



# PEMROGRAMAN VISUAL

LAYOUT MANAGER

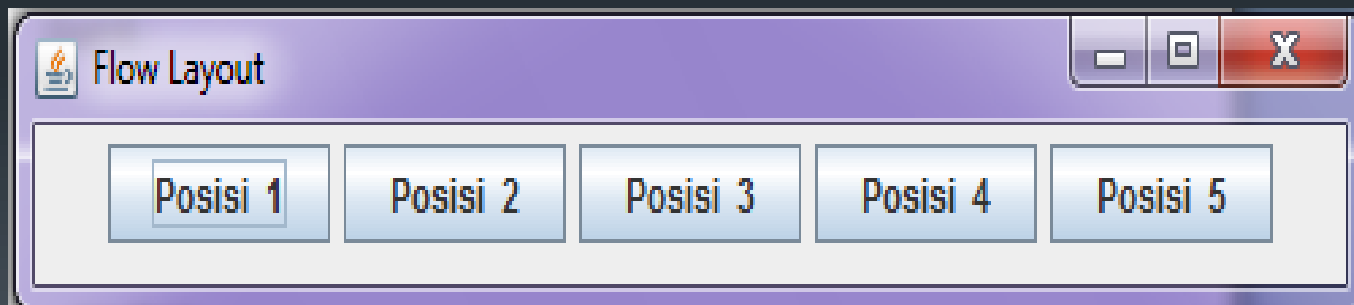


# LAYOUT MANAGER

- Pengaturan layout digunakan untuk mengatur posisi dari komponen visual penyusun program sesuai dengan desain user interface. Beberapa pilihan layout telah disediakan java, dimana keputusan untuk menggunakan jenis layout tertentu bergantung pada jenis aplikasi yang ingin dibuat serta tingkat kerapian yang diinginkan. Pada bab-bab terdahulu, manajemen layout yang digunakan adalah none layout, dimana pengaturan posisi komponen dalam frame dilakukan sendiri oleh programmer.

# FlowLayout

- FlowLayout adalah jenis pengaturan layout yang paling sederhana, dimana semua komponen akan tersusun dari kiri ke kanan sepanjang frame, dan akan pindah ke bawah bila telah sampai batas kanan frame.



# FlowLayout

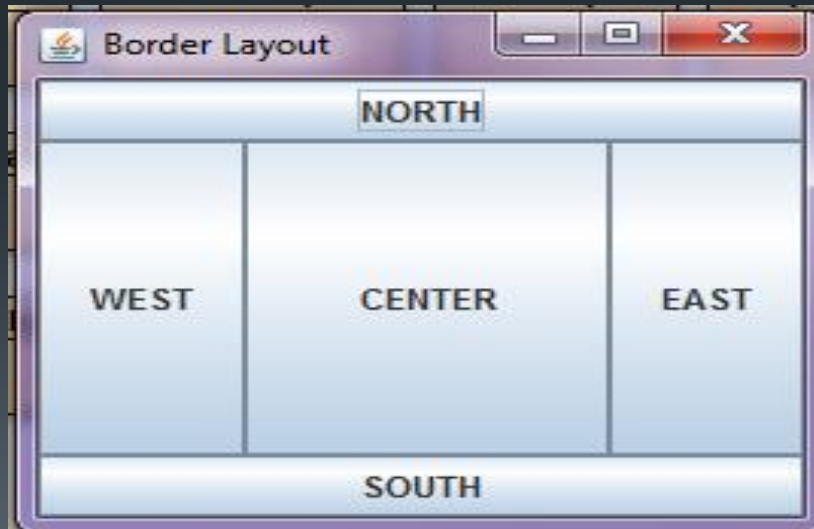
```
import javax.swing.*;
import java.awt.*;
class flow{
    public static void main (String [] args){
        JFrame f=new JFrame("Flow Layout");
        FlowLayout fl=new FlowLayout(FlowLayout.CENTER);

        JButton b1=new JButton("Posisi 1");
        JButton b2=new JButton("Posisi 2");
        JButton b3=new JButton("Posisi 3");
        JButton b4=new JButton("Posisi 4");
        JButton b5=new JButton("Posisi 5");

        f.getContentPane().setLayout(fl);
        f.getContentPane().add(b1);
        f.getContentPane().add(b2);
        f.getContentPane().add(b3);
        f.getContentPane().add(b4);
        f.getContentPane().add(b5);
        f.setSize(500,80);
        f.setVisible(true);
    }
}
```

# BorderLayout

- BorderLayout merupakan jenis layout yang bekerja dengan membagi frame menjadi lima bagian yaitu NORTH, EAST, SOUTH, WEST dan CENTER.



# BorderLayout

```
import javax.swing.*;
import java.awt.*;

class border{
    public static void main (String [] args){
        JFrame f=new JFrame("Border Layout");
        BorderLayout bl=new BorderLayout();
        f.getContentPane().setLayout(bl);

        JButton b1=new JButton("NORTH");
        JButton b2=new JButton("SOUTH");
        JButton b3=new JButton("EAST");
        JButton b4=new JButton("WEST");
        JButton b5=new JButton("CENTER");

        f.getContentPane().add(b1,BorderLayout.NORTH);
        f.getContentPane().add(b2,BorderLayout.SOUTH);
        f.getContentPane().add(b3,BorderLayout.EAST);
        f.getContentPane().add(b4,BorderLayout.WEST);
        f.getContentPane().add(b5,BorderLayout.CENTER);
        f.pack();
        f.setVisible(true);
    }
}
```

# GridLayout

- **GridLayout** adalah jenis layout yang bekerja berdasar baris dan kolom. Dengan layout ini kita dapat memberikan argumen banyaknya baris dan kolom sesuai dengan kebutuhan.



# GridLayout

```
import javax.swing.*;
import java.awt.*;

class grid{
    public static void main(String [] args){
        JFrame f=new JFrame("Grid Layout");
        JButton b1=new JButton("Satu");
        JButton b2=new JButton("Dua");
        JButton b3=new JButton("Tiga");
        JButton b4=new JButton("Empat");
        JButton b5=new JButton("Lima");

        GridLayout gl=new GridLayout(3,2);
        f.getContentPane().setLayout(gl);
        f.getContentPane().add(b1);
        f.getContentPane().add(b2);
        f.getContentPane().add(b3);
        f.getContentPane().add(b4);
        f.getContentPane().add(b5);

        f.pack();
        f.setVisible(true);
    }
}
```