LAPORAN PRATIKUM

ALGORITMA DAN PEMROGRAMAN

"Laporan Tugas Pekan 8 "

Disusun Oleh:

Zaki Misucha Andefi

2511533006

Informatika Kelas B

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

Asisten Praktikum : Rahmad Dwirizki Olders

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

1. Pseudocode

PROGRAM TugasPekan8_2511533006
{Program untuk demonstrasi operator assignment dalam GUI Java}

Deklarasi
Frame : JFrame
Panel : JPanel
Label1, Label2, Label3 : JLabel
TextField1, TextField2 : JTextField
TextArea : JTextArea
Button : JButton
ScrollPane : JScrollPane
a, b : integer
hasil : StringBuilder

Pseudocode

ALGORITMA

   BUAT Frame dengan judul "Operator Assignment"

   SET Panel dengan background biru

   TAMBAH Label "Bilangan 1", "Bilangan 2", "Hasil" ke Panel

   TAMBAH TextField1 dan TextField2 untuk input bilangan

   TAMBAH TextArea dengan ScrollPane untuk menampilkan hasil

   TAMBAH Button "Process" dengan event handler

   WHEN Button diklik:

     IF TextField1 kosong THEN

       TAMPILKAN pesan peringatan "Bilangan 1 harus diisi"

     ELSE IF TextField2 kosong THEN

       TAMPILKAN pesan peringatan "Bilangan 2 harus diisi"

     ELSE

       TRY

         a = Integer.parseInt(TextField1.getText())

         b = Integer.parseInt(TextField2.getText())

```
BUAT objek StringBuilder hasil
APPEND header "HASIL OPERATOR ASSIGNMENT" ke hasil
nilaiAwal = a
// Operator +=
a += b
APPEND "a += b → a = " + a ke hasil
a = nilaiAwal
// Operator -=
a -= b
APPEND "a -= b → a = " + a ke hasil
a = nilaiAwal
// Operator *=
a *= b
APPEND "a *= b → a = " + a ke hasil
a = nilaiAwal
// Operator /=
IF b ≠ 0 THEN
    a /= b
    APPEND "a /= b → a = " + a ke hasil
    a = nilaiAwal
ELSE
    APPEND "a /= b → Pembagian dengan nol tidak diperbolehkan" ke
hasil
ENDIF
// Operator %=
IF b ≠ 0 THEN
    a %= b
    APPEND "a %= b → a = " + a ke hasil
ELSE
    APPEND "a %= b → Modulus dengan nol tidak diperbolehkan" ke
hasil
ENDIF
```
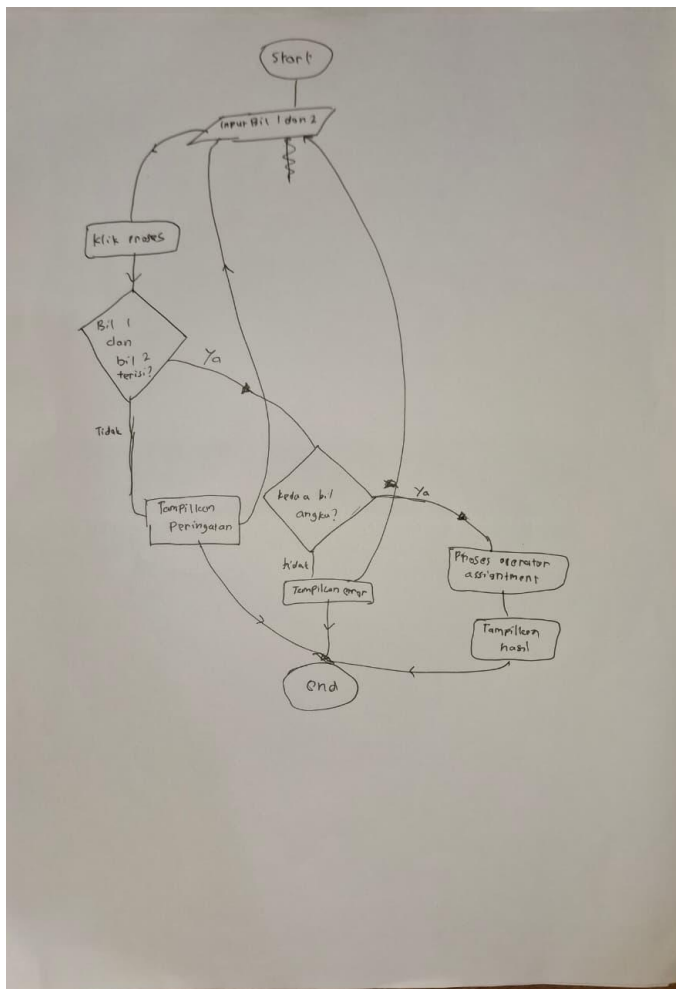
```
                SET TextArea dengan teks hasil
            CATCH NumberFormatException
                TAMPILKAN pesan error "Bilangan 1 dan Bilangan 2 harus angka"
                KOSONGKAN TextArea
            END TRY
        ENDIF
    END WHEN
END PROGRAM
```
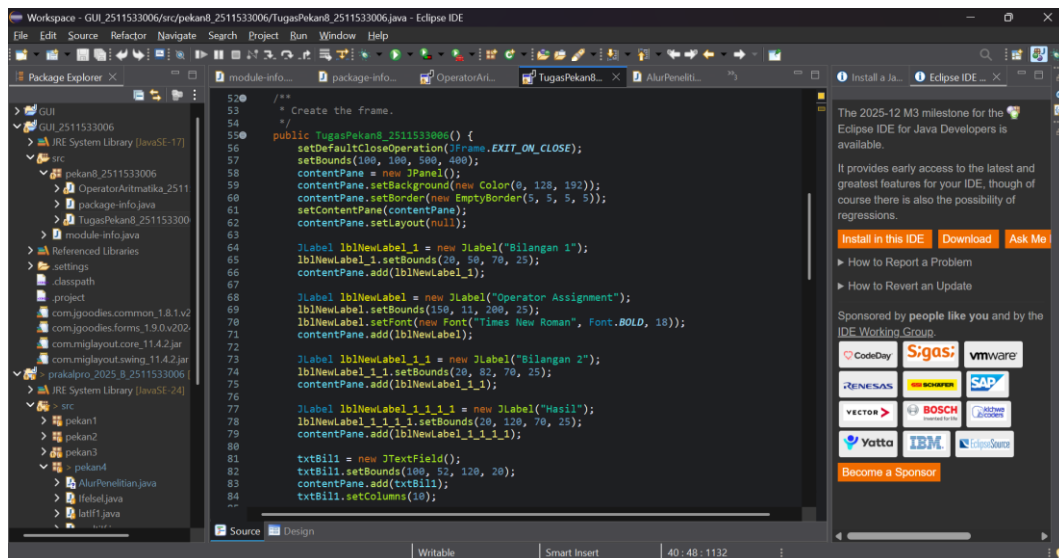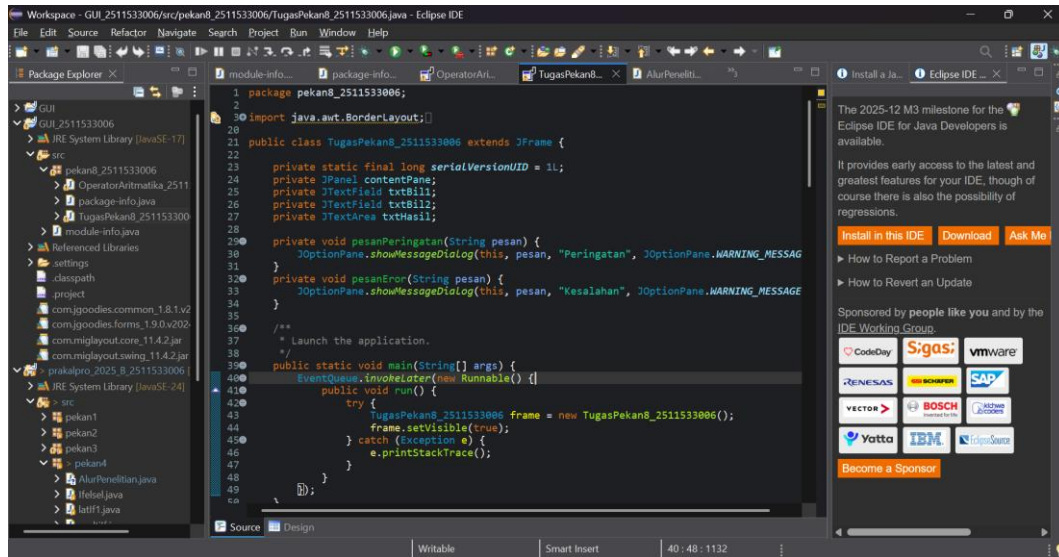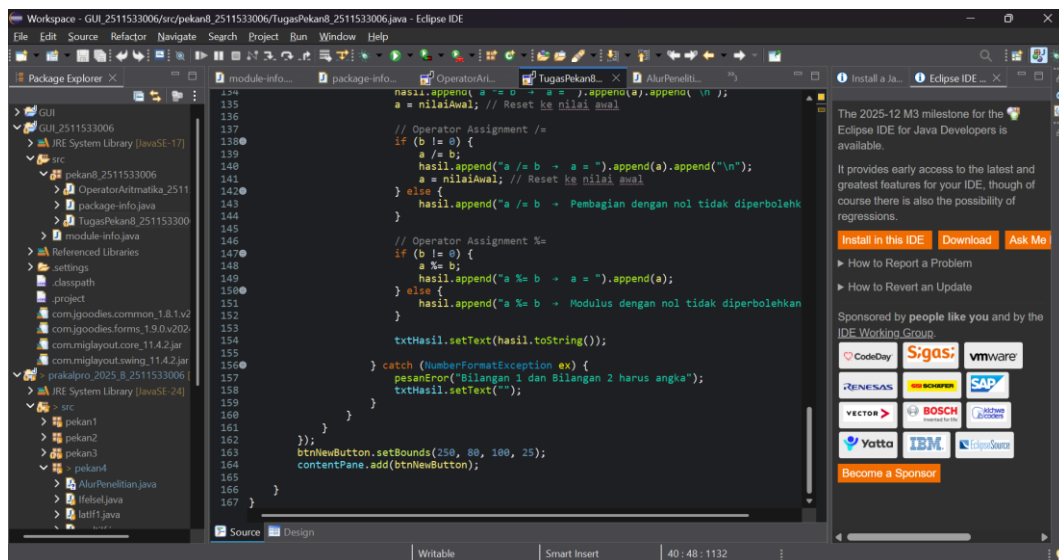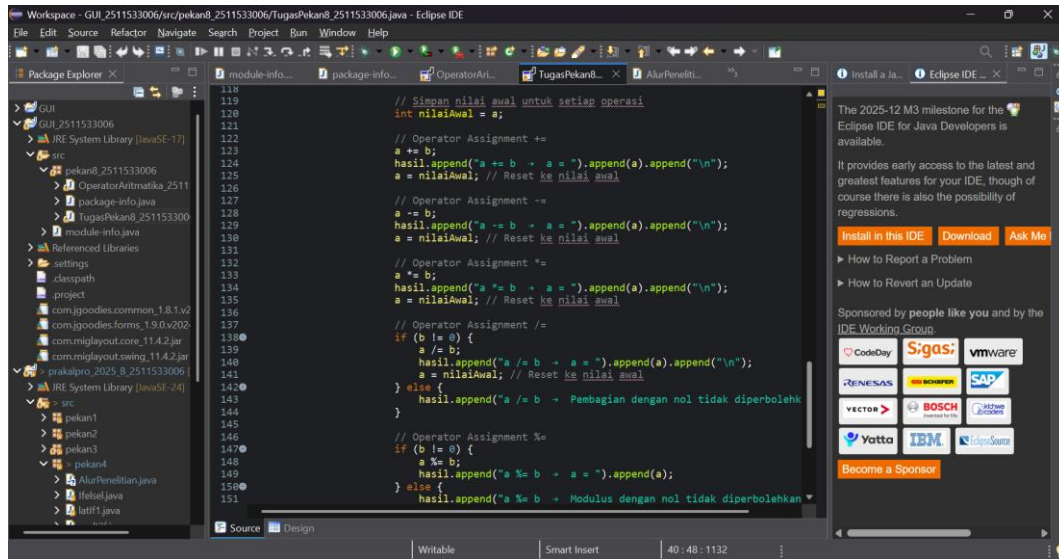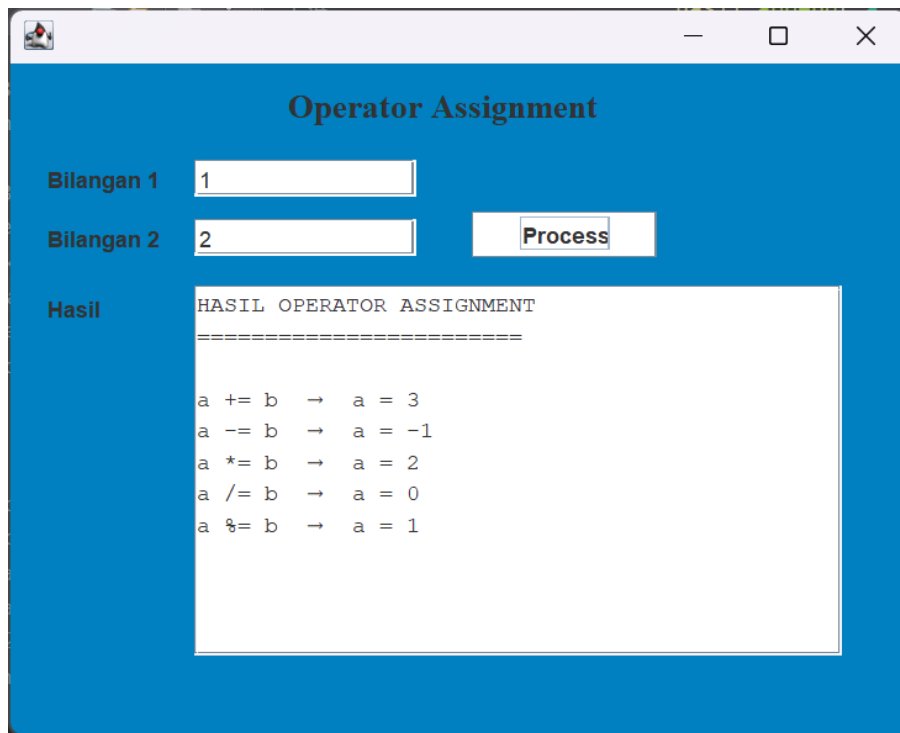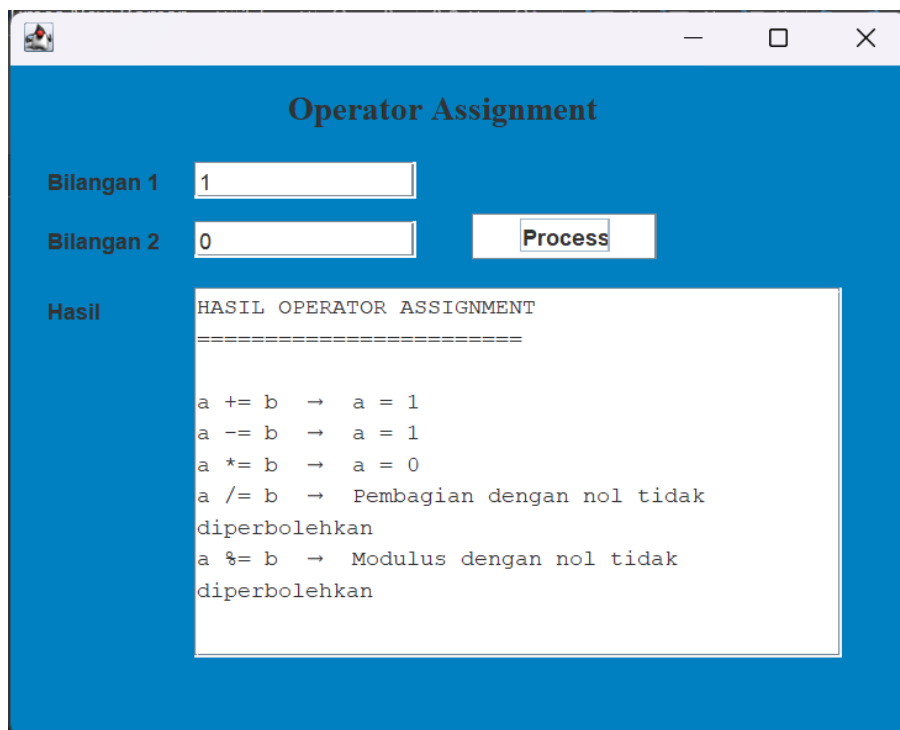
2. Flowchart

3. Kode Program


```java
package pekan8_2511533006;

import java.awt.BorderLayout;

public class TugasPekan8_2511533006 extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField txtBil1;
    private JTextField txtBil2;
    private JTextArea txtHasil;

    private void pesanPeringatan(String pesan) {
        JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAG
    }
    private void pesanError(String pesan) {
        JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.WARNING_MESSAGE
    }

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    TugasPekan8_2511533006 frame = new TugasPekan8_2511533006();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
```



```java
    /**
     * Create the frame.
     */
    public TugasPekan8_2511533006() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 500, 400);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(0, 128, 192));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel_1 = new JLabel("Bilangan 1");
        lblNewLabel_1.setBounds(20, 50, 70, 25);
        contentPane.add(lblNewLabel_1);

        JLabel lblNewLabel = new JLabel("Operator Assignment");
        lblNewLabel.setBounds(150, 11, 200, 25);
        lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 18));
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1_1 = new JLabel("Bilangan 2");
        lblNewLabel_1_1.setBounds(20, 82, 70, 25);
        contentPane.add(lblNewLabel_1_1);

        JLabel lblNewLabel_1_1_1_1 = new JLabel("Hasil");
        lblNewLabel_1_1_1_1.setBounds(20, 120, 70, 25);
        contentPane.add(lblNewLabel_1_1_1_1);

        txtBil1 = new JTextField();
        txtBil1.setBounds(100, 52, 120, 20);
        contentPane.add(txtBil1);
        txtBil1.setColumns(10);
```



```java
        txtBil2 = new JTextField();
        txtBil2.setColumns(10);
        txtBil2.setBounds(100, 84, 120, 20);
        contentPane.add(txtBil2);

        // Menggunakan JTextArea agar bisa menampilkan teks multi-line
        txtHasil = new JTextArea();
        txtHasil.setEditable(false);
        txtHasil.setLineWrap(true);
        txtHasil.setWrapStyleWord(true);
        txtHasil.setFont(new Font("Monospaced", Font.PLAIN, 12));

        JScrollPane scrollPane = new JScrollPane(txtHasil);
        scrollPane.setBounds(100, 120, 350, 200);
        contentPane.add(scrollPane);

        JButton btnNewButton = new JButton("Process");
        btnNewButton.setBackground(new Color(255, 255, 255));
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if(txtBil1.getText().trim().isEmpty()) {
                    pesanPeringatan("Bilangan 1 harus diisi");
                } else if (txtBil2.getText().trim().isEmpty()) {
                    pesanPeringatan("Bilangan 2 harus diisi");
                } else {
                    try {
                        int a = Integer.parseInt(txtBil1.getText());
                        int b = Integer.parseInt(txtBil1.getText());

                        StringBuilder hasil = new StringBuilder();
                        hasil.append("HASIL OPERATOR ASSIGNMENT\n");
                        hasil.append("=======================\n\n");
```

First screenshot:

```java
118              // Simpan nilai awal untuk setiap operasi
119              int nilaiAwal = a;
120
121
122              // Operator Assignment +=
123              a += b;
124              hasil.append("a += b  →  a = ").append(a).append("\n");
125              a = nilaiAwal; // Reset ke nilai awal
126
127              // Operator Assignment -=
128              a -= b;
129              hasil.append("a -= b  →  a = ").append(a).append("\n");
130              a = nilaiAwal; // Reset ke nilai awal
131
132              // Operator Assignment *=
133              a *= b;
134              hasil.append("a *= b  →  a = ").append(a).append("\n");
135              a = nilaiAwal; // Reset ke nilai awal
136
137              // Operator Assignment /=
138              if (b != 0) {
139                  a /= b;
140                  hasil.append("a /= b  →  a = ").append(a).append("\n");
141                  a = nilaiAwal; // Reset ke nilai awal
142              } else {
143                  hasil.append("a /= b  →  Pembagian dengan nol tidak diperbolehk
144              }
145
146              // Operator Assignment %=
147              if (b != 0) {
148                  a %= b;
149                  hasil.append("a %= b  →  a = ").append(a);
150              } else {
151                  hasil.append("a %= b  →  Modulus dengan nol tidak diperbolehkan
```

Second screenshot:

```java
134              hasil.append("a *= b  →  a = ").append(a).append("\n");
135              a = nilaiAwal; // Reset ke nilai awal
136
137              // Operator Assignment /=
138              if (b != 0) {
139                  a /= b;
140                  hasil.append("a /= b  →  a = ").append(a).append("\n");
141                  a = nilaiAwal; // Reset ke nilai awal
142              } else {
143                  hasil.append("a /= b  →  Pembagian dengan nol tidak diperbolehk
144              }
145
146              // Operator Assignment %=
147              if (b != 0) {
148                  a %= b;
149                  hasil.append("a %= b  →  a = ").append(a);
150              } else {
151                  hasil.append("a %= b  →  Modulus dengan nol tidak diperbolehkan
152              }
153
154              txtHasil.setText(hasil.toString());
155
156              } catch (NumberFormatException ex) {
157                  pesanEror("Bilangan 1 dan Bilangan 2 harus angka");
158                  txtHasil.setText("");
159              }
160          }
161      });
162      btnNewButton.setBounds(250, 80, 100, 25);
163      contentPane.add(btnNewButton);
164
165      }
166  }
167  }
```

4. Hasil Output Program

1. Program dengan input valid dan output berhasil

2. Program dengan Input tidak valid

5. Penjelasan Singkat Program

Deskripsi Program

Program ini merupakan aplikasi GUI Java Swing yang mendemonstrasikan penggunaan operator assignment dalam bentuk visual. Program memungkinkan pengguna untuk memasukkan dua bilangan dan melihat hasil dari berbagai operasi assignment yang diterapkan.

Struktur Program

Program terdiri dari satu class utama TugasPekan8_2511533006 yang extends JFrame dan mengimplementasikan komponen-komponen GUI berikut:

1. **JTextField** - Untuk input bilangan 1 dan bilangan 2
2. **JTextArea** - Untuk menampilkan hasil operasi dengan format multi-line
3. **JScrollPane** - Untuk membuat area hasil yang dapat di-scroll
4. **JButton** - Untuk memproses perhitungan
5. **JLabel** - Untuk label penanda

Operator Assignment yang Diimplementasikan

```
// 1. Operator += (Penambahan)
a += b;  // Setara dengan: a = a + b
// 2. Operator -= (Pengurangan)
a -= b;  // Setara dengan: a = a - b
// 3. Operator *= (Perkalian)
a *= b;  // Setara dengan: a = a * b
// 4. Operator /= (Pembagian)
a /= b;  // Setara dengan: a = a / b
// 5. Operator %= (Modulus)
a %= b;  // Setara dengan: a = a % b
```

Fitur Validasi Input

1. **Validasi Kelengkapan Data**
   - o   Memastikan kedua bilangan telah diisi sebelum diproses
   - o   Menampilkan pesan peringatan jika ada field yang kosong

2. **Validasi Tipe Data**
   - o   Memastikan input berupa angka menggunakan try-catch block
   - o   Menangani NumberFormatException dengan pesan error yang informatif

3. **Validasi Pembagian dengan Nol**
   - o   Mencegah error division by zero pada operasi /= dan %=
   - o   Menampilkan pesan yang jelas ketika terjadi percobaan pembagian dengan nol

Fitur Tambahan

1. **Reset Nilai** - Mengembalikan nilai variabel a ke nilai awal setelah setiap operasi untuk memastikan perhitungan yang independen

2. **Format Output** - Menggunakan StringBuilder untuk membuat output yang terstruktur dan rapi

3. **Scrollable Text Area** - Memungkinkan menampilkan hasil yang panjang dalam area terbatas

4. **User-Friendly Interface** - Desain GUI yang intuitif dengan warna dan layout yang jelas

Kelebihan Program

- **Robust**: Dilengkapi dengan error handling yang komprehensif
- **User-Friendly**: Interface GUI yang mudah digunakan
- **Edukatif**: Menampilkan proses step-by-step setiap operasi assignment
- **Structured**: Output yang terorganisir dan mudah dibaca

Program ini berhasil mendemonstrasikan konsep operator assignment dalam konteks pemrograman GUI, sekaligus menerapkan prinsip-prinsip good programming practice seperti validasi input dan error handling.