



# Layout

Praktikum Pemrograman II - 03



## 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.

```

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");
            }
        });

        buttonC.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.

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class HelloGridLayout extends JFrame implements ActionListener {
6      private JButton buttonA;
7      private JButton buttonB;
8      private JButton buttonC;
9      private JButton buttonD;
10     private JButton buttonE;
11     private JButton buttonF;
12     private JButton buttonG;
13     private JButton buttonH;
14     private JButton buttonI;
15
16     private JButton[] buttons;
17     private boolean gameOver;
18
19     public HelloGridLayout(){
20         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21
22         gameOver = false;
23
24         buttonA = new JButton("");
25         buttonB = new JButton("");
26         buttonC = new JButton("");
27         buttonD = new JButton("");
28         buttonE = new JButton("");
29         buttonF = new JButton("");
30         buttonG = new JButton("");
31         buttonH = new JButton("");
32         buttonI = new JButton("");
33

```

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

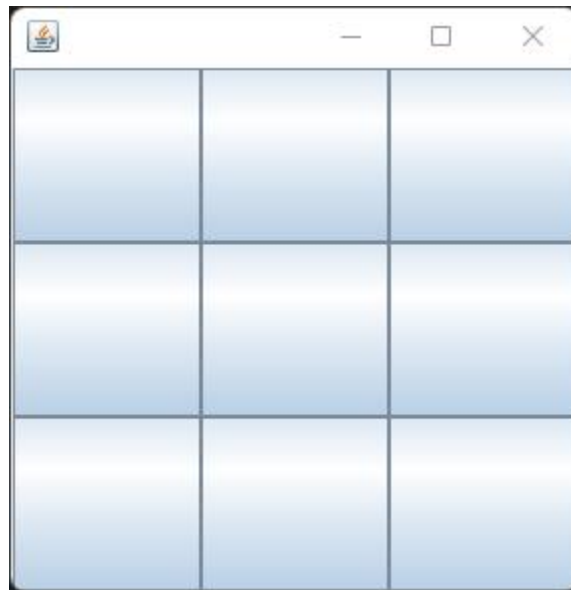
```

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  }

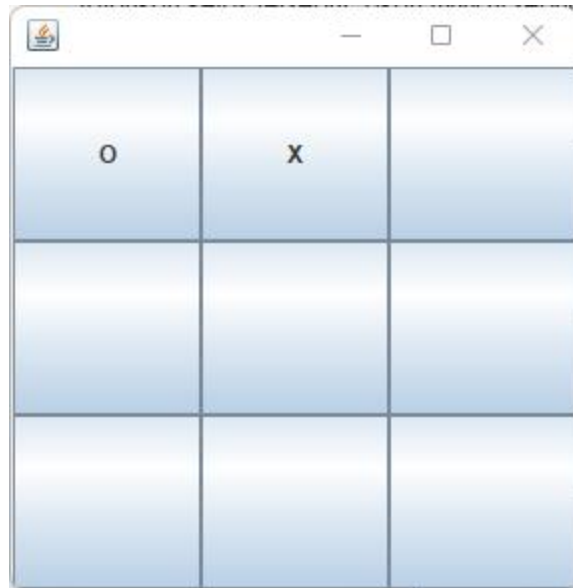
```

Gambar 11 Kode HelloGridLayout Bag 6

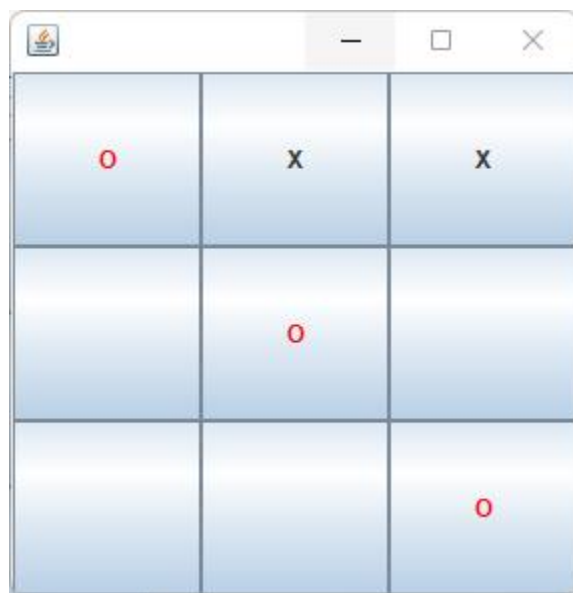
Jalankan kelas tersebut, akan muncul sebuah aplikasi untuk bermain tic-tac-toe sederhana seperti pada gambar berikut.



Gambar 12 Hasil Eksekusi HelloGridLayout Bag 1



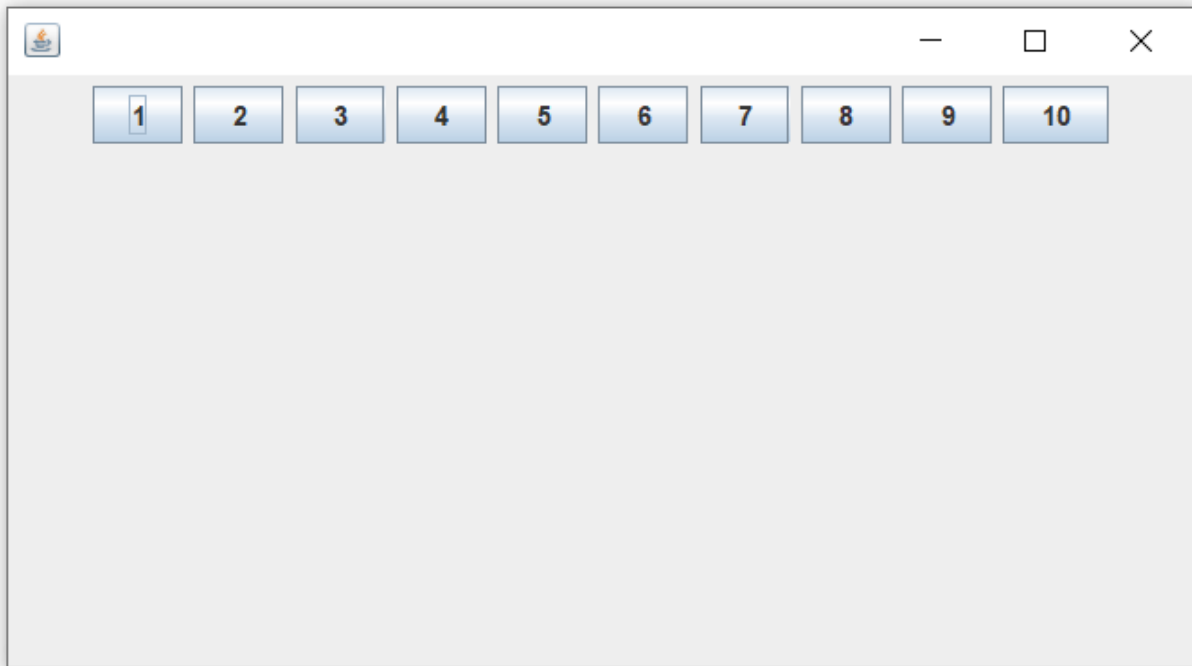
Gambar 13 Hasil Eksekusi HelloGridLayout Bag 2



Gambar 14 Hasil Eksekusi HelloGridLayout Bag 3

## FlowLayout

Kelas Java FlowLayout digunakan untuk mengatur komponen dalam satu baris, satu demi satu (dalam aliran). Kelas ini adalah tata letak default dari applet atau panel.



*Gambar 15 Ilustrasi FlowLayout*

### Latihan 3

Buatlah kelas dengan nama HelloFlowLayout dengan kode sebagai berikut.

```

1  import java.awt.*;
2  import javax.swing.*;
3
4  public class HelloFlowLayout extends JFrame {
5
6      public HelloFlowLayout(){
7          this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
8
9          JButton button1 = new JButton("1");
10         JButton button2 = new JButton("2");
11         JButton button3 = new JButton("3");
12         JButton button4 = new JButton("4");
13         JButton button5 = new JButton("5");
14         JButton button6 = new JButton("6");
15         JButton button7 = new JButton("7");
16         JButton button8 = new JButton("8");
17         JButton button9 = new JButton("9");
18         JButton button10 = new JButton("10");
19         this.setLayout(new FlowLayout());
20
21         this.add(button1);
22         this.add(button2);
23         this.add(button3);
24         this.add(button4);
25         this.add(button5);
26         this.add(button6);
27         this.add(button7);
28         this.add(button8);
29         this.add(button9);
30         this.add(button10);
31
32         this.setSize(400, 400);
33     }
34

```

Gambar 16 Kode HelloFlowLayout Bagian 1

```

34
35     public static void main(String[] args) {
36         javax.swing.SwingUtilities.invokeLater(new Runnable() {
37             public void run() {
38                 HelloFlowLayout h = new HelloFlowLayout();
39                 h.setVisible(true);
40             }
41         });
42     }
43 }
44 }

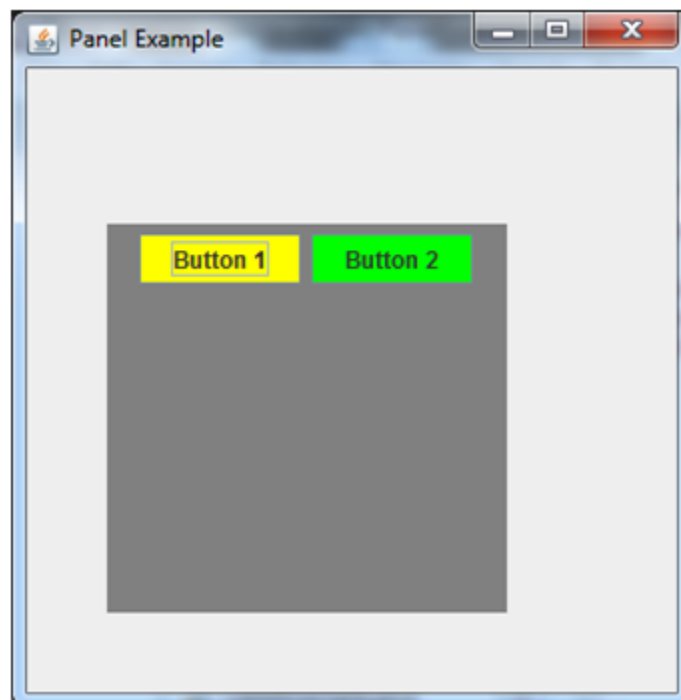
```

*Gambar 17 Kode HelloFlowLayout Bagian 2*

Kompilasi kemudian jalankan kelas tersebut, kemudian amati hasilnya.

## JPanel

JPanel adalah kelas kontainer paling sederhana. Kelas ini menyediakan ruang di mana aplikasi dapat melampirkan komponen lain. Kelas ini merupakan turunan dari kelas JComponents.



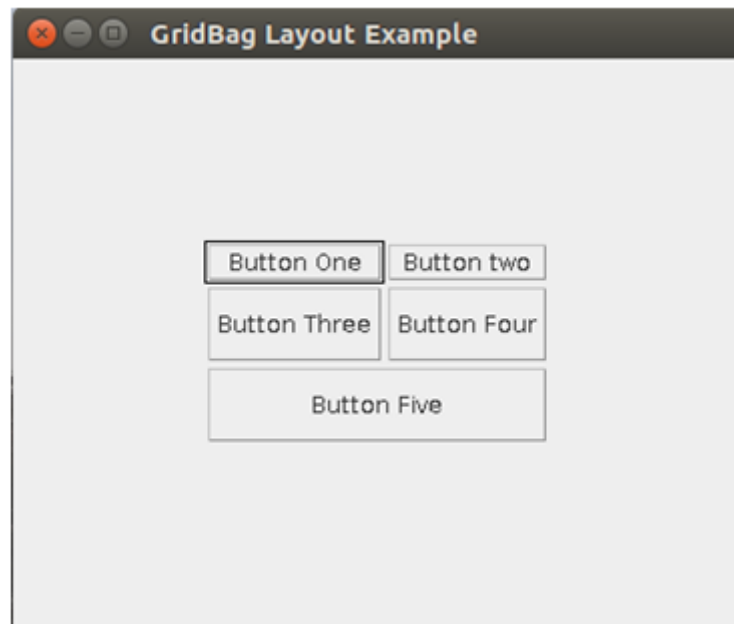
*Gambar 18 Ilustrasi Panel*



## GridBagLayout

Kelas Java GridBagLayout digunakan untuk menyelaraskan komponen secara vertikal, horizontal, atau sepanjang garis dasarnya.

Komponen GUI mungkin tidak berukuran sama. Setiap objek GridBagLayout mempertahankan grid sel kotak yang dinamis. Setiap komponen menempati satu atau lebih sel yang dikenal sebagai area tampilannya. Setiap komponen terkait dengan instance GridBagConstraints. Dengan bantuan objek constraint, anda dapat mengatur area tampilan komponen di grid. GridBagLayout mengelola ukuran minimum dan pilihan setiap komponen untuk menentukan ukuran komponen. Komponen GridBagLayout juga diatur dalam kotak persegi panjang tetapi dapat memiliki banyak ukuran yang berbeda dan dapat menempati beberapa baris atau kolom.



Gambar 19 Ilustrasi GridBagLayout

## Latihan 4

Buatlah kelas dengan nama HelloGridBagLayout dengan kode sebagai berikut.

```

J HelloGridBagLayout.java
1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class HelloGridBagLayout extends JFrame {
6
7      public HelloGridBagLayout(){
8          this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9
10         JLabel headerLabel = new JLabel("Layout in action: GridBagLayout", JLabel.CENTER);
11
12         JPanel controlPanel = new JPanel();
13         controlPanel.setLayout(new FlowLayout());
14
15         JPanel panel = new JPanel();
16         panel.setBackground(Color.darkGray);
17         panel.setSize(300,300);
18         GridBagLayout layout = new GridBagLayout();
19         panel.setLayout(layout);
20         GridBagConstraints gbc = new GridBagConstraints();
21
22         gbc.fill = GridBagConstraints.HORIZONTAL;
23         gbc.gridx = 0;
24         gbc.gridy = 0;
25         panel.add(new JButton("Button 1"),gbc);
26
27         gbc.gridx = 1;
28         gbc.gridy = 0;
29         panel.add(new JButton("Button 2"),gbc);
30
31         gbc.fill = GridBagConstraints.HORIZONTAL;
32         gbc.ipady = 20;
33         gbc.gridx = 0;
34         gbc.gridy = 1;
35         panel.add(new JButton("Button 3"),gbc);
36
37         gbc.gridx = 1;
38         gbc.gridy = 1;
39         panel.add(new JButton("Button 4"),gbc);
40

```

Gambar 20 Kode HelloGridBagLayout Bagian 1

```

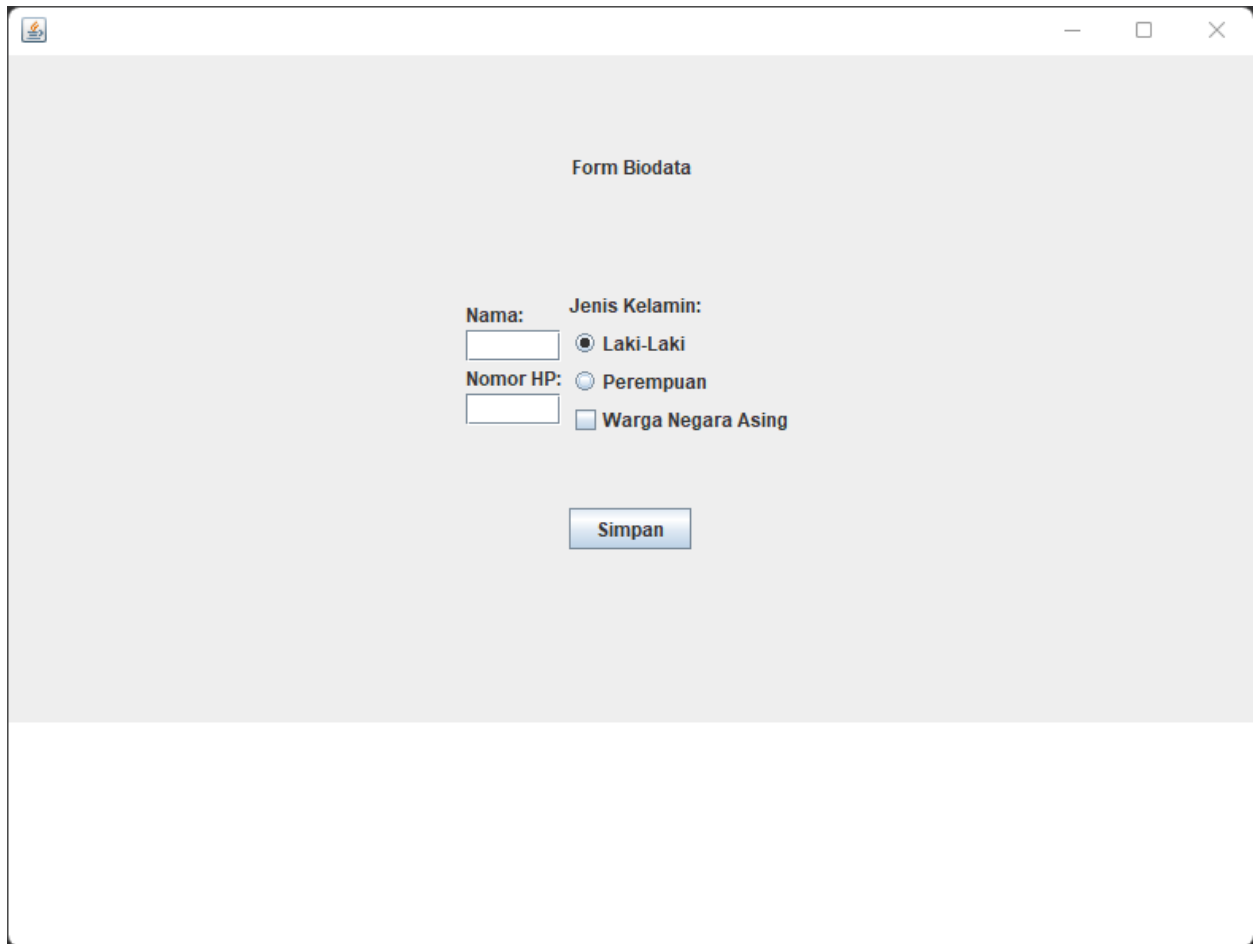
40
41     gbc.gridx = 0;
42     gbc.gridy = 2;
43     gbc.fill = GridBagConstraints.HORIZONTAL;
44     gbc.gridwidth = 2;
45     panel.add(new JButton("Button 5"),gbc);
46
47     controlPanel.add(panel);
48
49     this.setLayout(new GridLayout(2, 1));
50     this.add(headerLabel);
51     this.add(controlPanel);
52     this.setSize(400, 400);
53 }
54
55 public static void main(String[] args) {
56     javax.swing.SwingUtilities.invokeLater(new Runnable() {
57         public void run() {
58             HelloGridBagLayout h = new HelloGridBagLayout();
59             h.setVisible(true);
60         }
61     });
62
63 }
64 }

```

Gambar 21 Kode HelloGridBagLayout Bagian 2

## Latihan 5

Ubahlah aplikasi biodata yang telah anda buat sebelumnya untuk memanfaatkan panel dan beberapa layout lainnya yang telah dipelajari sehingga tampilannya menjadi seperti berikut.



Gambar 22 Hasil Eksekusi Form Biodata