

LAPORAN TUGAS  
ALGORITMA DAN PEMROGRAMAN  
IMPLEMENTASI GUI DALAM OPERATOR ASSIGNMENT  
MENGUNAKAN BAHASA JAVA



disusun Oleh:  
Annisa Layli Ramadhani  
2511532024  
Informatika Kelas B

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

DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
TAHUN 2025

## **KATA PENGANTAR**

Puji syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan praktikum pemrograman Java ini dengan baik. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum sekaligus untuk memperdalam pemahaman mengenai konsep dasar dan penerapan bahasa pemrograman Java. Dalam penyusunan laporan ini, penulis mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, serta rekan-rekan yang telah memberikan bimbingan, dukungan, dan kerja sama selama proses praktikum berlangsung. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi penulis maupun pembaca.

Padang, 22 November 2025

Penulis

## **DAFTAR ISI**

<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>4</b>
<b>1.1 Latar Belakang .....</b>	<b>4</b>
<b>1.2 Tujuan .....</b>	<b>5</b>
<b>1.3 Manfaat .....</b>	<b>5</b>
<b>BAB II PEMBAHASAN .....</b>	<b>6</b>
<b>2.1 Pseudocode .....</b>	<b>6</b>
<b>2.2 Flowchart .....</b>	<b>7</b>
<b>2.3 Kode Program.....</b>	<b>8</b>
<b>2.4 Output Program dan Analisis Program .....</b>	<b>13</b>
<b>BAB III PENUTUP .....</b>	<b>15</b>
<b>3.1 Kesimpulan .....</b>	<b>15</b>
<b>3.2 Saran.....</b>	<b>15</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pemrograman komputer merupakan bagian penting dalam proses pengolahan data dan pengembangan aplikasi yang berfungsi untuk membantu berbagai kebutuhan manusia. Pada bahasa pemrograman Java, pengolahan nilai numerik menjadi salah satu dasar yang harus dikuasai sejak awal, terutama terkait penggunaan operator dan ekspresi aritmatika. Salah satu materi fundamental yang sering digunakan dalam berbagai perhitungan adalah operator assignment, yaitu operator yang menggabungkan proses perhitungan dengan proses penugasan variabel, seperti `+=`, `-=`, `*=`, `/=`, dan `%=`. Operator ini memungkinkan proses perhitungan dilakukan dengan lebih ringkas, efisien, dan mudah dipahami.

Seiring dengan perkembangan teknologi dan kebutuhan aplikasi yang lebih interaktif, penggunaan Graphical User Interface (GUI) menjadi semakin penting. GUI memberikan tampilan visual yang lebih mudah digunakan dibandingkan antarmuka berbasis teks, sehingga pengguna dapat memasukkan data dan melihat hasil perhitungan dengan lebih intuitif. Oleh karena itu, penggabungan konsep operator assignment dengan pembuatan aplikasi GUI menjadi latihan yang relevan dalam meningkatkan kemampuan logika dan pemahaman struktur program.

Pembuatan aplikasi GUI sederhana untuk menghitung hasil operator assignment memberikan kesempatan untuk menerapkan konsep dasar Java Swing, seperti penggunaan `JFrame`, `JPanel`, `JLabel`, `JTextField`, dan `JButton`. Selain itu, proses ini juga kemungkinan kesalahan seperti input yang tidak valid. Dengan demikian, laporan ini disusun untuk menjelaskan proses pembuatan program GUI operator assignment, fungsi setiap komponen yang digunakan, serta bagaimana logika operasi dijalankan di dalam aplikasi.

## 1.2 Tujuan

Tujuan dari tugas praktikum antara lain sebagai berikut:

- 1.2.1 Menerapkan konsep operator assignment seperti  $+=$ ,  $-=$ ,  $*=$ ,  $/=$ , dan  $\%=$  dalam sebuah program nyata sehingga mahasiswa memahami cara kerja operator tersebut dalam pengolahan data numerik.
- 1.2.2 Mengembangkan keterampilan membuat aplikasi berbasis GUI menggunakan komponen Java Swing seperti JFrame, JLabel, JTextField, dan JButton.
- 1.2.3 Mengimplementasikan validasi input, khususnya dalam menangani kesalahan seperti input bukan angka atau pembagian dengan nol.

## 1.3 Manfaat

Manfaat dari tugas praktikum antara lain sebagai berikut:

- 1.3.1 Mempermudah pemahaman konsep operator assignment, karena pengguna dapat langsung melihat hasil perubahan nilai variabel melalui tampilan GUI.
- 1.3.2 Menambah pengalaman dalam pembuatan aplikasi GUI
- 1.3.3 Membantu mengasah kemampuan debugging dan error handling, terutama ketika terjadi input yang tidak valid.

## BAB II

### PEMBAHASAN

#### 2.1 Pseudocode

Pseudocode digunakan untuk menggambarkan langkah-langkah logis dari sebuah program secara terstruktur namun tanpa bergantung pada sintaks bahasa pemrograman tertentu. Penyajian pseudocode pada program GUI operator assignment ini bertujuan untuk memudahkan pemahaman alur kerja aplikasi, mulai dari penerimaan input, proses pengolahan menggunakan operator assignment, hingga menampilkan hasil perhitungan kepada pengguna. Dengan adanya pseudocode, proses analisis dan implementasi program dapat dilakukan dengan lebih jelas, sistematis, dan mudah ditelusuri.

<b>Judul</b> Program GUI Operator Assignment menggunakan Java Swing
<b>Deklarasi Variabel</b> A1, A2 : Integer Plus, minus, times, div, mod : Integer <b>Komponen GUI</b> JFrame, JPanel, JLabel, JTextField, JButton
<b>Algoritma</b> <ol style="list-style-type: none"><li>1. Mulai</li><li>2. Buat jendela GUI dengan komponen:<ul style="list-style-type: none"><li>• TextField untuk input A1 dan A2</li><li>• Tombol Proses</li><li>• TextField untuk menampilkan hasil operasi assignment</li></ul></li><li>3. Ketika tombol proses ditekan:<ol style="list-style-type: none"><li>a. Ambil nilai dari txtA1 dan txtA2</li><li>b. Konversikan nilai input menjadi tipe integer</li><li>c. Salin nilai A1 ke variabel:<ul style="list-style-type: none"><li>• Plus = A1</li><li>• Minus = A1</li></ul></li></ol></li></ol>

- Times = A1
- Div = A1
- Mod = A1

d. Lakukan operasi assignmen:

- Plus += A2
- Minus -= A2
- Times \*= A2
- Div /= A2
- Mod %= A2

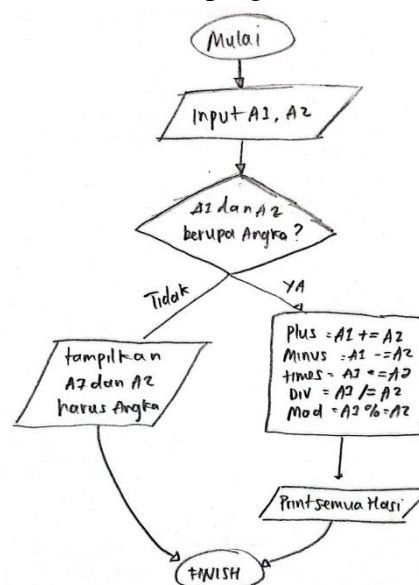
e. Tampilkan hasil masing-masing operasi ke textfield output

f. Jika input bukan angka, tampilkan pesan error.

4. Selesai

## 2.2 Flowchart

Flowchart digunakan untuk menggambarkan alur logika program secara visual sehingga setiap tahapan proses dapat dipahami dengan jelas. Melalui diagram ini, urutan operasi, percabangan keputusan, serta penanganan kesalahan pada program dapat ditinjau secara sistematis. Penyajian flowchart membantu memastikan bahwa logika perhitungan operator assignment telah disusun dengan benar sebelum diimplementasikan ke dalam kode program.



## 2.3 Kode Program

```

1.  package pekan8_2511532024;
2.
3.  import java.awt.EventQueue;
4.
5.  import javax.swing.JFrame;
6.  import javax.swing.JPanel;
7.  import javax.swing.border.EmptyBorder;
8.  import javax.swing.JLabel;
9.  import javax.swing.JOptionPane;
10.
11. import java.awt.Font;
12. import javax.swing.SwingConstants;
13. import java.awt.Color;
14. import javax.swing.JTextField;
15. import javax.swing.JButton;
16. import java.awt.event.ActionListener;
17. import java.awt.event.ActionEvent;
18.
19. public class tugasAlproPekan8_2511532024 extends JFrame {
20.
21.     private static final long serialVersionUID = 1L;
22.     protected static final int A1 = 0;
23.     protected static final int A2 = 0;
24.     private JPanel contentPane;
25.     private JTextField txtA1;
26.     private JTextField txtA2;
27.     private JTextField txtPlus;
28.     private JTextField txtMinus;
29.     private JTextField txtTimes;
30.     private JTextField txtDiv;
31.     private JTextField txtMod;
32.
33.     private void pesanPeringatan(String pesan) {
34.         JOptionPane.showMessageDialog(this, pesan,
35. "peringatan", JOptionPane.WARNING_MESSAGE);
36.     }
37.     private void pesanError(String pesan) {
38.         JOptionPane.showMessageDialog(this,
39 pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
40.     }
41.
42.     /**
43.      * Launch the application.
44.      */
45.     public static void main(String[] args) {
46.         EventQueue.invokeLater(new Runnable() {
47.             public void run() {
48.                 try {
49.                     tugasAlproPekan8_2511532024 frame = new
50. tugasAlproPekan8_2511532024();
51.                     frame.setVisible(true);
52.                 } catch (Exception e) {
53.                     e.printStackTrace();
54.                 }
55.             }
56.         });
57.     }
58. }

```



```

52.         }
53.     });
54. }
55.
56. /**
57.  * Create the frame.
58.  */
59. public tugasAlproPekan8_2511532024() {
60.     setTitle("OPERATOR ASSIGNMENT");
61.
62.     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
63.     setBounds(100, 100, 338, 349);
64.     contentPane = new JPanel();
65.     contentPane.setBackground(Color.GRAY);
66.     contentPane.setBorder(new EmptyBorder(5, 5, 5,
57. 5));
67.     setContentPane(contentPane);
68.     contentPane.setLayout(null);
69.
70.     JLabel lblNewLabel = new JLabel("OPERATOR
ASSIGNMENT");
71.     lblNewLabel.setForeground(Color.BLACK);
72.     lblNewLabel.setHorizontalAlignment(SwingConstants.CE
NTER);
73.     lblNewLabel.setFont(new Font("Times New
Roman", Font.BOLD, 19));
74.     lblNewLabel.setBounds(43, 11, 243, 14);
75.     contentPane.add(lblNewLabel);
76.
77.     JLabel lblNewLabel_1 = new JLabel("Bilangan
1");
78.     lblNewLabel_1.setForeground(Color.BLACK);
79.     lblNewLabel_1.setFont(new Font("Times New
Roman", Font.BOLD, 13));
80.     lblNewLabel_1.setHorizontalAlignment(SwingConstants.
CENTER);
81.     lblNewLabel_1.setBounds(35, 50, 80, 14);
82.     contentPane.add(lblNewLabel_1);
83.
84.     JLabel lblNewLabel_1_1 = new JLabel("Bilangan
2");
85.     lblNewLabel_1_1.setHorizontalAlignment(SwingConstant
s.CENTER);
86.     lblNewLabel_1_1.setForeground(Color.BLACK);
87.     lblNewLabel_1_1.setFont(new Font("Times New
Roman", Font.BOLD, 13));
88.     lblNewLabel_1_1.setBounds(35, 75, 80, 14);
89.     contentPane.add(lblNewLabel_1_1);
90.
91.     JLabel lblNewLabel_1_2 = new JLabel("Hasil
Operasi:");
92.     lblNewLabel_1_2.setHorizontalAlignment(SwingConstant
s.CENTER);

```

```

92.         lblNewLabel_1_2.setForeground(Color.BLACK);
93.         lblNewLabel_1_2.setFont(new Font("Times New
Roman", Font.BOLD, 14));
94.         lblNewLabel_1_2.setBounds(64, 110, 94, 14);
95.         contentPane.add(lblNewLabel_1_2);
96.
97.         JLabel lblNewLabel_1_3 = new
JLabel("Penjumlahan");
98.
        lblNewLabel_1_3.setHorizontalAlignment(SwingConstants
s.CENTER);
99.         lblNewLabel_1_3.setForeground(Color.BLACK);
100.        lblNewLabel_1_3.setFont(new Font("Times New
Roman", Font.BOLD, 13));
101.        lblNewLabel_1_3.setBounds(35, 147, 80, 14);
102.        contentPane.add(lblNewLabel_1_3);
103.
104.        JLabel lblNewLabel_1_4 = new
JLabel("Pengurangan");
105.
        lblNewLabel_1_4.setHorizontalAlignment(SwingConstants
s.CENTER);
106.        lblNewLabel_1_4.setForeground(Color.BLACK);
107.        lblNewLabel_1_4.setFont(new Font("Times New
Roman", Font.BOLD, 13));
108.        lblNewLabel_1_4.setBounds(35, 178, 80, 14);
109.        contentPane.add(lblNewLabel_1_4);
110.
111.        JLabel lblNewLabel_1_5 = new
JLabel("Perkalian");
112.
        lblNewLabel_1_5.setHorizontalAlignment(SwingConstants
s.CENTER);
113.        lblNewLabel_1_5.setForeground(Color.BLACK);
114.        lblNewLabel_1_5.setFont(new Font("Times New
Roman", Font.BOLD, 13));
115.        lblNewLabel_1_5.setBounds(35, 211, 80, 14);
116.        contentPane.add(lblNewLabel_1_5);
117.
118.        JLabel lblNewLabel_1_6 = new
JLabel("Pembagian");
119.
        lblNewLabel_1_6.setHorizontalAlignment(SwingConstants
s.CENTER);
120.        lblNewLabel_1_6.setForeground(Color.BLACK);
121.        lblNewLabel_1_6.setFont(new Font("Times New
Roman", Font.BOLD, 13));
122.        lblNewLabel_1_6.setBounds(35, 243, 80, 14);
123.        contentPane.add(lblNewLabel_1_6);
124.
125.        JLabel lblNewLabel_1_7 = new JLabel("Sisa
Hasil");
126.
        lblNewLabel_1_7.setHorizontalAlignment(SwingConstants
s.CENTER);
127.        lblNewLabel_1_7.setForeground(Color.BLACK);

```

```

128.         lblNewLabel_1_7.setFont(new Font("Times New
Roman", Font.BOLD, 13));
129.         lblNewLabel_1_7.setBounds(35, 275, 80, 14);
130.         contentPane.add(lblNewLabel_1_7);
131.
132.         txtA1 = new JTextField();
133.         txtA1.setBackground(Color.LIGHT_GRAY);
134.
135.         txtA1.setHorizontalAlignment(SwingConstants.CENTER);
136.         txtA1.setFont(new Font("Times New Roman",
Font.BOLD, 12));
137.         txtA1.setBounds(125, 47, 66, 20);
138.         contentPane.add(txtA1);
139.         txtA1.setColumns(10);
140.
141.         txtA2 = new JTextField();
142.         txtA2.setBackground(Color.LIGHT_GRAY);
143.
144.         txtA2.setHorizontalAlignment(SwingConstants.CENTER);
145.         txtA2.setFont(new Font("Times New Roman",
Font.BOLD, 12));
146.         txtA2.setBounds(125, 72, 66, 20);
147.         contentPane.add(txtA2);
148.         txtA2.setColumns(10);
149.
150.         txtPlus = new JTextField();
151.         txtPlus.setBackground(Color.LIGHT_GRAY);
152.
153.         txtPlus.setHorizontalAlignment(SwingConstants.CENTER
);
154.         txtPlus.setFont(new Font("Times New Roman",
Font.BOLD, 12));
155.         txtPlus.setEditable(false);
156.         txtPlus.setBounds(125, 144, 66, 20);
157.         contentPane.add(txtPlus);
158.         txtPlus.setColumns(10);
159.
160.         txtMinus = new JTextField();
161.         txtMinus.setBackground(Color.LIGHT_GRAY);
162.
163.         txtMinus.setHorizontalAlignment(SwingConstants.CENTE
R);
164.         txtMinus.setFont(new Font("Times New Roman",
Font.BOLD, 12));
165.         txtMinus.setEditable(false);
166.         txtMinus.setBounds(125, 175, 66, 20);
167.         contentPane.add(txtMinus);
168.         txtMinus.setColumns(10);
169.
170.         txtTimes = new JTextField();
171.         txtTimes.setBackground(Color.LIGHT_GRAY);
172.
173.         txtTimes.setHorizontalAlignment(SwingConstants.CENTE
R);
174.         txtTimes.setFont(new Font("Times New Roman",
Font.BOLD, 12));
175.         txtTimes.setEditable(false);

```

```

171.         txtTimes.setBounds(125, 208, 66, 20);
172.         contentPane.add(txtTimes);
173.         txtTimes.setColumns(10);
174.
175.         txtDiv = new JTextField();
176.         txtDiv.setBackground(Color.LIGHT_GRAY);
177.
178.         txtDiv.setHorizontalAlignment(SwingConstants.CENTER);
179.         txtDiv.setFont(new Font("Times New Roman",
Font.BOLD, 12));
180.         txtDiv.setEditable(false);
181.         txtDiv.setBounds(125, 240, 66, 20);
182.         contentPane.add(txtDiv);
183.         txtDiv.setColumns(10);
184.
185.         txtMod = new JTextField();
186.         txtMod.setBackground(Color.LIGHT_GRAY);
187.
188.         txtMod.setHorizontalAlignment(SwingConstants.CENTER);
189.         txtMod.setFont(new Font("Times New Roman",
Font.BOLD, 12));
190.         txtMod.setEditable(false);
191.         txtMod.setBounds(125, 272, 66, 20);
192.         contentPane.add(txtMod);
193.         txtMod.setColumns(10);
194.
195.         JButton btnNewButton = new JButton("PROSES");
196.         btnNewButton.setBackground(Color.LIGHT_GRAY);
197.         btnNewButton.addActionListener(new
ActionListener() {
198.             public void actionPerformed(ActionEvent
e) {
199.
200.                 if
201.                 (txtA1.getText().trim().isEmpty()) {
202.                     pesanPeringatan("Bilangan
203.                     1 harus diisi");
204.                 } else if
205.                 (txtA2.getText().trim().isEmpty()) {
206.                     pesanPeringatan("Bilangan
207.                     2 harus diisi");
208.                 }
209.
210.                 try {
211.                     int A1 =
212.                     Integer.parseInt(txtA1.getText());
213.                     int A2 =
214.                     Integer.parseInt(txtA2.getText());
215.
216.                     int plus = A1; plus += A2;
217.                     int minus = A1; minus -= A2;
218.                     int times = A1; times *= A2;
219.                     int div = A1; div /= A2;
220.                     int mod = A1; mod %= A2;
221.
222.                     txtPlus.setText(String.valueOf(plus));

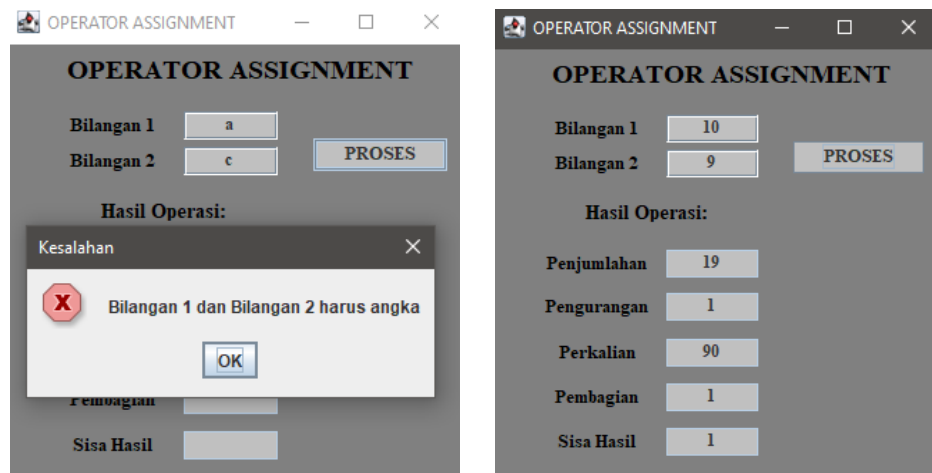
```

```

215.     txtMinus.setText(String.valueOf(minus));
216.     txtTimes.setText(String.valueOf(times));
217.         txtDiv.setText(String.valueOf(div));
218.         txtMod.setText(String.valueOf(mod));
219.
220.     } catch (NumberFormatException ex)
221.     {
222.         pesanError("Bilangan 1 dan
223.         Bilangan 2 harus angka");
224.     }
225. });
226.     btnNewButton.setFont(new Font("Times New
227.     Roman", Font.BOLD, 13));
228.     btnNewButton.setBounds(216, 66, 94, 23);
229.     contentPane.add(btnNewButton);
230. }
231.
232. }

```

## 2.4 Output Program dan Analisis Program



Program operator assignment yang dibuat menampilkan dua jenis output, yaitu output ketika input tidak valid dan output ketika input valid. Pada kondisi pertama, pengguna memasukkan nilai bilangan 1 dan bilangan 2 berupa huruf, sehingga proses konversi teks ke tipe data integer melalui perintah `Integer.parseInt()` gagal dilakukan. Kesalahan ini menimbulkan `NumberFormatException` yang kemudian ditangani oleh blok `catch`. Akibatnya, program tidak melakukan perhitungan apa pun dan langsung menampilkan pesan kesalahan “Bilangan 1 dan Bilangan 2

harus angka”. Hal ini menunjukkan bahwa mekanisme validasi input bekerja dengan benar, karena program mampu mendeteksi data yang tidak sesuai dan memberikan informasi yang jelas kepada pengguna sebelum melanjutkan proses perhitungan.

Pada kondisi kedua, pengguna memberikan input yang valid, yaitu angka 10 sebagai Bilangan 1 dan angka 9 sebagai Bilangan 2. Ketika tombol proses ditekan, program menjalankan lima operasi menggunakan operator assignment, yaitu  $+=$ ,  $-=$ ,  $*=$ ,  $/=$ ,  $\%=$ . Setiap operasi menggunakan nilai awal A1 sehingga hasil perhitungan tidak saling memengaruhi. Hasil penjumlahan menghasilkan 19, hasil pengurangan menghasilkan 1, dan hasil perkalian menghasilkan 90. Untuk operasi pembagian, tipe data integer menyebabkan hasil pembagian 10 dibagi 9 menjadi 1 karena pembulatan ke bawah. Sementara itu, operasi modulus menghasilkan sisa pembagian  $10 \bmod 9$  yaitu 1. Seluruh hasil perhitungan ditampilkan ke dalam field GUI sesuai dengan kolom masing-masing.

Secara keseluruhan, program berjalan sesuai logika yang telah dirancang. Validasi input bekerja efektif untuk mencegah kesalahan perhitungan, sementara seluruh operator assignment menghasilkan output konsisten dengan aturan perhitungan dalam bahasa pemrograman Java. Tampilan output juga tersaji dengan rapi sehingga memudahkan pengguna dalam memahami hasil setiap operasi

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Program operator assignment ini bekerja dengan baik karena mampu memvalidasi input sebelum melakukan perhitungan. Jika nilai yang dimasukkan bukan angka, program langsung menampilkan pesan kesalahan sehingga mencegah proses yang salah. Ketika input valid, semua operasi seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus dapat dihitung dan ditampilkan dengan benar. Flowchart yang dibuat juga menggambarkan alur kerja program secara jelas, mulai dari input, pengecekan kondisi, proses operasi, hingga hasil akhir.

#### **3.2 Saran**

Untuk meningkatkan kejelasan dan keandalan program, disarankan agar validasi input dibuat lebih rinci sehingga pengguna dapat memahami kesalahan dengan mudah. Selain itu, penyusunan flowchart sebaiknya dibuat konsisten pada setiap percabangan agar alur keputusan tidak membingungkan. Menambahkan contoh penggunaan dan penjelasan singkat pada tampilan GUI juga dapat membantu pengguna memahami fungsi setiap bagian program.

## DAFTAR PUSTAKA

- [1] Oracle, *Java Platform, Standard Edition Documentation*. Oracle Corporation, 2024. [Online]. Available: <https://docs.oracle.com/javase/>
- [2] TutorialsPoint, *Java Swing Tutorial: Java GUI Programming Made Easy*. 2021. [Online]. Available: <https://www.tutorialspoint.com/swing/index.htm>
- [3] IBM, *Flowcharting Techniques and Best Practices*. 2023. [Online]. Available: <https://www.ibm.com/docs/>