

**LAPORAN PRATIKUM PEKAN 8**  
**PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN**  
“Graphical User Interface”



Dosen Pengampu:  
DR. Wahyudi. .S.T.M.T

Asisten Lab:  
Jovantri Immanuel Gulo

Disusun Oleh:  
Marcello Bayu Denar Widjaksono  
2511532011

Fakultas Teknologi Informasi  
Departemen Informatika  
Universitas Andalas  
2025

## **KATA PENGANTAR**

Puji dan syukur senantiasa penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan ini guna memenuhi laporan pratikum mata kuliah Algoritma Pemrograman, dengan judul: “Graphical User Interface”.

Pada kesempatan ini, Penulis mengucapkan terima kasih Bapak DR. Wahyudi. .S.T.M.T yang telah memberikan tugas laporan pratikum. Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah mendukung serta membantu penulis dalam penyelesaian laporan pratikum Algoritma Pemrograman.

Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang penulis miliki. Oleh karena itu, penulis mengharapkan segala bentuk saran dan masukan serta kritik yang membangun dari berbagai pihak.

Akhir kata, penulis berharap Laporan ini dapat memberikan manfaat bagi perkembangan dunia pendidikan.

## **DAFTAR ISI**

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Manfaat Pratikum.....	1
BAB II PEMBAHASAN.....	2
2.1 Teori.....	2
2.2 Kode Program.....	3
BAB III KESIMPULAN.....	8
3.1 Kesimpulan.....	8
3.2 Saran.....	8
DAFTAR PUSTAKA.....	9

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Dalam mengembangkan sebuah software, pemisahan antara logika data (model/business logic) dan antarmuka (UI) merupakan prinsip yang fundamental. Java sebagai bahasa pemrograman berorientasi objek (OOP) menyediakan berbagai *tools* untuk membangun aplikasi, salah satunya adalah library Swing untuk pengembangan aplikasi desktop (GUI).

Pratikum ini adalah implementasi aplikasi Operasi Aritmatika menggunakan Java Swing. Aplikasi ini bertujuan untuk melakukan operasi aritmatika (operasinya dipilih oleh user) berdasarkan bilangan yang diinputkan user.

#### **1.2 Tujuan**

1. Menjelaskan konsep Graphical User Interface (GUI).
2. Menjelaskan dasar penggunaan Java Swing.
3. Menjelaskan mekanisme *event handling* ( respon interaksi pengguna saat mengklik tombol)

#### **1.3 Manfaat Pratikum**

1. Memahami dan mengimplementasikan konsep Graphical User Interface (GUI).
2. Memahami penggunaan java swing.
3. Mampu memahami dan mengimplementasikan mekanisme *event handling*

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Teori**

##### **A. Java Swing**

Java Swing adalah widget toolkit GUI (Graphical User Interface). Ini adalah bagian dari java Foundation class (JFC). Swing menyediakan komponen yang membangun antarmuka (UI) desktop. Komponen yang digunakan dalam praktik ini meliputi:

- JFrame: Jendela utama (container induk) aplikasi.
- JLabel: Komponen untuk menampilkan teks atau ikon.
- JTextField : Komponen input teks (satu baris).
- JComboBox : Komponen yang menyediakan daftar pilihan dalam bentuk drop-down box.
- JButton : Komponen tombol standar yang bisa diklik user.

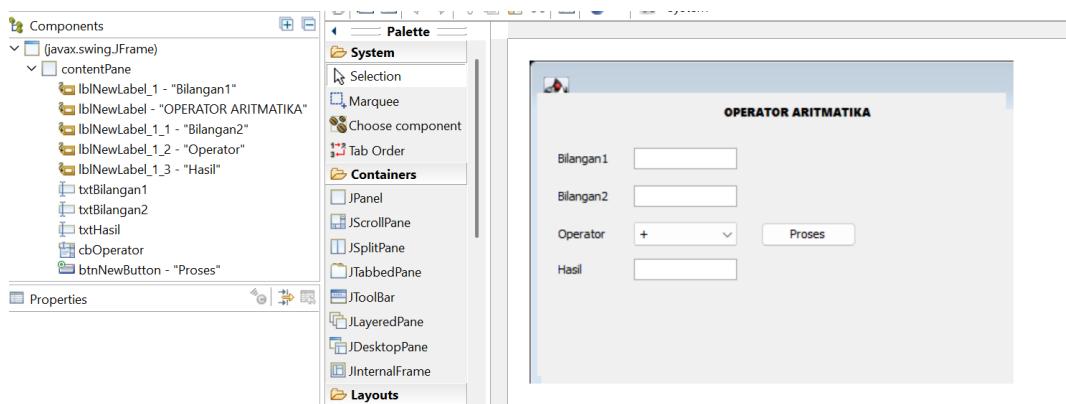
##### **B. Event Handling (Action Listener)**

Agar komponen swing dapat merespon interaksi user (seperti user mengklik tombol) Java menggunakan model delegasi acara (event delegation model). Sebuah Listener (ActionListener) didaftarkan ke sebuah komponen (misal JButton). Ketika event terjadi, maka metode yang relevan dalam listener (actionPerformed) akan dilaksanakan.

## 2.2 Kode Pemrograman

Saya akan menjelaskan penggunaan Graphical User Interface (GUI) dalam aplikasi desktop operator Aritmatika dan logika pemrograman untuk operasi aritmatika.

Tampilan Antarmuka(UI) Aplikasi Operator Aritmatika:



Kode untuk membuat komponen antarmuka(UI):

```

public OperatorAritmatikaGUI_2511532011() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel_1 = new JLabel("Bilangan1");
    lblNewLabel_1.setBounds(20, 54, 60, 14);
    contentPane.add(lblNewLabel_1);

    JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
    lblNewLabel.setBounds(174, 10, 142, 16);
    lblNewLabel.setFont(new Font("Segoe UI Black", Font.PLAIN, 11));
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1_1 = new JLabel("Bilangan2");
    lblNewLabel_1_1.setBounds(20, 89, 60, 14);
    contentPane.add(lblNewLabel_1_1);

    JLabel lblNewLabel_1_2 = new JLabel("Operator");
    lblNewLabel_1_2.setBounds(20, 124, 49, 14);
    contentPane.add(lblNewLabel_1_2);
}

```

```

JLabel lblNewLabel_1_3 = new JLabel("Hasil");
lblNewLabel_1_3.setBounds(20, 157, 49, 14);
contentPane.add(lblNewLabel_1_3);

txtBilangan1 = new JTextField();
txtBilangan1.setBounds(90, 51, 96, 20);
contentPane.add(txtBilangan1);
txtBilangan1.setColumns(10);
|
txtBilangan2 = new JTextField();
txtBilangan2.setColumns(10);
txtBilangan2.setBounds(90, 86, 96, 20);
contentPane.add(txtBilangan2);

txtHasil = new JTextField();
txtHasil.setColumns(10);
txtHasil.setBounds(90, 154, 96, 20);
contentPane.add(txtHasil);

JComboBox cbOperator = new JComboBox();
cbOperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}));
cbOperator.setBounds(90, 120, 96, 22);
contentPane.add(cbOperator);

```

## Method/Fungsi

```

private static final long serialVersionUID = 1L;
private JPanel contentPane;
private JTextField txtBilangan1;
private JTextField txtBilangan2;
private JTextField txtHasil;

private void pesanPeringatan(String pesan) {
    JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
}

private void pesanError(String pesan) {
    JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
}

```

### Penjelasan:

- Method pesanPeringatan() digunakan untuk memberitahu peringatan kepada user (jika ada suatu hal yang tidak sesuai dilakukan oleh user), yang mana cara menggunakan method ini cukup kita panggil pesanPeringatan("masukkan pesan yang ingin disampaikan").
- Method pesanError() digunakan untuk memberitahukan adanya Error kepada, cara menggunakan method ini cukup kita panggil pesanError("masukkan pesan yang ingin disampaikan")

Kode logika pada button:

```

JButton btnNewButton = new JButton("Proses");
btnNewButton.addActionListener(new ActionListener() {
    int hasil;
    public void actionPerformed(ActionEvent e) {
        if (txtBilangan1.getText().trim().isEmpty()) {
            pesanPeringatan("Silahkan Input Bilangan 1");
        } else if (txtBilangan2.getText().trim().isEmpty()) {
            pesanPeringatan("Silahkan Input Bilangan 2");
        } else if (txtBilangan2.getText().trim() == "0") {
            pesanPeringatan("Silahkan Input Bilangan selain 0");
        } else {
            try {
                int a = Integer.parseInt(txtBilangan1.getText());
                int b = Integer.parseInt(txtBilangan2.getText());
                int c = cbOperator.getSelectedIndex();

                if (c == 0) {
                    hasil = a+b;
                } else if (c==1) {
                    hasil = a-b;
                } else if (c==2) {
                    hasil = a*b;
                } else if (c==3) {
                    hasil = a/b;
                } else if (c==4) {
                    hasil = a%b;
                }

                txtHasil.setText(String.valueOf(hasil));
            } catch (NumberFormatException ex) {
                pesanError("Bilangan 1 Atau Bilangan 2 Harus Angka");
            }
        }
    });
});
```

btnNewButton.setBounds(208, 120, 89, 23);
contentPane.add(btnNewButton);

Penjelasan:

- Kita membuat objek btnNewButton (kita sesuaikan dengan nama komponen button yang kita buat yaitu “proses”)

- Kemudian kita tambahkan fungsi addActionListener ke btnNewButton (fungsi addActionListener akan memberikan respon (berupa mengeksekusi kode) saat user mencoba untuk berinteraksi (klik dll) dengan komponen).
- Inisialisasi var hasil dengan tipe int.
- Kita membuat method actionPerformed yang akan dihubungkan dengan komponen button.
- Sebelumnya kita membuat semacam kotak input (bilangan) yang memiliki nama txtBilangan1 dan txtBilangan2, disini kita akan melakukan pengecekan terlebih dahulu.
- Periksa value txtBilangan1, jika value txtBilangan1==kosong, maka method pesanPeringatan("masukkan bilangan1") akan dijalankan.
- Kemudian periksa value txtBilangan2, jika value txtBilangan2==kosong, maka method pesanPeringatan("masukkan bilangan2") akan dijalankan.
- Kemudian periksa value txtBilangan2, jika value txtBilangan2==0, maka method pesanPeringatan("masukkan bilangan2 selain 0") akan dijalankan.
- Jika dari ketiga kondisi di atas tadi tidak ada terpenuhi sama sekali maka kita akan mencoba menjalankan program sekali saja (meskipun ada kemungkinan error untuk terjadi sudah diketahui):
  - a. Kita melakukan inisialisasi var a dengan tipe int, diamana value a = input yang diinputkan pada kolom txtBilangan1.
  - b. Kita melakukan inisialisasi var b dengan tipe int, diamana value b = input yang diinputkan pada kolom txtBilangan2.
  - c. Kita melakukan inisialisasi var c (operasi aritmatika yang kita pilih dan ditempakan di jcomboBox).

Kemudian membuat kondisi berdasarkan index yang dipilih pada c:

- a. Jika c==0 maka lakukan a+b
- b. Jika c==1 maka lakukan a-b
- c. Jika c==2 maka lakukan a\*b
- d. Jika c==3 maka lakukan a/b

e. Jika  $c==4$  maka lakukan  $a\%b$

- Hasil operasi aritmatika tersebut akan disimpan kedalam variabel hasil yang kemudian akan ditampilkan di komponen textfield bernama “hasil”.
- Kita menampilkan pesan error jika input yang dimasukkan tipenya selain integer

## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Konsep Graphical User Interface memudahkan kita dalam membuat tampilan antarmuka (UI) aplikasi desktop sehingga memudahkan user untuk berinteraksi.

#### **3.2 Saran**

Saya harap kita sebagai mahasiswa informatika harus memahami mekanisme eventListener sehingga dapat memudahkan kita untuk membuat logika antarmukanya.

## **DAFTAR PUSTAKA**

- [1] Oracle, “Trail: Creating a GUI with Swing”, The Java™ Tutorials: Creating Graphical User Interfaces, Oracle Documentation, 2023 [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/uiswing/> [Diakses: 18-November-2025].
- [2] Oracle, “Trail: Creating a JavaFX GUI”, The Java™ Tutorials: Creating Graphical User Interfaces, Oracle Documentation, 2023 [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/javafx/> [Diakses: 18-November-2025].
- [3] Oracle, “Using JavaFX UI Controls: About This Tutorial”, JavaFX Documentation, Oracle Documentation, 2013 [Daring]. Tersedia pada: [https://docs.oracle.com/javafx/2/ui\\_controls/jfxpub-ui\\_controls.htm](https://docs.oracle.com/javafx/2/ui_controls/jfxpub-ui_controls.htm) [Diakses: 18-November-2025].

