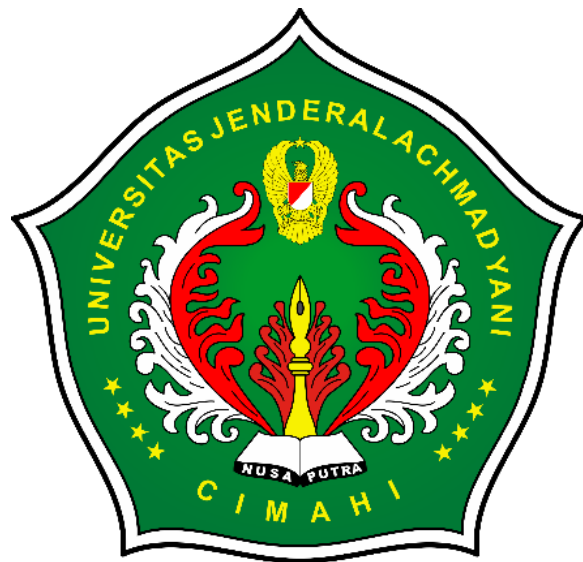


**LAPORAN PRAKTIKUM  
PEMOGRAMAN OBJEK 2**

**MODUL 4  
MEMBANGUN APLIKASI BERBASIS GUI**

**DISUSUN OLEH :  
NAUFAL FADHIL IHSAN FIKRI ASH SHIDQI - 2250081109**



**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN INFORMATIKA  
UNIVERSITAS JENDERAL ACHMAD YANI  
TAHUN 2024**



# DAFTAR ISI

DAFTAR GAMBAR .....	iv
BAB I. HASIL PRAKTIKUM.....	1
I.1 FrameExample.java.....	1
I.1.A. Source Code.....	1
I.1.B. Hasil .....	1
I.1.C. Analisa.....	2
I.2 FrameWithPanel.java .....	3
I.2.A. Source Code.....	3
I.2.B. Hasil .....	4
I.2.C. Analisa.....	4
I.3 GraphicPanel.java .....	5
I.3.A. Source Code.....	5
I.3.B. Hasil .....	6
I.3.C. Analisa.....	6
I.4 FrameWControls.java .....	7
I.4.A. Source Code.....	7
I.4.B. Hasil .....	9
I.4.C. Analisa.....	10
I.5 FlowLayoutDemo.java.....	11
I.5.A. Source Code.....	11
I.5.B. Hasil .....	11
I.5.C. Analisa.....	12
I.6 BorderLayoutDemo.java .....	13
I.6.A. Source Code.....	13
I.6.B. Hasil .....	13

I.6.C. Analisa.....	14
I.7 ComplexLayout.java .....	15
I.7.A. Source Code.....	15
I.7.B. Hasil .....	16
I.7.C. Analisa.....	17
I.8 MyContainer.java.....	18
I.8.A. Source Code.....	18
I.8.B. Hasil .....	18
I.8.C. Analisa.....	19
I.9 SwingDemo.java.....	20
I.9.A. Source Code.....	20
I.9.B. Hasil .....	21
I.9.C. Analisa.....	22
BAB II. TUGAS PRAKTIKUM .....	23
II.1 Tugas 4-1 .....	23
II.2 Tugas 4-2.....	23
II.2.A. Source Code.....	23
II.2.B. Hasil.....	24
II.2.C. Analisa .....	24
II.3 Tugas 4-3.....	25
II.3.A. Source Code.....	25
II.3.B. Hasil.....	26
II.3.C. Analisa .....	26
II.4 Tugas 4-2.....	27
II.4.A. Source Code.....	27
II.4.B. Hasil.....	28
II.5 Tugas Akhir .....	29

II.5.A.	Source Code.....	29
II.5.B.	Hasil.....	34
BAB III.	KESIMPULAN .....	35

## DAFTAR GAMBAR

Gambar I.1 Output Program FrameExample sebelum diperbaiki .....	2
Gambar I.1 Output Program FrameExample setelah diperbaiki.....	2
Gambar I.1 Output Program FrameWithPanel sebelum diperbaiki.....	4
Gambar I.1 Output Program FrameWithPanel setelah diperbaiki .....	4
Gambar I.1 Output Program GraphicPanel sebelum diperbaiki .....	6
Gambar I.1 Output Program GraphicPanel setelah diperbaiki .....	6
Gambar I.1 Output Program FrameWControls sebelum diperbaiki .....	9
Gambar I.1 Output Program FrameWControls setelah diperbaiki .....	9
Gambar I.1 Output Program FlowLayoutDemo sebelum diperbaiki.....	11
Gambar I.1 Output Program FlowLayoutDemo setelah diperbaiki.....	12
Gambar I.1 Output Program BorderLayoutDemo sebelum diperbaiki .....	13
Gambar I.1 Output Program BorderLayoutDemo setelah diperbaiki .....	14
Gambar I.1 Output Program ComplexLayout diperbaiki.....	16
Gambar I.1 Output Program ComplexLayout diperbaiki.....	16
Gambar I.1 Output Program MyContainer diperbaiki .....	18
Gambar I.1 Output Program MyContainer diperbaiki .....	19
Gambar I.1 Output Program SwingDemo diperbaiki .....	22
Gambar I.1 Output Program SwingDemo diperbaiki .....	22
Gambar II.1 Output GraphicPanel Tugas A .....	24
Gambar II.1 Output FlowLayoutDemo Tugas A.....	26
Gambar II.1 Output FlowLayoutDemo Tugas A.....	28
Gambar I.1 Output Program Instagram diperbaiki .....	34

# BAB I. HASIL PRAKTIKUM

## I.1 FrameExample.java

### I.1.A. Source Code

Sebelum diperbaiki:

```
import java.awt.*;

public class FrameExample {

    private Frame f;

    public FrameExample() {

        f = new Frame("Hello Out There!");

    }

    public void launchFrame() {

        f.setSize(300, 300);

        f.setBackground(Color.blue);

        f.setVisible(true);

    }

    public static void main(String[] args) {

        FrameExample guiWindow = new FrameExample();

        guiWindow.launchFrame();

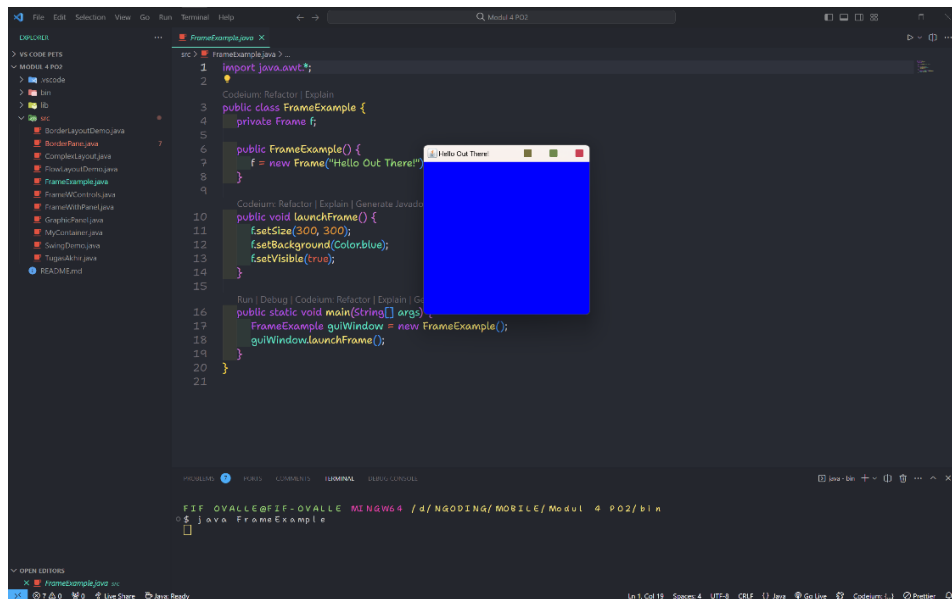
    }

}
```

Setelah diperbaiki

-

### I.1.B. Hasil



*Gambar 1.1 Output Program FrameExample sebelum diperbaiki*

-

*Gambar 1.2 Output Program FrameExample setelah diperbaiki*

### I.1.C. Analisa

Program tersebut merupakan contoh sederhana pembuatan GUI (Graphical User Interface) menggunakan Java AWT (Abstract Window Toolkit). Kelas `FrameExample` memiliki sebuah instance variabel bertipe `Frame` yang digunakan untuk membuat jendela. Di dalam konstruktor, sebuah jendela baru dengan judul "Hello Out There!" dibuat. Metode `launchFrame()` mengatur ukuran jendela menjadi 300x300 piksel dan mengubah warna latar belakangnya menjadi biru sebelum menampilkannya. Metode `main` membuat objek `FrameExample` dan memanggil `launchFrame()`, sehingga jendela dengan judul "Hello Out There!" ditampilkan dengan ukuran dan warna latar belakang yang telah ditentukan. Program ini memberikan contoh dasar tentang bagaimana membuat dan menampilkan jendela sederhana dalam Java menggunakan AWT.



## I.2 FrameWithPanel.java

### I.2.A. Source Code

Sebelum diperbaiki:

```
import java.awt.*;

public class FrameWithPanel {

    private Frame f;

    public FrameWithPanel(String title) {

        f = new Frame(title);

    }

    public void launchFrame() {

        f.setSize(200, 200);

        f.setBackground(Color.blue);

        f.setVisible(true);

        f.setLayout(null);

        Panel pan = new Panel();

        pan.setSize(100, 100);

        pan.setBackground(Color.yellow);

        f.add(pan);

        f.setVisible(true);

    }

    public static void main(String[] args) {

        FrameWithPanel guiWindow = new FrameWithPanel("Frame

        with Panel");

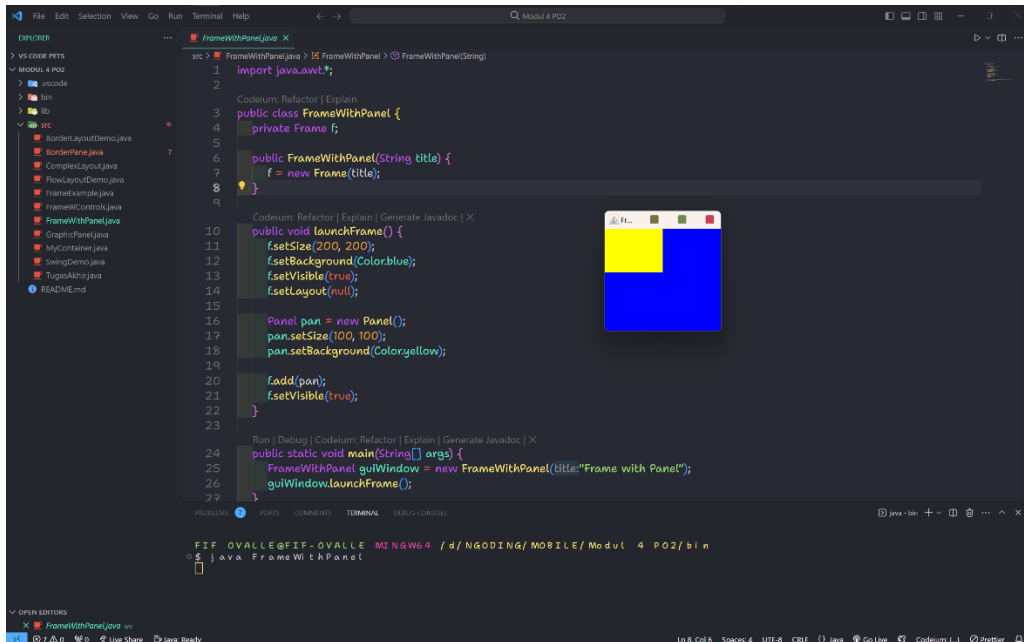
        guiWindow.launchFrame();

    }
}
```

Setelah diperbaiki

-

## I.2.B. Hasil



Gambar 1.3 Output Program FrameWithPanel sebelum diperbaiki

-

Gambar 1.4 Output Program FrameWithPanel setelah diperbaiki

## I.2.C. Analisa

Program ini merupakan contoh pembuatan GUI sederhana menggunakan Java AWT dengan menambahkan panel ke dalam jendela. Kelas 'FrameWithPanel' memiliki sebuah instance variabel bertipe 'Frame' yang digunakan untuk membuat jendela dengan judul yang diberikan dalam konstruktor. Metode 'launchFrame()' mengatur ukuran jendela, warna latar belakangnya, dan menampilkannya. Selain itu, sebuah panel dengan warna latar belakang kuning ditambahkan ke dalam jendela. Program ini memberikan contoh penggunaan panel dalam Java AWT untuk menambahkan komponen ke dalam jendela.

## I.3 GraphicPanel.java

### I.3.A. Source Code

Sebelum diperbaiki:

```
import java.awt.*;

public class GraphicPanel extends Panel {

    public GraphicPanel() {
        setBackground(Color.black);
    }

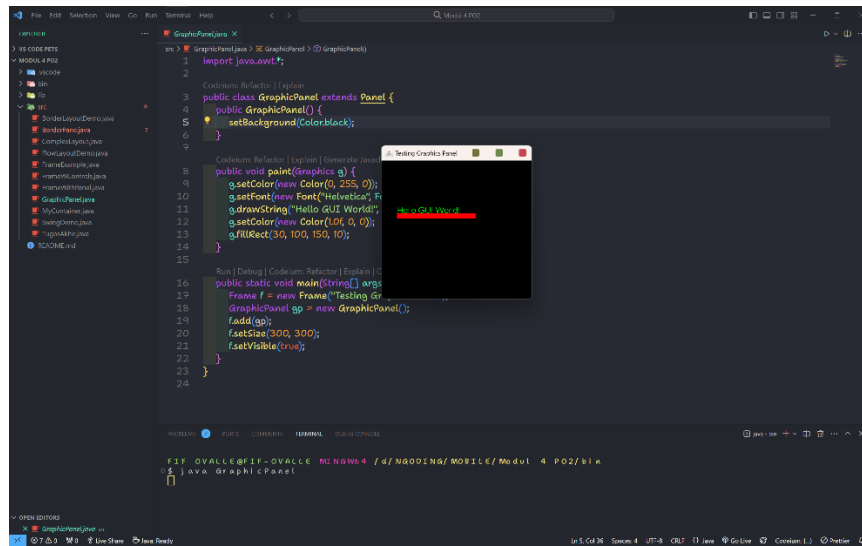
    public void paint(Graphics g) {
        g.setColor(new Color(0, 255, 0));
        g.setFont(new Font("Helvetica", Font.PLAIN, 16));
        g.drawString("Hello GUI World!", 30, 100);
        g.setColor(new Color(1.0f, 0, 0));
        g.fillRect(30, 100, 150, 10);
    }

    public static void main(String[] args) {
        Frame f = new Frame("Testing Graphics Panel");
        GraphicPanel gp = new GraphicPanel();
        f.add(gp);
        f.setSize(300, 300);
        f.setVisible(true);
    }
}
```

Setelah diperbaiki

-

### I.3.B. Hasil



Gambar I.5 Output Program GraphicPanel sebelum diperbaiki

Gambar I.6 Output Program GraphicPanel setelah diperbaiki

### I.3.C. Analisa

Program tersebut merupakan contoh implementasi sebuah panel grafis dalam Java menggunakan AWT (Abstract Window Toolkit). Kelas `GraphicPanel` adalah subkelas dari `Panel` yang digunakan untuk menampilkan grafis di dalamnya. Dalam konstruktor, warna latar belakang panel diatur menjadi hitam. Metode `paint(Graphics g)` digunakan untuk menggambar teks "Hello GUI World!" berwarna hijau pada koordinat (30, 100) dan sebuah persegi panjang merah pada koordinat (30, 100) dengan lebar 150 dan tinggi 10. Metode `main` membuat sebuah jendela dengan judul "Testing Graphics Panel", membuat objek `GraphicPanel`, menembarkannya ke dalam jendela, mengatur ukuran jendela menjadi 300x300 piksel, dan menampilkannya. Program ini memberikan contoh tentang cara membuat panel grafis sederhana dalam Java menggunakan AWT dan menambahkan objek panel tersebut ke dalam jendela.

## I.4 FrameWControls.java

### I.4.A. Source Code

Sebelum diperbaiki:

```
import java.awt.*;

public class FrameWControls extends Frame {

    public FrameWControls() {

        FrameWControls fwc = new FrameWControls();

        fwc.setLayout(new FlowLayout());

        fwc.setSize(600, 600);

        fwc.add(new Button("Test Me!"));

        fwc.add(new Label("Label"));

        fwc.add(new TextField());

        CheckboxGroup cbg = new CheckboxGroup();

        fwc.add(new Checkbox("chk1", cbg, true));

        fwc.add(new Checkbox("chk2", cbg, false));

        fwc.add(new Checkbox("chk3", cbg, false));

        List list = new List(3, false);

        list.add("MTV");

        list.add("V");

        fwc.add(list);

        Choice chooser = new Choice();

        chooser.add("Avril");

        chooser.add("Monica");

        chooser.add("Britney");

        fwc.add(chooser);

        fwc.add(new Scrollbar());
```

```
        fwc.setVisible(true);  
    }  
}
```

Setelah diperbaiki

```
import java.awt.*;  
  
public class FrameWControls extends Frame {  
    public FrameWControls() {  
        setLayout(new FlowLayout());  
        add(new Button("Test Me!"));  
        add(new Label("Label"));  
        add(new TextField());  
        CheckboxGroup cbg = new CheckboxGroup();  
        add(new Checkbox("chk1", cbg, true));  
        add(new Checkbox("chk2", cbg, false));  
        add(new Checkbox("chk3", cbg, false));  
        List list = new List(3, false);  
        list.add("MTV");  
        list.add("V");  
        add(list);  
        Choice chooser = new Choice();  
        chooser.add("Avril");  
        chooser.add("Monica");  
        chooser.add("Britney");  
        add(chooser);  
        add(new Scrollbar());  
        setSize(600, 600);  
    }  
}
```

```

setVisible(true);

addWindowListener(new java.awt.event.WindowAdapter() {

    public void windowClosing(java.awt.event.WindowEvent
        windowEvent) {

        System.exit(0);

    }

});

}

public static void main(String[] args) {

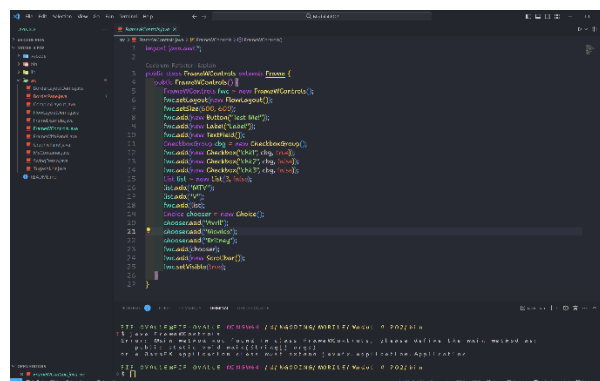
    FrameWControls frame = new FrameWControls();

}

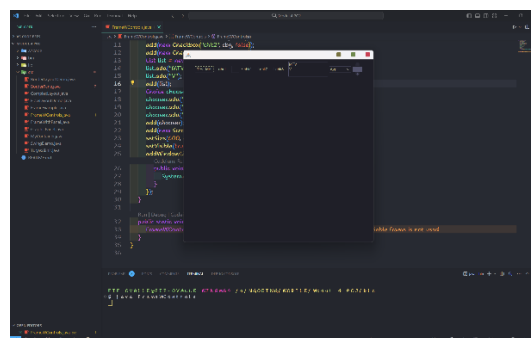
}

```

#### I.4.B. Hasil



Gambar 1.7 Output Program FrameWControls sebelum diperbaiki



Gambar 1.8 Output Program FrameWControls setelah diperbaiki

#### **I.4.C. Analisa**

Program tersebut merupakan contoh implementasi sebuah jendela dengan berbagai kontrol atau komponen GUI (Graphical User Interface) dalam Java menggunakan AWT (Abstract Window Toolkit). Kelas `FrameWControls` merupakan subkelas dari `Frame`, yang bertanggung jawab untuk menampilkan jendela dengan berbagai kontrol di dalamnya. Dalam konstruktor, sebuah objek baru dari kelas `FrameWControls` dibuat. Lalu, diatur tata letaknya menggunakan `FlowLayout` dan ukurannya menjadi 600x600 piksel. Berbagai kontrol ditambahkan ke jendela, termasuk tombol (`Button`), label (`Label`), kotak teks (`TextField`), grup checkbox (`CheckboxGroup`) dengan tiga kotak centang (`Checkbox`), daftar (`List`), pilihan (`Choice`), dan scrollbar (`Scrollbar`). Setelah semua kontrol ditambahkan, jendela ditampilkan ke layar dengan memanggil `setVisible(true)`. Program ini memberikan contoh tentang cara membuat jendela GUI dengan berbagai kontrol menggunakan AWT dalam Java.



## I.5 FlowLayoutDemo.java

### I.5.A. Source Code

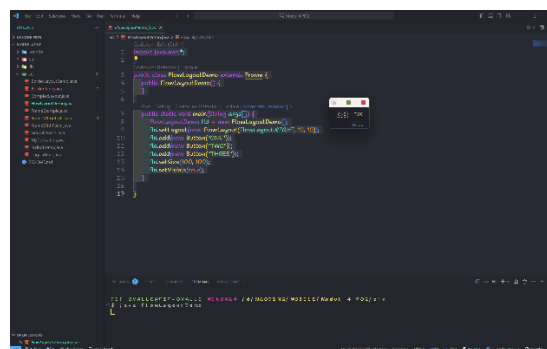
Sebelum diperbaiki:

```
import java.awt.*;  
  
public class FlowLayoutDemo extends Frame {  
  
    public FlowLayoutDemo() {  
  
    }  
  
    public static void main(String args[]) {  
  
        FlowLayoutDemo fld = new FlowLayoutDemo();  
  
        fld.setLayout(new FlowLayout(FlowLayout.RIGHT, 10,  
        10));  
  
        fld.add(new Button("ONE"));  
  
        fld.add(new Button("TWO"));  
  
        fld.add(new Button("THREE"));  
  
        fld.setSize(100, 100);  
  
        fld.setVisible(true);  
  
    }  
  
}
```

Setelah diperbaiki

-

### I.5.B. Hasil



Gambar 1.9 Output Program FlowLayoutDemo sebelum diperbaiki

### **I.5.C. Analisa**

Program `FlowLayoutDemo` adalah contoh penerapan tata letak aliran (`FlowLayout`) dalam Java AWT (Abstract Window Toolkit). Kelas `FlowLayoutDemo` merupakan subkelas dari `Frame` yang bertujuan untuk menampilkan jendela dengan tombol-tombol di dalamnya. Dalam metode `main`, sebuah objek `FlowLayoutDemo` dibuat. Kemudian, tata letak jendela diatur menggunakan `FlowLayout` dengan parameter `FlowLayout.RIGHT`, yang mengatur tata letak elemen-elemen dalam aliran ke arah kanan, serta jarak horizontal dan vertikal antar elemen sebesar 10 piksel. Tiga tombol dengan teks "ONE", "TWO", dan "THREE" ditambahkan ke jendela. Ukuran jendela diatur menjadi 100x100 piksel dan ditampilkan ke layar dengan memanggil `setVisible(true)`. Dengan demikian, program ini memberikan contoh tentang cara menggunakan tata letak aliran untuk menata elemen-elemen GUI dalam Java AWT.

## I.6 BorderLayoutDemo.java

### I.6.A. Source Code

Sebelum diperbaiki:

```
import java.awt.*;

public class BorderLayoutDemo extends Frame {

    public BorderLayoutDemo() {

    }

    public static void main(String args[]) {

        BorderLayoutDemo bld = new BorderLayoutDemo();

        bld.setLayout(new BorderLayout(10, 10));

        bld.add(new Button("SOUTH"), BorderLayout.SOUTH);

        bld.add(new Button("EAST"), BorderLayout.EAST);

        bld.add(new Button("WEST"), BorderLayout.WEST);

        bld.add(new Button("CENTER"), BorderLayout.CENTER);

        bld.setSize(200, 200);

        bld.setVisible(true);

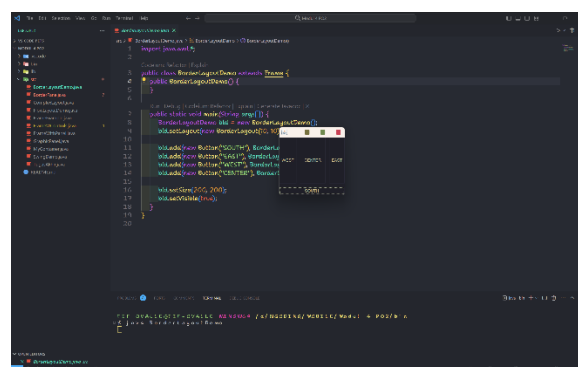
    }

}
```

Setelah diperbaiki

-

### I.6.B. Hasil



Gambar 1.11 Output Program BorderLayoutDemo sebelum diperbaiki

-

*Gambar 1.12 Output Program BorderLayoutDemo setelah diperbaiki*

### **I.6.C. Analisa**

Program `BorderLayoutDemo` adalah contoh penerapan tata letak border (`BorderLayout`) dalam Java AWT (Abstract Window Toolkit). Kelas `BorderLayoutDemo` merupakan subkelas dari `Frame` yang bertujuan untuk menampilkan jendela dengan tombol-tombol di dalamnya, masing-masing diletakkan di bagian-bagian utama dari tata letak border. Dalam metode `main`, sebuah objek `BorderLayoutDemo` dibuat. Kemudian, tata letak jendela diatur menggunakan `BorderLayout` dengan jarak horizontal dan vertikal antar elemen sebesar 10 piksel. Empat tombol dengan teks "SOUTH", "EAST", "WEST", dan "CENTER" ditambahkan ke jendela dengan menentukan lokasi penempatannya sesuai dengan bagian-bagian dari tata letak border menggunakan konstanta `BorderLayout.SOUTH`, `BorderLayout.EAST`, `BorderLayout.WEST`, dan `BorderLayout.CENTER`. Ukuran jendela diatur menjadi 200x200 piksel dan ditampilkan ke layar dengan memanggil `setVisible(true)`. Dengan demikian, program ini memberikan contoh tentang cara menggunakan tata letak border untuk menata elemen-elemen GUI dalam Java AWT.

## I.7 ComplexLayout.java

### I.7.A. Source Code

Sebelum diperbaiki:

```
import java.awt.*;

public class ComplexLayout extends Frame {

    public ComplexLayout() {

    }

    public static void main(String args[]) {

        ComplexLayout cl = new ComplexLayout();

        Panel panelNorth = new Panel();

        panelNorth.add(new Button("ONE"));
        panelNorth.add(new Button("TWO"));
        panelNorth.add(new Button("THREE"));

        Panel panelCenter = new Panel();

        panelCenter.setLayout(new GridLayout(4, 4));
        panelCenter.add(new TextField("1st"));
        panelCenter.add(new TextField("2nd"));
        panelCenter.add(new TextField("3rd"));
        panelCenter.add(new TextField("4th"));

        Panel panelSouth = new Panel();

        panelSouth.setLayout(new BorderLayout());
        panelSouth.add(new Checkbox("Choose me!"),
            BorderLayout.CENTER);

        panelSouth.add(new Checkbox("I'm here!"),
            BorderLayout.EAST);

        panelSouth.add(new Checkbox("Pick me!"),
```

```

        BorderLayout.WEST);

        cl.add(panelNorth, BorderLayout.NORTH);

        cl.add(panelCenter, BorderLayout.CENTER);

        cl.add(panelSouth, BorderLayout.SOUTH);

        cl.setSize(300, 300);

        cl.setVisible(true);

    }

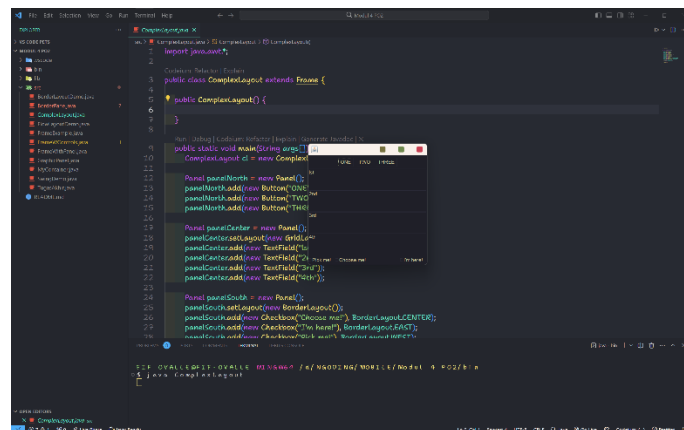
}

```

Setelah diperbaiki

-

### I.7.B. Hasil



Gambar 1.13 Output Program ComplexLayout diperbaiki

-

Gambar 1.14 Output Program ComplexLayout diperbaiki

### **I.7.C. Analisa**

Program ``ComplexLayout`` adalah contoh implementasi tata letak yang kompleks dalam Java AWT (Abstract Window Toolkit). Kelas ``ComplexLayout`` merupakan subkelas dari ``Frame`` yang bertujuan untuk menampilkan jendela dengan tata letak yang lebih rumit. Dalam metode ``main``, sebuah objek ``ComplexLayout`` dibuat. Kemudian, tiga panel dibuat untuk mewadahi elemen-elemen GUI dengan tata letak yang berbeda.

Panel utara (``panelNorth``) dibuat dan diisi dengan tiga tombol yang disusun secara horizontal.

Panel tengah (``panelCenter``) dibuat dan diatur menggunakan tata letak grid (``GridLayout``) dengan ukuran 4x4, lalu diisi dengan empat kotak teks.

Panel selatan (``panelSouth``) dibuat dan diatur menggunakan tata letak border (``BorderLayout``). Pada bagian tengah panel selatan ditempatkan sebuah checkbox, sedangkan checkbox lainnya ditempatkan di sisi kanan dan kiri.

Kemudian, ketiga panel tersebut ditambahkan ke jendela ``ComplexLayout`` dengan menentukan lokasi penempatannya sesuai dengan bagian-bagian dari tata letak border (``BorderLayout.NORTH``, ``BorderLayout.CENTER``, dan ``BorderLayout.SOUTH``).

Ukuran jendela diatur menjadi 300x300 piksel dan ditampilkan ke layar dengan memanggil ``setVisible(true)``. Dengan demikian, program ini memberikan contoh tentang cara menggunakan tata letak yang kompleks untuk menata elemen-elemen GUI dalam Java AWT.

## I.8 MyContainer.java

### I.8.A. Source Code

Sebelum diperbaiki:

```
import javax.swing.*;

public class MyContainer {

    public static void main(String args[]) {

        JFrame container = new JFrame("Window Utama");

        container.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        container.setSize(300, 100);

        container.setLocation(200, 200);

        container.setVisible(true);

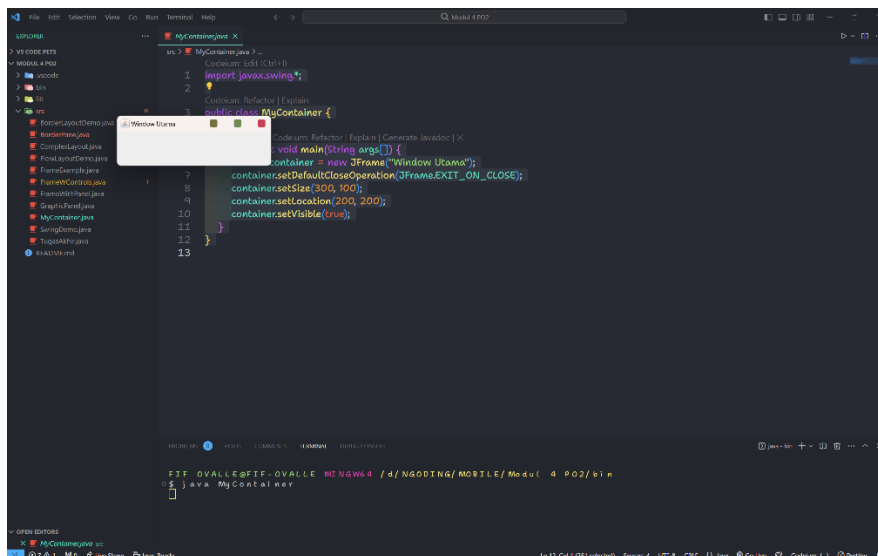
    }

}
```

Setelah diperbaiki

-

### I.8.B. Hasil



Gambar 1.15 Output Program MyContainer diperbaiki



=

*Gambar 1.16 Output Program MyContainer diperbaiki*

### **I.8.C. Analisa**

Program `MyContainer` adalah contoh sederhana dari pembuatan sebuah jendela menggunakan Swing, yaitu pustaka GUI bawaan dari Java yang lebih modern dan fleksibel daripada AWT. Kelas ini memiliki metode `main` yang merupakan titik masuk utama program.

Di dalam metode `main`, sebuah objek `JFrame` dengan judul "Window Utama" dibuat. Setelah itu, metode `setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE)` digunakan untuk menentukan perilaku aplikasi ketika jendela ditutup, dalam hal ini, aplikasi akan berhenti berjalan. Kemudian, ukuran jendela diatur menjadi 300x100 piksel menggunakan `setSize(300, 100)`, dan posisinya diatur menjadi (200, 200) piksel dari sudut kiri atas layar menggunakan `setLocation(200, 200)`. Terakhir, dengan memanggil `setVisible(true)`, jendela tersebut ditampilkan ke layar.

Dengan demikian, program ini memberikan contoh tentang cara membuat jendela sederhana menggunakan Swing dalam bahasa pemrograman Java.

## I.9 SwingDemo.java

### I.9.A. Source Code

Sebelum diperbaiki:

```
import java.awt.*;

import javax.swing.*;

public class SwingDemo {

    JFrame frame;

    JPanel panel;

    JButton[][] buttons;

    JLabel statusLabel;

    Container contentPane;

    public SwingDemo() {

    }

    void launchFrame() {

        frame = new JFrame("Tic Tac Toe");

        panel = new JPanel();

        buttons = new JButton[3][3];

        statusLabel = new JLabel("Player One's Turn");

        contentPane = frame.getContentPane();

        panel.setLayout(new GridLayout(3, 3));

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                buttons[i][j] = new JButton();

                buttons[i][j].setFont(new Font("Arial",
```

```

        Font.BOLD, 40));

        panel.add(buttons[i][j]);

    }

}

contentPane.add(panel, BorderLayout.CENTER);

JPanel statusPanel = new JPanel(new
FlowLayout(FlowLayout.CENTER));

statusPanel.add(statusLabel);

contentPane.add(statusPanel, BorderLayout.SOUTH);

frame.setSize(300, 300);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

frame.setVisible(true);

}

public static void main(String args[]) {

    SwingDemo ttt = new SwingDemo();

    ttt.launchFrame();

}

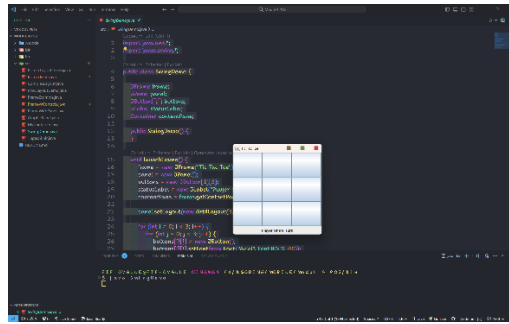
}

```

Setelah diperbaiki

-

### **I.9.B. Hasil**



Gambar 1.17 Output Program SwingDemo diperbaiki

Gambar 1.18 Output Program SwingDemo diperbaiki

### I.9.C. Analisa

Program `SwingDemo` adalah contoh implementasi permainan Tic Tac Toe menggunakan Java Swing. Kelas ini memiliki atribut untuk merepresentasikan jendela (`JFrame`), panel (`JPanel`), array tombol (`JButton[][]`), label status (`JLabel`), dan wadah konten (`Container`). Konstruktor `SwingDemo` tidak melakukan apa pun.

Metode `launchFrame` digunakan untuk menginisialisasi dan menampilkan jendela permainan. Dalam metode ini, sebuah objek `JFrame` dengan judul "Tic Tac Toe" dibuat. Sebuah panel dengan tata letak grid 3x3 juga dibuat, di mana setiap sel grid diisi dengan sebuah tombol yang direpresentasikan oleh array `buttons[][]`. Label status "Player One's Turn" juga ditambahkan ke panel.

Tombol-tombol diberi ukuran font yang besar dan tebal menggunakan `setFont(new Font("Arial", Font.BOLD, 40))`. Panel grid dan panel status kemudian ditambahkan ke konten utama jendela dengan menggunakan tata letak border (`BorderLayout`), di mana panel grid diletakkan di bagian tengah dan panel status diletakkan di bagian bawah.

Setelah mengatur seluruh tata letak, ukuran jendela diatur menjadi 300x300 piksel dan perilaku ketika jendela ditutup diatur agar aplikasi berhenti berjalan. Terakhir, jendela ditampilkan ke layar dengan memanggil `setVisible(true)`.

Dengan demikian, program ini memberikan contoh tentang cara membuat permainan Tic Tac Toe sederhana menggunakan Java Swing dengan tata letak yang disesuaikan.

## **BAB II. TUGAS PRAKTIKUM**

### **II.1 Tugas 4-1**

Sudah ada di hasil praktikum

### **II.2 Tugas 4-2**

#### **II.2.A. Source Code**

```
import java.awt.*;

public class GraphicPanel extends Panel {

    public GraphicPanel() {
        setBackground(Color.black);
    }

    public void paint(Graphics g) {
        g.setColor(new Color(0, 0, 255));
        g.setFont(new Font("Comic Sans MS", Font.BOLD, 20));
        g.drawString("Hello GUI World!", 30, 100);

        g.setColor(new Color(255, 0, 0));
        g.fillRect(30, 110, 150, 10);
    }

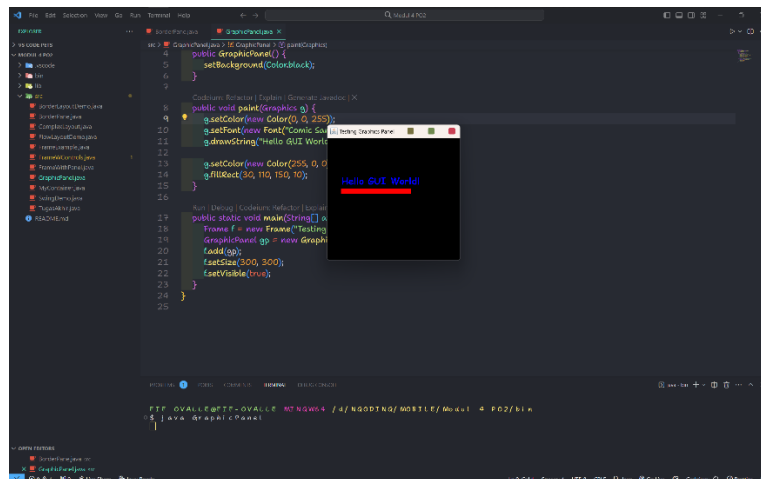
    public static void main(String[] args) {
        Frame f = new Frame("Testing Graphics Panel");
        GraphicPanel gp = new GraphicPanel();
        f.add(gp);
    }
}
```

```
f.setSize(300, 300);

f.setVisible(true);

}}
```

## II.2.B. Hasil



Gambar II.1 Output GraphicPanel Tugas A

## II.2.C. Analisa

Program `GraphicPanel` adalah contoh implementasi panel grafis dalam Java menggunakan AWT (Abstract Window Toolkit). Kelas `GraphicPanel` merupakan subkelas dari `Panel` yang bertujuan untuk menampilkan grafis di dalamnya. Dalam konstruktor, warna latar belakang panel diatur menjadi hitam.

Metode `paint(Graphics g)` digunakan untuk menggambar elemen-elemen grafis di dalam panel. Pertama, teks "Hello GUI World!" ditampilkan dengan warna biru (RGB: 0, 0, 255) dan ukuran font yang besar (Font "Comic Sans MS", Bold, ukuran 20), pada koordinat (30, 100). Kemudian, sebuah persegi panjang merah diisi pada koordinat (30, 110) dengan lebar 150 piksel dan tinggi 10 piksel.

Dalam metode `main`, sebuah jendela baru dengan judul "Testing Graphics Panel" dibuat. Objek `GraphicPanel` juga dibuat dan ditambahkan ke dalam jendela tersebut. Ukuran jendela diatur menjadi 300x300 piksel dan jendela ditampilkan ke layar.

Dengan demikian, program ini memberikan contoh tentang cara membuat dan menampilkan panel grafis sederhana dalam Java menggunakan AWT, yang menampilkan teks dan bentuk geometris dengan warna yang berbeda.

## II.3 Tugas 4-3

### II.3.A. Source Code

```
import java.awt.*;

public class FlowLayoutDemo extends Frame {

    public FlowLayoutDemo() {

    }

    public static void main(String args[]) {

        FlowLayoutDemo fld = new FlowLayoutDemo();

        fld.setLayout(new BorderLayout());

        Panel buttonPanel = new Panel(new

        FlowLayout(FlowLayout.CENTER));

        Button pressMeButton = new Button("PRESS ME");

        buttonPanel.add(pressMeButton);

        Button dontPressMeButton = new Button("DON'T PRESS ME");

        buttonPanel.add(dontPressMeButton);

        fld.add(buttonPanel, BorderLayout.CENTER);

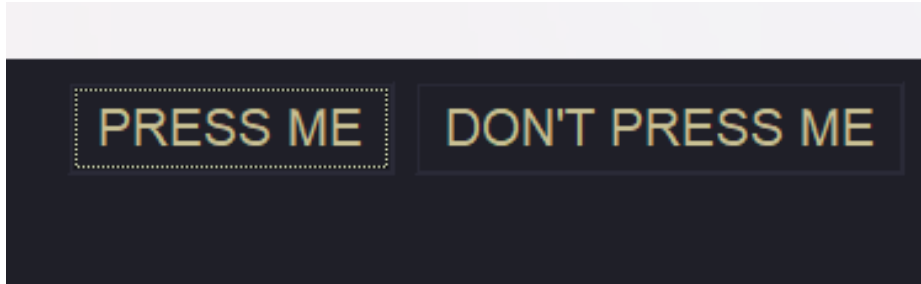
        fld.setSize(200, 150);

        fld.setVisible(true);

    }

}
```

### II.3.B. Hasil



*Gambar II.2 Output FlowLayoutDemo Tugas A*

### II.3.C. Analisa

Program `FlowLayoutDemo` adalah contoh penerapan tata letak yang berbeda dalam Java AWT (Abstract Window Toolkit). Kelas ini merupakan subkelas dari `Frame` yang bertujuan untuk menampilkan jendela dengan tombol-tombol di dalamnya. Dalam metode `main`, sebuah objek `FlowLayoutDemo` dibuat.

Kemudian, tata letak jendela diatur menggunakan `BorderLayout`, yang membagi jendela menjadi lima bagian: NORTH, SOUTH, EAST, WEST, dan CENTER.

Selanjutnya, dibuat sebuah panel tombol (`buttonPanel`) dengan tata letak `FlowLayout` yang terpusat (`FlowLayout.CENTER`). Dua tombol, "PRESS ME" dan "DON'T PRESS ME", ditambahkan ke dalam panel tersebut.

Panel tombol tersebut kemudian ditambahkan ke bagian tengah jendela menggunakan `BorderLayout.CENTER`.

Ukuran jendela diatur menjadi 200x150 piksel dan jendela ditampilkan ke layar.

Dengan demikian, program ini memberikan contoh tentang cara menggunakan tata letak yang berbeda, yaitu `FlowLayout` di dalam `BorderLayout`, untuk menata elemen-elemen GUI dalam Java AWT.



## II.4 Tugas 4-2

### II.4.A. Source Code

```
import java.awt.*;

public class FlowLayoutDemo extends Frame {

    public FlowLayoutDemo() {

    }

    public static void main(String args[]) {

        FlowLayoutDemo fld = new FlowLayoutDemo();

        fld.setLayout(new BorderLayout());

        Panel mainPanel = new Panel(new GridBagLayout());

        Panel buttonPanel = new Panel(new

        FlowLayout(FlowLayout.CENTER));

        Button pressMeButton = new Button("FILE");

        buttonPanel.add(pressMeButton);

        Button dontPressMeButton = new Button("HELP");

        buttonPanel.add(dontPressMeButton);

        Panel westPanel = new Panel(new

        FlowLayout(FlowLayout.CENTER));

        Button westButton = new Button("WEST");

        westPanel.add(westButton);

        Label workspaceLabel = new Label("WORK SPACE REGITION");

        westPanel.add(workspaceLabel);

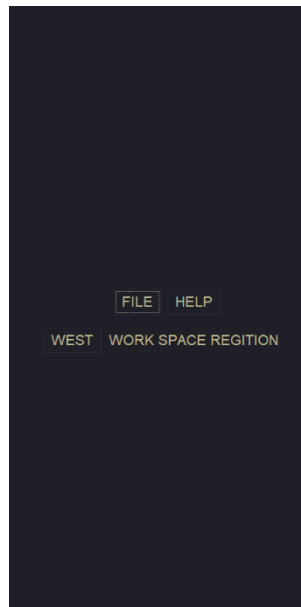
        GridBagConstraints gbc = new GridBagConstraints();

        gbc.gridx = 0;

        gbc.gridy = 0;
```

```
        mainPanel.add(buttonPanel, gbc);  
        gbc.gridy = 1;  
        mainPanel.add(westPanel, gbc);  
        fld.add(mainPanel, BorderLayout.CENTER);  
        fld.setSize(300, 200);  
        fld.setVisible(true);  
    }  
}
```

#### II.4.B. Hasil



*Gambar II.3 Output FlowLayoutDemo Tugas A*

## II.5 Tugas Akhir

### II.5.A. Source Code

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.IOException;
import java.net.URL;

public class TugasAkhir extends JFrame {

    private JLabel photoLabel;

    private JLabel titleLabel;

    private JLabel descriptionLabel;

    private ImageIcon loveIcon;

    private ImageIcon shareIcon;

    public TugasAkhir() {

        setTitle("Instagram");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setSize(600, 450);

        JPanel mainPanel = new JPanel(new GridBagLayout());

        GridBagConstraints gbc = new GridBagConstraints();

        gbc.fill = GridBagConstraints.BOTH;

        // Panel for card
```

```
        JPanel cardPanel = new JPanel(new BorderLayout());

cardPanel.setBorder(BorderFactory.createLineBorder(Color.GRAY, 1));


        JPanel photoPanel = new JPanel();
        photoLabel = new JLabel();
        photoPanel.add(photoLabel);
        cardPanel.add(photoPanel, BorderLayout.CENTER);


        JPanel textPanel = new JPanel(new BorderLayout());
        titleLabel = new JLabel("Title");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 16));
        textPanel.add(titleLabel, BorderLayout.NORTH);


        descriptionLabel = new JLabel("Description");
        textPanel.add(descriptionLabel, BorderLayout.CENTER);


        cardPanel.add(textPanel, BorderLayout.SOUTH);


        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.weighty = 0.9;
        mainPanel.add(cardPanel, gbc);


        JPanel iconPanel = new JPanel(new GridLayout(1, 2));
        gbc.gridy = 1;
```

```

gbc.weighty = 0.05;

gbc.insets = new Insets(10, 0, 10, 0);


loveIcon = new ImageIcon("love_icon.png");
shareIcon = new ImageIcon("share_icon.png");


JButton loveButton = new JButton(loveIcon);
loveButton.setBorderPainted(false);
loveButton.setFocusPainted(false);
loveButton.setContentAreaFilled(false);
loveButton.addMouseListener(new MouseAdapter() {

    @Override

    public void mouseEntered(MouseEvent e) {

        loveButton.setIcon(new

            ImageIcon("love_icon_hover.png"));

    }


    @Override

    public void mouseExited(MouseEvent e) {

        loveButton.setIcon(loveIcon);

    }


    @Override

    public void mouseClicked(MouseEvent e) {

    }

});

```

```

        JButton shareButton = new JButton(shareIcon);
        shareButton.setBorderPainted(false);
        shareButton.setFocusPainted(false);
        shareButton.setContentAreaFilled(false);
        shareButton.addMouseListener(new MouseAdapter() {

            @Override

            public void mouseEntered(MouseEvent e) {

                shareButton.setIcon(new

                    ImageIcon("share_icon_hover.png"));

            }

            @Override

            public void mouseExited(MouseEvent e) {

                shareButton.setIcon(shareIcon);

            }

            @Override

            public void mouseClicked(MouseEvent e) {

            }

        });

        iconPanel.add(loveButton);
        iconPanel.add(shareButton);
        mainPanel.add(iconPanel, gbc);

```

```

        // Panel for comment and like count
        JPanel infoPanel = new JPanel(new
        FlowLayout(FlowLayout.LEFT));
        JLabel likeCountLabel = new JLabel("100 likes");
        JLabel commentLabel = new JLabel("View all 20
        comments");
        infoPanel.add(likeCountLabel);
        infoPanel.add(commentLabel);
        gbc.gridxy = 2;
        gbc.weighty = 0.05;
        mainPanel.add(infoPanel, gbc);

        add(mainPanel);
        displayRandomPhoto();
        setVisible(true);
    }

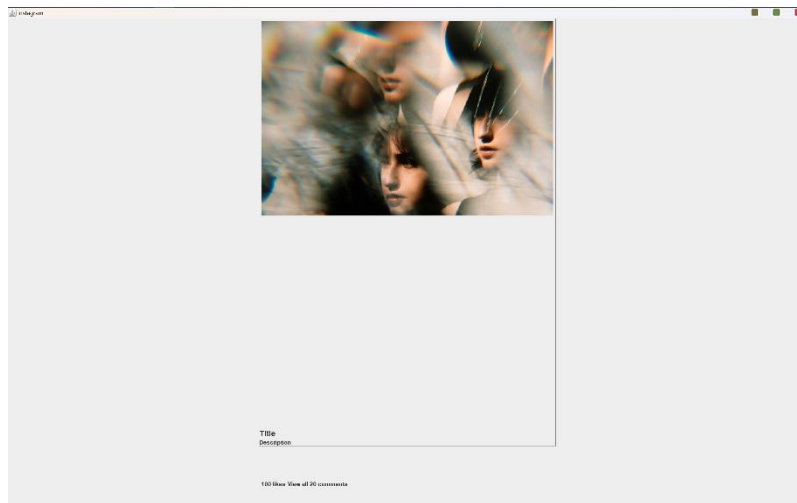
    private void displayRandomPhoto() {
        try {
            URL url = new
            URL("https://source.unsplash.com/random/600x400");
            ImageIcon imageIcon = new ImageIcon(url);
            photoLabel.setIcon(imageIcon);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

```

```
}

    public static void main(String[] args) {
        SwingUtilities.invokeLater(TugasAakhir::new);
    }
}
```

### II.5.B. Hasil



*Gambar II.4 Output Program Instagram diperbaiki*



### **BAB III. KESIMPULAN**

Dari praktikum tentang berbagai contoh program Java yang menggunakan AWT dan Swing untuk membuat antarmuka pengguna grafis (GUI), kita dapat menyimpulkan bahwa kedua pustaka tersebut menyediakan beragam komponen dan tata letak untuk membangun aplikasi GUI dalam Java. Program-program tersebut mengilustrasikan penggunaan berbagai elemen seperti jendela, tombol, panel, kotak teks, dan label, serta berbagai jenis tata letak seperti FlowLayout, BorderLayout, dan GridLayout. Dengan menggunakan AWT atau Swing, pengembang dapat membuat antarmuka pengguna yang intuitif dan interaktif dengan relatif mudah. AWT menyediakan pustaka GUI bawaan yang lebih sederhana dan lebih rendah tingkat abstraksi, sementara Swing menawarkan kontrol yang lebih lengkap dan fleksibel. Meskipun demikian, keduanya memungkinkan untuk membuat aplikasi GUI yang berfungsi dengan baik dalam lingkungan Java.