

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN**  
**LAPORAN PRAKTIKUM PEKAN 8**

**Disusun Oleh:**

Syasya Halwa Gazwani  
(2511531018)

**Dosen Pengampu:**

Dr. Wahyudi, S.T, M.T

**Asisten Praktikum:**

Muhammad Zaki Al Hafiz



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

## **KATA PENGANTAR**

Assalamu'alaikum warahmatullahi wabarakatuh,

Segala puji bagi Allah SWT yang telah memberikan kesempatan serta kemudahan kepada penulis untuk dapat menyelesaikan Laporan Praktikum pada mata kuliah Algoritma Pemrograman, sehingga Laporan Praktikum ini dapat dikumpulkan dengan tepat waktu. Atas rahmat dan karunianya Laporan Praktikum dapat terselesaikan dengan baik. Sholawat serta salam juga penulis sampaikan kepada baginda tercinta yaitu Nabi Muhammad SAW. yang kita nantikan syafa'atnya di akhirat nanti. Laporan Praktikum ini bertujuan untuk menambah wawasan para pembaca untuk lebih memperdalam ilmu yang ada pada makalah ini.

Dalam penyusunan Laporan Praktikum ini, penulis mengalami banyak kesulitan dan penulis menyadari bahwa Laporan Praktikum ini masih jauh dari kesempurnaan. Untuk itu, penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan pada Laporan Praktikum ini. Penulis mengucapkan terima kasih kepada bapak Dr. Wahyudi, S.T, M.T. selaku dosen mata kuliah Algoritma Pemrograman yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan sesuai dengan mata kuliah yang penulis tekuni. Penulis juga mengucapkan terima kasih kepada semua pihak yang telah membagi sebagian pengetahuannya sehingga penulis dapat menyelesaikan Laporan Praktikum ini.

Padang, November 2025

**Penulis**

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	i
<b>DAFTAR ISI .....</b>	ii
<b>BAB I PENDAHULUAN.....</b>	1
1.1    Latar Belakang.....	1
1.2    Tujuan .....	1
1.3    Manfaat Praktikum .....	1
<b>BAB II PEMBAHASAN.....</b>	3
2.1    Operator Aritmatika .....	3
2.1.1    Kode Program .....	3
2.1.2    Langkah Kerja .....	5
2.1.3    Analisis Hasil .....	7
<b>KESIMPULAN.....</b>	12
<b>DAFTAR PUSTAKA.....</b>	13

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Pemrograman atau *programming* merupakan sebuah proses menulis, menguji dan memperbaiki (*debug*), serta memelihara kode yang dapat membangun suatu program komputer. Kode ini ditulis dalam berbagai bahasa pemrograman atau sering disebut juga bahasa komputer. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan atau “pekerjaan” sesuai dengan keinginan pemrograman. Untuk melakukan pemrograman, diperlukan keterampilan dalam algoritma, logika, bahasa pemrograman, dan pada banyak kasus, pengetahuan-pengetahuan lain seperti matematika. Bahasa Pemrograman, atau biasanya disebut bahasa komputer atau *computer language programming*, yaitu sebuah instruksi standar untuk mengendalikan sebuah komputer. Seperangkat aturan *syntax* dan *semantic* yang digunakan untuk memberikan sebuah definisi pada program komputer dikenal sebagai bahasa pemrograman. Dengan menggunakan bahasa ini, seorang *programmer* dapat dengan tepat menentukan data yang akan diproses oleh komputer, bagaimana penyimpanan dan transfernya, dan tindakan apa yang harus diambil dalam berbagai keadaan.

### 1.2 Tujuan

- 1.2.1 Memahami konsep dasar tipe data dalam bahasa pemrograman Java serta penerapannya dalam pembuatan program sederhana.
- 1.2.2 Mengembangkan keterampilan analisis terhadap hasil keluaran program sesuai dengan logika yang diterapkan.

### 1.3 Manfaat Praktikum

- 1.2.3 Menambah wawasan dan keterampilan mahasiswa dalam mengelola data menggunakan bahasa pemrograman Java sebagai dasar untuk pemrograman yang lebih lanjut.

1.2.4 Memberikan pemahaman dasar kepada mahasiswa mengenai penggunaan tipe data dalam bahasa pemrograman Java, sehingga dapat menjadi bekal dalam mengembangkan program yang lebih kompleks di tahap berikutnya.

## **BAB II PEMBAHASAN**

## 2.1 Operator Aritmatika

### 2.1.1 Kode Program

```
1 package pekan8_2511531018;
2
3 import java.awt.BorderLayout;
4 import java.awt.EventQueue;
5
6 import javax.swing.JFrame;
7 import javax.swing.JPanel;
8 import javax.swing.border.EmptyBorder;
9 import java.awt.Font;
10 import javax.swing.JLabel;
11 import javax.swing.JOptionPane;
12 import javax.swing.JTextField;
13 import javax.swing.JComboBox;
14 import javax.swing.DefaultComboBoxModel;
15 import javax.swing.SwingConstantsConstants;
16 import javax.swing.JButton;
17 import java.awt.event.ActionListener;
18 import java.awt.event.ActionEvent;
19 import java.awt.Color;
20
21 public class OperatorAritmatikaGUI_211531018 extends JFrame {
22
23     private static final long serialVersionUID = 1L;
24     private JPanel contentPane;
25     private JTextField txtBil1;
26     private JTextField txtBil2;
27     private JTextField txtHasil;
28
29
30
31
32
33     private void pesanPeringatan(String pesan) {
34         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
35     }
36
37     private void pesanError(String pesan) {
38         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
39     }
40
41     /**
42      * Launch the application.
43      */
44     public static void main(String[] args) {
45         EventQueue.invokeLater(new Runnable() {
46             public void run() {
47                 try {
48                     OperatorAritmatikaGUI_211531018 frame = new OperatorAritmatikaGUI_211531018();
49                     frame.setVisible(true);
50                 } catch (Exception e) {
51                     e.printStackTrace();
52                 }
53             }
54         });
55     }
56
57     /**
58      * Create the frame.
59      */
60     public OperatorAritmatikaGUI_211531018() {
61         setFont(new Font("Times New Roman", Font.BOLD, 12));
62         setTitle("Operator Aritmatika\n");
63         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
64         setBounds(100, 100, 450, 302);
65         contentPane = new JPanel();
66         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
67         setContentPane(contentPane);
68         contentPane.setLayout(null);
69
70         JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
71         lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 12));
72         lblNewLabel.setBounds(149, 21, 151, 14);
73         contentPane.add(lblNewLabel);
74     }
75 }
```

```

73     JLabel lblNewLabel_1 = new JLabel("Bilangan 1");
74     lblNewLabel_1.setFont(new Font("Times New Roman", Font.PLAIN, 12));
75     lblNewLabel_1.setBounds(24, 60, 65, 14);
76     contentPane.add(lblNewLabel_1);
77
78     JLabel lblNewLabel_1_1 = new JLabel("Bilangan 2");
79     lblNewLabel_1_1.setFont(new Font("Times New Roman", Font.PLAIN, 12));
80     lblNewLabel_1_1.setBounds(24, 88, 65, 14);
81     contentPane.add(lblNewLabel_1_1);
82
83     JLabel lblNewLabel_1_2 = new JLabel("Operator");
84     lblNewLabel_1_2.setFont(new Font("Times New Roman", Font.PLAIN, 12));
85     lblNewLabel_1_2.setBounds(24, 135, 65, 14);
86     contentPane.add(lblNewLabel_1_2);
87
88     JLabel lblNewLabel_1_2_1 = new JLabel("Hasil");
89     lblNewLabel_1_2_1.setFont(new Font("Times New Roman", Font.PLAIN, 12));
90     lblNewLabel_1_2_1.setBounds(24, 181, 65, 14);
91     contentPane.add(lblNewLabel_1_2_1);
92
93     JTextField txtBill = new JTextField();
94     txtBill.setHorizontalTextPosition(SwingConstants.CENTER);
95     txtBill.setBounds(90, 57, 51, 20);
96     contentPane.add(txtBill);
97     txtBill.setColumns(10);
98
99     JTextField txtBill2 = new JTextField();
100    txtBill2.setHorizontalTextPosition(SwingConstants.CENTER);
101    txtBill2.setColumns(10);
102    txtBill2.setBounds(90, 85, 51, 20);
103    contentPane.add(txtBill2);
104
105    JComboBox cbOperator = new JComboBox();
106    cbOperator.setModel(new DefaultComboBoxModel(new String[]{"+", "-", "*", "/", "%"}));
107    cbOperator.setBounds(90, 131, 51, 22);
108    contentPane.add(cbOperator);
109
110    JTextField txtHasil = new JTextField();
111    txtHasil.setForeground(Color.BLACK);
112    txtHasil.setBackground(new Color(255, 255, 255));
113    txtHasil.setHorizontalTextPosition(SwingConstants.CENTER);
114    txtHasil.setEnabled(false);
115    txtHasil.setEditable(false);
116    txtHasil.setBounds(90, 178, 57, 20);
117    contentPane.add(txtHasil);
118    txtHasil.setColumns(10);
119
120    JButton btnNewButton = new JButton("Proses");
121    btnNewButton.addActionListener(new ActionListener() {
122        int hasil;
123        public void actionPerformed(ActionEvent e) {
124
125            if(txtBill.getText().trim().isEmpty()) {
126                pesanPeringatan("Bilangan 1 harus diisi");
127            } else if (txtBill2.getText().trim().isEmpty()) {
128                pesanPeringatan("Bilangan 2 harus diisi");
129            } else {
130                try {
131                    int a = Integer.parseInt(txtBill.getText());
132                    int b = Integer.parseInt(txtBill2.getText());
133                    int c = cbOperator.getSelectedIndex(); //operator
134                    if (c==0) {
135                        hasil = a+b;
136                    }
137                    if (c==1) {
138                        hasil = a-b;
139                    }
140                    if (c==2) {
141                        hasil = a*b;
142                    }
143                    if (c==3) {
144                        hasil = a/b;
145                    }
146                    if (c==4) {
147                        hasil = a%b;
148                    }
149                } catch (NumberFormatException ex) {
150                    pesanError("Bilangan 1 dan Bilangan 2 harus angka");
151                }
152            }
153            txtHasil.setText(String.valueOf(hasil));
154        }
155    });
156    btnNewButton.setBounds(179, 131, 89, 23);
157    contentPane.add(btnNewButton);
158

```

Gambar 2.1.1 Kode program Operator Aritmatika.

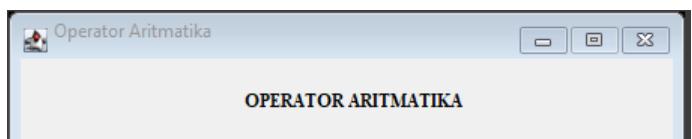
## 2.1.2 Langkah Kerja

- 2.1.2.1 Membuat *class* bernama Operator Aritmatika, sehingga pada bagian atas desain akan muncul tulisan “Operator Aritmatika.”



Gambar 2.1.2.1 Membuat *class* Operator Aritmatika.

- 2.1.2.2 Pada bagian **Palette**, tambahkan **JLabel** di bagian **Components** dan menuliskan “Operator Aritmatika” sebagai judul.



Gambar 2.1.2.2 Menambahkan judul pada desain.

- 2.1.2.3 Menambahkan **JLabel** untuk menuliskan Bilangan 1 dan Bilangan 2 dan menambahkan **JTextField** sebagai *input* angka yang akan dimasukkan oleh pengguna. Pada **JTextField**, ubah text pada variabel menjadi **txtBil1** dan **txtBil2**.



Gambar 2.1.2.3 Menambahkan Bilangan 1 dan Bilangan 2.

- 2.1.2.4 Menambahkan **JLabel** untuk menuliskan Operator.
- 2.1.2.5 Menambahkan **JComboBox** sebagai daftar pilihan dalam bentuk *dropdown* agar pengguna dapat memilih salah satu item dalam daftar pilihan. Dalam program ini digunakan tanda penambahan (+), pengurangan (-), perkalian (\*), pembagian (/), dan sisa bagi (%) (dimasukkan pada model). Dan mengubah *editable* menjadi *false*.
- 2.1.2.6 Menambahkan **JButton** yang menghasilkan aksi ketika ditekan oleh pengguna (Proses).



Gambar 2.1.2.4 Menambahkan Operator.

- 2.1.2.7 Menambahkan **JLabel** untuk menuliskan Hasil.
- 2.1.2.8 Menambahkan **JTextField** untuk menampilkan hasil dari angka yang diinputkan oleh pengguna. Mengubah *editable* dan *enabled* menjadi *false* dan mengubah text variabel menjadi **txtHasil**.



Gambar 2.1.2.8 Menambahkan Hasil.

- 2.1.2.9 Untuk **JTextField**, **JComboBox**, dan **JButton** pada bagian **horizontalAlignment** diubah menjadi **CENTER**.

- 2.1.2.10 Deklarasi variabel hasil dan menambahkan **Method actionPerformed(ActionEvent e)** sebagai aksi yang dijalankan ketika tombol ditekan.

```
125          int hasil;
126  public void actionPerformed(ActionEvent e) {
```

Gambar 2.1.2.10 Deklarasi variabel dan Method actionPerformed.

- 2.1.2.11 Validasi input yaitu jika bilangan 1 atau bilangan 2 kosong, maka muncul peringatan.

```
128          if(txtBil1.getText().trim().isEmpty()) {
129              pesanPeringatan("Bilangan 1 harus diisi");
130          } else if (txtBil2.getText().trim().isEmpty()) {
131              pesanPeringatan("Bilangan 2 harus diisi");
```

Gambar 2.1.2.11 Validasi input.

- 2.1.2.12 Int a berguna untuk mengambil angka dari **TextField 1**, int b berguna untuk mengambil angka dari **TextField 2**, dan int c untuk mengetahui operator mana yang dipilih di **JComboBox** berdasarkan *index*.

```
134          int a = Integer.parseInt(txtBil1.getText());
135          int b = Integer.parseInt(txtBil2.getText());
136          int c = cbOperator.getSelectedIndex(); //operator
```

Gambar 2.1.2.12 Nilai integer.

- 2.1.2.13 Logika operasi (menggunakan *index* operator). 0 → tambah, 1 → kurang, 2 → kali, 3 → bagi, dan 4 → modulus (sisa bagi).

```

137          if (c==0) {
138              hasil = a+b;
139          }
140          if (c==1) {
141              hasil = a-b;
142          }
143          if (c==2) {
144              hasil = a*b;
145          }
146          if (c==3) {
147              hasil = a/b;
148          }
149          if (c==4) {
150              hasil = a%b;

```

Gambar 2.1.2.13 Logika Operasi.

2.1.2.14 Penanganan error jika angka tidak valid (jika pengguna menginput huruf atau simbol).

```

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

```

Gambar 2.1.2.14 Penanganan error.

2.1.2.15 Method pesan Peringatan yang menampilkan kotak dialog yang isinya peringatan dan Method pesan Error yang menampilkan dialog error.

```

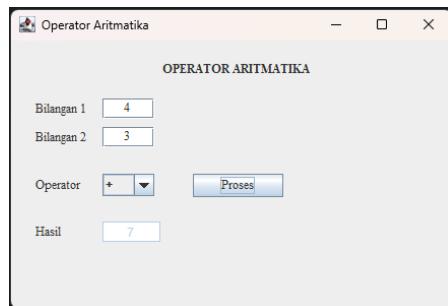
33 private void pesanPeringatan(String pesan) {
34     JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
35 }
36 private void pesanError(String pesan) {
37     JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
38 }

```

Gambar 2.1.2.15 Pesan Peringatan dan pesan Error.

## 2.1.3 Analisis Hasil

### 2.1.3.1 Penjumlahan

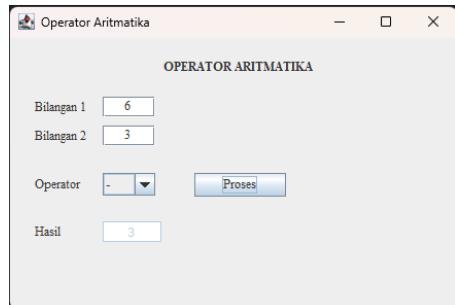


Gambar 2.1.3.1 Output Penjumlahan.

Pada output diatas, pengguna menginput angka 4 pada Bilangan 1 dan menginput angka 3 pada Bilangan 2. Pada Operator dipilih penjumlahan untuk menjumlahkan angka pada Bilangan 1 dan Bilangan 2. Pengguna menekan tombol

Proses untuk mencetak hasil dari penjumlahan Bilangan 1 dan Bilangan 2.

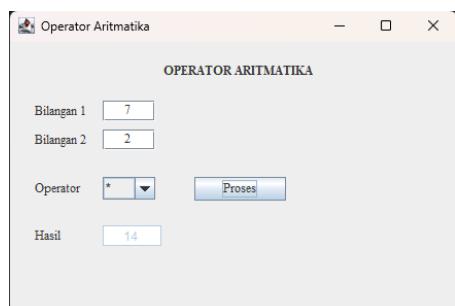
#### 2.1.3.2 Pengurangan



Gambar 2.1.3.2 Output Pengurangan.

Pada output diatas, pengguna menginput angka 6 pada Bilangan 1 dan menginput angka 3 pada Bilangan 2. Pada Operator dipilih pengurangan untuk mengurangkan angka pada Bilangan 1 dan Bilangan 2. Pengguna menekan tombol Proses untuk mencetak hasil dari pengurangan Bilangan 1 dan Bilangan 2.

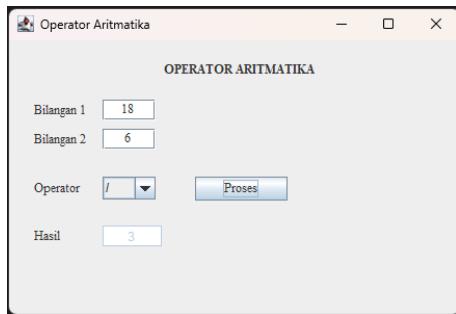
#### 2.1.3.3 Perkalian



Gambar 2.1.3.3 Output Perkalian.

Pada output diatas, pengguna menginput angka 7 pada Bilangan 1 dan menginput angka 2 pada Bilangan 2. Pada Operator dipilih perkalian untuk mengalikan angka pada Bilangan 1 dan Bilangan 2. Pengguna menekan tombol Proses untuk mencetak hasil dari perkalian Bilangan 1 dan Bilangan 2.

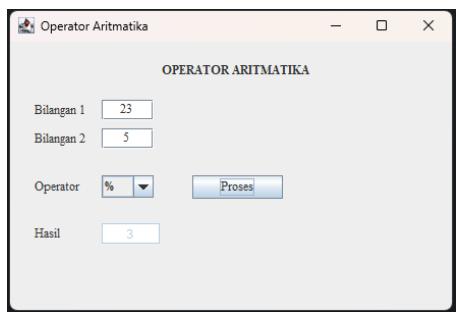
#### 2.1.3.4 Pembagian



Gambar 2.1.3.4 Output Pembagian.

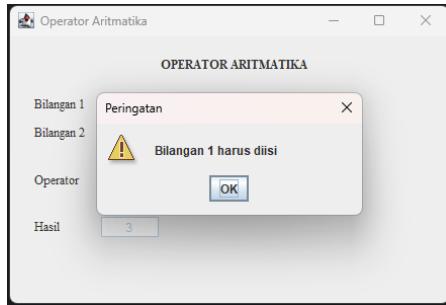
Pada output diatas, pengguna menginput angka 18 pada Bilangan 1 dan menginput angka 6 pada Bilangan 2. Pada Operator dipilih pembagian untuk membagi angka pada Bilangan 1 dan Bilangan 2. Pengguna menekan tombol Proses untuk mencetak hasil dari pembagian Bilangan 1 dan Bilangan 2.

#### 2.1.3.5 Modulus (sisa bagi)



Pada output diatas, pengguna menginput angka 23 pada Bilangan 1 dan menginput angka 5 pada Bilangan 2. Pada Operator dipilih modulus (sisa bagi) untuk mencari sisa bagi angka pada Bilangan 1 dan Bilangan 2. Pengguna menekan tombol Proses untuk mencetak hasil dari modulus Bilangan 1 dan Bilangan 2.

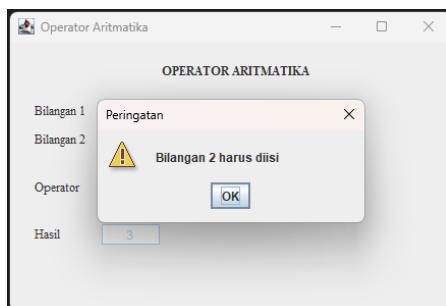
#### 2.1.3.6 Pesan Peringatan Bilangan 1



Gambar 2.1.3.6 Output Pesan Peringatan Bilangan 1.

Pada output diatas, pengguna tidak menginput angka pada Bilangan 1, tetapi menginput angka pada Bilangan 2. Sehingga, ketika pengguna menekan tombol Proses, muncul program mencetak pesan peringatan “Bilangan 1 harus diisi.”

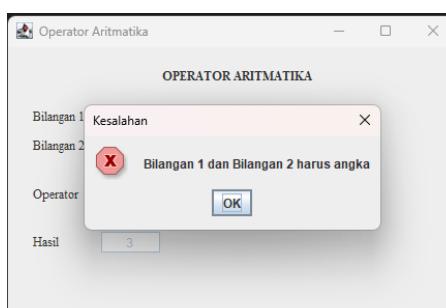
#### 2.1.3.7 Pesan Peringatan Bilangan 2



Gambar 2.1.3.7 Output Pesan Peringatan Bilangan 2.

Pada output diatas, pengguna menginput angka pada Bilangan 1, namun tidak menginput angka pada Bilangan 2. Sehingga, ketika pengguna menekan tombol Proses, muncul program mencetak pesan peringatan “Bilangan 2 harus diisi.”

#### 2.1.3.8 Pesan Error



#### Gambar 2.1.3.8 Output Pesan Error.

Pada output diatas, program akan mencetak pesan error jika pengguna menginput huruf atau simbol pada Bilangan 1 atau Bilangan 2. Pengguna diharuskan untuk menginput angka.

## KESIMPULAN

Pembuatan aplikasi Operator Aritmatika ini berhasil menunjukkan bagaimana komponen-komponen GUI pada Java, seperti **JLabel**, **JTextField**, **JComboBox**, dan **JButton**, dapat digunakan secara terpadu untuk membangun sebuah program perhitungan sederhana. Seluruh proses dimulai dari penambahan elemen tampilan, pengaturan variabel, hingga implementasi logika operasi berdasarkan pilihan operator yang diberikan pengguna. Program juga dilengkapi dengan validasi input untuk memastikan data yang dimasukkan sesuai, serta penanganan kesalahan apabila pengguna memasukkan karakter yang tidak valid. Melalui rangkaian implementasi tersebut, aplikasi mampu menghasilkan keluaran yang benar untuk operasi penjumlahan, pengurangan, perkalian, pembagian, dan modulus, serta mampu memberikan peringatan atau pesan kesalahan ketika terjadi kondisi input yang tidak sesuai. Hal ini menunjukkan bahwa desain antarmuka dan logika fungsional telah berjalan dengan baik dan saling melengkapi.

**DAFTAR PUSTAKA**

- [1] H. M. Deitel dan P. J. Deitel, *Java: How to Program*, 10th ed. Boston, MA: Pearson, 2015.
- [2] Oracle, “Creating a GUI With Swing,” *Oracle Java Documentation*. Diakses: 21 Nov. 2025. [Daring].
- [3] J. Nino dan F. A. Hosch, *An Introduction to Programming and Object-Oriented Design Using Java*. New York: John Wiley & Sons, 2008.
- [4] D. Eck, *Introduction to Programming Using Java*, 8th ed. 2022. [Daring].