

## **Laporan praktikum pekan 8**



**ARYA PRATAMA HENDRI  
2411533008**

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

## **KATA PENGANTAR**

Sebuah laporan tentang pelaksanaan praktikum dari matkul praktikum algoritma dan pemograman yang berisi penjelasan tentang code yang dikerjakan. Laporan ini dibuat sebagai bentuk pertanggungjawaban atas pelaksanaan praktikum yang telah dilakukan. Isi laporan ini membahas penerapan percabangan (if-else) dan penggunaan struktur kontrol switch-case dalam pemrograman Java. Konsep tersebut diimplementasikan melalui program yang kita kerjakan dan yang bertujuan untuk melatih logika dasar dalam pengambilan keputusan pada sebuah program komputer.

Padang, 2025

Tim Penyusun

## DAFTAR PUSTAKA

<b>KATA PENGANTAR .....</b>	i
<b>DAFTAR PUSTAKA .....</b>	ii
<b>DAFTAR LAMPIRAN.....</b>	iii
<b>BAB I PENDAHULUAN .....</b>	1
1.1 Pengertian Pratikum.....	1
1.2 Tujuan Pratikum .....	1
1.3 Persyaratan Pratikum .....	1
1.4 Waktu dan Tempat Pratikum .....	2
<b>BAB II PELAKSANAAN PRAKTIKUM .....</b>	3
2.1 Dasar teori.....	3
2.2 alat yang digunakan .....	3
2.3 Penjelasan coding .....	3
2.4 Kesimpulan .....	4
2.5 Daftar pustaka.....	5

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Pengertian Pratikum**

Praktikum Java merupakan kegiatan pembelajaran yang bertujuan untuk melatih kemampuan mahasiswa dalam memahami dan menerapkan konsep dasar pemrograman berbasis Java. Kegiatan ini memfokuskan pada latihan langsung, mulai dari penulisan kode, proses kompilasi, hingga pengujian output program. Melalui praktikum, mahasiswa dapat menghubungkan teori yang dipelajari di kelas dengan penerapannya dalam bentuk program nyata, sekaligus membangun keterampilan teknis dan ketelitian dalam bekerja dengan bahasa pemrograman Java.

#### **1.2 Tujuan Pratikum**

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

1. Mengerti tentang apa code yang dikerjakan.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (*problem solving*) dengan pendekatan algoritmik.
4. Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum.
5. Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.

#### **1.3 Persyaratan Pratikum**

Agar praktikum berjalan lancar, mahasiswa perlu memenuhi beberapa persyaratan berikut:

1. Telah mengikuti perkuliahan teori Pemrograman Java sebagai dasar pemahaman.
2. Membawa perlengkapan yang diperlukan, antara lain laptop atau komputer yang sudah terpasang Java Development Kit (JDK) dan Integrated Development Environment (IDE) yang direkomendasikan.

3. Mengikuti setiap sesi praktikum sesuai jadwal yang ditetapkan dan hadir minimal sesuai ketentuan program studi.
4. Mematuhi tata tertib laboratorium, termasuk menjaga keamanan data, perangkat, serta lingkungan kerja.
5. Menyusun laporan praktikum dengan format dan aturan yang telah ditetapkan dalam pedoman ini.

#### 1.4 Waktu dan Tempat Pratikum

Pelaksanaanya di sesi c di hari selasa jam 10.00 dilaksanakan di labor if

## BAB II

### PENULISAN LAPORAN PRATIKUM

#### 2.1 Dasar teori

Praktikum ini berfokus pada penggunaan komponen GUI pada Java Swing, proses event handling, serta penggunaan operator aritmatika untuk melakukan perhitungan pada input pengguna.

##### 1. Java Swing

Java Swing adalah library Java yang digunakan untuk membangun antarmuka grafis (GUI). Swing menyediakan komponen seperti:

- JFrame → jendela utama aplikasi
  - JPanel → wadah komponen
  - JLabel → menampilkan teks
  - JTextField → menerima input pengguna
  - JButton → tombol untuk menjalankan aksi
  - JComboBox → memilih opsi seperti operator aritmatika
- Pada praktikum ini Swing digunakan untuk membuat tampilan perhitungan aritmatika secara interaktif.

##### 2. Event Handling

Event handling adalah mekanisme untuk menangani aksi pengguna, seperti menekan tombol.

Dalam Java Swing, event ditangani dengan:

- **ActionListener**
  - Method **actionPerformed()**
- Ketika tombol "Hitung" ditekan, program membaca input, memproses operator, dan menampilkan hasil perhitungan.

##### 3. Validasi Input

Validasi input dilakukan untuk memastikan data yang dimasukkan sudah benar sebelum dihitung.

Validasi mencakup:

- Pemeriksaan apakah kolom input kosong
  - Menangani kesalahan format angka menggunakan try { ... } catch (NumberFormatException)
- Jika terjadi kesalahan, program menampilkan pesan menggunakan JOptionPane.

##### 4. Operator Aritmatika

Operator aritmatika adalah operator yang digunakan untuk melakukan perhitungan matematis. Java menyediakan beberapa operator dasar, yaitu:

- + → penjumlahan
- - → pengurangan
- \* → perkalian
- / → pembagian
- % → modulus (sisa hasil bagi)

Pada praktikum ini operator dipilih melalui JComboBox, kemudian dihitung berdasarkan input bilangan pertama dan kedua.

## 5. Exception Handling

Exception handling digunakan untuk mencegah program berhenti ketika terjadi kesalahan input.

Misalnya:

- Pengguna memasukkan huruf pada kolom angka
  - Terjadi kesalahan pembagian
- Kesalahan ditangani menggunakan blok try–catch, dan program menampilkan pesan error menggunakan JOptionPane.

---

## 2.2 Alat yang digunakan

1. Perangkat computer atau laptop
2. Jaringan internet
3. IDE (Integreted Development Environment) direkomendasikan  
elipse IDE Java JDK
4. Kit)

## 2.3 Penjelasan coding

1. Membuat class utama
  - Membuat class bernama OperatorAritmatikaGUI\_2411533008 yang menjadi frame utama dengan pewarisan dari JFrame.
2. Menyiapkan komponen GUI
  - Program membuat komponen seperti:
    - JTextField txtBil1, txtBil2, dan txtHasil
    - JComboBox cbOperator
    - JButton btnHitung
    - Semua komponen ditempatkan menggunakan setBounds().
3. Membuat method pesan peringatan

Syntax logika:

```
private void pesanPeringatan(String pesan)
```

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

Digunakan ketika input kosong.

---

4. Membuat method pesan error

Syntax logika:

```
private void PesanEror(String pesan)
```

```
    private void PesanEror(String pesan) {  
        JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.ERROR_MESSAGE);  
    }
```

Digunakan ketika input bukan angka (NumberFormatException).

---

5. Validasi input menggunakan logika percabangan

Program mengecek apakah field kosong sebelum dihitung.

Syntax logika:

```

// VALIDASI INPUT KOSONG
if (txtBil1.getText().trim().isEmpty()) {
    pesanPeringatan("Bilangan 1 Harus diisi");
    return;
}
else if (txtBil2.getText().trim().isEmpty()) {
    pesanPeringatan("Bilangan 2 Harus diisi");
    return;
}

```

Logika:

- Jika Bilangan1 kosong → tampilkan pesan
  - Jika Bilangan2 kosong → tampilkan pesan
  - Jika keduanya terisi → lanjut proses perhitungan
- 

## 6. Mengubah input String menjadi angka

Syntax logika:

```

int a = Integer.valueOf(txtBil1.getText());
int b = Integer.valueOf(txtBil2.getText());
int c = cbOperator.getSelectedIndex();

```

Digunakan agar bisa dihitung secara matematis.

---

## 7. Menentukan operator menggunakan logika switch (percabangan berbasis indeks)

Program membaca indeks operator dari ComboBox:

```
int c = cbOperator.getSelectedIndex();
```

Logika operator:

```

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 hasil = a % b;

```

Penjelasan logika:

- Jika combo memilih index 0 → lakukan penjumlahan
  - Jika index 1 → kurangi
  - Jika index 2 → bagi
  - Jika index 3 → kali
  - Jika index 4 → modulus
- 

#### 8. Menampilkan hasil

Syntax logika:

```
txtHasil.setText(String.valueOf(hasil));
```

---

#### 9. Menangani input invalid dengan try-catch

Logika error:

```
} catch (NumberFormatException ex) {
    PesanError("Bilangan 1 dan Bilangan 2 harus angka");
}
};
```

Jika pengguna memasukkan huruf atau karakter lain, maka tampil pesan error.

---

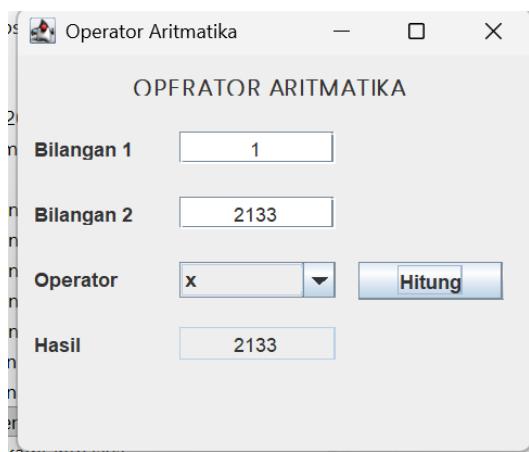
#### 10. Menampilkan GUI dengan EventQueue

Syntax:

```
EventQueue.invokeLater(new Runnable() {
    public void run() {
        try {
            OperatorAritmatikaGUI_2411533008 frame = new OperatorAritmatikaGUI_2411533008();
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
```

Logika: memastikan GUI berjalan aman di thread event Swing.

#### 11. outputnya



## 2.4 Kesimpulan

Program GUI Operator Aritmatika ini menyajikan antarmuka yang interaktif dan ringkas untuk melakukan operasi matematika dasar (penjumlahan, pengurangan, perkalian, pembagian, dan modulus) antara dua bilangan, dilengkapi dengan validasi input dan penanganan error untuk mencegah kesalahan format, serta menggunakan logika percabangan berdasarkan pilihan operator dari pengguna agar hasilnya dapat ditampilkan secara tepat di tampilan GUI.

## 2.5 Daftar pustaka

[https://ilmukomputer.org/wp-content/uploads/2018/07/sofwan\\_java-15.pdf](https://ilmukomputer.org/wp-content/uploads/2018/07/sofwan_java-15.pdf)



