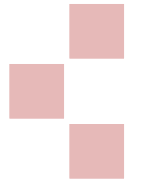
- Pembagian dengan 0
- Mengakses elemen diluar jangkauan array
- Input tidak benar
- Membuka file yang tidak ada
- dll



contoh

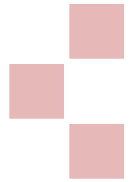
```
class DivByZero {
        public static void main(String args[]) {
                System.out.println(3/0);
                System.out.println("Cetak.");
        }
}
```

```
Exception in thread "main" java.lang.ArithmeticException: / by
zero at DivByZero.main(DivByZero.java:3)
```



Menangkap exception dg try...catch

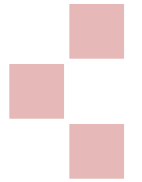
```
try {
        <code to be monitored for exceptions>
} catch (<ExceptionType1> <ObjName>) {
        <handler if ExceptionType1 occurs>
}
...
} catch (<ExceptionTypeN> <ObjName>) {
        <handler if ExceptionTypeN occurs>
}
```



contoh

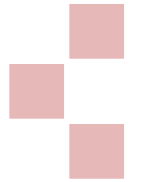
```
class DivByZero {
        public static void main(String args[]) {
                try {
                        System.out.println(3/0);
                        System.out.println("Cetak.");
                } catch (ArithmeticException exc) {
                        //Reaksi atas kejadian
                        System.out.println(exc);
                }
                System.out.println("Setelah Exception.");
        }
}
```

```
java.lang.ArithmeticException: / by zero
After exception.
```



Keyword finally

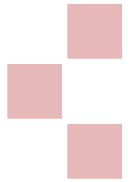
```
try {
        <kode monitor exception>
} catch (<ExceptionType1> <ObjName>) {
        <penanganan jika ExceptionType1 terjadi>
} ...
} finally {
        <kode yang akan dieksekusi saat blok try berakhir>
}
```



Keyword throw

```
throw <exception object>;
```

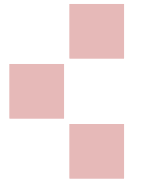
```
class ThrowDemo {
    public static void main(String args[]){
            String input = "invalid input";
            try {
                    if (input.equals("invalid input")) {
                            throw new RuntimeException("throw demo");
                    } else {
                            System.out.println(input);
                    }
                    System.out.println("After throwing");
                    } catch (RuntimeException e) {
                            System.out.println("Exception caught here.");
                            System.out.println(e);
            }
    }
}
```



Keyword throws

```
<type> <methodName> (<parameterList>) throws <exceptionList> {
      <methodBody>
}
```

```
class ThrowingClass {
       static void myMethod() throws ClassNotFoundException {
              throw new ClassNotFoundException ("just a demo");
       }
}

class ThrowsDemo {
       public static void main(String args[]) {
              try {
                     ThrowingClass.myMethod();
              } catch (ClassNotFoundException e) {
                     System.out.println(e);
              }
       }
}
```



Langkah akses DB dengan JDBC

- Mendaftarkan driver sesuai jenis database yang akan diakses.
  Untuk MS Access menggunakan ODBC:

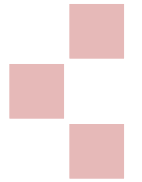
```
DriverManager.registerDriver(new
  sun.jdbc.odbc.JdbcOdbcDriver());
```

- Membuat koneksi ke database yang dituju.

```
Connection conn = DriverManager.getConnection(String
  url, String user, String password);
```

- Format url untuk database Ms Access sebagai berikut:
  “jdbc:odbc:DSN”

  Ket: DSN merupakan data source name yang didefinisikan dan diarahkan ke database yang akan diakses.



JDBC (2)

- Membuat objek statement dari koneksi yang telah dibuat.

```
Statement stmt = conn.createStatement();
```

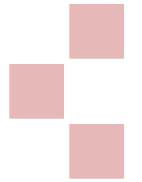
- Memanggil method untuk mengeksekusi query.

  Untuk query select:

```
ResultSet rs = stmt.executeQuery("select *
  from tableName");
```

  Untuk query insert, update, delete:

```
stmt.executeUpdate("DELETE FROM TABLENAME
  WHERE FIELD = VALUE");
```



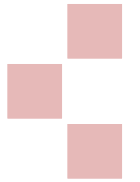
JDBC (3)

- Mengekstrak resultset yang didapat dari query (untuk query select)

```
while (rs.next()) {
  System.out.println(rs.getString(1)+"
  "+rs.getString(2));
}
```

- Menutup koneksi, statement dan resultset (untuk query select).

```
conn.close();
stmt.close();
rs.close();
```



Tugas

- Buat aplikasi Java GUI dengan database, kasus tentukan sendiri!
- Kelompok 6 mhs
- Presentasikan pertemuan ke 13 14

Pertanyaan???