

**LAPORAN PRATIKUM  
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN**

**LAPORAN PRAKTIKUM PEKAN 8**

**Disusun Oleh:  
Muhammad Aufa Rafiki**

**2511531012**

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

**Asisten Praktikum: Muhammad Zaki Al Hafiz**



**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**2025**

## **KATA PENGANTAR**

Pedoman ini disusun sebagai rujukan resmi bagi mahasiswa Departemen Informatika dalam penyusunan laporan praktikum pada mata kuliah *Pemrograman Dasar dengan Java*. Dokumen ini tidak hanya memberikan gambaran umum mengenai format penulisan, tetapi juga menguraikan secara rinci sistematika laporan, tata cara penyajian isi, serta contoh penulisan kode program yang dilengkapi dengan referensi ilmiah. Melalui panduan ini, mahasiswa diharapkan mampu menyusun laporan yang tidak sekadar memenuhi aspek administratif, tetapi juga mencerminkan ketelitian, keteraturan, dan penerapan kaidah penulisan akademik pada tingkat dasar. Dengan demikian, laporan praktikum yang dihasilkan dapat berfungsi sebagai media pembelajaran, dokumentasi kegiatan, sekaligus sarana untuk melatih keterampilan menulis ilmiah yang akan bermanfaat dalam jenjang studi selanjutnya.

## **DAFTAR ISI**

<b>KATA PENGANTAR .....</b>	<b>i</b>
<b>DAFTAR ISI .....</b>	<b>ii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
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 PEMBAHASAN .....</b>	<b>3</b>
2.1    Desing .....	3
2.2    Kode Program.....	4
2.3    Penjelasan Kerja Program.....	4
2.4    Output .....	6
<b>BAB III KESIMPULAN .....</b>	<b>8</b>
<b>DAFTAR PUSTAKA .....</b>	<b>9</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Pengertian Pratikum**

Praktikum Java adalah kegiatan pembelajaran yang dilakukan di laboratorium komputer untuk mengasah keterampilan mahasiswa dalam memahami serta menerapkan konsep pemrograman Java. Kegiatan ini tidak hanya menekankan pada penguasaan teori, tetapi juga pada latihan penyusunan kode program, pengujian, hingga analisis hasil eksekusi. Praktikum dipandang sebagai wahana latihan yang menjembatani pemahaman konseptual dengan kemampuan teknis pemrograman.

### **1.2 Tujuan Pratikum**

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.** Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
- 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.

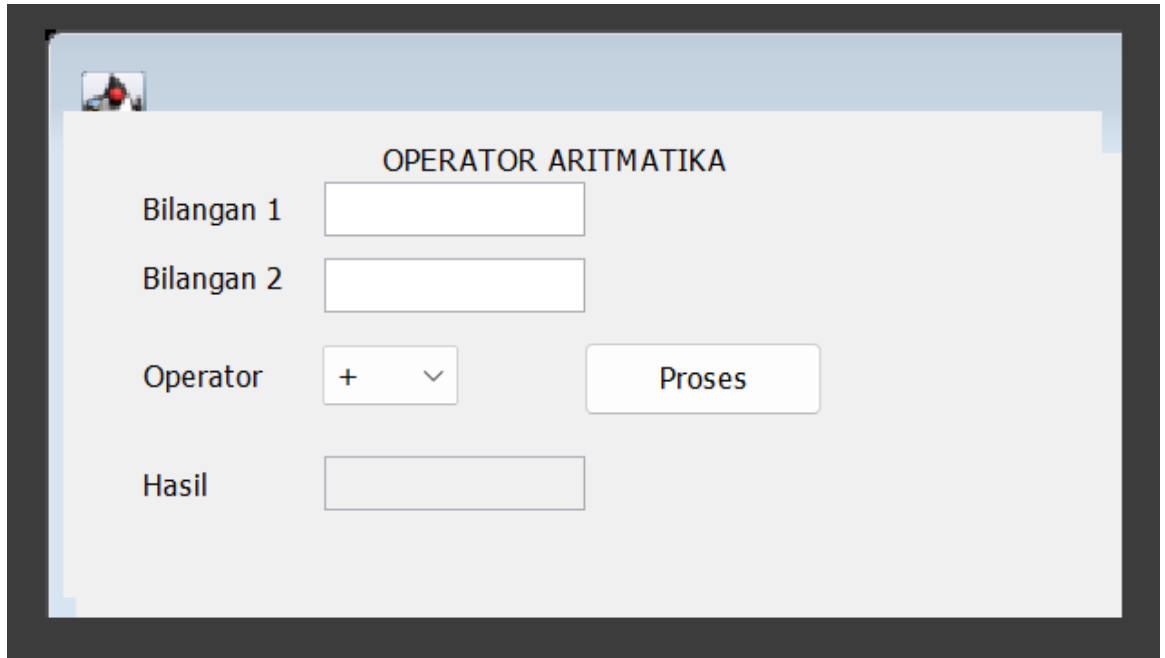
3. Mengikuti setiap sesi praktikum sesuai jadwal yang ditetapkan dan hadir minimal sesuai ketentuan program studi.
4. Mematuhi tata tertib laboratorium, termasuk menjaga keamanan data, perangkat, serta lingkungan kerja.
5. Menyusun laporan praktikum dengan format dan aturan yang telah ditetapkan dalam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

#### 1.4 Waktu dan Tempat Pratikum

Pelaksanaan praktikum Java mengikuti kalender akademik yang berlaku pada program studi. Setiap sesi praktikum dilaksanakan sesuai jadwal yang ditentukan oleh dosen pengampu. Tempat kegiatan umumnya berlangsung di laboratorium komputer, namun pada kondisi tertentu dapat dilaksanakan secara mandiri dengan perangkat masing-masing, selama memenuhi syarat teknis yang ditetapkan.

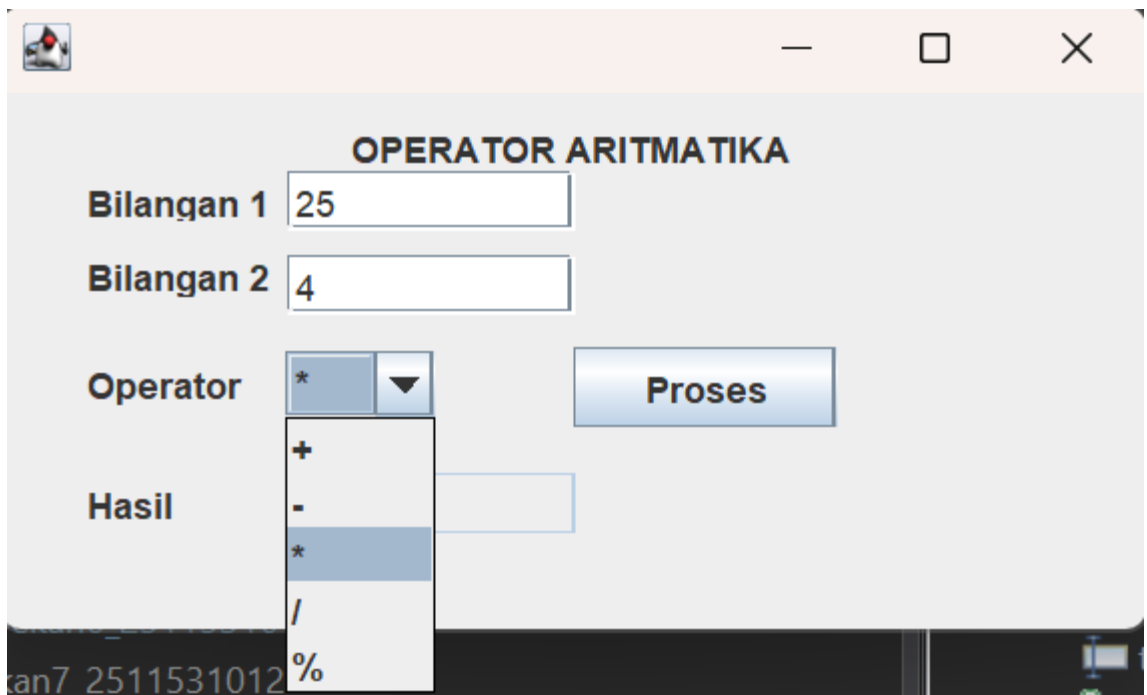
## BAB II PEMBAHASAN

### 2.1 Design



The screenshot shows a window titled "OPERATOR ARITMATIKA". It contains four input fields: "Bilangan 1", "Bilangan 2", "Operator", and "Hasil". The "Operator" field is a dropdown menu currently showing a "+" sign. To the right of the "Operator" field is a button labeled "Proses". The "Hasil" field is currently empty.

Gambar 2.1



The screenshot shows the same "OPERATOR ARITMATIKA" window after user input. "Bilangan 1" contains the value "25" and "Bilangan 2" contains the value "4". The "Operator" dropdown menu is open, showing a list of operators: "+", "-", "\*", and "/". The "\*" operator is currently selected and highlighted. The "Hasil" field is empty. The "Proses" button is visible to the right of the operator dropdown.

Gammbar 2.2

## 2.2 Kode Program

```
17 public class OperatorAritmatikaGUI_2511531012 extends JFrame {
18
19     private static final long serialVersionUID = 1L;
20     private JPanel contentPane;
21     private JTextField txtBil1;
22     private JTextField txtBil2;
23     private JTextField txtHasil;
```

Gambar 2.3

```
28 private void pesanPeringatan(String pesan) {
29     JOptionPane.showMessageDialog(this, pesan, "peringatan", JOptionPane.WARNING_MESSAGE);
30 }
31 private void pesanError(String pesan) {
32     JOptionPane.showMessageDialog(this, pesan, "kesalahan", JOptionPane.ERROR_MESSAGE);
33 }
```

Gambar 2.4

```
99
100     JButton btnNewButton = new JButton("Proses");
101     btnNewButton.addActionListener(new ActionListener() {
102         int hasil ;
103         public void actionPerformed(ActionEvent e) {
104             if (txtBil1.getText().trim().isEmpty()) {
105                 pesanPeringatan ("Bilangan 1 harus diisi");
106             } else if (txtBil2.getText().trim().isEmpty()) {
107                 pesanPeringatan ("Bilangan 2 harus diisi");
108             }
109             else
110             {
111                 try {
112                     int a= Integer.parseInt(txtBil1.getText());
113                     int b= Integer.parseInt(txtBil2.getText());
114                     int c= cbOperator.getSelectedIndex();
115                     if (c==0) {hasil = a+b;}
116                     if (c==1) {hasil = a-b;}
117                     if (c==2) {hasil = a*b;}
118                     if (c==3) {hasil = a/b;}
119                     if (c==4) {hasil = a%b;}
120                 } catch (NumberFormatException ex) {
121                     pesanError("Bilangan 1 dan Bilangan 2 harus angka");
122                 }
123             }
124             txtHasil.setText(String.valueOf(hasil));
125         }
126     });
```

Gambar 2.5

## 2.3 Penjelasan kerja program

1. Program ini memiliki komponen utama di antarmuka penggunaanya:
  - Input Bilangan 1 & 2: Dua text field (txtBil1 dan txtBil2) untuk memasukkan bilangan pertama dan kedua.
  - Operator: Sebuah combo box yang berisi operator aritmatika yang tersedia (+, -, \*, /, %). Indeks operator ini sangat penting dalam logika program.
    - + (Indeks 0)
    - - (Indeks 1)
    - \* (Indeks 2)

- / (Indeks 3)
  - % (Indeks 4)
  - Tombol Proses: Sebuah tombol (Proses) yang ketika diklik akan memicu eksekusi logika perhitungan.
  - Hasil: Sebuah text field (txtHasil) untuk menampilkan hasil perhitungan.
2. Logika Aksi (Saat Tombol "Proses" Diklik)
- Ketika pengguna menekan tombol "Proses" (`btnNewButton.addActionListener(...)`), program akan menjalankan langkah-langkah berikut:
- 1) Validasi Input
 

Program terlebih dahulu memeriksa apakah kedua text field bilangan sudah terisi.

    - Jika txtBil1 kosong, ia akan memanggil pesanPeringatan("Bilangan 1 harus diisi").
    - Jika txtBil1 terisi, tetapi txtBil2 kosong, ia akan memanggil pesanPeringatan("Bilangan 2 harus diisi").
  - 2) Konversi dan Validasi Angka
 

Jika kedua text field sudah terisi, program akan masuk ke blok try-catch:

    - Konversi: Nilai dari txtBil1 dan txtBil2 diambil sebagai String dan dicoba dikonversi menjadi tipe data Integer (`int a` dan `int b`) menggunakan `Integer.parseInt()`.
    - Indeks Operator: Indeks operator yang dipilih dari combo box disimpan dalam variabel `int c` menggunakan `cbOperator.getSelectedIndex()`.
    - Penanganan Error: Jika konversi gagal (misalnya, pengguna memasukkan teks non-angka seperti "lima"), `NumberFormatException` akan terpicu, dan program akan memanggil pesanError("Bilangan 1 dan Bilangan 2 harus angka").
  - 3) Pemilihan dan Eksekusi Operasi
 

Setelah bilangan dikonversi dengan sukses, program menggunakan serangkaian if statement berdasarkan indeks operator (`c`) untuk menentukan operasi yang akan dijalankan:

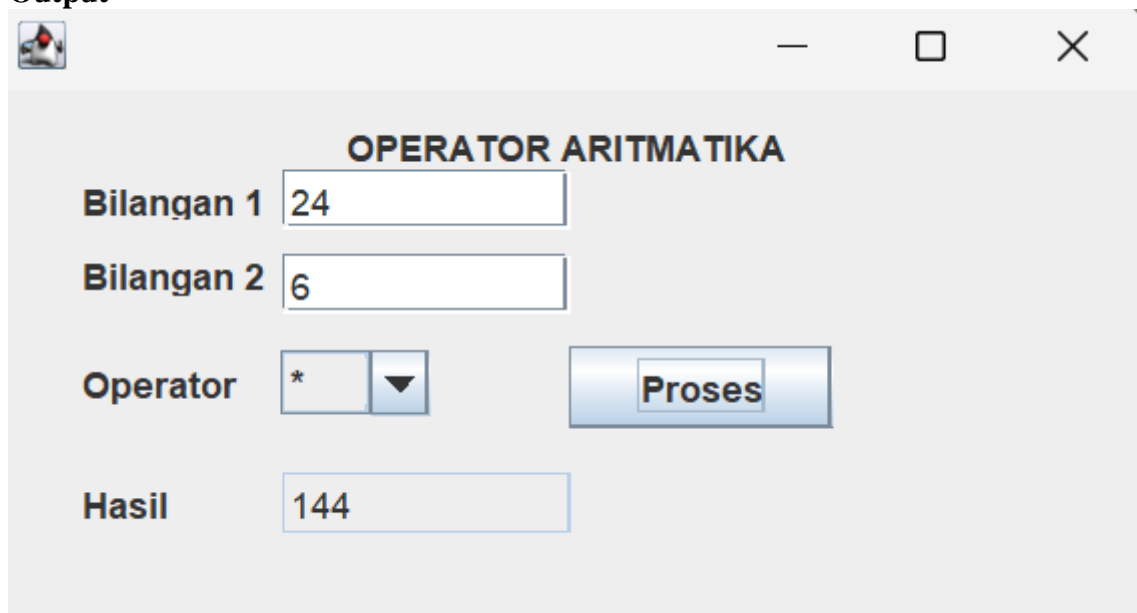
    - Jika `c = 0` (Operator `+`): `hasil = a + b`
    - Jika `c = 1` (Operator `-`): `hasil = a - b`
    - Jika `c = 2` (Operator `*`): `hasil = a * b`
    - Jika `c = 3` (Operator `/`): `hasil = a / b` (Perlu diingat, ini adalah pembagian integer, hasilnya akan dibulatkan ke bawah)
    - Jika `c = 4` (Operator `%`): `hasil = a mod b` (Operasi modulus atau sisa bagi)
  - 4) Menampilkan Hasil
 

Nilai akhir dari variabel hasil dikonversi kembali menjadi String dan ditampilkan di text field hasil (`txtHasil.setText(String.valueOf(hasil))`).
3. Fungsi Pesan (Peringatan & Error)
- Program memiliki dua fungsi bantu untuk menampilkan pesan pop-up menggunakan `JOptionPane`:
- 1) `private void pesanPeringatan(String pesan) { ... }`
    - Tujuan: Fungsi ini dipanggil ketika terjadi kesalahan yang relatif ringan, terutama yang berkaitan dengan validasi data yang hilang (input kosong).
    - Parameter: Menerima satu parameter, yaitu String pesan, yang berisi

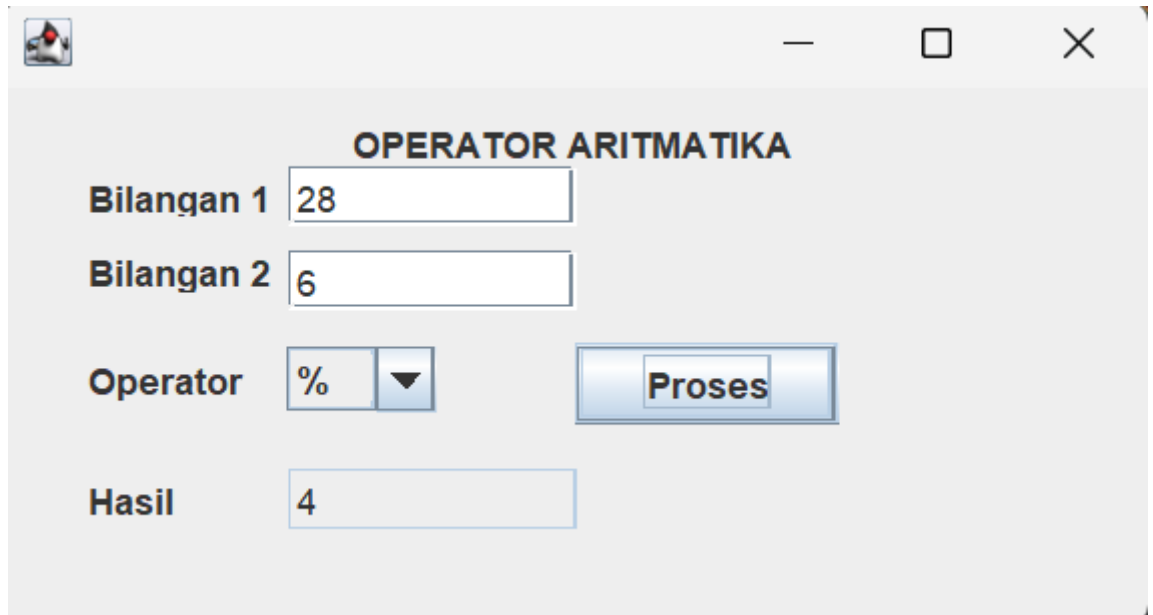


- teks spesifik yang ingin ditampilkan, misalnya "Bilangan 1 harus diisi".
- Cara Kerja JOptionPane.showMessageDialog():
    - this: Menunjukkan bahwa dialog akan berpusat pada frame utama aplikasi.
    - pesan: Teks pesan yang akan ditampilkan.
    - "peringatan": Judul yang muncul di bagian atas jendela dialog.
    - JOptionPane.WARNING\_MESSAGE: Ini adalah konstanta penting yang menentukan jenis ikon yang ditampilkan di dialog. Untuk pesan peringatan, ini biasanya menampilkan ikon berbentuk segitiga kuning.
- 2) private void pesanError(String pesan) {...}
- Tujuan: Fungsi ini dipanggil ketika terjadi kesalahan yang lebih serius yang berkaitan dengan format data yang salah, khususnya ketika terjadi NumberFormatException.
  - Parameter: Menerima satu parameter, yaitu String pesan, yang berisi teks spesifik kesalahan, misalnya "Bilangan 1 dan Bilangan 2 harus angka".
  - Cara Kerja JOptionPane.showMessageDialog():
    - this: Sama, dialog berpusat pada frame utama.
    - pesan: Teks pesan kesalahan yang akan ditampilkan.
    - "kesalahan": Judul yang muncul di jendela dialog.
    - JOptionPane.ERROR\_MESSAGE: Konstanta ini menentukan jenis ikon yang ditampilkan. Untuk pesan error, ini biasanya menampilkan ikon berbentuk tanda silang merah atau simbol berhenti, menandakan kegagalan proses.

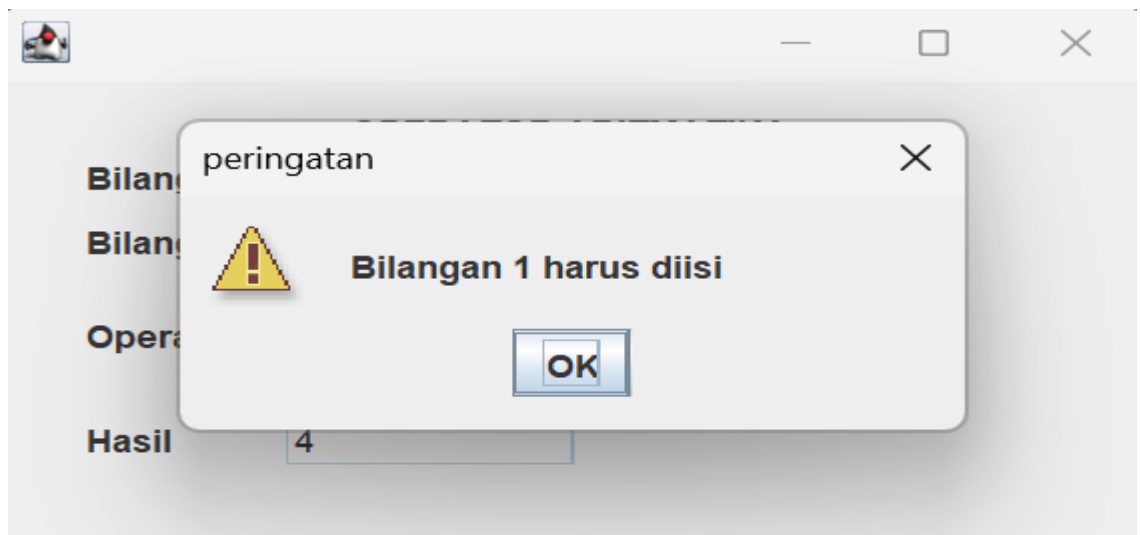
## 2.4 Output



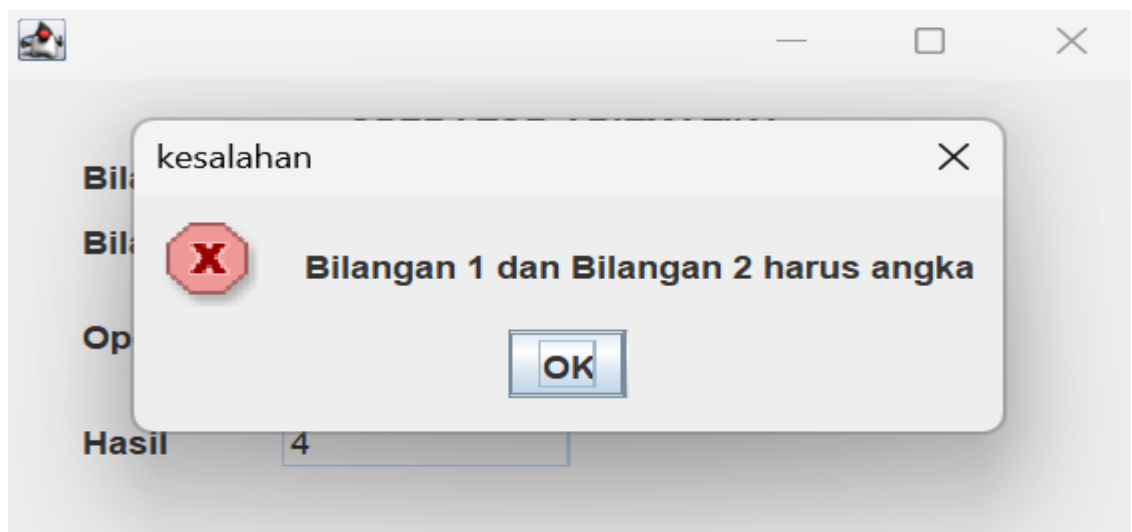
Gambar 2.6



Gambar 2.7



Gambar 2.8



Gambar 2.9

### **BAB III**

### **KESIMPULAN**

Program Operator Aritmatika ini dirancang sebagai aplikasi GUI (Graphical User Interface) menggunakan pustaka Java Swing untuk memfasilitasi perhitungan dua bilangan. Alur kerja sistem dimulai ketika pengguna memasukkan nilai pada kolom input dan memilih jenis operator aritmatika melalui komponen combo box. Logika inti program dieksekusi saat tombol "Proses" ditekan, di mana sistem akan membaca indeks operator yang dipilih (0 hingga 4) untuk menentukan rumus matematika yang sesuai, mulai dari penjumlahan hingga operasi modulus, lalu menampilkan hasilnya kembali ke antarmuka pengguna.

Program ini dilengkapi dengan dua fungsi pesan kustom yang memanfaatkan JOptionPane. Fungsi pesanPeringatan bertugas memvalidasi kelengkapan data dengan menampilkan dialog peringatan (WARNING\_MESSAGE) jika pengguna membiarkan kolom input kosong. Sementara itu, fungsi pesanError bekerja di dalam blok penanganan eksepsi (catch) untuk menampilkan dialog kesalahan (ERROR\_MESSAGE) apabila input yang dimasukkan bukan berupa angka. Mekanisme ini memastikan program berjalan secara robust (tangguh) dengan menangani potensi kesalahan input manusia secara efektif.

## DAFTAR PUSTAKA

- [1] H. Schildt, *Java: The Complete Reference*, 12th ed. New York, NY, USA: McGraw-Hill Education, 2021.
- [2] Oracle, "Class JOptionPane," *Java™ Platform, Standard Edition 8 API Specification*. [Online]. Tersedia: <https://docs.oracle.com/javase/8/docs/api/javax/swing/JOptionPane.html>. [Diakses: 20 Nov. 2025].