

**LAPORAN PRAKTIKUM PEKAN 8**  
**ALGORITMA DAN PEMROGRAMAN**  
**PEMROGRAMAN GUI**

Disusun Oleh:

Ihsanul Zaky EL-Muhammady

NIM:2511533001

Dosen Pengampu: Dr.Wahyudi S.T , M.T

Asisten Laboratorium: Aufan Taufiqurrahman



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**  
**PADANG 2025**

## **KATA PENGANTAR**

Dengan penuh rasa syukur, saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa, yang telah memberikan kemudahan dan kesempatan sehingga laporan praktikum ini dapat selesai dengan baik.

Saya juga ingin menyampaikan penghargaan yang setinggi-tingginya kepada Dr. Wahyudi, S.T., M.T. selaku dosen pengampu Mata Kuliah Algoritma dan Pemrograman, yang telah memberikan bimbingan dan motivasi selama perkuliahan. Ucapan terima kasih juga saya sampaikan kepada Uda Aufan Tafiqurrahman, asisten laboratorium, yang dengan sabar dan penuh perhatian telah membimbing kami dalam praktikum ini.

Laporan ini disusun sebagai salah satu tugas dalam Mata Kuliah Pemrograman Dasar yang membahas mengenai konsep dasar GUI Pada java dalam pemrograman.

Saya menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, segala masukan dan saran yang membangun sangat saya harapkan untuk perbaikan di masa mendatang.

Padang, 2025

IHSANUL ZAKY EL-MUHAMMADY

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	
<b>DAFTAR ISI.....</b>	
<b>BAB I PENDAHULUAN .....</b>	
1.1 Latar Belakang .....	
1.2 Tujuan.....	
1.3 Manfaat.....	
<b>BAB II PEMBAHASAN .....</b>	
2.1 Langkah Pengerjaan .....	
<b>BAB III KESIMPULAN .....</b>	
3.1 Kesimpulan .....	
3.2 Saran.....	
<b>DAFTAR PUSTAKA.....</b>	

# **BAB I**

## **PENDALULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi perangkat lunak mendorong kebutuhan akan aplikasi yang mampu memberikan kemudahan dan kenyamanan dalam penggunaan. Salah satu pendekatan yang banyak digunakan untuk mencapai hal tersebut adalah penerapan Graphical User Interface (GUI). Melalui GUI, pengguna dapat berinteraksi dengan aplikasi secara visual tanpa harus memahami perintah berbasis teks. Elemen-elemen seperti tombol, kolom input, dan menu memberikan pengalaman penggunaan yang lebih langsung dan responsif.

Java merupakan salah satu bahasa pemrograman yang mendukung pembangunan antarmuka grafis dengan stabil dan konsisten. Keunggulan utama Java terletak pada sifatnya yang tidak bergantung pada platform tertentu, sehingga aplikasi dapat dijalankan di berbagai sistem operasi selama tersedia Java Virtual Machine. Selain itu, Java menyediakan pustaka khusus seperti Swing yang memungkinkan perancangan antarmuka secara fleksibel. Kemampuan ini menjadikan Java relevan untuk dipelajari, terutama dalam pengembangan aplikasi desktop yang memerlukan tampilan interaktif.

### **1.2 Tujuan**

1. Memahami prinsip dasar pembuatan antarmuka grafis dan cara kerja pemrograman berbasis event pada Java.
2. Mengenali komponen-komponen utama Swing serta memahami fungsi dan penerapannya dalam aplikasi.
3. Mampu merancang dan membangun aplikasi GUI sederhana yang dapat menjalankan operasi aritmatika secara tepat.

### **1.3 Manfaat**

1. Memberikan pengalaman langsung dalam merancang tampilan dan alur kerja aplikasi desktop.
2. Meningkatkan kemampuan dalam mengelola komponen dan event dalam lingkungan GUI Java.
3. Menjadi bekal awal untuk mengembangkan aplikasi dengan antarmuka yang lebih kompleks dan profesional.

## **BAB II**

### **PEMBAHASAN**

#### 2.1 Langkah pengerjaan

##### A. Struktur Program

###### 1. Kelas Utama dan Komponen

Program dibangun dengan membuat kelas `OperatorAritmatikaGUI_2511532029` yang merupakan turunan dari `JFrame`. Struktur antarmuka terdiri atas:

- ❖ `ControlPane`, panel yang digunakan sebagai wadah seluruh komponen.
- ❖ `txtBil1` dan `txtBil2`, tempat input angka dari pengguna.
- ❖ `txtHasil`, kolom untuk menampilkan hasil operasi.
- ❖ `JComboBox`, komponen yang berisi pilihan operator aritmatika.

###### 2. Fungsi Pendukung

Untuk memberikan umpan balik kepada pengguna, disediakan dua method:

- ❖ `pesanPeringatan(String pesan)`, menampilkan pesan peringatan.
- ❖ `pesanError(String pesan)`, menampilkan pesan kesalahan.

###### 3. Eksekusi Utama Program

Method `main()` menjalankan antarmuka menggunakan `EventQueue.invokeLater()`, memastikan proses GUI berada pada thread yang sesuai. Di dalamnya dibuat objek frame yang kemudian ditampilkan.

##### B. Perancangan GUI Melalui Konstruktor

###### 1. Pengaturan Jendela

Pada konstruktor, jendela program disiapkan dengan langkah berikut:

- ❖ Menentukan judul aplikasi sebagai “OperatorAritmatika”.
- ❖ Mengatur agar program berhenti ketika jendela ditutup.
- ❖ Memberikan ukuran awal serta pengaturan visual lain yang diperlukan.

###### 2. Penambahan Komponen

Beberapa elemen ditambahkan untuk membentuk antarmuka kalkulator:

- ❖ Label judul serta label penanda seperti Bilangan 1, Bilangan 2, Operator, dan Hasil.
- ❖ Kolom input dan output nilai bilangan.

- ❖ ComboBox berisi operator dasar, yaitu: penjumlahan, pengurangan, perkalian, pembagian, dan modulus.
- ❖ Tombol Hitung sebagai pemicu proses aritmatika.

### C. Pemrosesan Event dan Logika Program

#### 1. Pemeriksaan Input

Program memastikan kolom input tidak dibiarkan kosong. Pemeriksaan dilakukan menggunakan fungsi `trim().isEmpty()`. Bila terdapat kolom yang belum terisi, sistem memberikan peringatan agar pengguna melengkapi datanya.

#### 2. Penanganan Kesalahan Input

Jika pengguna memasukkan nilai selain angka, program akan mendeteksi hal tersebut melalui blok try-catch. Kesalahan tipe `NumberFormatException` ditangani dengan menampilkan dialog error.

#### 3. Pelaksanaan Operasi Aritmatika

- Nilai input dikonversi dari teks menjadi bilangan menggunakan `Integer.parseInt()`.
- Operator dipilih berdasarkan indeks pada ComboBox:
  - 0 → Penjumlahan ( $a + b$ )
  - 1 → Pengurangan ( $a - b$ )
  - 2 → Perkalian ( $a * b$ )
  - 3 → Pembagian ( $a / b$ )
  - 4 → Modulus ( $a \% b$ )
- Perhitungan dilakukan sesuai pilihan pengguna.

#### 4. Penampilan Hasil

Setelah operasi selesai, hasil diubah menjadi string menggunakan `String.valueOf(hasil)` dan ditampilkan ke dalam `txtHasil`.

#### Kode Program

```

1 package Pekand_2511533001;
2
3 import java.awt.EventQueue;
4
5 public class OperatorAritmatikaGUI_2511533001 extends JFrame {
6
7     private static final long serialVersionUID = 1L;
8     private JPanel contentPane;
9     private JTextField txtB1;
10    private JTextField txtB2;
11    private JTextField txtHasil;
12
13    private void pesanPeringatan(String pesan) {
14        JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
15    }
16    private void pesanError(String pesan) {
17        JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
18    }
19
20    /**
21     * Launch the application.
22     */
23    public static void main(String[] args) {
24        EventQueue.invokeLater(new Runnable() {
25            public void run() {
26                try {
27                    OperatorAritmatikaGUI_2511533001 frame = new OperatorAritmatikaGUI_2511533001();
28                    frame.setVisible(true);
29                } catch (Exception e) {
30                    e.printStackTrace();
31                }
32            }
33        });
34    }
35
36}

```

Gambar 2.1

```

42@         } catch (Exception e) {
43             e.printStackTrace();
44         }
45     }
46     });
47 }
48
49@ /**
50 * Create the frame.
51 */
52@ public OperatorAritmatikaGUI_2511533001() {
53     setTitle("OperatorAritmatika");
54     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
55     setBounds(100, 100, 450, 300);
56     contentPane = new JPanel();
57     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
58     setContentPane(contentPane);
59     contentPane.setLayout(null);
60
61     JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
62     lblNewLabel.setFont(new Font("Segoe UI Light", Font.BOLD, 15));
63     lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
64     lblNewLabel.setBounds(109, 11, 176, 14);
65     contentPane.add(lblNewLabel);
66
67     JLabel lblNewLabel_1 = new JLabel("Bilangan 1");
68     lblNewLabel_1.setBounds(10, 43, 75, 14);

```

Gambar 2.2

```

70
71     JLabel lblNewLabel_1_1 = new JLabel("Bilangan 2");
72     lblNewLabel_1_1.setBounds(10, 73, 75, 14);
73     contentPane.add(lblNewLabel_1_1);
74
75     JLabel lblNewLabel_1_2 = new JLabel("Operator");
76     lblNewLabel_1_2.setBounds(10, 109, 75, 14);
77     contentPane.add(lblNewLabel_1_2);
78
79     JLabel lblNewLabel_1_3 = new JLabel("Hasil");
80     lblNewLabel_1_3.setBounds(10, 160, 75, 14);
81     contentPane.add(lblNewLabel_1_3);
82
83     txtBil1 = new JTextField();
84     txtBil1.setHorizontalAlignment(SwingConstants.CENTER);
85     txtBil1.setBounds(95, 40, 86, 20);
86     contentPane.add(txtBil1);
87     txtBil1.setColumns(10);
88
89     txtBil2 = new JTextField();
90     txtBil2.setHorizontalAlignment(SwingConstants.CENTER);
91     txtBil2.setColumns(10);
92     txtBil2.setBounds(95, 70, 86, 20);
93     contentPane.add(txtBil2);
94
95     txtHasil = new JTextField();
96     txtHasil.setHorizontalAlignment(SwingConstants.CENTER);

```

Gambar 2.3

Output yang dihasilkan:

Penjumlahan

The screenshot shows a Java Swing window titled "OperatorAritmatika". Inside, there's a title "OPERATOR ARITMATIKA". Below it, there are two input fields labeled "Bilangan 1" and "Bilangan 2" with values 19 and 25 respectively. There's a dropdown menu for "Operator" showing a plus sign (+), and a "Hitung" button. At the bottom, there's a "Hasil" field showing the result 44.

Gambar 2.4

Pengurangan

The screenshot shows the same Java Swing window. The "Bilangan 1" field now contains 12 and "Bilangan 2" contains 18. The "Operator" dropdown now shows a minus sign (-). The "Hitung" button is still present. The "asil" (likely Hasil) field now shows the result -6.

Gambar 2.5

## Perkalian

OPERATOR ARITMATIKA

Bilangan 1

Bilangan 2

Operator

Hasil

Gambar 2.6

## Pembagian

OPERATOR ARITMATIKA

Bilangan 1

Bilangan 2

Operator

Hasil

Gambar 2.7

## Sisa Bagi

OPERATOR ARITMATIKA

Bilangan 1

Bilangan 2

Operator

Hasil

Gambar 2.8



## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Kegiatan praktikum ini memberikan pemahaman mengenai cara membangun antarmuka grafis menggunakan Java Swing. Melalui implementasi kalkulator sederhana, terlihat bahwa komponen seperti JFrame, JComboBox, JTextField, serta JButton dapat digunakan untuk membangun antarmuka yang dapat berinteraksi dengan pengguna. Proses event-driven berhasil diterapkan melalui penggunaan ActionListener, sehingga setiap tindakan pengguna dapat memicu proses perhitungan.

Sistem juga dilengkapi dengan mekanisme pemeriksaan input serta penanganan error untuk memastikan program tetap berjalan stabil meski pengguna melakukan kesalahan. Munculnya dialog informasi, peringatan, dan kesalahan melalui JOptionPane membantu membuat aplikasi lebih komunikatif. Secara keseluruhan, praktikum ini memberikan pengalaman yang jelas mengenai bagaimana mengelola komponen, event, dan logika pemrograman dalam pembuatan aplikasi GUI berbasis Java.

#### **3.2 Saran**

Sebaiknya dosen menyelenggarakan sesi pra-praktikum di kelas sebagai pengantar materi, agar mahasiswa memperoleh pemahaman awal yang lebih baik. Dengan demikian, potensi kepanikan atau kesalahan selama praktikum dapat diminimalkan. Disarankan agar materi praktikum dibagikan terlebih dahulu melalui platform iLearn, sehingga mahasiswa memiliki waktu yang cukup untuk mempelajari dan mempersiapkan diri sebelum pelaksanaan praktikum.

## DAFTAR PUSTAKA

1. Deitel, P. & Deitel, H. (2015). *Java: How to Program*. Prentice Hall.
2. Oracle. (2024). *Java Platform, Standard Edition Documentation*. Oracle Corporation.
3. Schildt, H. (2019). *Java: The Complete Reference*. McGraw-Hill.