



Komponen Swing Dasar

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);

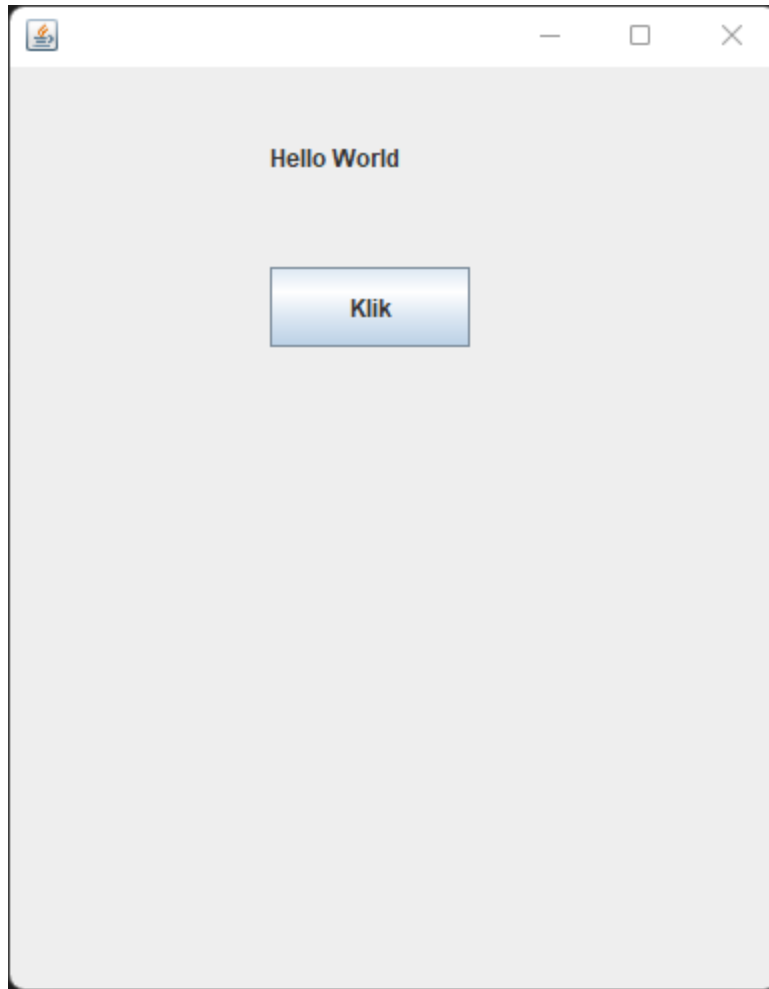
        this.setSize(400,500);
        this.setLayout(null);
    }

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

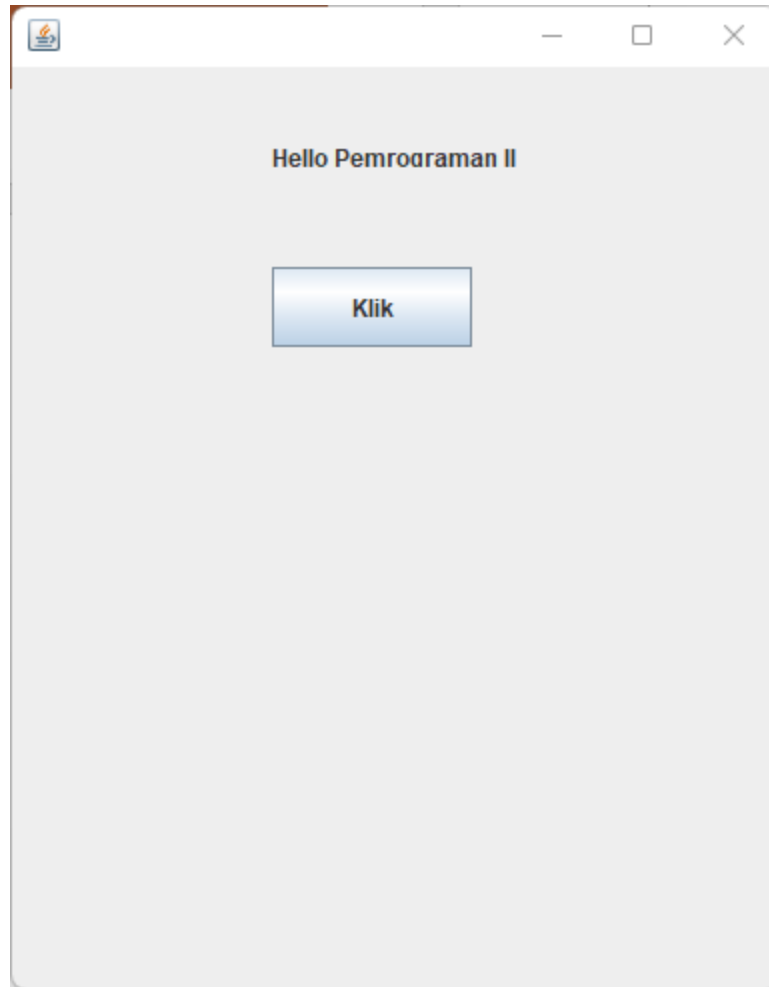
```

Gambar 1 Kode Kelas ButtonEventSwing

Kode diatas memanfaatkan button dan memperlihatkan cara kerja event aksi button. Jika anda kompilasi dan jalankan, kode tersebut 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);

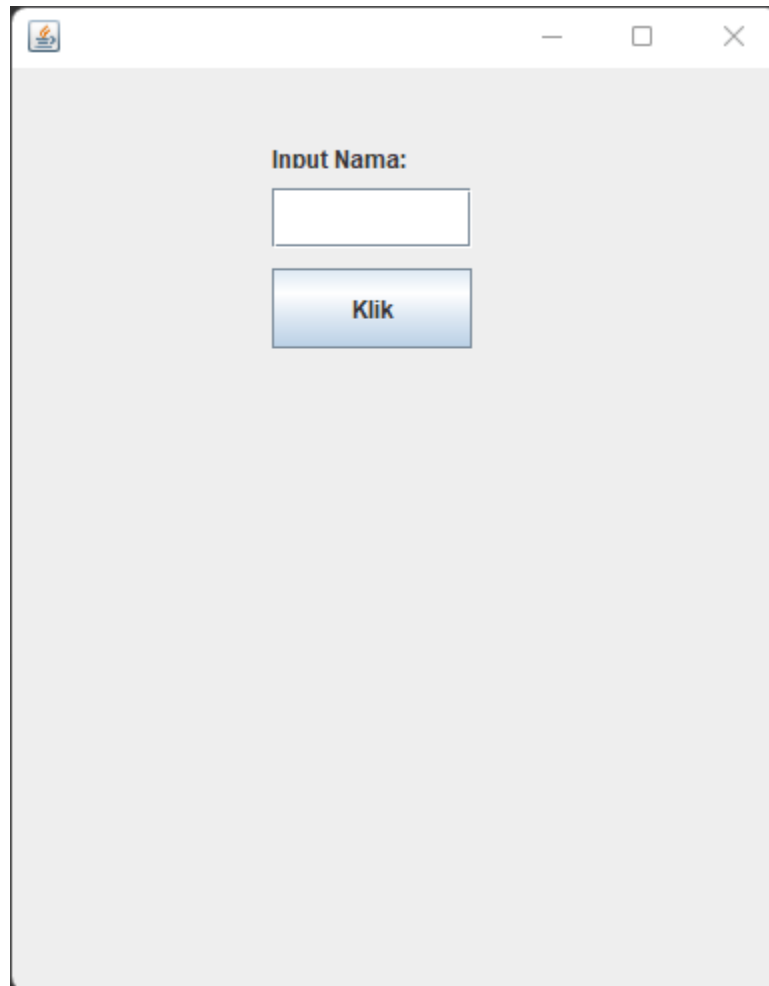
        this.setSize(400,500);
        this.setLayout(null);
    }

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

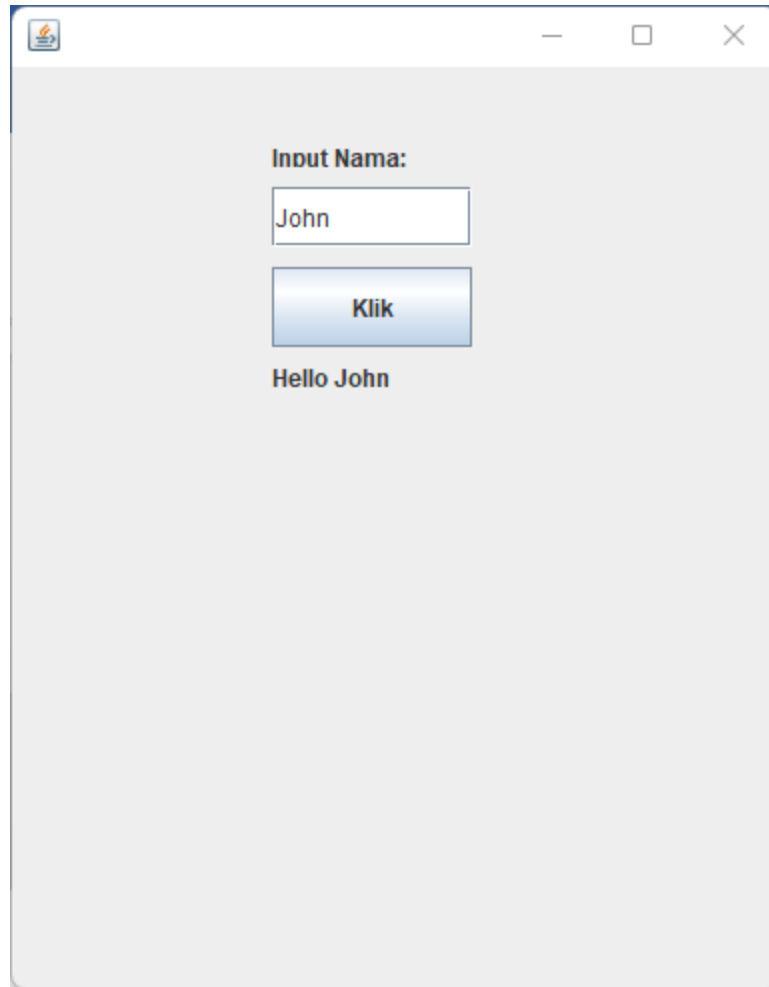
```

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 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

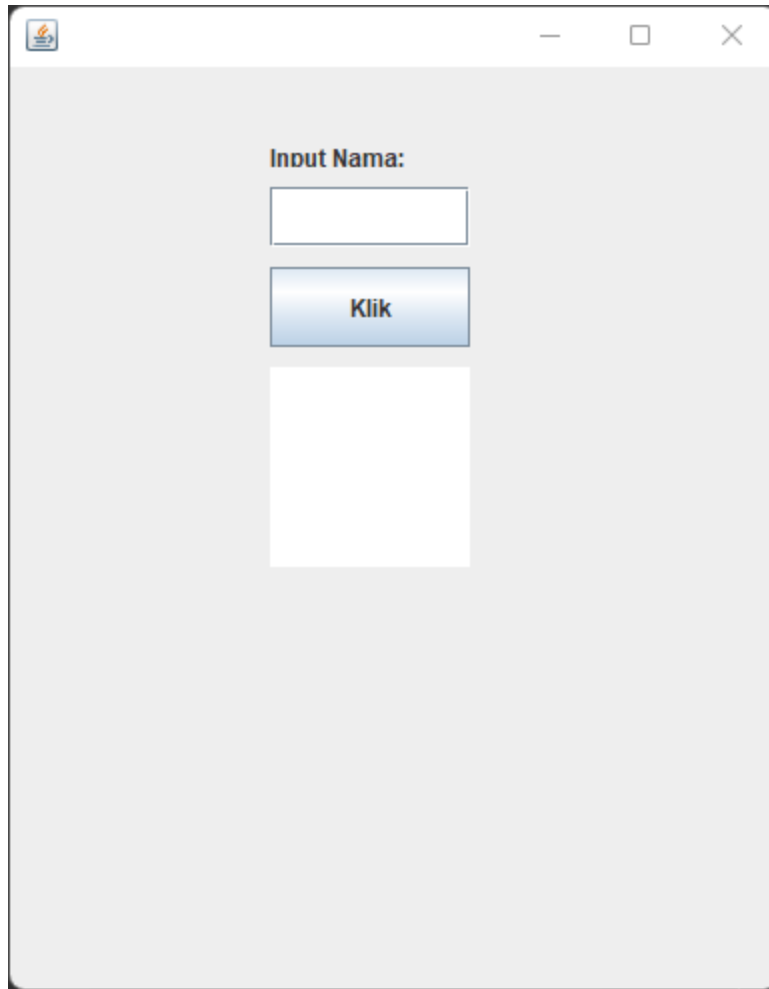
```

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

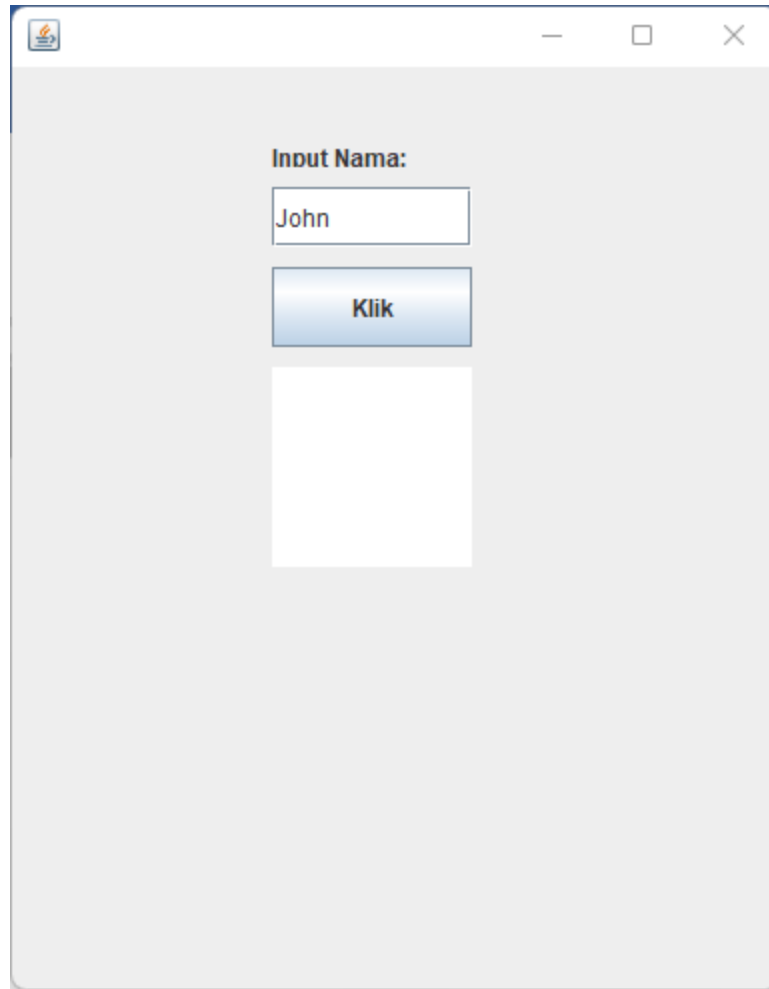
```

Gambar 8 Kode HelloTextArea Bagian 2

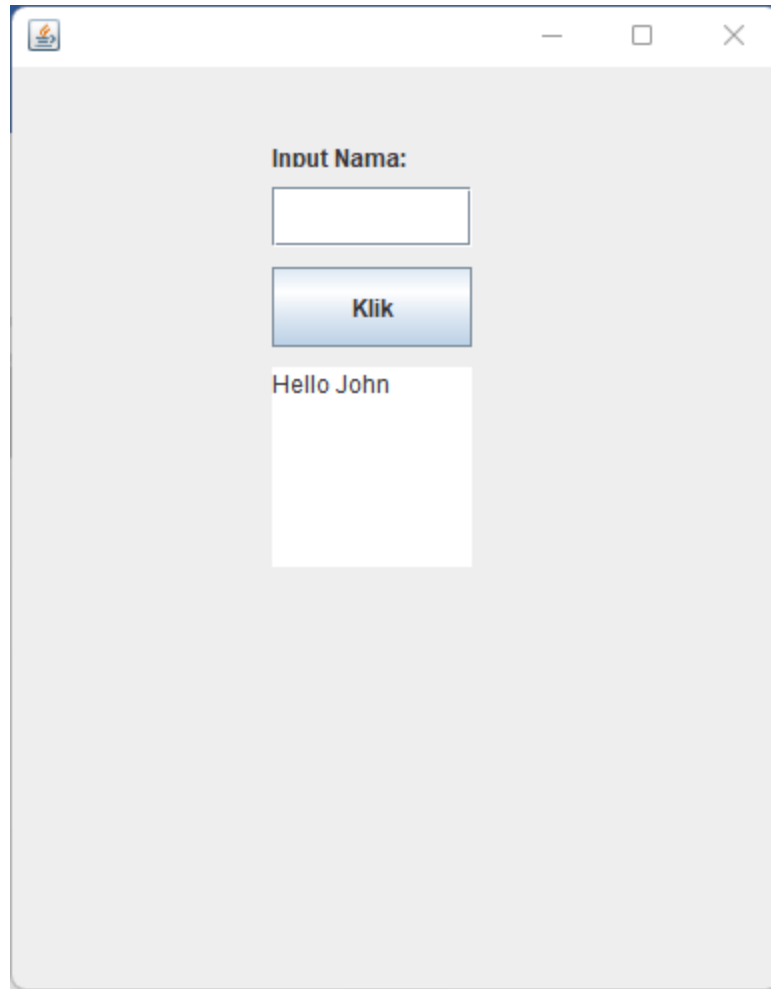
Kode diatas memanfaatkan button dan memperlihatkan cara kerja event aksi button dan inputan dari textfield lalu outputnya ditambahkan (append) ke textarea. Jika anda kompilasi dan jalankan, kode tersebut akan menampilkan aplikasi seperti pada gambar 9, ketika tombol Klik ditekan, akan muncul tulisan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



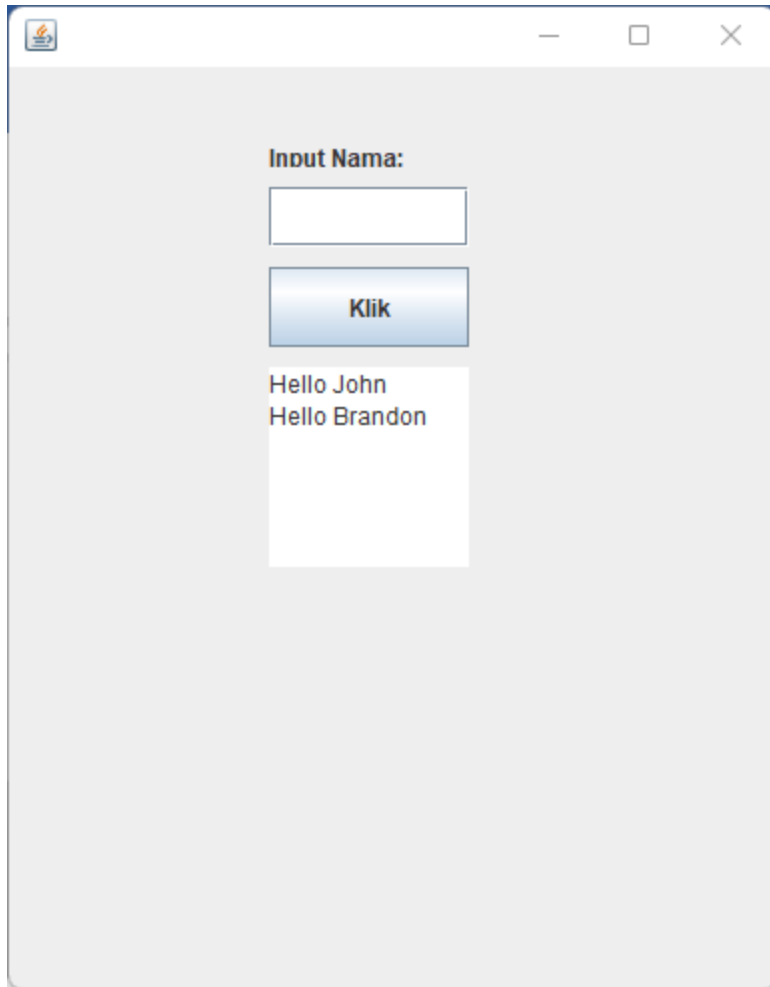
Gambar 9 Hasil Running HelloTextArea



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.