Layout 1

Praktikum Pemrograman II - 04

Wanda Gusdya

2023

LayoutManagers

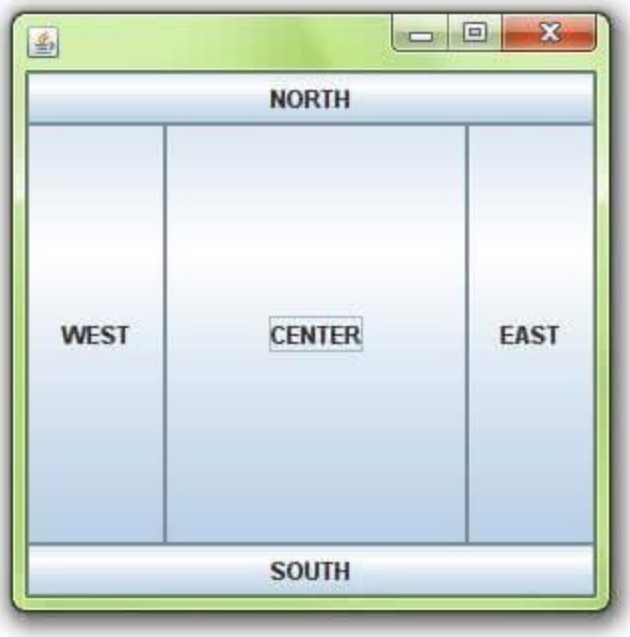
LayoutManager dapat digunakan untuk mengatur komponen dengan cara tertentu. LayoutManagers memfasilitasi kita untuk mengontrol posisi dan ukuran komponen GUI. LayoutManager adalah interface yang diimplementasikan oleh semua kelas pengelola tata letak. Berikut merupakan kelas-kelas yang merupakan kelas pengelola tata letak:

1. java.awt.BorderLayout
2. java.awt.FlowLayout
3. java.awt.GridLayout
4. java.awt.CardLayout
5. java.awt.GridBagLayout
6. javax.swing.BoxLayout
7. javax.swing.BoxLayout
8. javax.swing.BoxLayout
9. javax.swing.BoxLayout
10. dll

BorderLayout

BorderLayout digunakan untuk mengatur komponen di lima wilayah: utara, selatan, timur, barat, dan tengah. Setiap region (area) dapat berisi satu komponen saja. Layout ini adalah layout default dari frame atau window. BorderLayout menyediakan lima konstanta untuk setiap wilayah:

1. public static final int NORTH
2. public static final int SOUTH
3. public static final int EAST
4. public static final int WEST
5. public static final int CENTER



Gambar 1 Ilustrasi Layout

Latihan 1

Buatlah sebuah kelas dengan nama HelloBorderLayout dan kode berikut ini.

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

public class HelloBorderLayout extends JFrame {

    public HelloBorderLayout(){
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel labelPertanyaan = new JLabel("Apakah ibukota Indonesia?");
        JLabel labelHasil = new JLabel("Jawab pertanyaan di atas");

        JButton buttonA = new JButton("Jakarta");
        JButton buttonB = new JButton("Bandung");
        JButton buttonC = new JButton("Surabaya");

        buttonA.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                labelHasil.setText("Jawaban anda benar");
            }
        });

        buttonB.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                labelHasil.setText("Jawaban anda salah");
            }
        });

        buttonB.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                labelHasil.setText("Jawaban anda salah");
            }
        });

        this.add(labelPertanyaan, BorderLayout.NORTH);
        this.add(labelHasil, BorderLayout.SOUTH);
        this.add(buttonA, BorderLayout.WEST);
        this.add(buttonB, BorderLayout.CENTER);
        this.add(buttonC, BorderLayout.EAST);
```

Gambar 2 Kode HelloBorderLayout Bag 1

```
        this.setSize(400, 200);
    }

    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                HelloBorderLayout h = new HelloBorderLayout();
                h.setVisible(true);
            }
        });

    }
}
```

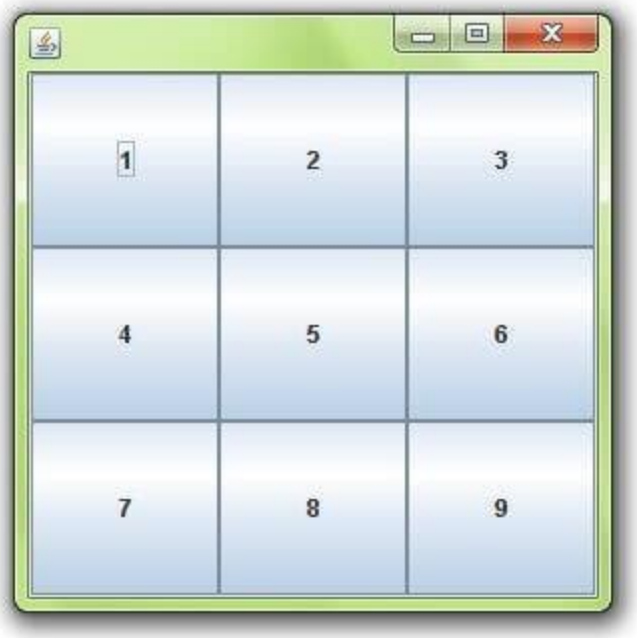
Gambar 3 Kode HelloBorderLayout Bag 2

Kompilasi kemudian jalankan kelas tersebut, sehingga akan menghasilkan aplikasi seperti berikut.

Gambar 4 Hasil Eksekusi HelloBorderLayout

GridLayout

Kelas Java GridLayout digunakan untuk mengatur komponen dalam sebuah grid yang terdiri dari baris dan kolom. Satu komponen ditampilkan di setiap sel/kotak dari grid tersebut.



Gambar 5 Ilustrasi Grid

Latihan 2

Buatlah sebuah kelas dengan nama HelloGridLayout kemudian isi dengan kode berikut.

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

public class HelloGridLayout extends JFrame implements ActionListener {
    private JButton buttonA;
    private JButton buttonB;
    private JButton buttonC;
    private JButton buttonD;
    private JButton buttonE;
    private JButton buttonF;
    private JButton buttonG;
    private JButton buttonH;
    private JButton buttonI;

    private JButton[] buttons;
    private boolean gameOver;

    public HelloGridLayout(){
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        gameOver = false;

        buttonA = new JButton("");
        buttonB = new JButton("");
        buttonC = new JButton("");
        buttonD = new JButton("");
        buttonE = new JButton("");
        buttonF = new JButton("");
        buttonG = new JButton("");
        buttonH = new JButton("");
        buttonI = new JButton("");

```

Gambar 6 Kode HelloGridLayout Bag 1

```
34          buttons = new JButton[9];
35          buttons[0] = buttonA;
36          buttons[1] = buttonB;
37          buttons[2] = buttonC;
38          buttons[3] = buttonD;
39          buttons[4] = buttonE;
40          buttons[5] = buttonF;
41          buttons[6] = buttonG;
42          buttons[7] = buttonH;
43          buttons[8] = buttonI;
44
45          buttonA.addActionListener(this);
46          buttonB.addActionListener(this);
47          buttonC.addActionListener(this);
48          buttonD.addActionListener(this);
49          buttonE.addActionListener(this);
50          buttonF.addActionListener(this);
51          buttonG.addActionListener(this);
52          buttonH.addActionListener(this);
53          buttonI.addActionListener(this);
54
55          this.setLayout(new GridLayout(3, 3));
56
57          this.add(buttonA);
58          this.add(buttonB);
59          this.add(buttonC);
60          this.add(buttonD);
61          this.add(buttonE);
62          this.add(buttonF);
63          this.add(buttonG);
64          this.add(buttonH);
65          this.add(buttonI);
66
67          this.setSize(300, 300);
68      }
69
```

Gambar 7 Kode HelloGridLayout Bag 2

```
69
70      public void actionPerformed(ActionEvent e){
71          if (!gameOver)
72          {
73              JButton button = (JButton) e.getSource();
74              if (button.getText().isEmpty())
75              {
76                  button.setText("O");
77                  checkWinner();
78                  if (!gameOver)
79                  {
80                      for (int i = 0; i < buttons.length; i++)
81                      {
82                          if (buttons[i].getText().isEmpty())
83                          {
84                              buttons[i].setText("X");
85                              break;
86                          }
87                      }
88                      checkWinner();
89                  }
90
91              }
92          }
93      }
94
```

Gambar 8 Kode HelloGridLayout Bag 3

```
94
95      private void checkWinner()
96      {
97          String winner = "";
98          if (!buttonA.getText().isEmpty()
99              && buttonA.getText().equals(buttonB.getText())
100             && buttonA.getText().equals(buttonC.getText()))
101         {
102             winner = buttonA.getText();
103             buttonA.setForeground(Color.RED);
104             buttonB.setForeground(Color.RED);
105             buttonC.setForeground(Color.RED);
106         } else if (!buttonD.getText().isEmpty()
107             && buttonD.getText().equals(buttonE.getText())
108             && buttonD.getText().equals(buttonF.getText()))
109         {
110             winner = buttonD.getText();
111             buttonD.setForeground(Color.RED);
112             buttonE.setForeground(Color.RED);
113             buttonF.setForeground(Color.RED);
114         } else if (!buttonG.getText().isEmpty()
115             && buttonG.getText().equals(buttonH.getText())
116             && buttonG.getText().equals(buttonI.getText()))
117         {
118             winner = buttonG.getText();
119             buttonG.setForeground(Color.RED);
120             buttonH.setForeground(Color.RED);
121             buttonI.setForeground(Color.RED);
122         } else if (!buttonA.getText().isEmpty()
123             && buttonA.getText().equals(buttonD.getText())
124             && buttonA.getText().equals(buttonG.getText()))
125         {
126             winner = buttonA.getText();
127             buttonA.setForeground(Color.RED);
128             buttonD.setForeground(Color.RED);
129             buttonG.setForeground(Color.RED);
130         } else if (!buttonB.getText().isEmpty()
131             && buttonB.getText().equals(buttonE.getText())
132             && buttonB.getText().equals(buttonH.getText()))
```

Gambar 9 Kode HelloGridLayout Bag 4

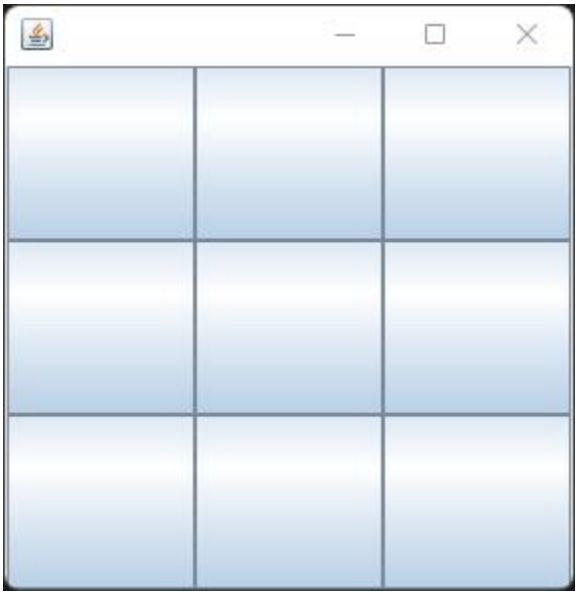
```
130         } else if (!buttonB.getText().isEmpty()
131             && buttonB.getText().equals(buttonE.getText())
132             && buttonB.getText().equals(buttonH.getText()))
133         {
134             winner = buttonB.getText();
135             buttonB.setForeground(Color.RED);
136             buttonE.setForeground(Color.RED);
137             buttonH.setForeground(Color.RED);
138         } else if (!buttonC.getText().isEmpty()
139             && buttonC.getText().equals(buttonF.getText())
140             && buttonC.getText().equals(buttonI.getText()))
141         {
142             winner = buttonC.getText();
143             buttonC.setForeground(Color.RED);
144             buttonF.setForeground(Color.RED);
145             buttonI.setForeground(Color.RED);
146         } else if (!buttonA.getText().isEmpty()
147             && buttonA.getText().equals(buttonE.getText())
148             && buttonA.getText().equals(buttonI.getText()))
149         {
150             winner = buttonA.getText();
151             buttonA.setForeground(Color.RED);
152             buttonE.setForeground(Color.RED);
153             buttonI.setForeground(Color.RED);
154         } else if (!buttonC.getText().isEmpty()
155             && buttonC.getText().equals(buttonE.getText())
156             && buttonC.getText().equals(buttonG.getText()))
157         {
158             winner = buttonC.getText();
159             buttonC.setForeground(Color.RED);
160             buttonE.setForeground(Color.RED);
161             buttonG.setForeground(Color.RED);
162         }
163 
164         gameOver = !winner.isEmpty();
165     }
166 
```

Gambar 10 Kode HelloGridLayout Bag 5

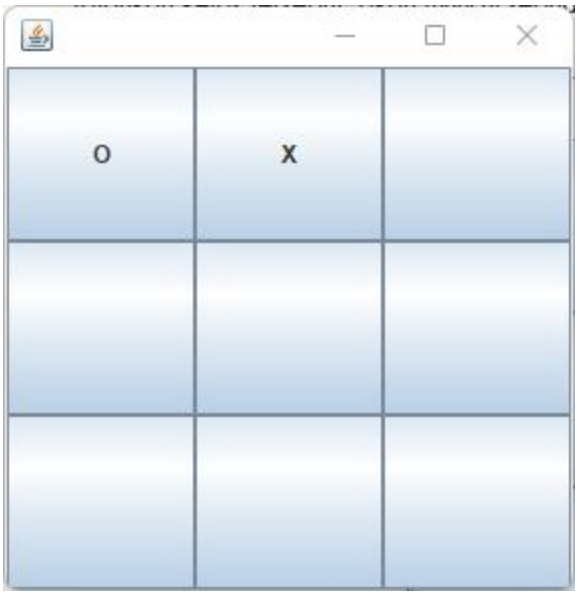
```
166
167     public static void main(String[] args) {
168         javax.swing.SwingUtilities.invokeLater(new Runnable() {
169             public void run() {
170                 HelloGridLayout h = new HelloGridLayout();
171                 h.setVisible(true);
172             }
173         });
174
175     }
176
177
178 }
```

Gambar 11 Kode HelloGridLayout Bag 6

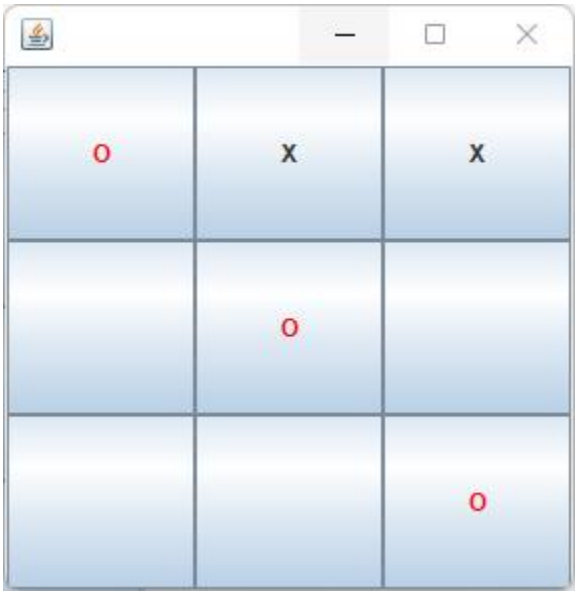
Jalankan kelas tersebut, akan muncul sebuah aplikasi untuk bermain tic-tac-toe sederhana seperti pada gambar berikut.



Gambar 12 Hasil Eksekusi HelloGridLayout Bag 1



Gambar 13 Hasil Eksekusi HelloGridLayout Bag 2



Gambar 14 Hasil Eksekusi HelloGridLayout Bag 3