GUI in Java

Presented by HCI Team

Ali Ridho Barakbah

Umi Sa'adah

Nur Rosyid Mubtada'i

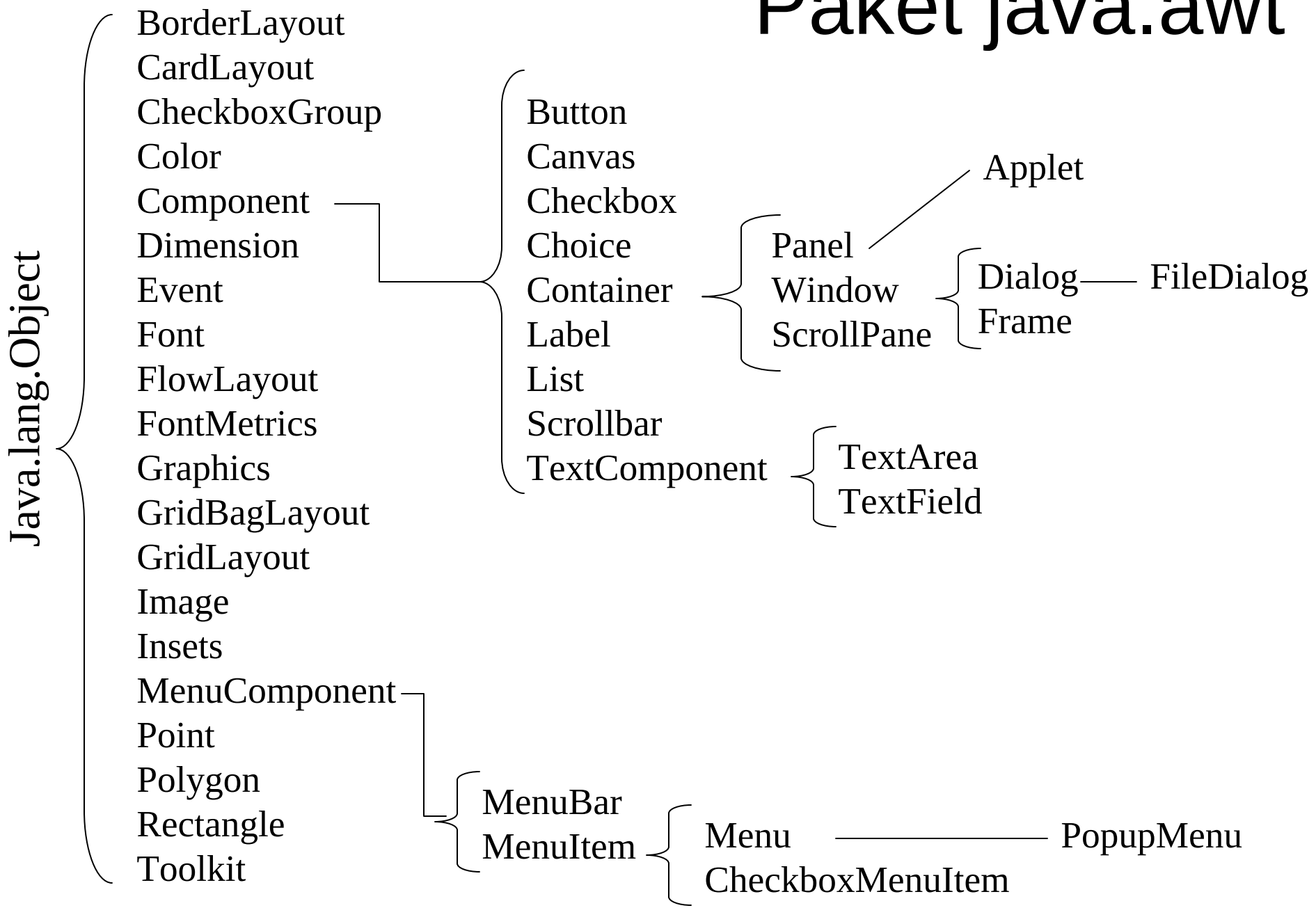
Supervised by

Prof. Kohei Arai

Abstract Window Toolkit

- Menyediakan komponen-komponen GUI yang digunakan di semua aplikasi Java & Java applet
- Berisi class-class yang dapat diturunkan dan propertis-propertisnya dapat diwariskan
- Memastikan bahwa setiap komponen GUI yang dapat ditampilkan di layar adalah merupakan suatu subclass Component atau MenuComponent
- Mempunyai Container, yaitu suatu abstract subclass dari Component dan termasuk di dalamnya adalah 2 subclass
  - Panel
  - Window

Paket java.awt



Containers

- Menambah komponen dengan metode add()
- Mempunyai 2 tipe kontainer
  - Window
  - Panel
- Suatu Window adalah objek window yang dapat melayang
- Suatu Panel adalah dapat diisikan sejumlah komponen GUI.

Memposisikan Komponen

- Posisi dan ukuran dari suatu komponen pada suatu Container adalah ditentukan oleh suatu layout manager
- Posisi dan ukuran dari komponen dapat diset dengan men-disable layout manager
- setLocation() dan setSize() dapat dipakai untuk memposisikan dan mengeset ukuran dari suatu komponen pada suatu container

Frames

- Adalah suatu subclass dari Window
- Mempunyai title dan tepi yang dapat diubah ukurannya
- Default-nya di-set invisible dan dapat di-set visible dengan setVisible(true)
- Mempunyai border layout sebagai layout manager default
- Menggunakan metode setLayout untuk merubah layout manager default

Contoh Frame

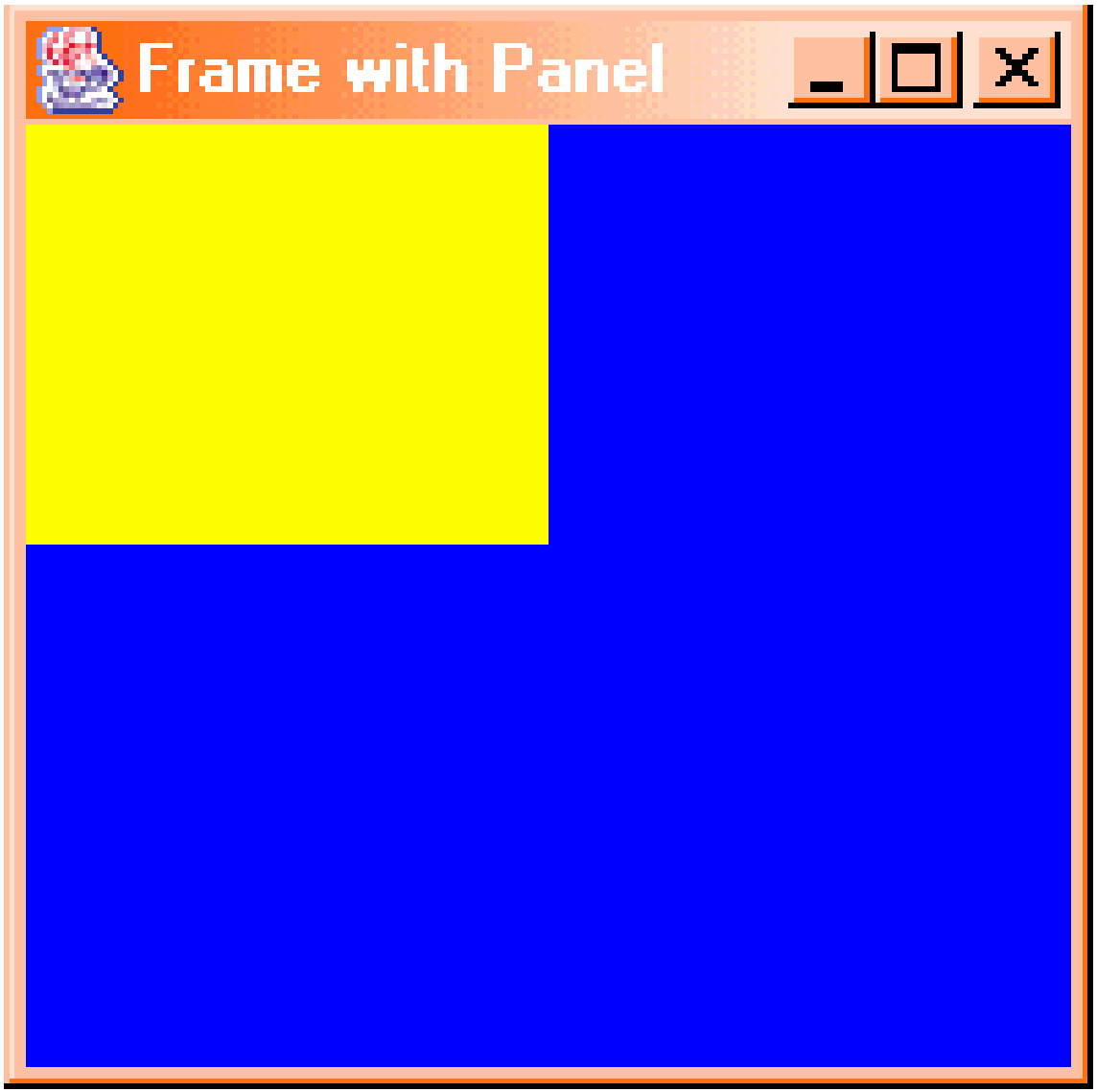
```
import java.awt.*;

public class FrameExample {
        private Frame f;

        public FrameExample() {
                f=new Frame("Coba frame");
                f.setSize(170,170);
                f.setBackground(Color.blue);
                f.setVisible(true);
        }

        public static void main(String args[]) {
                FrameExample guiWindow=
                        new FrameExample();
        }
}
```


Coba frame

Contoh Frame dengan Panel

```
import java.awt.*;

public class FrameWithPanel {
        private Frame f;
        private Panel pan;

        public FrameWithPanel(String title) {
                f=new Frame(title);
                pan=new Panel();

                f.setSize(200,200);
                f.setBackground(Color.blue);
                f.setLayout(null);
```

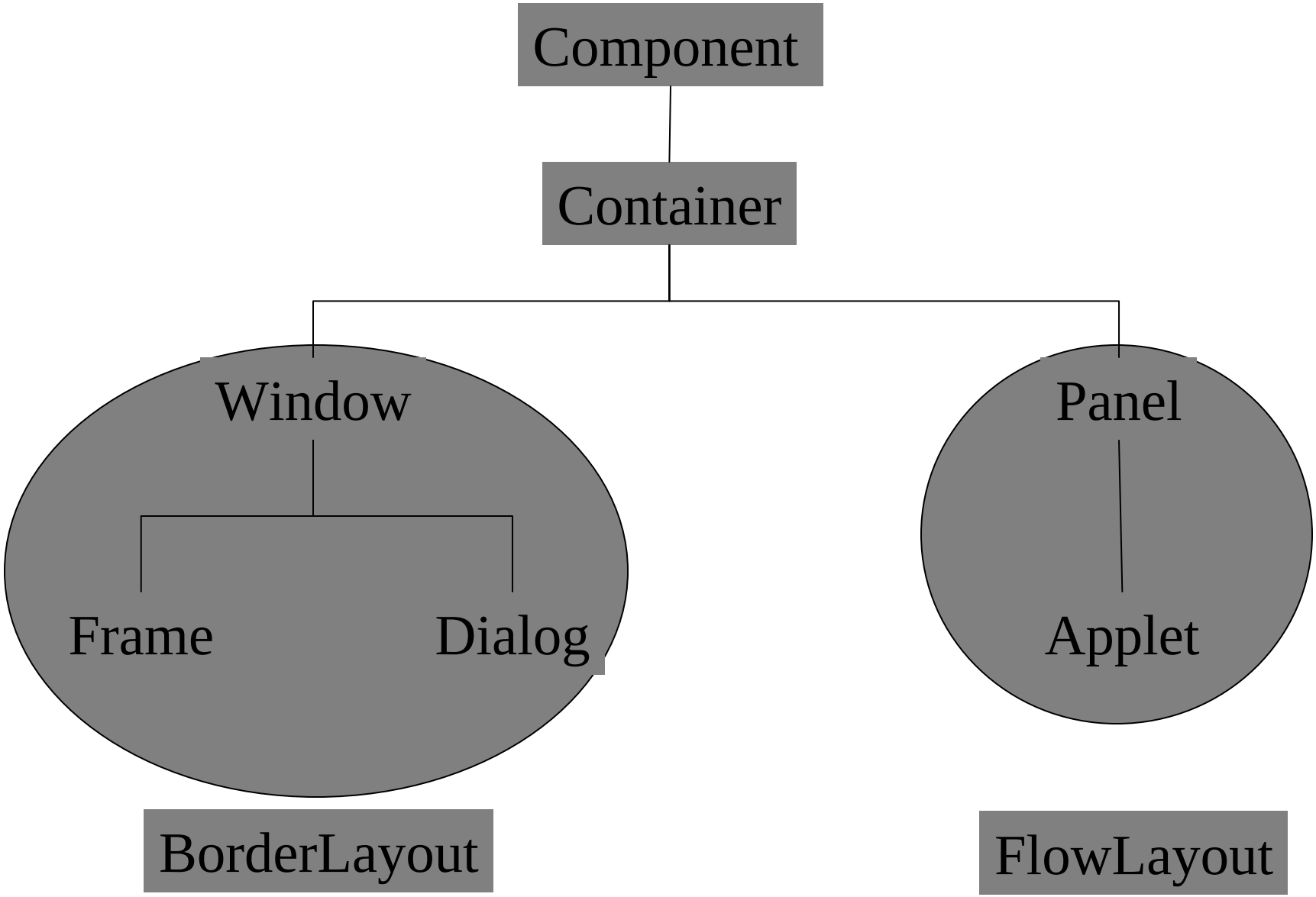
```
                pan.setSize(100,100);
                pan.setBackground(Color.yellow);
                f.add(pan);
                f.setVisible(true);
        }

        public static void main(String args[]) {
                FrameWithPanel guiWindow=
                        new FrameWithPanel("Frame with Panel");
        }
}
```


Frame with Panel

Container Layouts

- FlowLayout
- BorderLayout
- GridLayout
- CardLayout
- GridBagLayout

Default Layout Managers



Contoh FlowLayout sederhana

```
import java.awt.*;

public class LayoutExample {
        private Frame f;
        private Button b1;
        private Button b2;

        public LayoutExample() {
                f=new Frame("Layout example");
                b1=new Button("Press me");
                b2=new Button("Don't Press me");
        }
```

```
        public void launchFrame() {
                f.setLayout(new FlowLayout());
                f.add(b1);
                f.add(b2);
                f.pack();
                f.setVisible(true);
        }

        public static void main(String args[]) {
                LayoutExample guiWindow=
                        new LayoutExample();
                guiWindow.launchFrame();
        }
}
```


Layout example
Press me
Don't Press me

FlowLayout Manager

- Default layout untuk class Panel
- Komponen-komponen ditambahkan dari kiri ke kanan
- Default alignment adalah rata tengah
- Menggunakan ukuran komponen yang sesuai
- Menggunakan constructor untuk mengeset behavior

Format FlowLayout

setLayout(new FlowLayout(int align, int hgap, int vgap));

Dimana :

align : rata baris
FlowLayout.LEFT → rata kiri
FlowLayout.CENTER → rata tengah
FlowLayout.RIGHT → rata kanan

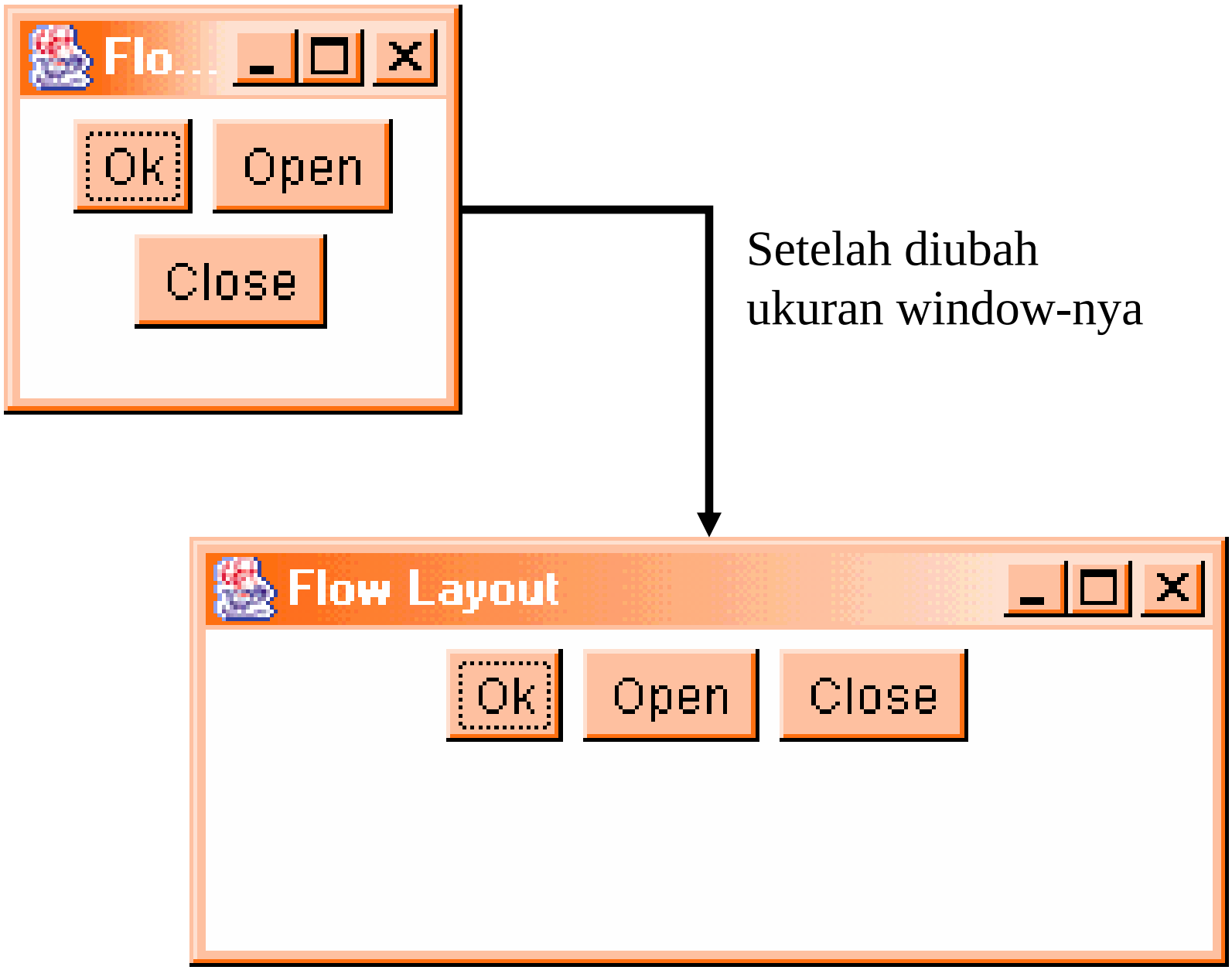
hgap : horizontal gap

vgap : vertical gap

Contoh FlowLayout

```
import java.awt.*;

public class FlowExample {
        private Frame f;
        private Button b1;
        private Button b2;
        private Button b3;

        public FlowExample() {
                f=new Frame("Flow Layout");
                b1=new Button("Ok");
                b2=new Button("Open");
                b3=new Button("Close");
        }
```

```
        public void launchFrame() {
                f.setLayout(new FlowLayout());
                f.add(b1);
                f.add(b2);
                f.add(b3);
                f.setSize(100,100);
                f.setVisible(true);
        }

        public static void main(String args[]) {
                FlowExample guiWindow=
                        new FlowExample();
                guiWindow.launchFrame();
        }
}
```

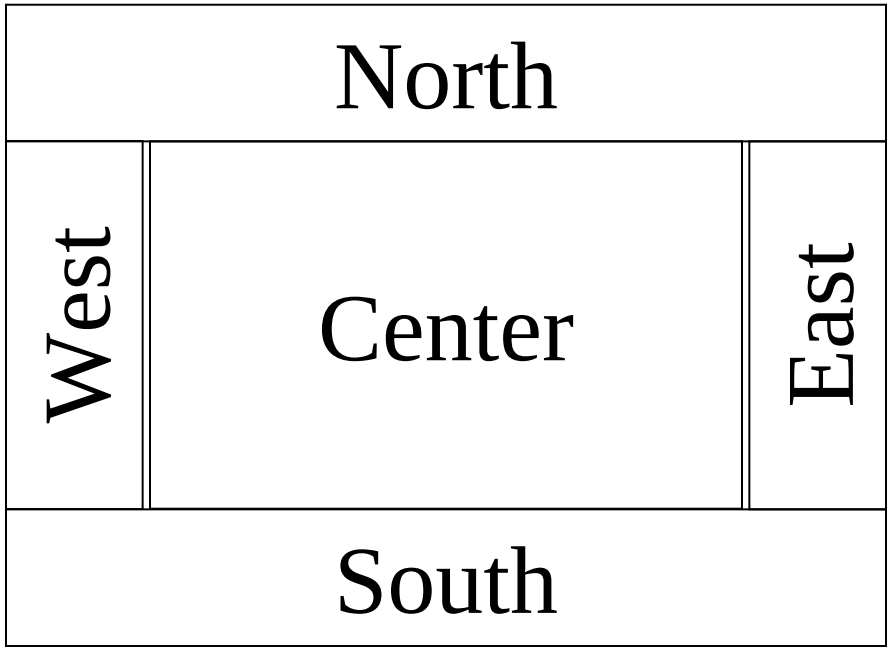

Flo...
Ok
Open
Close
Setelah diubah
ukuran window-nya
Flow Layout
Ok
Open
Close

BorderLayout Manager

- Default layout untuk class Frame
- Komponen ditambahkan pada daerah tertentu
- Terbagi menjadi 5 daerah :

  North, South, Center, East, West
- Perubahan ukuran window :
  - North, South, Center → secara horizontal
  - East, West, Center → secara vertikal

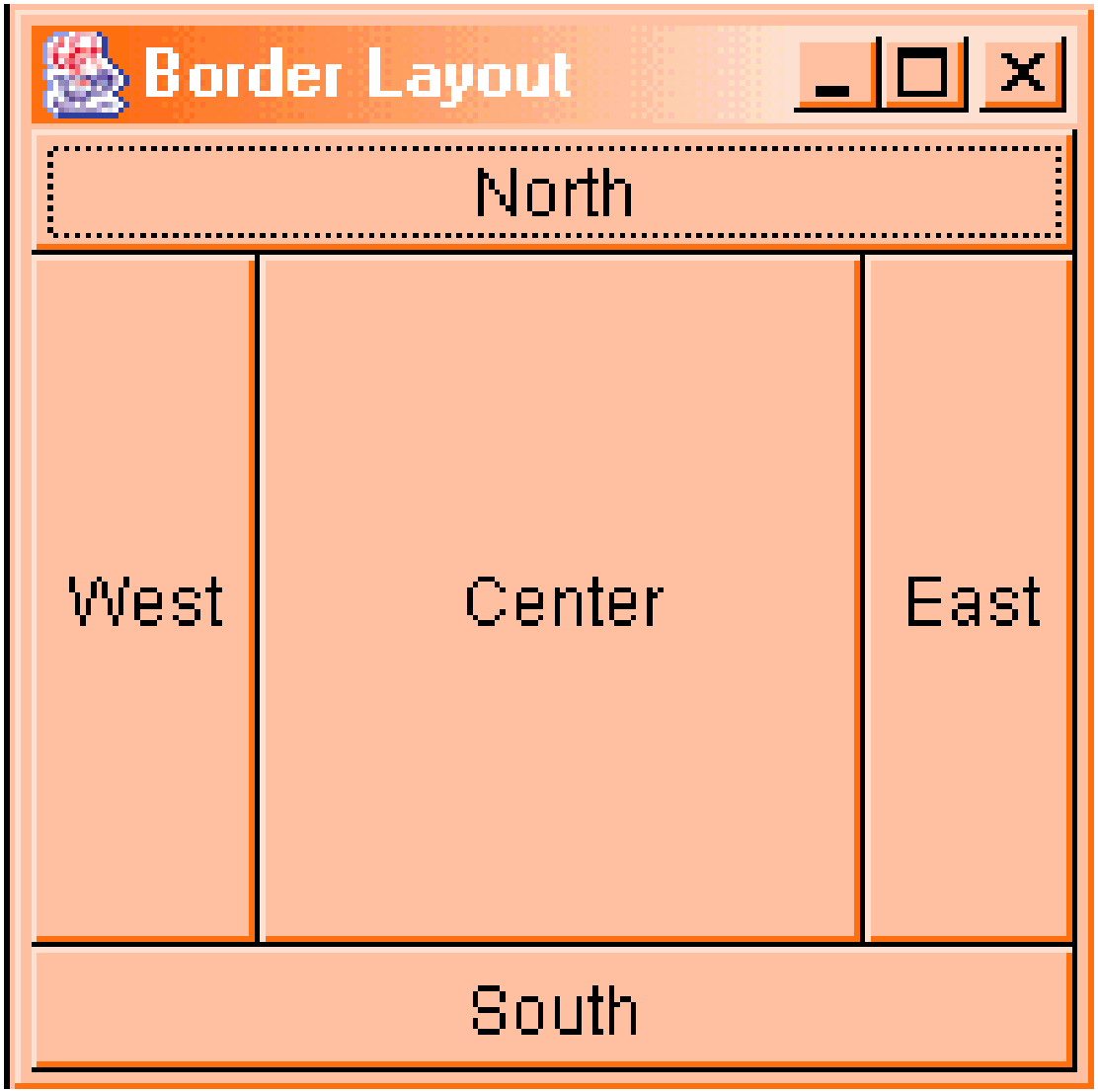
Daerah BorderLayout



Contoh BorderLayout

```
import java.awt.*;

public class BorderExample {
        private Frame f;
        private Button bNorth;
        private Button bSouth;
        private Button bWest;
        private Button bEast;
        private Button bCenter;

        public BorderExample() {
                f=new Frame ("Border Layout");
                bNorth=new Button("North");
                bSouth=new Button("South");
                bWest=new Button("West");
                bEast=new Button("East");
                bCenter=new Button("Center");
        }
```

```
        public void launchFrame() {
                f.add(bNorth, BorderLayout.NORTH);
                f.add(bSouth, BorderLayout.SOUTH);
                f.add(bWest, BorderLayout.WEST);
                f.add(bEast, BorderLayout.EAST);
                f.add(bCenter, BorderLayout.CENTER);
                f.setSize(200,200);
                f.setVisible(true);
        }

        public static void main(String args[]) {
                BorderExample guiWindow=
                        new BorderExample();
                guiWindow.launchFrame();
        }
}
```


Border Layout
North
West
Center
East
South

GridLayout Manager

- Komponen ditambahkan dari kiri ke kanan, atas ke bawah
- Daerah-daerah bagian di-set sama besar
- Constructor menspesifikasikan sejumlah baris dan kolom

Format GridLayout

setLayout(new GridLayout(int rows, int cols, int hgap, int vgap));

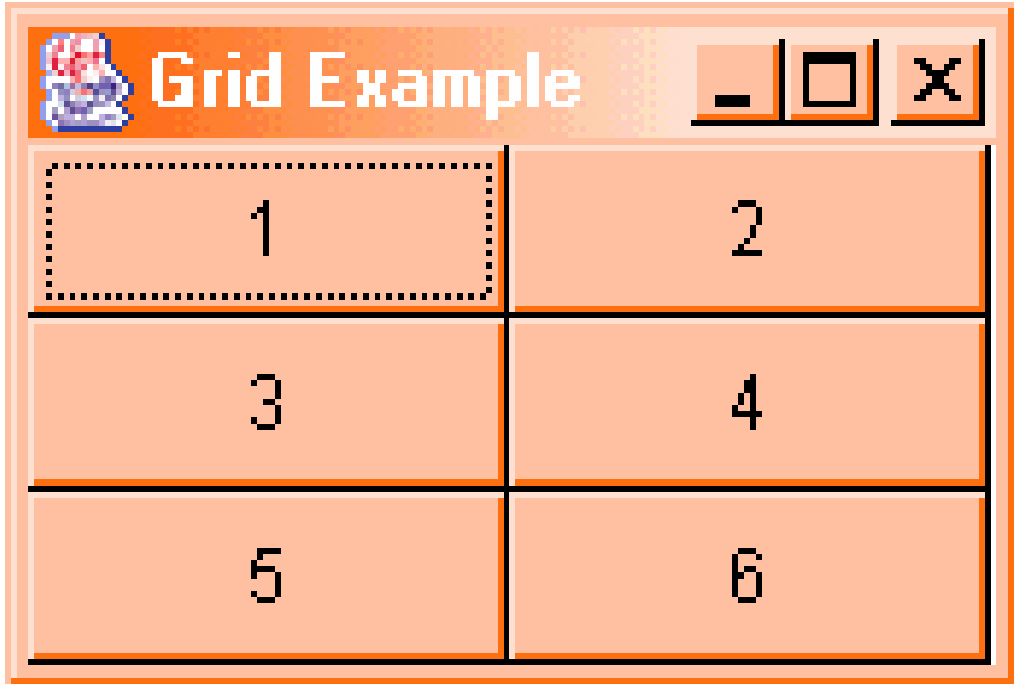
Dimana :

rows → jumlah baris
cols → jumlah kolom
hgap → horizontal gap
vgap → vertical gap

Contoh GridLayout

```
import java.awt.*;

public class GridExample {
        private Frame f;
        private Button b1, b2, b3, b4, b5, b6;

        public GridExample() {
                f=new Frame("Grid Example");
                b1=new Button("1");
                b2=new Button("2");
                b3=new Button("3");
                b4=new Button("4");
                b5=new Button("5");
                b6=new Button("6");
        }
```

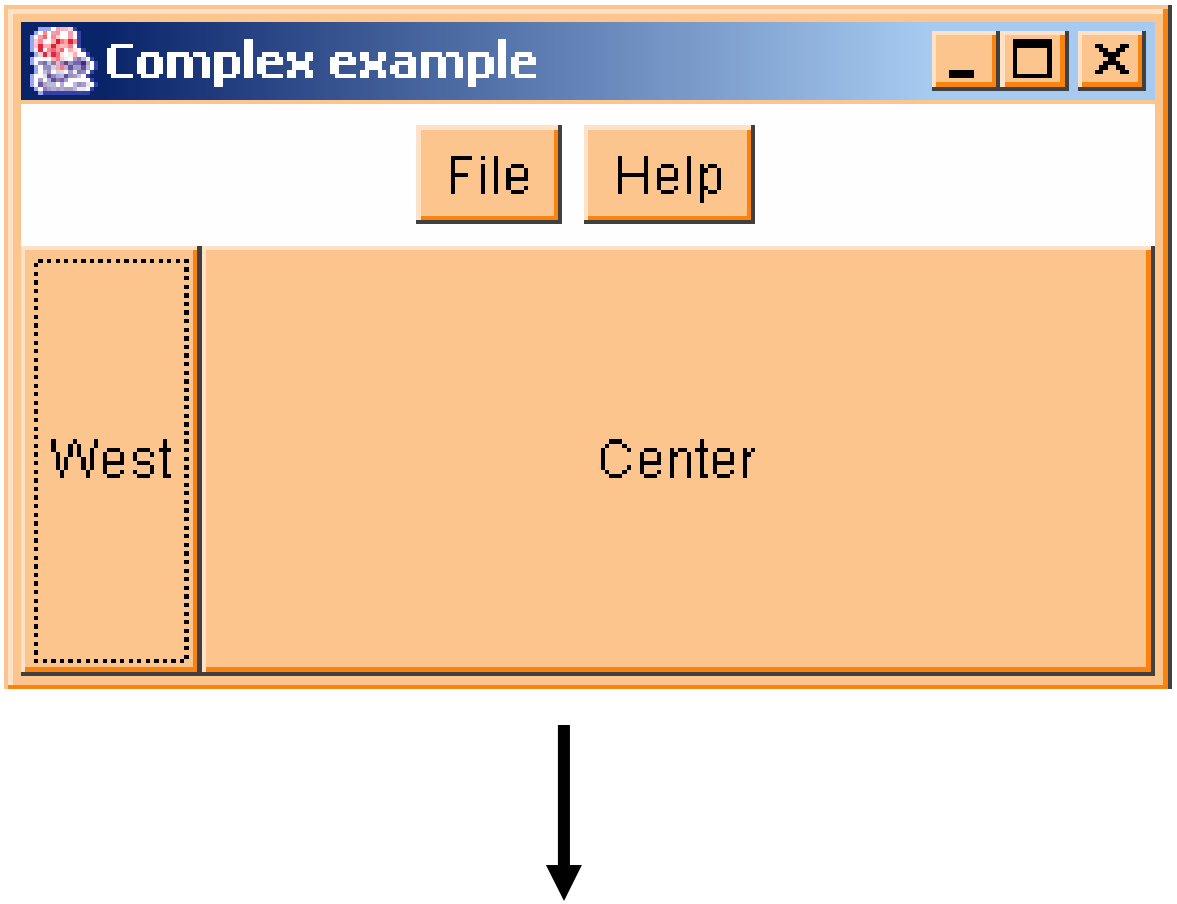
```
        public void launchFrame() {
                f.setLayout(new GridLayout(3,2));
                f.add(b1);
                f.add(b2);
                f.add(b3);
                f.add(b4);
                f.add(b5);
                f.add(b6);
                f.pack();
                f.setVisible(true);
        }

        public static void main(String args[]) {
                GridExample guiWindow=new GridExample();
                guiWindow.launchFrame();
        }
}
```


Grid Example
1
2
3
4
5
6

Contoh Complex Layout

```
import java.awt.*;

public class ComplexExample {
        private Frame f;
        private Panel p;
        private Button bWest, bCenter;
        private Button bFile, bHelp;

        public ComplexExample() {
                f=new Frame("Complex example");
                bWest=new Button("West");
                bCenter=new Button("Center");
                bFile=new Button("File");
                bHelp=new Button("Help");
        }
```

```
        public void launchFrame() {
                f.add(bWest, BorderLayout.WEST);
                f.add(bCenter, BorderLayout.CENTER);
                p=new Panel();
                p.add(bFile);
                p.add(bHelp);
                f.add(p,BorderLayout.NORTH);
                f.pack();
                f.setVisible(true);
        }

        public static void main(String args[]) {
                ComplexExample guiWindow=
                        new ComplexExample();
                guiWindow.launchFrame();
        }
}
```



Combining FlowLayout & BorderLayout