

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
PENERAPAN PROGRAM *OPERASI ASSIGMENT DENGAN GRAPHICAL*
USER INTERFACE (GUI) PADA JAVA



Disusun Oleh:

Haliya Isma Husna Putri Ahmadi
2511532002

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

Asisten Praktikum:
Muhammad Zaky Al Hafiz

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji dan Syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas Rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan tugas Praktikum Algoritma dan Pemrograman dengan judul “Penerapan Program Operasi Assigment Dengan *Graphical User Interface (Gui)* pada Java” pekan ke-8 tepat pada waktunya.

Penyusunan laporan ini bertujuan untuk memenuhi salah satu tugas dalam mata kuliah Praktikum Algoritma dan Pemrograman, sekaligus sebagai sarana pembelajaran bagi penulis dalam memahami dasar logika pemrograman dan penerapan operasi aritmatika pada bahasa pemrograman *Java*. Melalui praktikum ini, penulis mempelajari bagaimana membangun sebuah program perhitungan yang dapat menampilkan hasil penjumlahan, pengurangan, perkalian, pembagian, dan modulo dari dua bilangan menggunakan antarmuka grafis (*GUI*) berbasis *Java Swing*.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang membangun demi perbaikan di masa yang akan datang.

Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada dosen pengampu, asisten praktikum, serta semua pihak yang terlibat dalam pelaksanaan praktikum dan penyusunan laporan ini. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi penulis maupun pembaca.

Padang, 20 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN	4
1.1 Latar Belakang.....	4
1.2 Tujuan	4
1.3 Manfaat.....	4
BAB II PEMBAHASAN.....	5
2.1 Uraian Kode Program Java	5
2.1.1 Penjelasan Kode.....	5
2.2 Langkah Kerja	9
2.2.1 Bahasa Natural	10
2.2.2 <i>Flowchart</i>	11
2.2.3 <i>Pseudocode</i>	12
2.3 Analisis Hasil	13
BAB III KESIMPULAN.....	17
3.1 Hasil Praktikum	17
3.2 Saran	17
DAFTAR PUSTAKA	18

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam mempelajari bahasa pemrograman, pemahaman terhadap konsep dasar pengaplikasikan *Graphical User Interface (GUI)* dalam mengembangkan suatu program agar memungkinkan *user* berinteraksi dengan program melalui tampilan visual yang mudah dipahami, merupakan salah satu aspek penting dalam pengembangan perangkat lunak modern. Dalam *Java*, *GUI* dapat dibangun menggunakan *Java Swing*, sebuah *library* yang menyediakan berbagai komponen seperti *button*, *label*, *text field*, serta fitur *event handling* untuk merespons aksi pengguna.

Pada tugas praktikum pekan ke-8 ini, mahasiswa mempelajari cara mengimplementasikan *GUI* menggunakan *Java Swing* melalui pembuatan program “*Operator Assignment*”. Melalui program ini, mahasiswa belajar bagaimana mengaplikasikan *GUI* dengan menerima *input*, memproses operasi aritmatika, dan menampilkan hasil.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

- 1.2.1 Memahami konsep dasar pembuatan *GUI* menggunakan *Java Swing*, termasuk penggunaan *label*, *text field*, dan *button*.
- 1.2.2 Mengimplementasikan *event handling (ActionListener)* untuk memproses *input user* untuk diproses program.
- 1.2.3 Membangun logika pemrograman dalam menyusun program *GUI* yang mudah dipahami dengan algoritma yang terstruktur.

1.3 Manfaat

Manfaat yang diharapkan dari pelaksanaan praktikum ini antara lain:

- 1.3.1 Memberikan pengalaman praktis dalam membuat aplikasi sederhana berbasis *GUI* menggunakan *Java Swing*.
- 1.3.2 Meningkatkan pemahaman mengenai komponen *GUI* dan *event listener* dalam pemrosesan data untuk program perhitungan.

BAB II

PEMBAHASAN

2.1 Uraian Kode Program Java

Berikut merupakan uraian kode program yang dibuat untuk mengimplementasikan operasi aritmatika menggunakan antarmuka *Graphical User Interface (GUI)* pada Java. Program ini disusun dengan memanfaatkan komponen-komponen dasar Java Swing seperti *JLabel*, *JTextField*, dan *JButton*, serta mekanisme *event handling (ActionListener)* untuk memprogram *input user*.

Berikut merupakan penjelasan lebih lanjut terkait kode program “*tugasAlproPekan8_2511532002*”.

2.1.1 Penjelasan Kode

a) Akun.java (Class data)

1. *package pekan8_2511532002;*

Menjelaskan posisi file berada pada *package* bernama “*pekan8_2511532002*”.

2. *Impor library* dari Java untuk membuat aplikasi *GUI*.

```
3  import java.awt.EventQueue;
4
5  import javax.swing.JFrame;
6  import javax.swing.JPanel;
7  import javax.swing.border.EmptyBorder;
8  import java.awt.Color;
9  import javax.swing.JLabel;
10 import javax.swing.JOptionPane;
11
12 import java.awt.Font;
13 import javax.swing.SwingConstantsConstants;
14 import javax.swing.JTextField;
15 import javax.swing.JButton;
16 import java.awt.event.ActionListener;
17 import java.awt.event.ActionEvent;
```

2.2.1. 1 *Impor library* dari Java,

3. *public class tugasAlproPekan8_2511532002 extends JFrame{}*

Menunjukkan bahwa kelas ini merupakan bagian dari *JFrame*, sehingga menjadi jendela utama aplikasi *GUI*

4. Deklarasi *Variabel*

```
private static final long serialVersionUID = 1L;
private JPanel contentPane;
private JTextField tfBil1;
private JTextField tfBil2;
private JTextField txtPenjumlahan;
private JTextField txtPengurangan;
private JTextField txtPerkalian;
private JTextField txtPembagian;
private JTextField txtMod;
```

2.2.1. 2 Deklarasi Variabel

Berikut merupakan penjelasan untuk tiap-tiap variabel yang digunakan :

1) *private JPanel contentPane;*

Variabel *string* ini digunakan untuk menyimpan *username* yang dimasukkan nantinya pada *class utama*.

2) *Pivate JTextField tfBil1;*

private JTextField tfBil2;;

Merupakan *text field* untuk *input* bilangan.

3) *Private JTextField txtPenjumlahan;*

private JTextField txtPengurangan;

private JTextField txtPerkalian;

private JTextField txtPembagian;

private JTextField txtMod;

Merupakan *text field* untuk menampilkan hasil operasi atau *output*.

5. Methode Pesan Peringatan dan Errror

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

2.2.1. 3 Methode Pesan Peringatan dan Errror

Merupakan *methode* yang digunakan untuk menampilkan pesan *errror* atau *warning*.

6. Methode Main

```
/*
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                tugasAlproPekan8_2511532002 frame = new tugasAlproPekan8_2511532002();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

2.2.1. 4 Methode Main

7. JButton (ActionListener/ Logika button)

```
btnProses.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
```

2.2.1. 5 JButton ActionListener

“*addActionListener()*” digunakan agar tombol dapat merespons ketika diklik nantinya dan menjalankan perhitungan.

```

if (tfBil1.getText().trim().isEmpty()) {
    pesanPeringatan("Bilangan 1 harus diisi!");
    return;
}
if (tfBil2.getText().trim().isEmpty()) {
    pesanPeringatan("Bilangan 2 harus diisi");
    return;
}

try {
    int a = Integer.parseInt(tfBil1.getText());
    int b = Integer.parseInt(tfBil2.getText());

    //Operasi
    txtPenjumlahan.setText(String.valueOf(a + b));
    txtPengurangan.setText(String.valueOf(a - b));
    txtPerkalian.setText(String.valueOf(a * b));

    if (b == 0) {
        txtPembagian.setText("Error");
        txtMod.setText("Error");
        pesanError("Bilangan 2 tidak boleh 0!");
    }else {
        txtPembagian.setText(String.valueOf(a/b));
        txtMod.setText(String.valueOf(a%b));
    }
}catch (NumberFormatException ex) {
    pesanError("Bilangan 1 dan 2 harus angka!");
}

```

2.2.1. 6 Uraian Syntax JButton

Berikut adalah penjelasan lengkap terkait uraian *syntax* diatas:

- 1) Mengecek *input* kosong

```

if (tfBil1.getText().trim().isEmpty()) {
    pesanPeringatan("Bilangan 1 harus diisi!");
    return;
}
if (tfBil2.getText().trim().isEmpty()) {
    pesanPeringatan("Bilangan 2 harus diisi");
    return;
}

```

- 2) Mengambil nilai yang di *input user* menjadi *integer* untuk di proses/ dihitung.

```

try {
    int a = Integer.parseInt(tfBil1.getText());
    int b = Integer.parseInt(tfBil2.getText());

```

- 3) Operasi aritmatika.

```
//Operasi
txtPenjumlahan.setText(String.valueOf(a + b));
txtPengurangan.setText(String.valueOf(a - b));
txtPerkalian.setText(String.valueOf(a * b));
```

Melakukan perhitungan penjumlahan, pengurangan, dan perkalian.

```
if (b == 0) {
    txtPembagian.setText("Error");
    txtMod.setText("Error");
    pesanError("Bilangan 2 tidak boleh 0!");
} else {
    txtPembagian.setText(String.valueOf(a/b));
    txtMod.setText(String.valueOf(a%b));
}
}catch (NumberFormatException ex) {
    pesanError("Bilangan 1 dan 2 harus angka!");
}
```

Selanjutnya pada bagian ini untuk perhitungan pembagian dan *mod*/ sisa bagi, dimana untuk menghindari eror, pembagian dan modulo tidak dapat dilakukan jika bilangan kedua bernilai nol.

- 4) Eror jika *input* bukan angka.

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

2.2 Langkah Kerja

Untuk menyusun sebuah program tentunya perlu untuk merancang algoritma dari program tersebut terlebih dahulu, agar logika program dapat dipahami dengan baik sebelum diubah ke bahasa pemrograman. Algoritma tersebut dapat dibuat dalam bentuk bahasa natural terlebih dahulu, lalu *flowchart*, dan *pseudocode* baru selanjutnya diaplikasikan ke bahasa pemrograman.

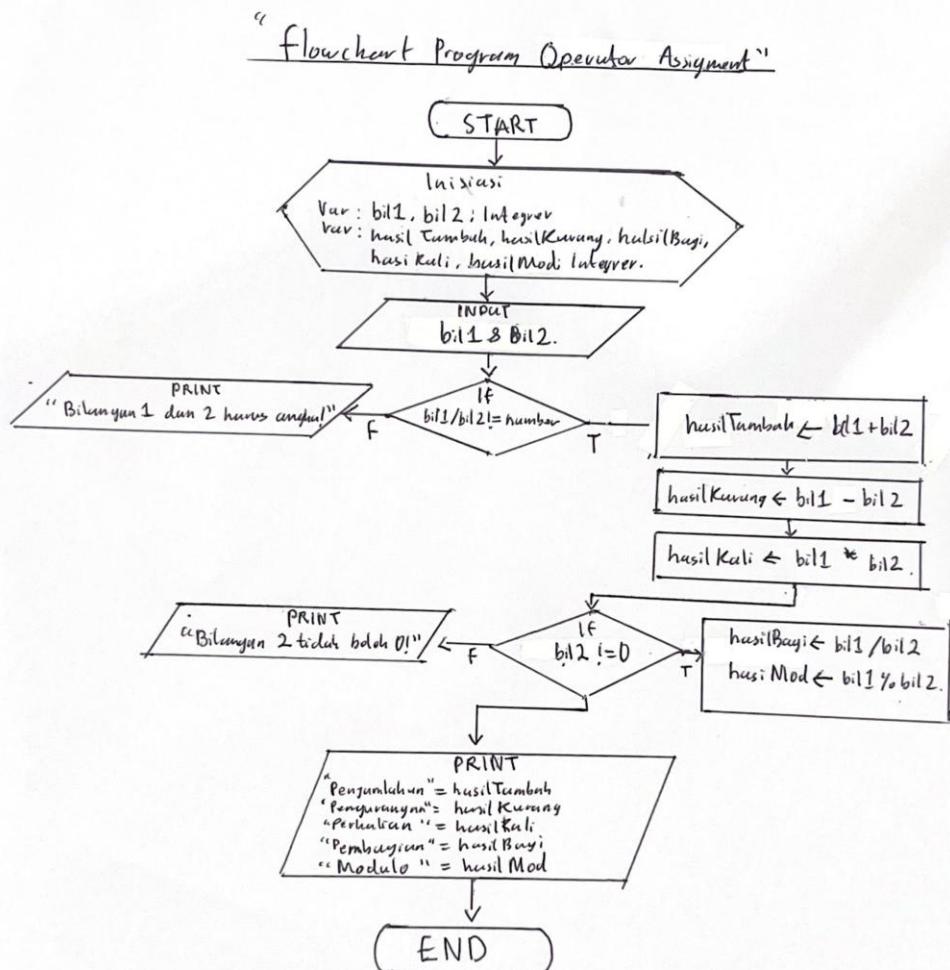
2.2.1 Bahasa Natural

1. *Mulai.*
2. *Inisiasi: bil1, bil2, hasilTambah, hasilKurang, hasilKali, hasilBagi, hasilMod.*
3. *Input bil1, bil2.*
4. *IF bil1 atau bil2 bukan angka, THEN*
5. *PRINT “Bilangan 1 dan 2 harus angka!”*
6. *} ELSE {*
7. *PRINT hasilTambah <- bil1 + bil2;*
8. *PRINT hasilKurang <- bil1 - bil2;*
9. *PRINT hasilKali <- bil1 * bil2;*
10. *IF bil2 != 0, THEN*
11. *PRINT hasilBagi <- bil1 / bil2;*
12. *PRINT hasilMod <- bil1 % bil2;*
13. *} ELSE {*
14. *PRINT “Bilangan 2 tidak boleh 0!”.*
15. *}*
16. *Selesai.*

2.2.2 Flowchart

Nama : Haliya Isma Husna Putri A.
NIM : 2911532002

21/11/2025



2.2.1. 7 Flowchart Program Operator Assigment

2.2.3 Pseudocode

Judul
Program Operator Assigment
{ Program digunakan untuk menampilkan hasil perhitungan aritmataika (jumlahan, pengurangan, perkalian, pembagian, dan modulo) dari dua bilangan yang di <i>input user</i> . }
Deklarasi
Var = bil1, bil2 : Integer Var = hasilTambah, hasilKurang, hasilKali, hasilBagi, hasilMod: Integer
Pseudocode
<ol style="list-style-type: none"> 1. INPUT bil1, bil2; 2. IF bil1, bil2 != number THEN { 3. PRINT "Bilangan 1 dan 2 harus angka!" 4. } ELSE { 5. hasilTambah <- bil1 + bil2; 6. hasilKurang <- bil1 - bil2; 7. hasilKali <- bil1 * bil2; 8. IF bil2 != 0, THEN 9. hasilBagi <- bil1 / bil2; 10. hasilMod <- bil1 % bil2; 11. } ELSE { 12. PRINT "Bilangan 2 tidak boleh 0!". 13. }END IF 14. }END IF 15. PRINT hasilTambah, hasilKurang, hasilKali, hasilBagi, hasilMod. 16. END

2.2.1. 8 Pseudocode Program Assigment

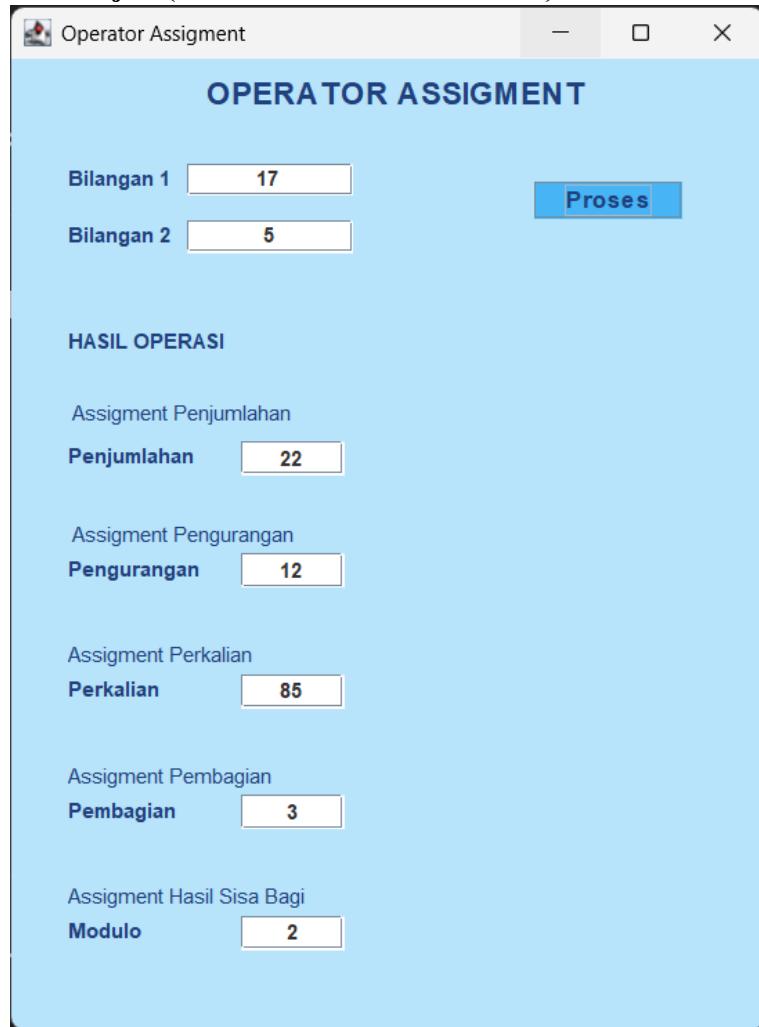
2.3 Analisis Hasil

a) **Hasil Uji 1 (INPUT VALID, MOD 0)**

OPERATOR ASSIGMENT	
Bilangan 1	18
Bilangan 2	6
HASIL OPERASI	
Assigment Penjumlahan	
Penjumlahan	24
Assigment Pengurangan	
Pengurangan	12
Assigment Perkalian	
Perkalian	108
Assigment Pembagian	
Pembagian	3
Assigment Hasil Sisa Bagi	
Modulo	0

2.2.1. 9 Hasil Uji 1 (INPUT VALID, MOD 0)

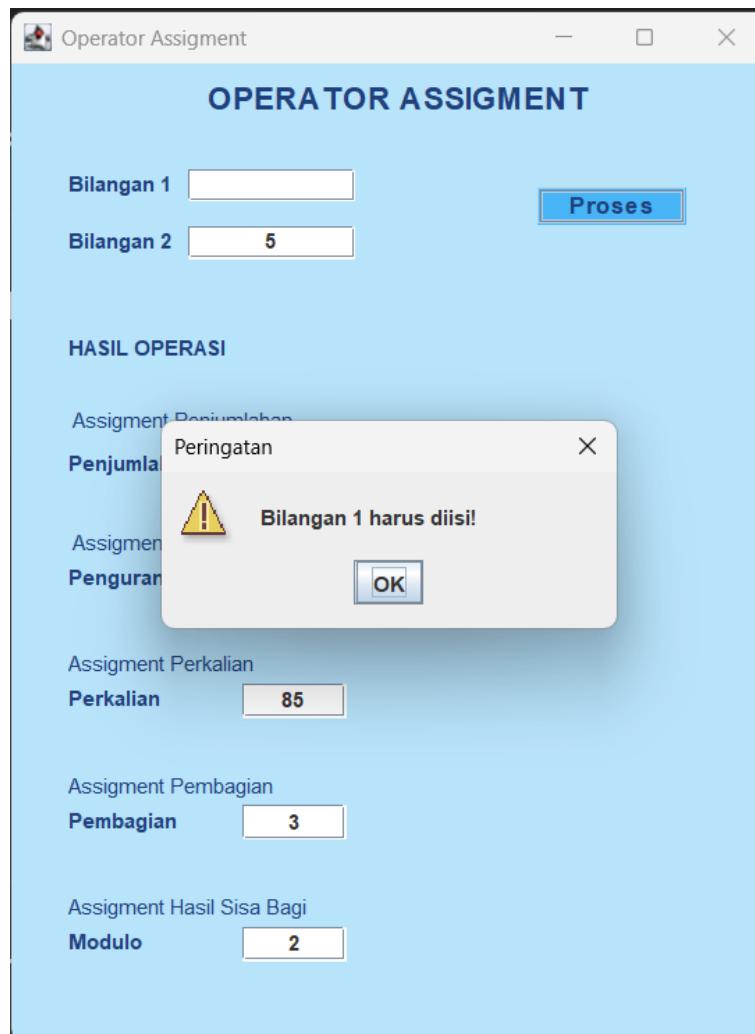
Program menampilkan *output* yang sesuai, dengan *user input* bilangan 1= 18 dan bilangan 2 = 6, lalu hasil operasi aritmatika yaitu penjumlahan, pengurangan, perkalian, pembagian, dan modulo menghasilkan perhitungan yang tepat.

b) Hasil Uji 2 (INPUT VALID, ADA MOD)

2.2.1. 10 Hasil Uji 2 (INPUT VALID, ADA MOD)

Program menampilkan *output* yang sesuai, dengan *user input* bilangan 1= 17 dan bilangan 2 = 5, lalu hasil operasi aritmatika yaitu penjumlahan, pengurangan, perkalian, pembagian, dan modulo menghasilkan perhitungan yang tepat.

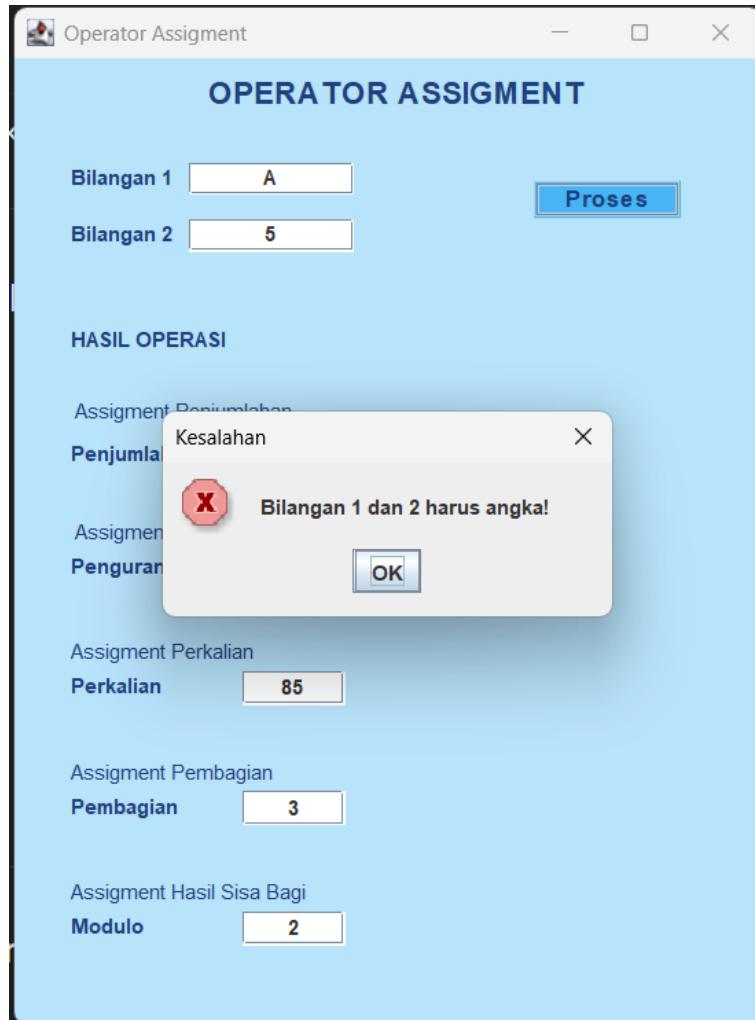
c) Hasil Uji 3 (Kasus Bilangan 1 tidak diisi/ kosong)



2.2.1. 11 Hasil Uji 3 (Kasus Bilangan 1 tidak diisi/ kosong)

Program menampilkan *output* yang sesuai, dengan *user input* bilangan 1= “ ” dan bilangan 2 = 5 program langsung memunculkan pesan peringatan: “*Bilangan 1 harus diisi!*”.

d) Hasil Uji 4 (Kasus Bilangan 1 Bukan Angka)



2.2.1. 12 Hasil Uji 4 (Kasus Bilangan 1 Bukan Angka)

Program menampilkan *output* yang sesuai, dengan *user input* bilangan 1 = “A” dan bilangan 2 = 5 program langsung memunculkan pesan peringatan: “Bilangan 1 dan 2 harus angka!”.

Berdasarkan hasil pengujian keseluruhan, dapat disimpulkan bahwa program berjalan dengan baik dan sesuai sesuai dengan logika yang telah dirancang. Validasi *input*, perhitungan aritmatika, penanganan kesalahan, dan tampilan *output* telah sesuai. Program juga mampu menangani berbagai kondisi, seperti *input kosong*, *input* bukan angka, dan pembagian dengan nol, sehingga membuktikan bahwa fitur *GUI*, validasi, dan “*Operasi Assigmen*” telah diimplementasikan secara benar dan efektif.

BAB III

KESIMPULAN

3.1 Hasil Praktikum

Berdasarkan hasil praktikum dan pengujian program “*Operasi Assigment*” basis *Graphical User Interface* (GUI) yang telah dilakukan, dapat disimpulkan bahwa program berhasil dijalankan dengan baik sesuai ketentuan yang diberikan. Program mampu menerima *input* dua bilangan, melakukan proses perhitungan *operasi assignment* (penjumlahan, pengurangan, perkalian, pembagian, dan modulo atau program yang memproses operasi aritmatika), serta menampilkan hasil perhitungan secara langsung pada antarmuka *user*.

Program bekerja dengan baik dan sesuai ketentuan dimana:

1. Penerapan *GUI* menggunakan *library Java Swing* melalui komponen seperti *JFrame*, *JPanel*, *JLabel*, *JTextField*, dan *JButton*.
2. Program mampu melakukan operasi aritmatika dasar, yaitu penjumlahan, pengurangan, perkalian, pembagian, dan modulo.
3. Terdapat validasi *input* yang dilakukan, pengecekan *input* kosong, *input* bukan angka, serta penanganan kasus ketika bilangan 2 bernilai 0 (untuk pembagian dan modulo).

Secara keseluruhan, program berjalan dengan baik, menghasilkan *output* yang sesuai dan mampu menerapkan semua ketentuan yang diminta.

3.2 Saran

Diharapkan mahasiswa dapat lebih teliti dalam memahami dan menyusun logika perhitungan serta memahami penggunaan komponen *GUI* secara lebih mendalam agar tampilan program semakin optimal. Penulisan *syntax* juga perlu diperhatikan dengan baik, khususnya pada bagian validasi *input* untuk mencegah *error* yang tidak diinginkan.

DAFTAR PUSTAKA

- [1] Eclipse Foundation, “WindowBuilder – Eclipse Project.” [Daring]. Tersedia pada: <https://eclipse.dev/windowbuilder/> . [Diakses: 11-Nov-2025].
- [2] E. H. Chua, “Java GUI Programming – Swing (JFC).” [Daring]. Tersedia pada: https://www3.ntu.edu.sg/home/ehchua/programming/java/J4a_GUI.html. [Diakses: 11-Nov-2025].
- [3] GeeksforGeeks, “Introduction to Java Swing.” [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/java/introduction-to-java-swing/>. [Diakses: 11-Nov-2025].