

LAPORAN PRATIKUM PEKAN 7
ALGORITMA DAN PEMROGRAMAN

disusun Oleh:

ALIF MUHAMMAD IHSAN

NIM 2511531004

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

Asisten Laboratorium: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Swt atas limpahan rahmat dan karunia-Nya sehingga praktikum pemrograman Java ini dapat diselesaikan dengan baik dan laporan ini dapat disusun sebagai bentuk pertanggungjawaban akademik.

Laporan praktikum ini disusun untuk mendokumentasikan proses pembelajaran yang telah dilakukan, khususnya dalam memahami struktur dasar pemrograman Java, penggunaan metode, serta penerapan logika prosedural dalam bentuk program sederhana. Praktikum ini diharapkan dapat memperkuat pemahaman peserta terhadap konsep-konsep dasar pemrograman serta meningkatkan kemampuan teknis dalam menulis kode yang terstruktur dan efisien.

Penulis menyadari bahwa penyusunan laporan ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada dosen pembimbing, asisten praktikum, serta rekan-rekan yang telah memberikan dukungan dan masukan selama pelaksanaan praktikum.

Akhir kata, penulis berharap laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pembaca dalam memahami dasar-dasar pemrograman Java. Kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Padang, November 2025

Alif Muhammad Ihsan

Daftar Isi

KATA PENGANTAR.....	i
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Manfaat Praktikum	2
BAB II.....	3
PEMBAHASAN.....	3
2.1 Kode Program.....	3
2.2 Langkah Kerja.....	11
2.3 Analisis Hasil	14
BAB III	16
PENUTUP.....	16
3.1 Kesimpulan.....	16
3.2 Saran.....	16
DAFTAR PUSTAKA	18

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era perkembangan teknologi informasi yang semakin pesat, pemahaman terhadap dasar-dasar pemrograman menjadi hal yang esensial bagi setiap individu yang ingin berkontribusi dalam bidang rekayasa perangkat lunak. Salah satu bahasa pemrograman yang banyak digunakan dalam dunia akademik maupun industri adalah Java. Bahasa ini dikenal karena sifatnya yang berorientasi objek, portabel, serta memiliki struktur sintaks yang relatif mudah dipahami oleh pemula.

Praktikum ini dilaksanakan sebagai bagian dari upaya untuk memperkuat pemahaman mahasiswa terhadap konsep dasar pemrograman Java, khususnya dalam hal struktur program, penggunaan metode, serta logika eksekusi program. Melalui kegiatan praktikum, peserta diharapkan mampu menerapkan teori yang telah dipelajari dalam bentuk implementasi nyata, sehingga tercipta keterkaitan antara aspek konseptual dan praktik langsung.

1.2 Tujuan

1. Memahami struktur dasar program Java, termasuk penggunaan metode, pemanggilan fungsi, dan alur eksekusi program secara sistematis.
2. Melatih kemampuan dalam menyusun algoritma prosedural, melalui implementasi langkah-langkah logis yang merepresentasikan proses nyata, seperti pembuatan kue.
3. Meningkatkan keterampilan teknis dalam menulis kode yang terstruktur dan efisien, serta membiasakan diri dengan praktik penulisan program yang baik dan mudah dipahami.

4. Mendorong kemampuan berpikir kritis dan analitis, khususnya dalam mengidentifikasi urutan proses, menyederhanakan logika, dan menerjemahkan aktivitas sehari-hari ke dalam bentuk algoritma.

1.3 Manfaat Praktikum

1. **Memperkuat pemahaman konsep dasar pemrograman Java**, sehingga peserta mampu mengaitkan teori dengan implementasi nyata.
2. **Mengasah keterampilan penyusunan algoritma prosedural**, melalui pembiasaan merinci setiap langkah kerja sebelum menerjemahkannya ke dalam kode.
3. **Meningkatkan kemampuan menulis kode yang modular dan efisien**, dengan memanfaatkan metode sebagai unit kerja yang terpisah dan terstruktur.
4. **Memberikan pengalaman langsung dalam penerjemahan proses sehari-hari ke dalam program**, sehingga peserta dapat melihat relevansi pemrograman dalam konteks kehidupan nyata.
5. **Melatih kedisiplinan, kerapian, dan kemampuan dokumentasi**, yang menjadi landasan penting bagi profesional di bidang rekayasa perangkat lunak.

BAB II

PEMBAHASAN

2.1 Kode Program

```
package Pekan8_2511531004;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JButton;
import java.awt.Color;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
```

```

public class OperatorAritmatikaGUI_2511531004 extends JFrame {

    private static final long serialVersionUID = 1L;

    private JPanel contentPane;

    private JTextField TextBil1;

    private JTextField TextBil2;

    private JTextField TextHasil;


    private void pesanPeringatan(String pesan) {

        JOptionPane.showMessageDialog(this,pesan,"Peringatan",
JOptionPane.WARNING_MESSAGE);

    }

    private void pesanError(String pesan) {

        JOptionPane.showMessageDialog(this,pesan,"Kesalahan",
JOptionPane.WARNING_MESSAGE);

    }

    /**
     * Launch the application.
     */

    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable() {

            public void run() {

                try {

```

```

        OperatorAritmatikaGUI_2511531004 frame
= new OperatorAritmatikaGUI_2511531004();

        frame.setVisible(true);

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

});

}

/**
 * Create the frame.
 */

public OperatorAritmatikaGUI_2511531004() {

    setTitle("OPERATOR ARITMATIKA");

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setBounds(100, 100, 450, 300);

    contentPane = new JPanel();

    contentPane.setBackground(new Color(255, 255, 255));

    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);

    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("Operator Aritmatika");

```



```
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);  
lblNewLabel.setFont(new Font("Times New Roman", Font.PLAIN,  
14));  
lblNewLabel.setBounds(150, 10, 118, 36);  
contentPane.add(lblNewLabel);
```

```
JLabel bilangan2 = new JLabel("Bilangan 2");  
bilangan2.setFont(new Font("Times New Roman", Font.PLAIN,  
12));  
bilangan2.setBounds(27, 85, 74, 12);  
contentPane.add(bilangan2);
```

```
JLabel bilangan1 = new JLabel("Bilangan 1");  
bilangan1.setFont(new Font("Times New Roman", Font.PLAIN,  
12));  
bilangan1.setBounds(27, 63, 74, 12);  
contentPane.add(bilangan1);
```

```
JLabel operator = new JLabel("Operator");  
operator.setFont(new Font("Times New Roman", Font.PLAIN,  
12));  
operator.setBounds(27, 144, 74, 12);  
contentPane.add(operator);
```

```
JLabel head = new JLabel("Head");
```

```

head.setFont(new Font("Times New Roman", Font.PLAIN, 12));

head.setBounds(27, 166, 74, 12);

contentPane.add(head);


TextBil1 = new JTextField();

TextBil1.setBounds(111, 60, 74, 18);

contentPane.add(TextBil1);

TextBil1.setColumns(10);


TextBil2 = new JTextField();

TextBil2.setColumns(10);

TextBil2.setBounds(111, 82, 74, 18);

contentPane.add(TextBil2);


TextHasil = new JTextField();

TextHasil.setColumns(10);

TextHasil.setBounds(111, 166, 74, 18);

contentPane.add(TextHasil);


JComboBox CbOperator = new JComboBox();

CbOperator.setModel(new DefaultComboBoxModel(new String[]
{"* ", "/", "+", "-", "%"}));

CbOperator.setBounds(111, 140, 35, 20);

contentPane.add(CbOperator);

```

```

JButton proccess = new JButton("Proccess");

proccess.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        int hasil = 0;

        if( TextBil1.getText().trim().isEmpty()){

            pesanPeringatan("Bilangan 1 harus diisi");

        } else if ( TextBil2.getText(). trim(). isEmpty()) {

            pesanPeringatan("Bilangan 2 harus diisi");

        } else {

            try {

                int a = Integer.parseInt(TextBil1.getText());

                int b = Integer.parseInt(TextBil2.getText());

                int c = CbOperator.getSelectedIndex(); // memilih
operator

                if ( c == 0 ) {

                    hasil = a * b;

                }

            }

        }

    }

});

```

```

        if (c == 1) {

            hasil = a / b;

        }

        if (c == 2) {

            hasil = a + b;

        }

        if ( c == 3) {

            hasil = a - b;

        }

        if ( c == 4 ) {

            hasil = a % b;

        }

    } catch (NumberFormatException ex ) {

        pesanError("Bilangan 1 dan Bilangan 2 harus diisi
angka");

    }

}

TextHasil.setText(String.valueOf(hasil));

```

```
        }  
    });  
    proccess.setBounds(202, 140, 84, 20);  
    contentPane.add(proccess);  
  
    }  
}
```

2.2 Langkah Kerja

Judul

Program Operator Aritmatika Berbasis Java Swing

Deklarasi

- **Variabel**

- TextBil1 : kotak teks untuk input bilangan pertama (int)
- TextBil2 : kotak teks untuk input bilangan kedua (int)
- TextHasil : kotak teks untuk menampilkan hasil perhitungan (int)
- CbOperator : kotak pilihan (combo box) untuk memilih jenis operator aritmatika (String)
- hasil : variabel sementara untuk menyimpan hasil perhitungan (int)

- **Metode Tambahan**

- pesanPeringatan(String pesan) : menampilkan pesan peringatan jika input kosong.
- pesanError(String pesan) : menampilkan pesan kesalahan jika input bukan angka.

- **Komponen Antarmuka**

- JFrame : jendela utama program.
- JPanel : panel konten.
- JLabel : label teks untuk keterangan (Bilangan 1, Bilangan 2, Operator, Head).
- JButton : tombol "Proccess" untuk menjalankan perhitungan.

Algoritma

1. Inisialisasi Program

- Membuat jendela utama (JFrame) dengan judul "Operator Aritmatika".
- Menentukan ukuran jendela dan tata letak.
- Menambahkan panel (JPanel) dengan latar putih.

2. Membuat Komponen Input

- Menambahkan label "Bilangan 1" dan kotak teks TextBil1.
- Menambahkan label "Bilangan 2" dan kotak teks TextBil2.
- Menambahkan label "Operator" dan combo box CbOperator berisi pilihan: *, /, +, -, %.

3. Membuat Komponen Output

- Menambahkan label "Head" dan kotak teks TextHasil untuk menampilkan hasil perhitungan.

4. Tombol Proses

- Menambahkan tombol "Process".
- Saat ditekan, program akan:
- Mengecek apakah TextBil1 atau TextBil2 kosong. Jika kosong, tampilkan pesan peringatan.
- Jika tidak kosong, coba konversi input menjadi bilangan bulat (int).
- Jika input bukan angka, tampilkan pesan error.
- Jika valid, ambil operator yang dipilih dari CbOperator.
- Lakukan perhitungan sesuai operator:
 - * → perkalian
 - / → pembagian

- $+$ \rightarrow penjumlahan
- $-$ \rightarrow pengurangan
- $\%$ \rightarrow modulus
- Simpan hasil ke variabel hasil.
- Tampilkan hasil pada TextHasil.

5. Menjalankan Program

- Program dijalankan melalui `main()` dengan memanggil konstruktor `OperatorAritmatikaGUI_2511531004`.
- Jendela kalkulator ditampilkan dan siap digunakan.

2.3 Analisis Hasil

1. Antarmuka Grafis (GUI)

- Program berhasil menampilkan jendela dengan label, kotak teks, combo box, dan tombol proses.
- Input bilangan pertama, bilangan kedua, serta hasil ditampilkan dalam kotak teks (JTextField).
- Operator dipilih melalui combo box (JComboBox) dengan opsi: *, /, +, -, %.

2. Validasi Input

- Jika kotak teks bilangan pertama atau kedua kosong, program menampilkan pesan peringatan.
- Jika input bukan angka, program menampilkan pesan kesalahan.
- Hal ini membuat program lebih aman dan user-friendly karena mencegah error runtime.

3. Fungsi Perhitungan

- Program membaca nilai bilangan pertama (a) dan bilangan kedua (b).
- Operator dipilih berdasarkan indeks combo box (c).
- Perhitungan dilakukan sesuai operator:
 - * → perkalian
 - / → pembagian
 - + → penjumlahan
 - - → pengurangan
 - % → modulus
- Hasil perhitungan ditampilkan pada kotak teks hasil (TextHasil).

4. Kelebihan Program

- Antarmuka sederhana dan mudah dipahami.
- Validasi input mencegah kesalahan penggunaan.
- Mendukung lima operasi aritmatika dasar.

5. Keterbatasan Program

- Program hanya menerima input bilangan bulat (int), belum mendukung bilangan desimal.
- Tidak ada penanganan khusus untuk pembagian dengan nol (akan menimbulkan error).
- Tampilan masih sederhana, belum ada fitur tambahan seperti riwayat perhitungan atau desain interaktif.

Referensi

- Oracle. (2023). *Java Swing Tutorial*. Oracle Documentation. Diakses dari: <https://docs.oracle.com/javase/tutorial/uiswing/>
- Schildt, H. (2018). *Java: The Complete Reference* (11th ed.). New York: McGraw-Hill Education.
- Deitel, P. J., & Deitel, H. M. (2017). *Java How to Program* (10th ed.). Pearson Education.
- W3Schools. (2023). *Java Swing Components*. Diakses dari: https://www.w3schools.com/java/java_swing.asp

BAB III

PENUTUP

3.1 Kesimpulan

1. Program `OperatorAritmatikaGUI_2511531004` berhasil menjalankan operasi aritmatika dasar (perkalian, pembagian, penjumlahan, pengurangan, dan modulus) melalui antarmuka grafis berbasis Java Swing.
2. Antarmuka yang dibuat sederhana dan cukup jelas, dengan input bilangan, pilihan operator, serta hasil yang ditampilkan pada kotak teks.
3. Program sudah dilengkapi validasi input berupa pesan peringatan jika kotak teks kosong dan pesan kesalahan jika input bukan angka, sehingga lebih ramah pengguna.
4. Hasil perhitungan ditampilkan secara langsung setelah tombol "Proccess" ditekan, sesuai operator yang dipilih.
5. Walaupun sudah berfungsi dengan baik, program masih memiliki keterbatasan, seperti belum mendukung bilangan desimal dan belum ada penanganan khusus untuk pembagian dengan nol.

3.2 Saran

1. **Validasi** **Lebih** **Lanjut**
Tambahkan pengecekan khusus untuk kasus pembagian dengan nol agar program tidak error.
2. **Dukungan** **Bilangan** **Desimal**
Ubah tipe data dari int menjadi double agar program dapat menghitung bilangan pecahan.
3. **Pengembangan** **Antarmuka**
Gunakan tata letak yang lebih modern (misalnya GridLayout) dan tambahkan warna atau ikon agar tampilan lebih menarik.

4. **Fitur Tambahan**

- a. Menyediakan riwayat perhitungan (history).
- b. Menambahkan tombol reset untuk menghapus semua input sekaligus.
- c. Memberikan opsi format hasil (integer atau desimal).

5. **Optimasi**

Kode

Struktur kode dapat diperbaiki dengan membuat metode khusus untuk setiap operasi agar lebih modular dan mudah dipelihara.

DAFTAR PUSTAKA