



Komponen GUI Dasar di JFC

Praktikum Pemrograman II - 02



Button

Button dalam JFC direpresentasikan dengan kelas JButton. Kelas JButton digunakan untuk membuat tombol berlabel yang dapat implementasinya independen terhadap platform. Aplikasi dapat melakukan tindakan ketika tombol ditekan yang merupakan sebuah Event. Kelas JButton merupakan turunan kelas AbstractButton. Kelas JButton memiliki 3 (tiga) konstruktor yang umum digunakan seperti dijelaskan pada tabel 1 berikut.

Tabel 1 Konstruktor JButton

Konstruktor	Keterangan
JButton()	Membangun button tanpa teks dan ikon
JButton(String s)	Membangun button dengan teks label
JButton(Icon i)	Membangun button dengan ikon

Sedangkan method kelas JButton yang sering digunakan adalah sebagai berikut.

Tabel 2 Method JButton

Method	Keterangan
void setText(String s)	Untuk menentukan teks pada button
String getText()	Untuk mengembalikan teks yang tertera pada button
void setEnabled(boolean b)	Digunakan untuk mengaktifkan atau menonaktifkan button (jika tidak diaktifkan, button tidak dapat menghasilkan event klik)
void setIcon(Icon b)	Untuk menentukan ikon
Icon getIcon()	Untuk mengembalikan ikon yang ada pada button
void setMnemonic(int a)	Untuk menentukan mnemonic pengingat aksi button
void addActionListener(ActionListener a)	Untuk menambahkan listener untuk event aksi klik

Latihan 1

Buatlah sebuah kelas bernama ButtonEventSwing dengan kode seperti pada gambar 1 berikut.

```

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

public class ButtonEventSwing extends JFrame {

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

        JLabel label = new JLabel("Hello World");
        label.setBounds(130,40,400,10);

        JButton button = new JButton("Klik");
        button.setBounds(130,100,100, 40);

        button.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                label.setText("Hello Pemrograman II");
            }
        });

        this.add(button);
        this.add(label);

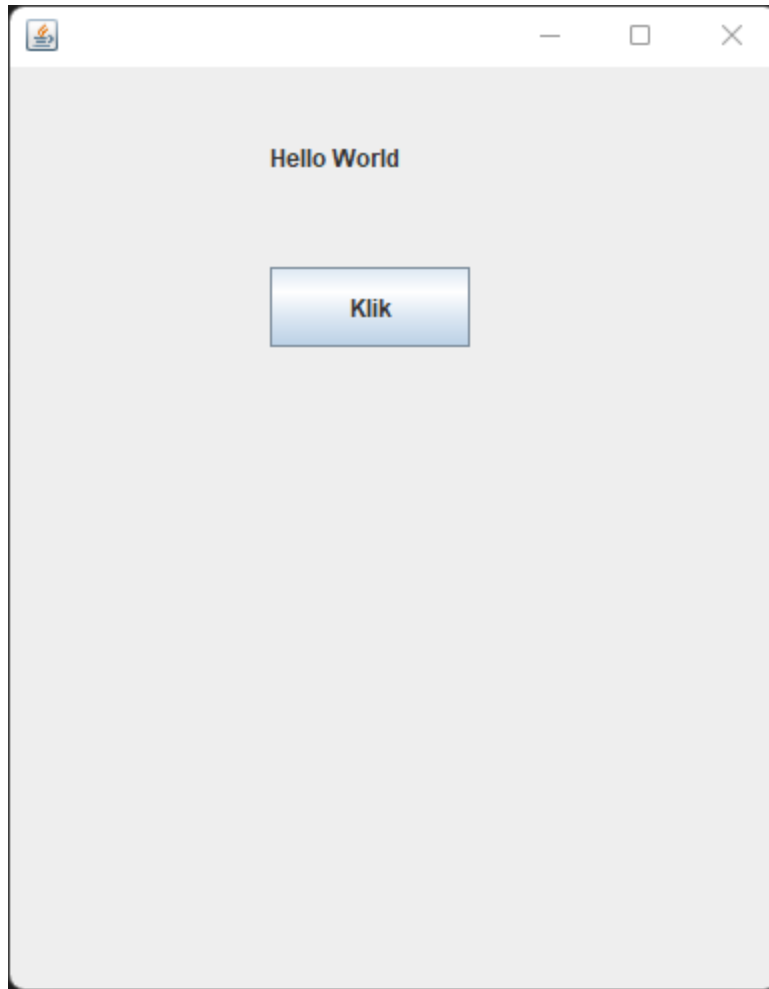
        this.setSize(400,500);
        this.setLayout(null);
    }

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

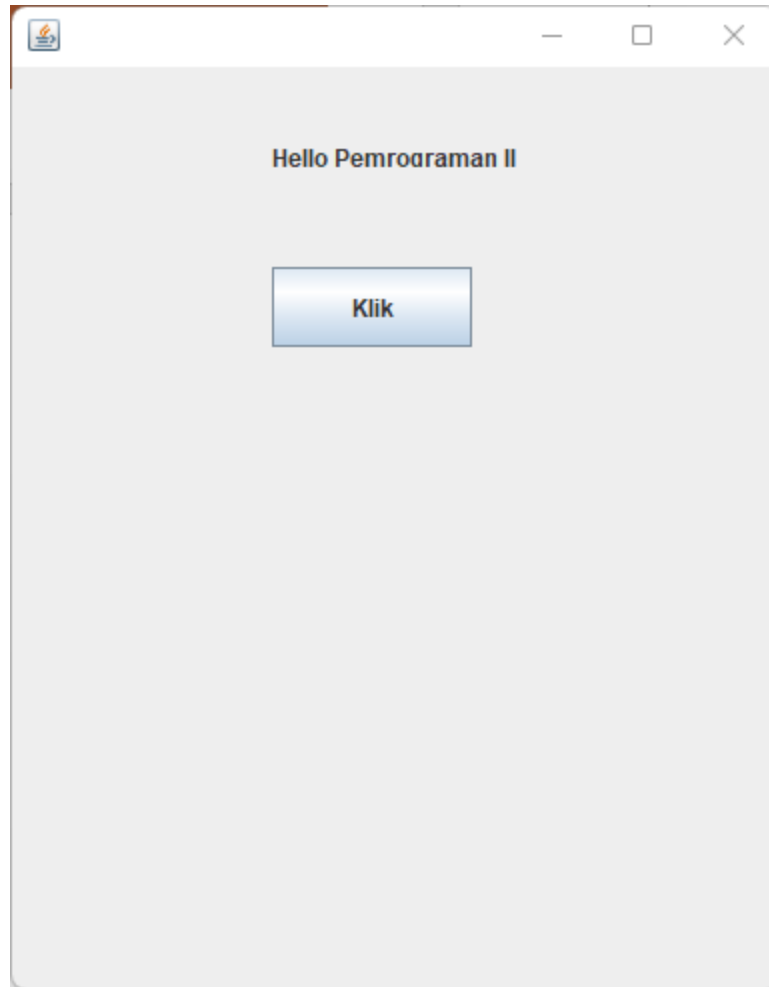
```

Gambar 1 Kode Kelas ButtonEventSwing

Kode diatas memanfaatkan button dan memperlihatkan cara kerja event aksi button. Jika anda kompilasi dan jalankan kode tersebut (dengan menggunakan perintah javac dan java), kode tersebut akan menampilkan aplikasi seperti pada gambar 2, ketika tombol Klik ditekan, label Hello World akan berubah menjadi Hello Pemrograman II seperti pada gambar 3.



Gambar 2 Hasil Run ButtonEventSwing



Gambar 3 Hasil Ketika Button Diklik

TextField

TextField pada JFC direpresentasikan dengan kelas `TextField`. Objek kelas `TextField` adalah komponen teks yang memungkinkan pengeditan teks dalam satu baris. Kelas `TextField` merupakan turunan kelas `TextComponent`. Kelas `TextField` memiliki 4 (empat) konstruktor yang paling sering digunakan dijelaskan pada tabel 3 berikut ini.

Tabel 3 Konstruktor `TextField`

Konstruktor	Keterangan
<code>TextField()</code>	Membuat textfield baru
<code>TextField(String text)</code>	Membuat textfield baru dengan teks inisial
<code>TextField(String text, int columns)</code>	Membuat textfield baru dengan teks inisial dan kolom
<code>TextField(int columns)</code>	Membuat textfield baru dengan jumlah kolom yang ditentukan

Tabel 4 Method JTextField

Method	Keterangan
void addActionListener(ActionListener l)	Digunakan untuk menambahkan listener untuk event aksi
Action getAction()	Mengembalikan Action dari sumber aksi. Biasanya dijalankan dalam sebuah Action Listener.
void setFont(Font f)	Digunakan untuk menentukan jenis font
void removeActionListener(ActionListener l)	Digunakan untuk menghapus listener untuk event aksi.

Latihan 2

Buatlah sebuah kelas bernama HelloInput dengan kode seperti pada gambar 4 berikut.

```

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

public class HelloInput extends JFrame {

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

        JLabel labelInput = new JLabel("Input Nama:");
        labelInput.setBounds(130,40,100,10);

        JTextField textField = new JTextField();
        textField.setBounds(130,60,100,30);

        JButton button = new JButton("Klik");
        button.setBounds(130,100,100, 40);

        JLabel labelOutput = new JLabel("");
        labelOutput.setBounds(130,150,100,10);

        button.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                String nama = textField.getText();
                labelOutput.setText("Hello "+nama);
            }
        });

        this.add(button);
        this.add(textField);
        this.add(labelInput);
        this.add(labelOutput);

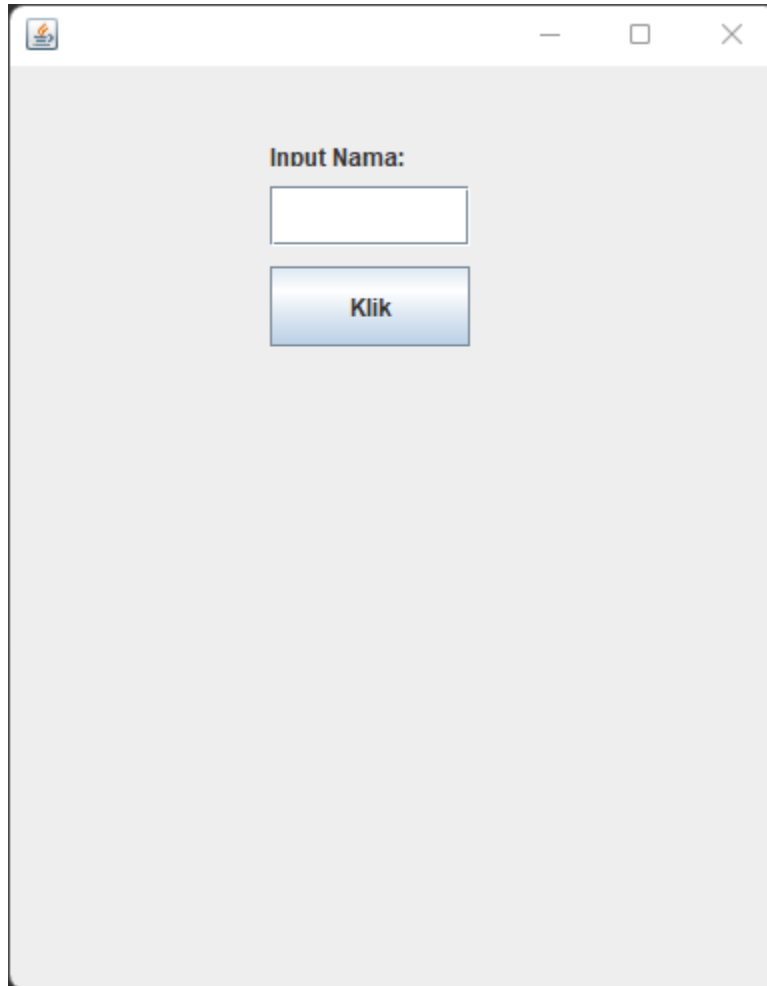
        this.setSize(400,500);
        this.setLayout(null);
    }

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

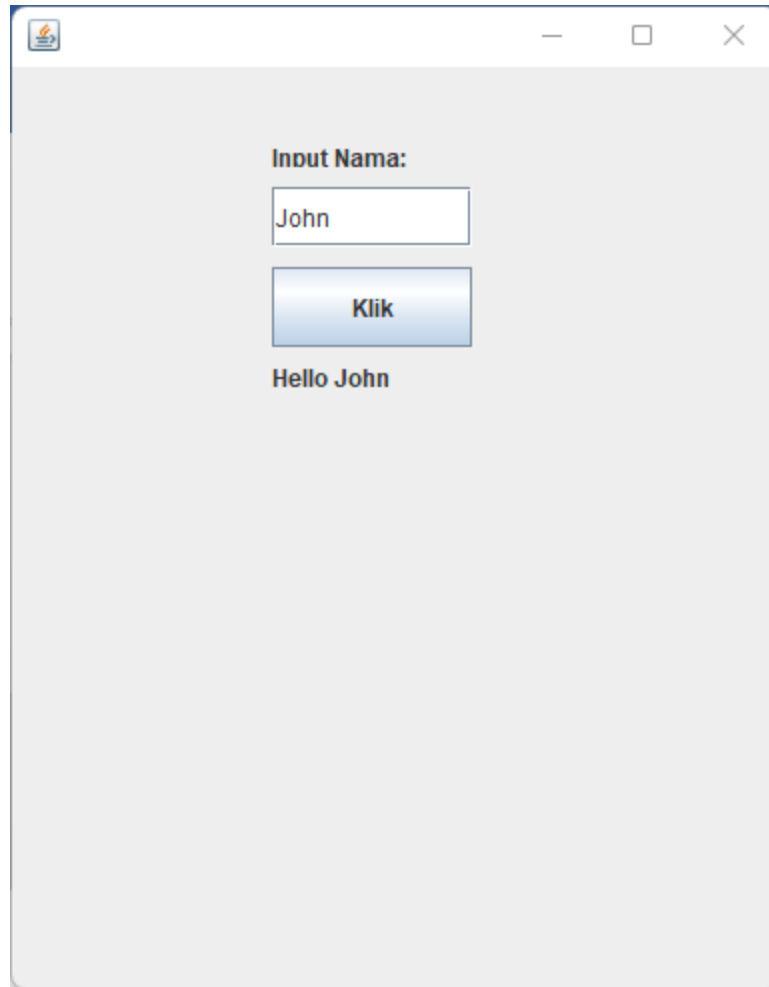
```

Gambar 4 Kode Kelas HelloInput

Kode diatas memanfaatkan button dan memperlihatkan cara kerja event aksi button dan inputan dari textfield. Jika anda kompilasi dan jalankan kode tersebut (dengan menggunakan perintah javac dan java), kode tersebut akan menampilkan aplikasi seperti pada gambar 5, ketika tombol Klik ditekan, akan muncul tulisan Hello ditambah dengan teks inputan dari textfield seperti pada gambar 6.



Gambar 5 Hasil Running HelloInput



Gambar 6 Hasil Output Kelas HelloInput

TextArea

TextArea pada JFC direpresentasikan dengan kelas JTextArea. Objek kelas JTextArea adalah wilayah multi baris yang menampilkan teks. Komponen ini memungkinkan pengeditan beberapa baris teks. Kelas JTextArea merupakan turunan dari kelas JTextComponent. Kelas ini memiliki 4 (empat) konstruktor yang sering digunakan, seperti yang dijelaskan pada tabel 5 berikut ini.

Tabel 5 Konstruktor JTextArea

Konstruktor	Keterangan
JTextArea()	Membuat textarea tanpa teks inisial
JTextArea(String s)	Membuat textarea dengan teks inisial
JTextArea(int row, int column)	Membuat textarea dengan jumlah baris dan jumlah kolom inisial
JTextArea(String s, int row, int column)	Membuat textarea dengan teks, jumlah baris, dan jumlah kolom inisial

Sedangkan method yang sering digunakan pada kelas JTextArea dijelaskan pada tabel 6 berikut.

Tabel 6 Method JTextArea

Method	Keterangan
<code>void setRows(int rows)</code>	Digunakan untuk menentukan jumlah baris
<code>void setColumns(int cols)</code>	Digunakan untuk menentukan jumlah kolom
<code>void setFont(Font f)</code>	Digunakan untuk menentukan jenis font
<code>void insert(String s, int position)</code>	Digunakan untuk menyisipkan teks pada posisi tertentu
<code>void append(String s)</code>	Digunakan untuk menambahkan teks di akhir

Latihan 3

Buatlah sebuah kelas bernama HelloTextArea dengan kode seperti pada gambar 4 berikut.

```

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

public class HelloTextArea extends JFrame {

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

        JLabel labelInput = new JLabel("Input Nama:");
        labelInput.setBounds(130,40,100,10);

        JTextField textField = new JTextField();
        textField.setBounds(130,60,100,30);

        JButton button = new JButton("Klik");
        button.setBounds(130,100,100, 40);

        JTextArea txtOutput = new JTextArea("");
        txtOutput.setBounds(130,150,100,100);

        button.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                String nama = textField.getText();
                txtOutput.append("Hello "+nama+"\n");
                textField.setText("");
            }
        });

        this.add(button);
        this.add(textField);
        this.add(labelInput);
        this.add(txtOutput);

        this.setSize(400,500);
        this.setLayout(null);
    }
}

```

Gambar 7 Kode HelloTextArea Bagian 1

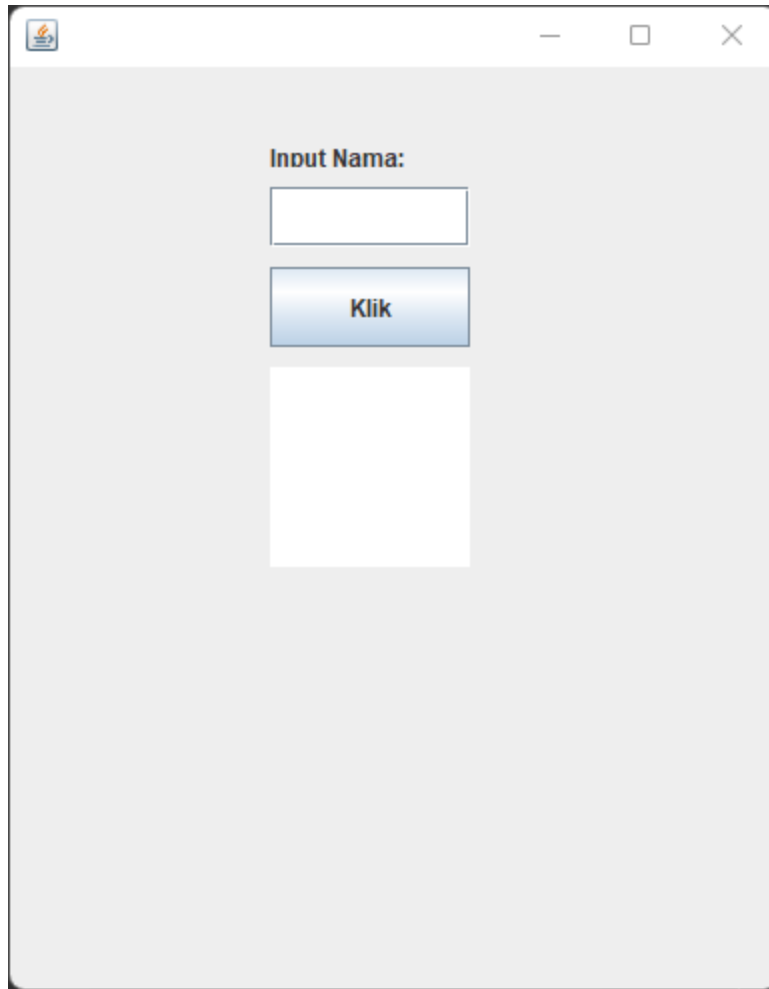
```

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

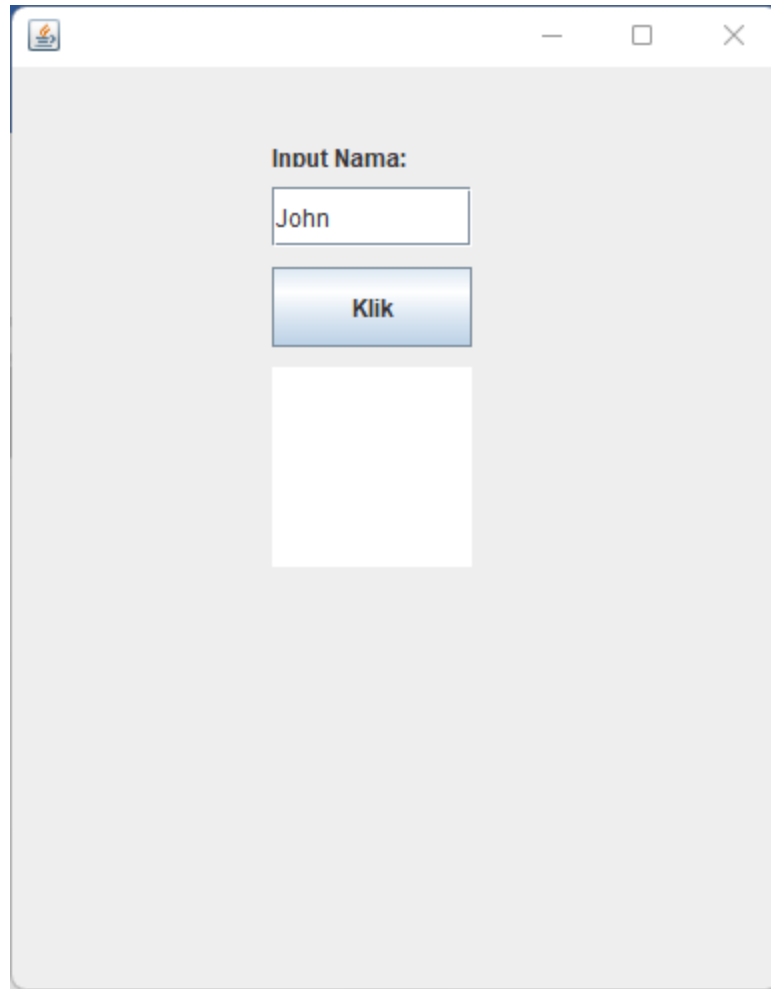
```

Gambar 8 Kode HelloTextArea Bagian 2

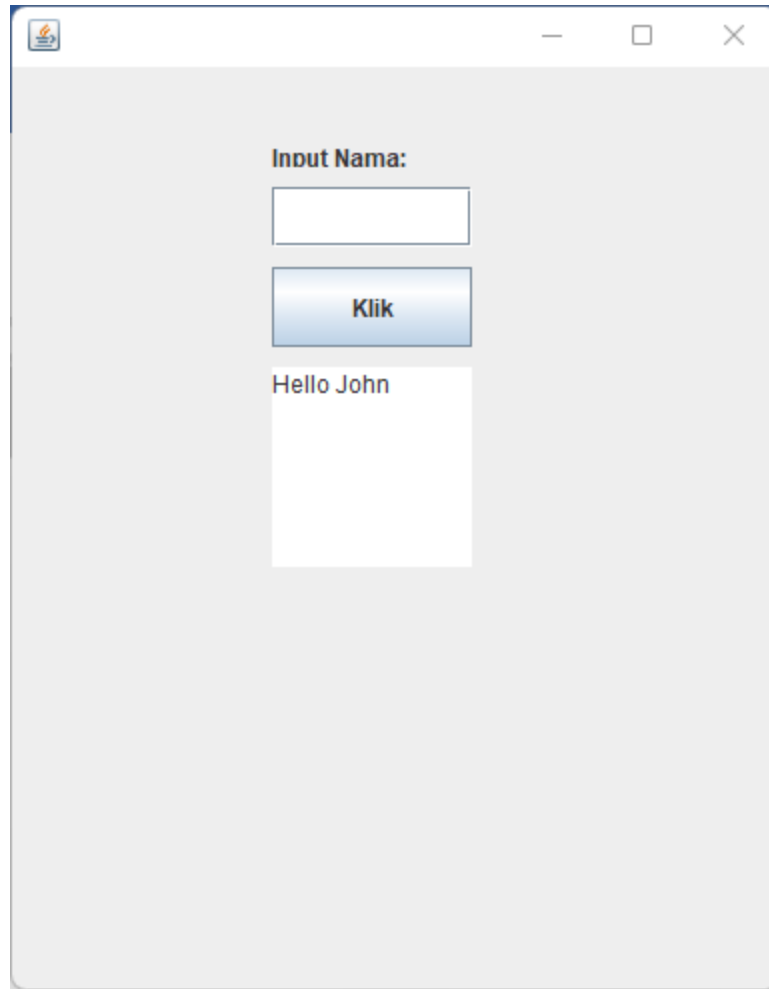
Kode diatas memanfaatkan button dan memperlihatkan cara kerja event aksi button dan inputan dari textfield lalu outputnya ditambahkan (append) ke textarea. Jika anda kompilasi dan jalankan kode tersebut (dengan menggunakan perintah javac dan java), kode tersebut akan menampilkan aplikasi seperti pada gambar 9, ketika tombol Klik ditekan, akan muncul tulisan Hello ditambah dengan teks inputan dari textfield seperti pada gambar 10 dan 11 pada textarea.



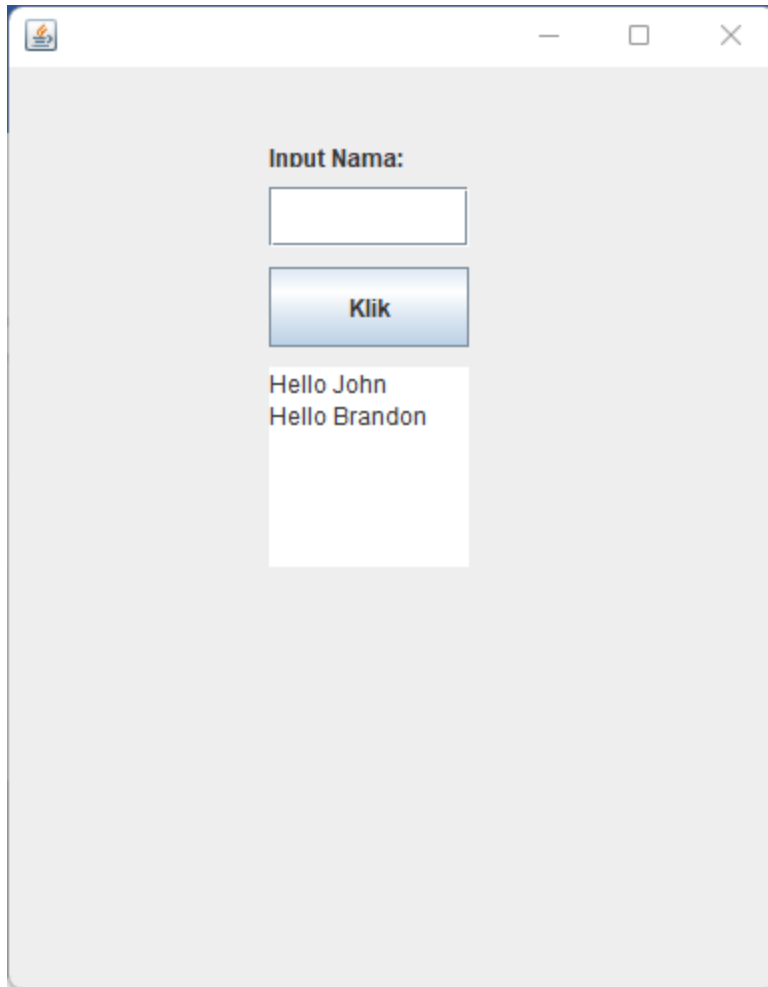
Gambar 9 Hasil Running HelloTextArea



Gambar 10 Inputan HelloTextArea



Gambar 11 Hasil Klik HelloTextArea



Gambar 12 Hasil 2 Kali Input HelloTextArea

Latihan 4

Buatlah sebuah aplikasi untuk menampung biodata teman (nama dan nomor telepon) menggunakan 2 textfield. Biodata disimpan dalam textarea, dipisahkan dengan garis (menggunakan -, _, atau =) untuk setiap data.

CheckBox

CheckBox dalam JFC direpresentasikan dengan kelas JCheckBox. Kelas JCheckBox digunakan untuk membuat kotak yang dapat dicentang. Kotak ini biasanya digunakan jika kita ingin mendapatkan inputan yang bersifat *toggle* (aktif-nonaktif). Komponen ini juga mempunyai kemampuan untuk menghasilkan event ketika diklik atau nilainya berubah. Kelas JButton merupakan turunan kelas JToggleButton. Kelas JButton memiliki 4 (empat) konstruktor yang umum digunakan seperti dijelaskan pada tabel 1 berikut.

Tabel 7 Konstruktor JCheckBox

Konstruktor	Keterangan
JCheckBox()	Membuat kotak yang tidak diceklis dan tanpa teks
JCheckBox(String s)	Membuat kotak tidak diceklis namun dengan teks
JCheckBox(String text, boolean selected)	Membuat kotak dengan teks dan pilihan ceklis atau tidak melalui parameter boolean
JCheckBox(Action a)	Membuat kotak dengan parameter Action

Sedangkan method kelas JCheckBox yang sering digunakan adalah sebagai berikut.

Tabel 8 Method JCheckBox

Method	Keterangan
AccessibleContext getAccessibleContext()	Untuk mengembalikan AccessibleContext (kelas yang menampung informasi mengenai komponen accessible seperti JCheckBox) yang digunakan pada JCheckBox
protected String paramString()	Untuk mengembalikan representasi string dari JCheckBox

Latihan 5

Buatlah sebuah kelas bernama HelloCheckBox dengan kode seperti pada gambar 1 dan 2 berikut.

```

J HelloCheckBox.java
1  import java.awt.event.*;
2  import javax.swing.*;
3
4  public class HelloCheckBox extends JFrame {
5
6      private boolean checkBoxSelected;
7
8      public HelloCheckBox(){
9          this.checkBoxSelected = false;
10         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11
12         JLabel labelInput = new JLabel("Input Nama:");
13         labelInput.setBounds(15,40,350,10);
14
15         JTextField textField = new JTextField();
16         textField.setBounds(15,60,350,30);
17
18         JCheckBox checkBox = new JCheckBox("Saya menyetujui syarat dan ketentuan yang berlaku");
19         checkBox.setBounds(15,100,350,30);
20
21         JButton button = new JButton("Simpan");
22         button.setBounds(15,150,100,40);
23
24         JTextArea txtOutput = new JTextArea("");
25         txtOutput.setBounds(15,200,350,100);
26
27         checkBox.addItemListener(new ItemListener() {
28             public void itemStateChanged(ItemEvent e) {
29                 checkBoxSelected = e.getStateChange() == 1;
30             }
31         });
32

```

Gambar 13 Kode Kelas HelloCheckBox Bagian 1

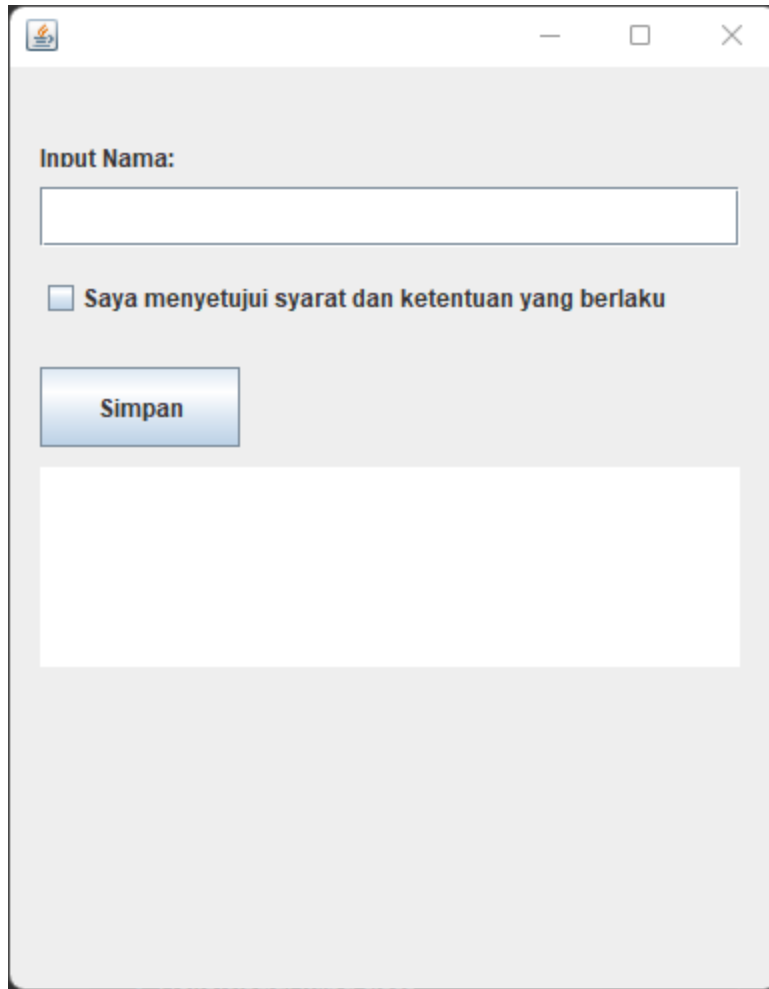
```

32
33     button.addActionListener(new ActionListener(){
34         public void actionPerformed(ActionEvent e){
35             if (checkBoxSelected) {
36                 String nama = textField.getText();
37                 txtOutput.append("Hello "+nama+"\n");
38                 textField.setText("");
39             } else {
40                 txtOutput.append("Anda tidak mencentang kotak persetujuan\n");
41             }
42         }
43     });
44
45     this.add(button);
46     this.add(textField);
47     this.add(checkBox);
48     this.add(labelInput);
49     this.add(txtOutput);
50
51     this.setSize(400,500);
52     this.setLayout(null);
53 }
54
55 public static void main(String[] args) {
56     javax.swing.SwingUtilities.invokeLater(new Runnable() {
57         public void run() {
58             HelloCheckBox h = new HelloCheckBox();
59             h.setVisible(true);
60         }
61     });
62 }
63
64 }

```

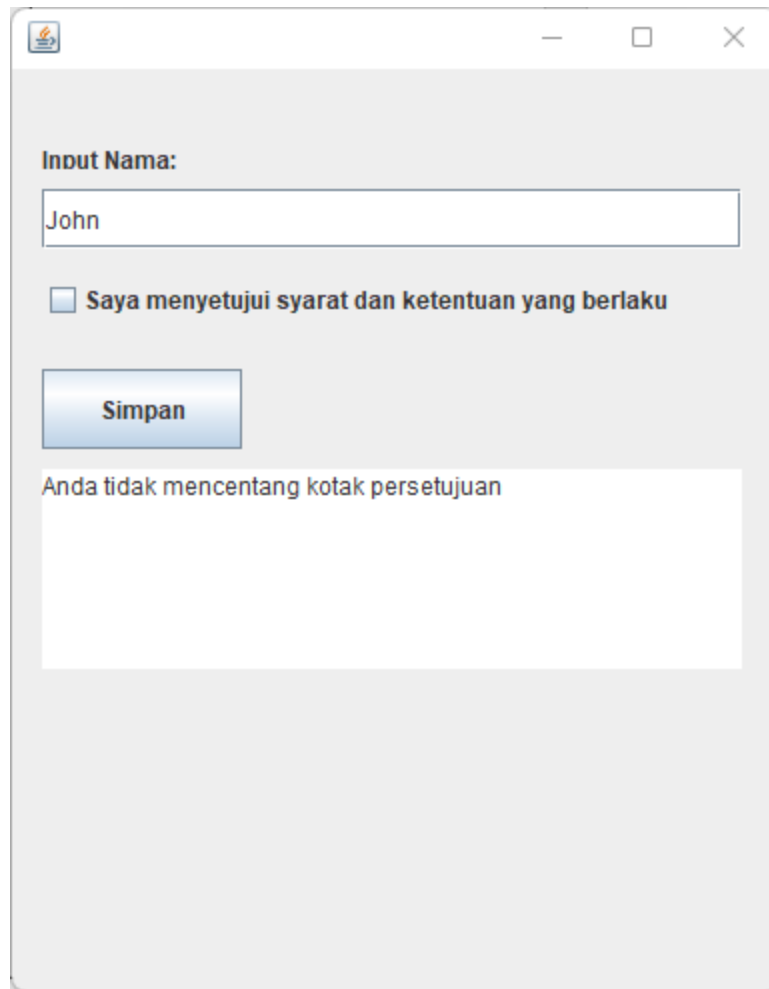
Gambar 14 Kode Kelas HelloCheckBox Bagian 2

Kode diatas memanfaatkan checkbox dan memperlihatkan cara kerja event ketika nilai checkbox berubah. Jika anda kompilasi dan jalankan, kode tersebut akan menampilkan aplikasi seperti pada gambar 3. Aplikasi hanya akan menerima inputan jika checkbox diceklis seperti yang ditunjukkan pada gambar 4 dan 5.



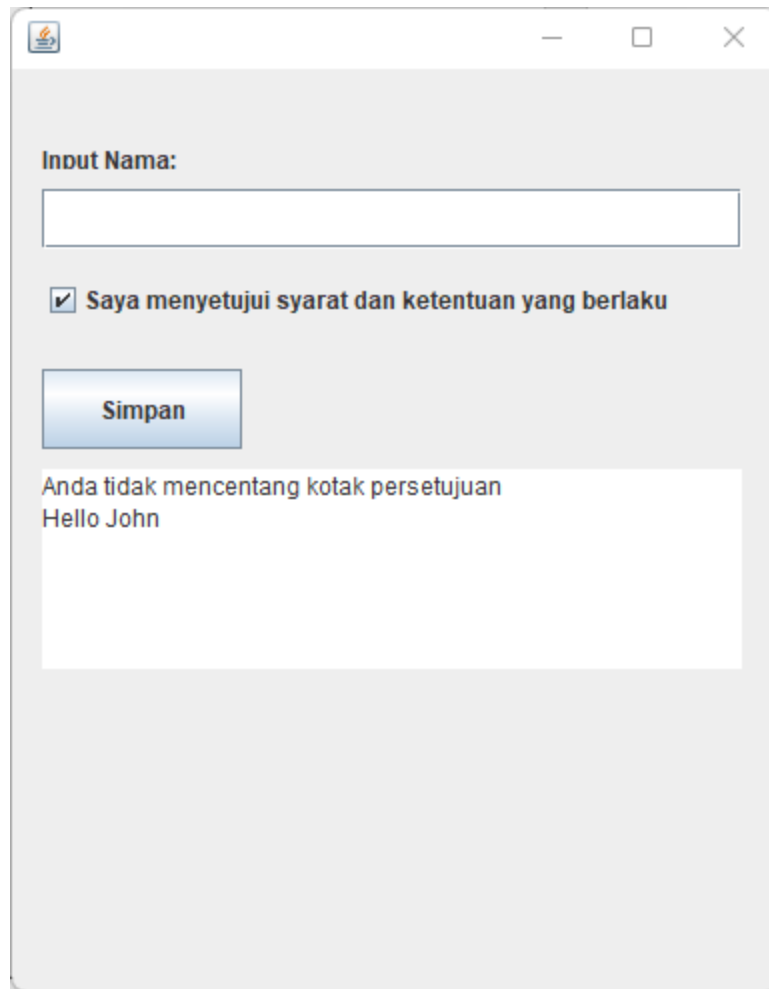
A screenshot of a web application window. The window has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The main content area has a light gray background. It contains the following elements from top to bottom: a label "Input Nama:" in bold black text; a white rectangular text input field with a thin gray border; a checkbox with a small square icon followed by the text "Saya menyetujui syarat dan ketentuan yang berlaku"; a blue rectangular button with the word "Simpan" in white text; and a large, empty white rectangular text area.

Gambar 15 Hasil Eksekusi Aplikasi



The image shows a web browser window with a form titled "Input Nama:". The form contains a text input field with the name "John". Below the input field is a checkbox labeled "Saya menyetujui syarat dan ketentuan yang berlaku", which is currently unchecked. A blue "Simpan" button is positioned below the checkbox. At the bottom of the form, a message states "Anda tidak mencentang kotak persetujuan" (You did not check the consent box), followed by a large, empty white rectangular area.

Gambar 16 User Hanya Menginput Nama



Gambar 17 User Mencentang CheckBox

RadioButton

RadioButton dalam JFC direpresentasikan dengan kelas `JRadioButton`. Kelas `JRadioButton` digunakan untuk membuat tombol berbentuk lingkaran seperti tombol pada radio. RadioButton ini biasanya digunakan jika kita ingin mendapatkan inputan yang bersifat pilihan disajikan dengan beberapa RadioButton. Komponen ini juga mempunyai kemampuan untuk menghasilkan event ketika diklik atau nilainya berubah. Kelas `JRadioButton` merupakan turunan kelas `JToggleButton`. Kelas `JRadioButton` memiliki 3 (tiga) konstruktor yang umum digunakan seperti dijelaskan pada tabel 3 berikut.

Tabel 9 Konstruktor `JRadioButton`

Konstruktor	Keterangan
<code>JRadioButton()</code>	Membuat RadioButton yang tidak dipilih dan tanpa teks

JRadioButton(String s)	Membuat RadioButton tidak dipilih namun dengan teks
JRadioButton(String text, boolean selected)	Membuat RadioButton dengan teks dan status pilihan melalui parameter boolean

Sedangkan method kelas JRadioButton yang sering digunakan adalah sebagai berikut.

Tabel 10 Method JRadioButton

Method	Keterangan
void setText(String s)	Untuk menentukan teks dalam RadioButton
String getText()	Untuk mengembalikan teks dalam RadioButton
void setEnabled(boolean b)	Untuk menentukan apakah RadioButton dipilih atau tidak
void setIcon(Icon b)	Untuk menentukan ikon dari RadioButton
Icon getIcon()	Untuk mengembalikan ikon dari RadioButton
void setMnemonic(int a)	Untuk menentukan kode mnemonic dari RadioButton
void addActionListener(ActionListener a)	Untuk menambahkan action listener dari RadioButton

Latihan 6

Buatlah kelas HelloRadioButton seperti pada gambar 6, 7, dan 8 berikut ini.

J HelloRadioButton.java

```
1  import java.awt.event.*;
2  import javax.swing.*;
3
4  public class HelloRadioButton extends JFrame {
5
6      public HelloRadioButton(){
7          this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
8
9          JLabel labelInput = new JLabel("Nama:");
10         labelInput.setBounds(15,40,350,10);
11
12         JTextField textField = new JTextField();
13         textField.setBounds(15,60,350,30);
14
15         JLabel labelRadio = new JLabel("Jenis Member:");
16         labelRadio.setBounds(15,100,350,10);
17
18         JRadioButton radioButton1 = new JRadioButton("Silver", true);
19         radioButton1.setBounds(15,115,350,30);
20
21         JRadioButton radioButton2 = new JRadioButton("Gold");
22         radioButton2.setBounds(15,145,350,30);
23
24         JRadioButton radioButton3 = new JRadioButton("Platinum");
25         radioButton3.setBounds(15,175,350,30);
26
27         ButtonGroup bg = new ButtonGroup();
28         bg.add(radioButton1);
29         bg.add(radioButton2);
30         bg.add(radioButton3);
31
32         JButton button = new JButton("Simpan");
33         button.setBounds(15,205,100,40);
34
35         JTextArea txtOutput = new JTextArea("");
36         txtOutput.setBounds(15,250,350,100);
```

Gambar 18 Kode HelloRadioButton Bagian 1


```

37
38     button.addActionListener(new ActionListener(){
39         public void actionPerformed(ActionEvent e){
40             String jenisMember = "";
41             if(radioButton1.isSelected()){
42                 jenisMember = radioButton1.getText();
43             }
44             if(radioButton2.isSelected()){
45                 jenisMember = radioButton2.getText();
46             }
47             if(radioButton3.isSelected()){
48                 jenisMember = radioButton3.getText();
49             }
50
51             String nama = textField.getText();
52             txtOutput.append("Hello "+nama+"\n");
53             txtOutput.append("Anda adalah member "+jenisMember+"\n");
54             textField.setText("");
55         }
56     });
57
58     this.add(button);
59     this.add(textField);
60     this.add(labelRadio);
61     this.add(radioButton1);
62     this.add(radioButton2);
63     this.add(radioButton3);
64     this.add(labelInput);
65     this.add(txtOutput);
66
67     this.setSize(400,500);
68     this.setLayout(null);
69 }

```

Gambar 19 Kode HelloRadioButton Bagian 2

```
70
71     public static void main(String[] args) {
72         javax.swing.SwingUtilities.invokeLater(new Runnable() {
73             public void run() {
74                 HelloRadioButton h = new HelloRadioButton();
75                 h.setVisible(true);
76             }
77         });
78     }
79 }
80 }
```

Gambar 20 Kode HelloRadioButton Bagian 3

Kelas HelloRadioButton memiliki 3 (tiga) JRadioButton yang masing-masing digunakan untuk mewakili satu pilihan input (Silver, Gold, dan Platinum). Untuk menghubungkan ketiga JRadioButton, Anda perlu menggunakan ButtonGroup. Hal tersebut bertujuan untuk memastikan hanya ada satu JRadioButton yang dipilih dari semua JRadioButton yang tergabung dalam ButtonGroup. Anda dapat mengetahui apakah salah satu JRadioButton dipilih menggunakan method isSelected() yang akan mengembalikan nilai boolean. Jalankan kelas HelloRadioButton tersebut kemudian perhatikan hasilnya.

Nama:

Jenis Member:

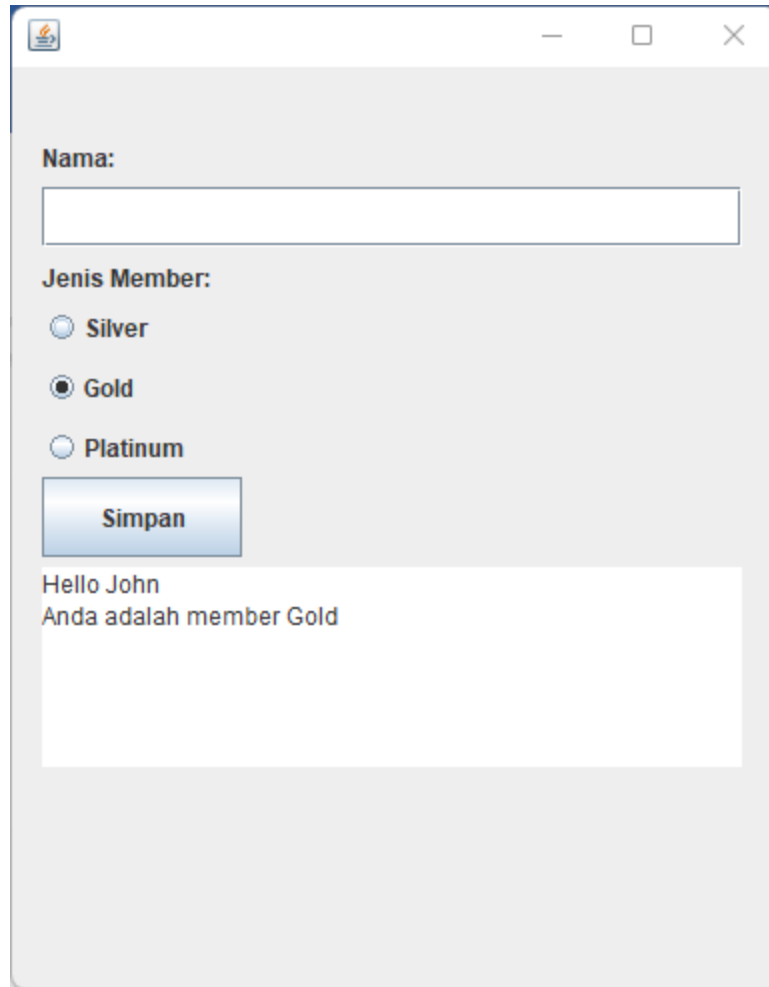
☒ Silver

☐ Gold

☐ Platinum

Simpan

Gambar 21 Hasil Eksekusi Aplikasi



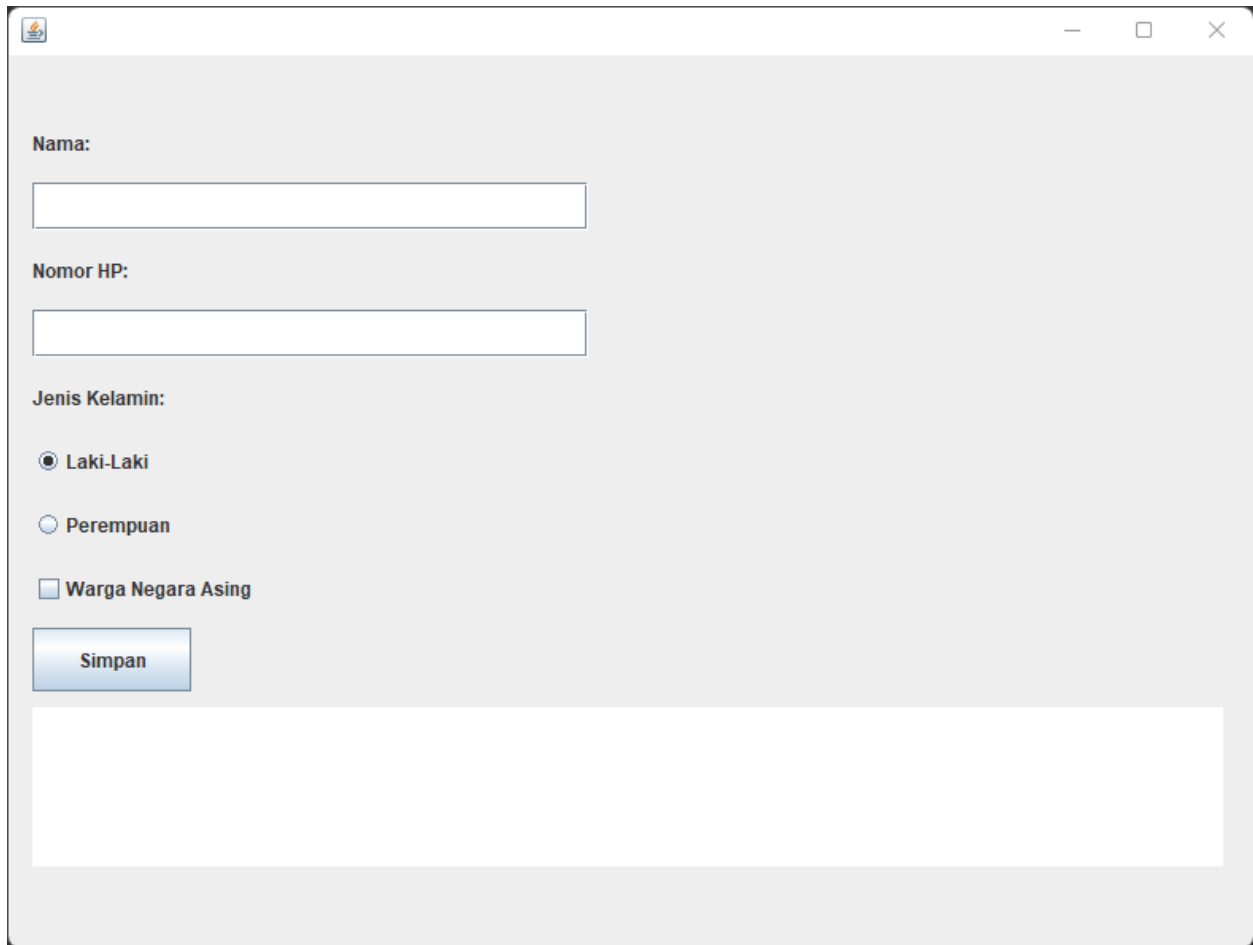
The image shows a Java Swing window with a light gray background. At the top left is a small icon, and at the top right are standard window controls (minimize, maximize, close). The form contains the following elements:

- A label "Nama:" followed by a single-line text input field.
- A label "Jenis Member:" followed by three radio buttons labeled "Silver", "Gold", and "Platinum". The "Gold" radio button is selected.
- A blue "Simpan" button.
- A text area containing the text "Hello John" and "Anda adalah member Gold".

Gambar 22 Hasil Inputan Menggunakan RadioButton

Latihan 7

Lengkapi aplikasi biodata teman dari Modul 2 sehingga memiliki tampilan seperti pada gambar 11 berikut.



The image shows a window titled "Biodata" with a standard Windows-style title bar (minimize, maximize, close buttons). The interface is light gray and contains the following elements:

- Nama:** A text label followed by a single-line text input field.
- Nomor HP:** A text label followed by a single-line text input field.
- Jenis Kelamin:** A text label followed by three radio button options:
 - ☒ Laki-Laki
 - ☐ Perempuan
- ☐ Warga Negara Asing
- Simpan**: A blue button with white text.
- A large, empty white rectangular area at the bottom of the form.

Gambar 23 Tampilan Aplikasi Biodata

Dengan input dan output seperti pada gambar-gambar berikut.

The image shows a web browser window with a form titled "Contoh Input 1". The form contains the following fields and controls:

- Nama:** A text input field containing "John".
- Nomor HP:** A text input field containing "081122334455".
- Jenis Kelamin:** Two radio button options: "Laki-Laki" (selected) and "Perempuan".
- Warga Negara Asing:** A checkbox that is checked.
- Simpan:** A blue button to save the data.

Below the form, a summary of the entered data is displayed in a white box:

Nama : John
Nomor HP : 081122334455
Jenis Kelamin : Laki-Laki
WNA : Ya

Below the summary, there is a dashed line indicating a separator.

Gambar 24 Contoh Input 1

The image shows a web browser window with a form titled "Contoh Input 2". The form contains the following fields and controls:

- Nama:** A text input field containing "Wati".
- Nomor HP:** A text input field containing "085544332211".
- Jenis Kelamin:** A group of radio buttons with the following options:
 - ☐ Laki-Laki
 - ☒ Perempuan
- ☐ Warga Negara Asing
- Simpan**: A blue button to submit the form.

Below the form, a summary of the input is displayed in a white box:

Nama : Wati
Nomor HP : 085544332211
Jenis Kelamin : Perempuan
WNA : Bukan

=====

Gambar 25 Contoh Input 2