

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN**  
**“laporan Praktikum Pekan 8”**

Disusun Oleh:

Faiz Fikri Satria

2511533026

Dosen Pengampu : Dr. Wahyudi, S.T, M.T.  
Asisten Praktikum : Rahmad Dwirizki Olders



DAPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
2025

## KATA PENGANTAR

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga dapat menyelesaikan laporan praktikum ini dengan baik. Tidak lupa juga saya ucapkan terima kasih kepada bapak Wahyudi. Dr. S.T.M.T sebagai dosen pembimbing dan Rahmad Dwirizki Olders yang telah membantu dalam pelaksanaan praktikum ini. Laporan ini saya susun untuk memenuhi tugas praktikum Algoritma dan Pemrograman dari praktikum pertemuan ke delapan mengenai pemahaman dasar pemrograman bahasa Java menggunakan IDE Eclipse. Praktikum ini bertema Penggunaan Jframe melalui GUI Eclipse. Saya berharap laporan ini dapat memberikan manfaat bagi pembaca dalam memahami konsep dasar pemrograman Java.

Laporan ini dibuat dengan harapan dapat memberikan pemahaman dasar mengenai penggunaan Bahasa pemrograman Java, khususnya mengenai hal struktur dasar program dan fungsi.

Saya menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu, kritik dan saran sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Padang 23 November, 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR .....	ii
DAFTAR ISI .....	iii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan .....	1
1.3 Manfaat .....	1
BAB II PEMBAHASAN .....	2
<b>2.1 Aplikasi Kalkulator Aritmatika Sederhana Menggunakan JFrame dan Swing .....</b>	<b>2</b>
2.1.1 Langkah-langkah.....	2
2.1.2 Contoh Pemrograman .....	4
2.1.3 Hasil Output .....	5
2.1.4 Analisis .....	6
2.1.5 Teori Mengenai Kode .....	6
BAB III KESIMPULAN .....	7
DAFTAR PUSTAKA .....	8

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Tujuan**

1. Memahami struktur dasar pembuatan aplikasi GUI menggunakan JFrame dan Swing.
2. Mampu menggunakan komponen JTextField, JComboBox, JButton, dan JOptionPane.
3. Mengimplementasikan event handling dengan ActionListener pada tombol.
4. Melakukan validasi input dan penanganan exception (NumberFormatException).
5. Membuat aplikasi kalkulator aritmatika (+, -, ×, ÷, %) yang interaktif dan user-friendly.

#### **1.2 Manfaat**

1. Mahasiswa mampu beralih dari pemrograman console ke pemrograman berbasis GUI.
2. Menguasai dasar-dasar Java Swing sebagai fondasi aplikasi desktop.
3. Melatih logika pemrograman sekaligus desain antarmuka dan penanganan kesalahan pengguna.
4. Menjadi bekal untuk membuat aplikasi edukasi, tools produktivitas, atau sistem informasi berbasis desktop.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Aplikasi Kalkulator Aritmatika Sederhana Menggunakan JFrame dan Swing**

Kode ini merupakan implementasi lengkap aplikasi kalkulator aritmatika sederhana berbasis GUI yang dibangun menggunakan Java Swing. Aplikasi ini memiliki dua input angka (JTextField), satu JComboBox untuk memilih operator (+, -, ×, ÷, %), satu tombol "Hitung", dan satu JTextField read-only untuk menampilkan hasil. Ketika tombol "Hitung" ditekan, program melakukan validasi apakah field angka kosong, kemudian mencoba mengonversi input menjadi integer. Jika berhasil, operasi aritmatika dilakukan sesuai operator yang dipilih; jika gagal (input bukan angka), muncul JOptionPane berupa pesan error. Aplikasi juga menangani pembagian dengan benar (menggunakan integer division) dan modulus (%). Penggunaan method terpisah pesanPeringatan() dan pesanError() menunjukkan penerapan modularitas kode, sementara EventQueue.invokeLater() memastikan GUI berjalan pada thread yang tepat (Event Dispatch Thread). Secara keseluruhan, kode ini menjadi contoh terbaik untuk mempelajari integrasi antara komponen GUI, event handling, validasi input, exception handling, dan logika aritmatika dalam satu aplikasi fungsional.

##### **2.1.1 Langkah-langkah**

1. Buat project Java baru di Eclipse dengan package  
pekan\_delapan\_2511533026.
2. Buat class OperatorAritmatikaGUI\_2511533026 yang extends JFrame.
3. Buat variabel instance untuk JTextField dan JComboBox.
4. Buat dua method bantu: pesanPeringatan() dan pesanError() menggunakan JOptionPane.
5. Atur JFrame: ukuran, title tersirat dari class, close operation, dan contentPanel dengan null layout.
6. Tambahkan JLabel untuk judul dan keterangan field.
7. Tambahkan dua JTextField untuk angka 1 dan angka 2, satu JTextField hasil (editable = false), dan satu JComboBox berisi operator.

8. Tambahkan JButton "Hitung" dengan ActionListener.

9. Di dalam actionPerformed:

- Cek apakah input angka kosong → tampilkan peringatan
- Gunakan try-catch untuk konversi string ke integer
- Ambil indeks operator yang dipilih dan lakukan perhitungan sesuai indeks
- Tampilkan hasil di txtHasil
- Tangkap NumberFormatException → tampilkan pesan error

10. Jalankan melalui EventQueue.invokeLater() di method main.

11. Uji aplikasi dengan berbagai input valid dan tidak valid.

## 2.1.2 Contoh Pemrograman

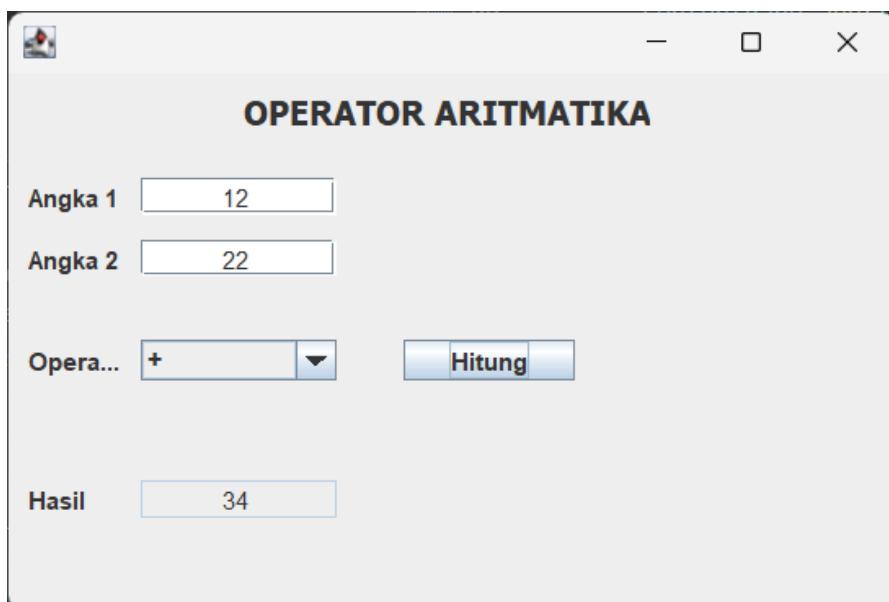
```
1 package pekan_delapan_2511533026;
2
3 import java.awt.BorderLayout;
4
5 public class OperatorAritmatikaGUI_2511533026 extends JFrame {
6
7     private static final long serialVersionUID = 1L;
8     private JPanel contentPane;
9     private JTextField txtAngka1;
10    private JTextField txtAngka2;
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     * Launch the application.
21     */
22    public static void main(String[] args) {
23        EventQueue.invokeLater(new Runnable() {
24            public void run() {
25                try {
26                    OperatorAritmatikaGUI_2511533026 frame = new OperatorAritmatikaGUI_2511533026();
27                    frame.setVisible(true);
28                } catch (Exception e) {
29                    e.printStackTrace();
30                }
31            }
32        });
33    }
34
35    /**
36     * Create the frame.
37     */
38    public OperatorAritmatikaGUI_2511533026() {
39        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
40        setBounds(100, 100, 450, 300);
41        contentPane = new JPanel();
42        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
43        setContentPane(contentPane);
44        contentPane.setLayout(null);
45
46        JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
47        lblNewLabel.setBounds(115, 10, 200, 20);
48        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 16));
49        lblNewLabel.setHorizontalTextPosition(SwingConstants.CENTER);
50        contentPane.add(lblNewLabel);
51
52        JLabel lblNewLabel_1_1 = new JLabel("Angka 2");
53        lblNewLabel_1_1.setBounds(10, 81, 45, 20);
54        contentPane.add(lblNewLabel_1_1);
55
56        JLabel lblNewLabel_1_2 = new JLabel("Operator");
57        lblNewLabel_1_2.setBounds(10, 128, 45, 20);
58        contentPane.add(lblNewLabel_1_2);
59
60        JLabel lblNewLabel_1_2_1 = new JLabel("Hasil");
61        lblNewLabel_1_2_1.setBounds(10, 196, 45, 26);
62        contentPane.add(lblNewLabel_1_2_1);
63
64        txtAngka1 = new JTextField();
65        txtAngka1.setHorizontalAlignment(SwingConstants.CENTER);
66        txtAngka1.setBounds(65, 52, 96, 18);
67        contentPane.add(txtAngka1);
68        txtAngka1.setColumns(10);
69
70        txtAngka2 = new JTextField();
71        txtAngka2.setHorizontalAlignment(SwingConstants.CENTER);
72        txtAngka2.setColumns(10);
73        txtAngka2.setBounds(65, 82, 96, 18);
74        contentPane.add(txtAngka2);
75    }
76}
```

```

90     JLabel lblNewLabel_1_1_1 = new JLabel("Angka 1");
91     lblNewLabel_1_1_1.setBounds(10, 51, 45, 20);
92     contentPane.add(lblNewLabel_1_1_1);
93
94     txtHasil = new JTextField();
95     txtHasil.setEditable(false);
96     txtHasil.setHorizontalAlignment(SwingConstants.CENTER);
97     txtHasil.setColumns(10);
98     txtHasil.setBounds(65, 200, 96, 18);
99     contentPane.add(txtHasil);
100
101    JComboBox cbOperator = new JComboBox();
102    cbOperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}));
103    cbOperator.setBounds(65, 131, 96, 20);
104    contentPane.add(cbOperator);
105
106    JButton btnNewButton = new JButton("Hitung");
107    btnNewButton.addActionListener(new ActionListener() {
108        int hasil;
109        public void actionPerformed(ActionEvent e) {
110            if(txtAngka1.getText().trim().isEmpty()) {
111                pesanPeringatan("Inputkan angka 1");
112            } else if(txtAngka2.getText().trim().isEmpty()) {
113                pesanPeringatan("Inputkan angka 2");
114            } else {
115                try {
116                    int a = Integer.valueOf(txtAngka1.getText());
117                    int b= Integer.valueOf(txtAngka2.getText());
118                    int c= cbOperator.getSelectedIndex();
119                    if (c==0) { hasil= a+b; }
120                    if (c==1) { hasil=a-b; }
121                    if (c==2) { hasil=a*c; }
122                    if (c==3) { hasil=a/b; }
123                    if (c==4) { hasil=a%b; }
124                    txtHasil.setText (String.valueOf(hasil));
125                }catch (NumberFormatException ex) {
126                    pesanError("Angka 1 dan 2 tidak valid");
127                }
128            }
129        }
130    });
131    btnNewButton.setBounds(194, 131, 84, 20);
132    contentPane.add(btnNewButton);
133
134}
135}
136

```

### 2.1.3 Hasil Output



#### **2.1.4 Analisis**

- Penggunaan null layout memberikan kontrol penuh terhadap posisi komponen, cocok untuk aplikasi sederhana.
- Validasi input dilakukan secara bertahap: kosong → konversi → operasi.
- Penggunaan JOptionPane membuat feedback kepada pengguna sangat jelas dan profesional.
- Pemisahan method pesanPeringatan() dan pesanError() menunjukkan prinsip DRY (Don't Repeat Yourself).
- Penggunaan Integer.parseInt() lebih disarankan daripada Integer.valueOf() untuk kasus sederhana.
- Belum menangani pembagian dengan nol (ArithmetricException) — dapat ditambahkan catch tambahan.

#### **2.1.5 Teori Mengenai Kode**

JFrame adalah container utama dalam Swing, sedangkan JPanel berfungsi sebagai container dalam container. Semua komponen ditambahkan ke contentPane dari JFrame. Event-driven programming menjadi inti GUI: tombol menghasilkan ActionEvent yang ditangkap oleh ActionListener. Exception handling (try-catch) sangat penting dalam GUI karena input pengguna tidak dapat diprediksi. JOptionPane merupakan cara standar untuk menampilkan dialog message, input, atau konfirmasi. Penggunaan EventQueue.invokeLater() memastikan semua operasi Swing dilakukan pada Event Dispatch Thread (EDT) agar thread-safe. Operator aritmatika (+, -, ×, /, %) pada tipe int mengikuti aturan matematika dasar, dengan pembagian dan modulus menghasilkan integer (pembagian bulat).

## BAB III

### KESIMPULAN

Praktikum pertemuan kedua ini berhasil mengantarkan mahasiswa dari pemrograman berbasis console menuju pengembangan aplikasi desktop berbasis Graphical User Interface (GUI) menggunakan Java Swing dan JFrame. Aplikasi kalkulator aritmatika yang dibuat tidak hanya berfungsi dengan baik, tetapi juga menerapkan praktik pemrograman yang baik seperti validasi input, penanganan exception, modularitas kode, dan pemberian feedback yang jelas kepada pengguna melalui JOptionPane. Mahasiswa kini memahami bahwa aplikasi GUI bersifat event-driven, sehingga logika program dijalankan berdasarkan aksi pengguna, bukan urutan prosedural seperti pada console.

Dari sisi teknis, praktikum ini memperkuat pemahaman tentang komponen-komponen dasar Swing (JTextField, JComboBox, JButton, JOptionPane), cara menangani event tombol, serta pentingnya menangani kesalahan input pengguna agar aplikasi tetap stabil dan user-friendly. Penggunaan layout null memberikan kebebasan posisi, namun juga mengajarkan bahwa untuk aplikasi yang lebih besar sebaiknya menggunakan layout manager agar responsif terhadap berbagai ukuran layar. Aplikasi ini menjadi contoh nyata bagaimana logika aritmatika sederhana dapat dikemas dalam antarmuka yang interaktif dan edukatif.

Ke depan, mahasiswa diharapkan dapat mengembangkan aplikasi GUI yang lebih kompleks dengan menambahkan fitur seperti riwayat perhitungan, dukungan bilangan desimal (menggunakan double), tombol clear, atau bahkan mengintegrasikan dengan database dan file. Praktikum ini telah membuktikan bahwa Java Swing tetap menjadi pilihan yang sangat baik untuk pengembangan aplikasi desktop edukasi dan enterprise sederhana hingga menengah. Saran untuk pertemuan berikutnya adalah membuat aplikasi manajemen data mahasiswa atau sistem login dengan multiple window dan penggunaan layout manager yang lebih advanced seperti GridBagLayout atau MigLayout.

## **DAFTAR PUSTAKA**

1. Oracle. (2023). *Creating a GUI With Swing - The Java™ Tutorials*.
2. Schildt, Herbert. (2019). *Java: The Complete Reference, 11th Edition*. McGraw-Hill Education.
3. Deitel, P. & Deitel, H. (2020). *Java How to Program, 11th Edition*. Pearson.
4. Horstmann, Cay S. (2019). *Core Java Volume I – Fundamentals, 11th Edition*. Prentice Hall.
5. Eck, David J. (2021). *Introduction to Programming Using Java, Version 9*. Hobart and William Smith Colleges.