

LAPORAN PRAKTIKUM

PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

**“PENGGUNAAN GRAPHICAL USER INTERFACE (GUI) DALAM**

**PEMBUATAN PROGRAM OPERASI ARITMATIKA DENGAN BAHASA**

**PEMROGRAMAN JAVA”**

disusun oleh

Hafizh Habibullah

2511531002

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

Asisten Praktikum: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

TAHUN 2025

## KATA PENGANTAR

Puji syukur kita panjatkan ke hadirat Allah SWT. karena atas rahmat dan karunia-Nya laporan praktikum Algoritma dan Pemrograman ini dapat diselesaikan dengan baik.

Laporan ini disusun untuk memenuhi tugas praktikum pada mata kuliah Algoritma dan Pemrograman dengan topik Penggunaan *Graphical User Interface (GUI)* dalam Bahasa Pemrograman Java. Melalui laporan ini, penulis berusaha menjelaskan hasil praktikum yang meliputi penjelasan mengenai Objek yang digunakan, cara kerja, serta hasil output dari program.

Penulis sangat mengharapkan kritik dan saran yang membangun agar dapat menjadi bahan perbaikan di kemudian hari. Semoga laporan ini dapat memberikan manfaat bagi penulis maupun pembaca.

Padang, 22 November 2025

Hafizh Habibullah

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	i
<b>DAFTAR ISI.....</b>	ii
<b>DAFTAR LAMPIRAN .....</b>	iii
<b>BAB I PENDAHULUAN.....</b>	1
1.1    Latar Belakang.....	1
1.2    Tujuan.....	1
1.3    Manfaat.....	1
<b>BAB 2 PEMBAHASAN.....</b>	2
2.1    Pengenalan Antarmuka Grafis .....	2
2.2    Pembuatan <i>GUI</i> .....	5
2.3    Penerapan Logika Dasar Program .....	6
2.4    Hasil Output dari Aplikasi .....	9
<b>BAB III KESIMPULAN.....</b>	11
3.1    Kesimpulan.....	11
3.2    Saran Pengembangan.....	11
<b>DAFTAR PUSTAKA .....</b>	12

## **DAFTAR LAMPIRAN**

Gambar 2. 1 - Kode Program pada Awal Project .....	2
Gambar 2. 2 - Tampilan Layar Bagian "Design" .....	3
Gambar 2. 3 - Panel "Properties" .....	4
Gambar 2. 4 - Nama Aplikasi.....	5
Gambar 2. 5 – Bagian-bagian yang dimasukkan .....	5
Gambar 2. 6 - Tampilan opsi "Model" .....	6
Gambar 2. 7 - Kode Program Berisikan Logika untuk Operasi Aritmetika.....	7
Gambar 2. 8 - Program yang berisikan Pesan kesalahan pengguna.....	7
Gambar 2. 9 - Tampilan Aplikasi yang Sudah Selesai dan Dijalankan.....	9
Gambar 2. 10 - Pesan Peringatan.....	10
Gambar 2. 11 - Pesan Error.....	10

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Pada pengembangan aplikasi, *Antarmuka Grafis*, atau *Graphical User Interface (GUI)* digunakan untuk meningkatkan kenyamanan dan mempermudah pengguna dalam melakukan interaksi dengan program. Dengan *GUI*, pengguna dapat berinteraksi dengan tombol, kotak teks, menu, dan elemen visual lainnya tanpa harus melakukan perintah secara manual melalui *console*. Penerapan *GUI* juga membantu meminimalkan kesalahan input, karena setiap aksi yang dilakukan pengguna dapat dibantu dengan kontrol visual yang terarah. Oleh karena itu, Penggunaan *GUI* menjadi salah satu kemampuan dasar yang perlu dipelajari dalam pembuatan aplikasi *modern*.

#### **1.2 Tujuan**

Adapun tujuan dari praktikum ini adalah:

1. Memahami penggunaan antarmuka grafis (*GUI*) dalam pembuatan aplikasi Java.
2. Memahami cara menerima input, menampilkan output, dan memberikan interaksi melalui elemen *GUI*.
3. Menerapkan konsep logika dasar dalam program berbasis tampilan grafis.
4. Melatih kemampuan dalam menangani kesalahan input pada aplikasi.

#### **1.3 Manfaat**

Manfaat dari Praktikum ini adalah:

1. Memberikan pengalaman dalam merancang dan mengelola komponen *GUI* dengan baik.
2. Membantu meningkatkan pemahaman pemrograman secara praktis melalui visualisasi proses program.
3. Menjadi dasar pengembangan aplikasi yang lebih kompleks.

## BAB 2

### PEMBAHASAN

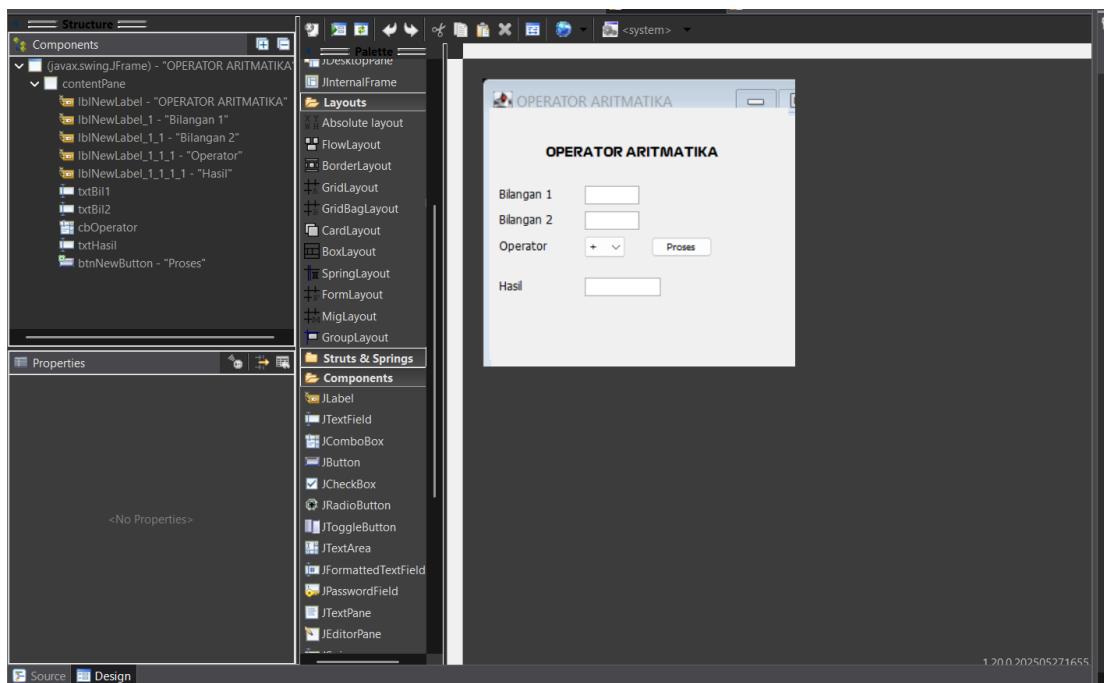
*Plugin WindowBuilder* adalah *Plugin* yang terdapat pada *Software Eclipse IDE* yang memungkinkan pengguna untuk membuat *program* berbasis Antarmuka (GUI). Pada praktikum ini, dilakukan pemrograman Operasi Aritmatika menggunakan *GUI*. Terdapat beberapa langkah-langkah pembuatan program dan Grafis Antarmuka, seperti penggunaan Komponen *GUI*, Penerapan logika dasar, dan Validasi serta penanganan kesalahan input.

#### 2.1 Pengenalan Antarmuka Grafis

```
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                OperatorAritmatikaGUI_2511531002 frame = new OperatorAritmatikaGUI_2511531002();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

Gambar 2. 1 - Kode Program pada Awal Project

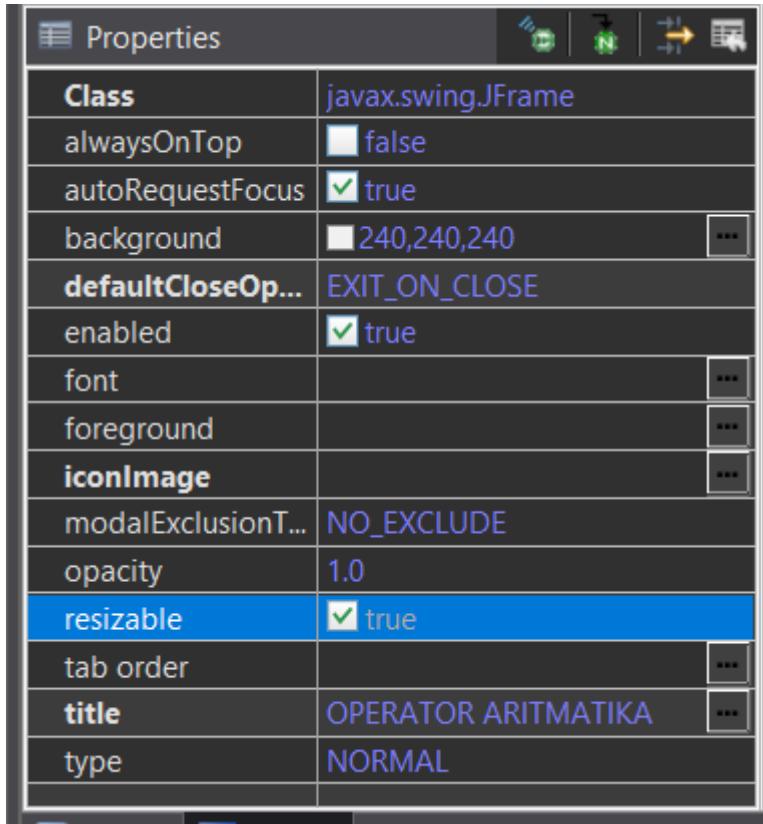
Sebelum membuat *GUI*, terlebih dahulu dibuat *project* baru dengan menggunakan *WindowBuilder*. Setelah itu, program akan secara otomatis membuat program dasar dalam pembuatan *GUI* dengan pemrograman Java. Lalu, pada bagian bawah layar, terdapat opsi “**Design**” yang dibuka untuk mengganti pemrograman dari mode *Code-based* (menulis *layout* dengan kode) ke mode *Visual-based* (mengatur *layout* dengan *editor* visual). Tampilannya akan menjadi seperti berikut.



Gambar 2. 2 - Tampilan Layar Bagian "Design"

Di bagian kanan layar terdapat *canvas* yang merupakan *wadah* bagi *GUI* yang akan dibuat. Ukuran *canvas* dapat diatur sesuai keinginan. Di bagian kiri layar, terdapat beberapa panel, yaitu panel *Components* (yang berisikan komponen yang digunakan pada pembuatan *GUI*), *Properties* (untuk mengatur isi dari komponen tertentu), dan *Pallete* (jenis-jenis komponen yang dapat dipilih sesuai kebutuhan).

Sebelum memulai pembuatan, *layout Canvas* terlebih dahulu diubah ke *Absolute Layout*, yang bisa ditemukan di panel *Pallete*. Pengubahan *layout* ini dilakukan supaya komponen yang nantinya akan dimasukkan bisa diletakkan dimanapun pada *canvas*. Setelah itu, pada panel *Properties*, terdapat opsi *resizable* yang secara otomatis diatur menjadi *true* yang memungkinkan *Programmer* untuk merubah ukuran *canvas* sesuai keinginannya. Jika opsi diubah menjadi *false*, maka ukuran *canvas* akan dikunci dan tidak dapat diubah-ubah.



Gambar 2. 3 - Panel "Properties"

Setelah itu, *Programmer* dapat memilih komponen yang ingin dimasukkan ke dalam *canvas*. Terdapat banyak jenis Komponen yang bisa digunakan dalam pembuatan program. Beberapa contoh komponen yang digunakan pada Program ini adalah sebagai berikut:

a. JLabel

Komponen yang digunakan untuk menampilkan teks pada *GUI* yang dibuat. Grafis ini bisa diletakkan di kanvas pada posisi yang kita inginkan.

b. JTextField

Komponen yang digunakan untuk memasukkan kolom input yang dapat digunakan oleh pengguna nantinya.

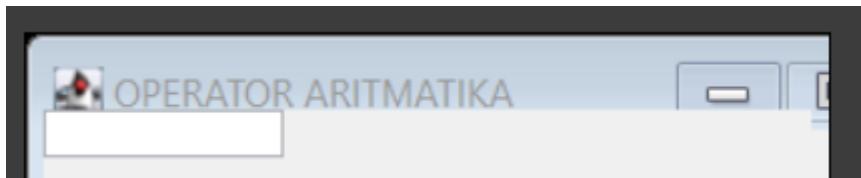
c. JComboBox

Komponen yang digunakan untuk memasukkan pilihan-pilihan berbeda.

d. JButton

Komponen yang digunakan untuk menampilkan tombol yang bisa ditekan oleh pengguna untuk menjalankan operasi tertentu.

## 2.2 Pembuatan *GUI*



Gambar 2. 4 - Nama Aplikasi

Proses Pembuatan *GUI* diawali dengan mengubah nama Aplikasi yang akan dibuat. Pada contoh ini, nama Aplikasi diubah menjadi “OPERATOR ARITMATIKA”.

Selanjutnya, dimasukkan beberapa *Label* (Penanda Merah) yang berfungsi memberikan keterangan kepada pengguna mengenai tiap-tiap bagian atau *input* apa yang harus diisi pada bagian tertentu. Pada aplikasi ini, terdapat beberapa *Label* yang berbeda. **Bilangan 1** disimpan ke **txtBil1**, **Bilangan 2** disimpan ke **txtBil2**, **Operator** disimpan ke **cbOperator**, dan **Hasil** disimpan ke **txtHasil**. Dengan adanya *Label* ini, pengguna dapat memahami setiap bagian-bagian yang ditampilkan.

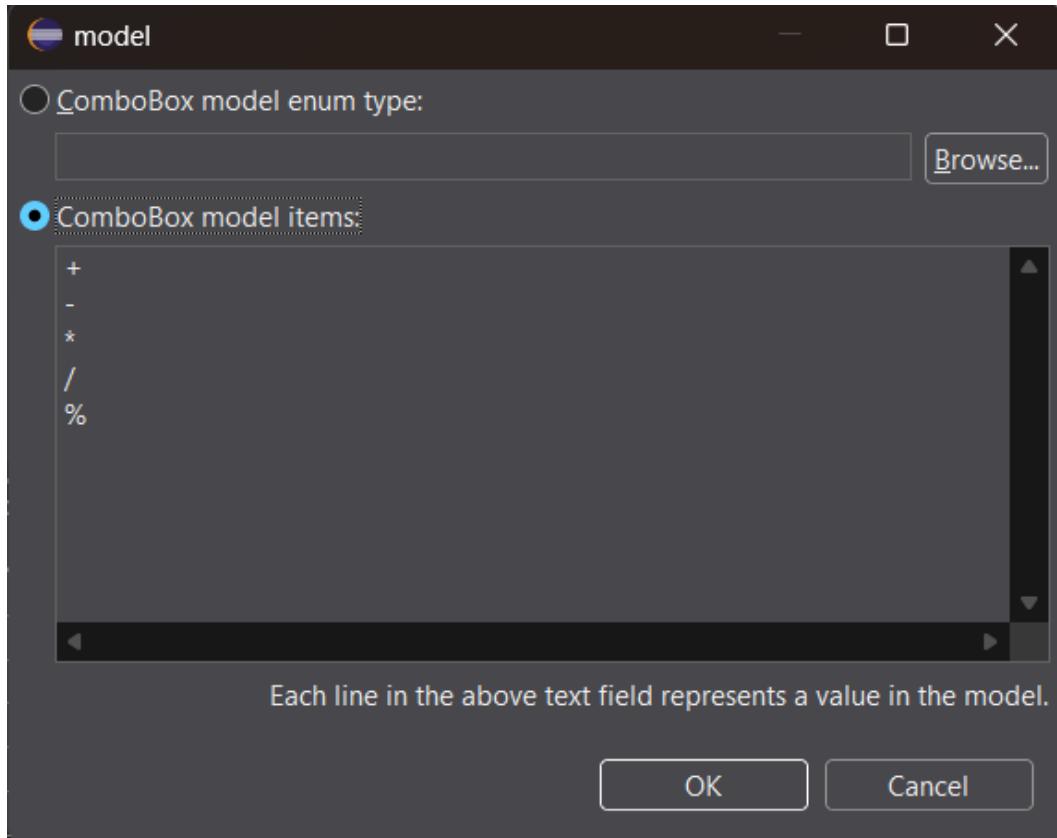


Gambar 2. 5 – Bagian-bagian yang dimasukkan

Selanjutnya, dimasukkan bagian-bagian *Text Field* (Penanda Kuning) yang dapat digunakan sebagai penerima *input* dari pengguna. Pada Aplikasi ini, *Text Field* diletakkan setelah bagian **Bilangan 1**, **Bilangan 2**, dan **Hasil**.

Bagian *Combo Box* (Penanda Hijau) berisikan jenis-jenis Operator yang dapat digunakan oleh pengguna. Program nantinya akan berjalan sesuai dengan Operator yang dipilih oleh pengguna. Pilihan Operator dapat diatur di bagian *model*

yang terdapat di panel *Properties*. Pada program ini, ditambahkan 5 jenis Operator yang dapat dipilih, yaitu Penjumlahan (+), Pengurangan (-), Perkalian (\*), Pembagian (/), dan Sisa bagi (%).



Gambar 2. 6 - Tampilan opsi "Model"

Terakhir, terdapat tombol *Proses* (Penanda Biru) yang nantinya akan berfungsi menjalankan operasi perhitungan setelah pengguna memilih Jenis Operator yang diinginkan.

### 2.3 Penerapan Logika Dasar Program

Antarmuka Grafis memang sudah dibuat sebelumnya, namun Operasi yang terdapat didalamnya masih belum bisa dijalankan. Hal ini karena program masih belum memiliki Kode yang berisikan logika yang dapat menjalankan Operasi.

```

JButton btnNewButton = new JButton("Proses");
btnNewButton.addActionListener(new ActionListener() {
    int hasil;
    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(txtBil2.getText());
                int c = cbOperator.getSelectedIndex(); //memilih operator
                if(c == 0) {
                    hasil= a+b;
                }
                if(c == 1) {
                    hasil = a-b;
                }
                if(c == 2) {
                    hasil = a*b;
                }
                if (c == 3) {
                    hasil = a/b;
                }
                if (c == 4) {
                    hasil = a%b;
                }
            } catch (NumberFormatException ex) {
                pesanError("Bilangan 1 dan Bilangan 2 harus angka");
            }
        }

        txtHasil.setText(String.valueOf(hasil));
    }
});
btnNewButton.setBounds(166, 132, 63, 21);
contentPane.add(btnNewButton);

```

Gambar 2. 7 - Kode Program Berisikan Logika untuk Operasi Aritmetika

Sesuai nama Aplikasi, program diatas dibuat untuk menjalankan Operasi Perhitungan sesuai dengan jenis Operasi yang dipilih oleh pengguna. Awalnya, program akan menyimpan Bilangan yang merupakan *input* yang diterima dari pengguna. **txtBil1** disimpan ke variabel **a**, **txtBil2** disimpan ke variabel **b**, dan **cbOperator** disimpan ke variabel **c**.

```

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

```

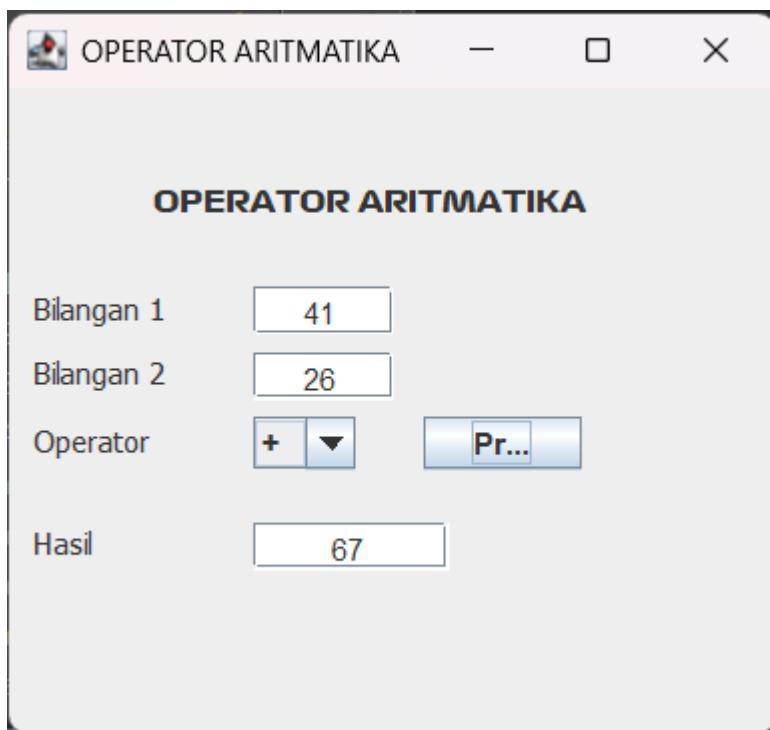
Gambar 2. 8 - Program yang berisikan Pesan kesalahan pengguna

Program juga ditambahkan kode yang berfungsi sebagai pesan peringatan jika terjadi kesalahan dalam memasukkan *input*. Program akan memeriksa input yang dimasukkan oleh pengguna. Jika **Bilangan 1** atau **Bilangan 2** tidak diisi, maka program akan menampilkan pesan peringatan yang memberitahu pengguna bahwa Kolom Bilangan 1 atau Bilangan 2 harus diisi. Program juga akan menampilkan pesan *error* jika pengguna tidak mengisi Kolom Bilangan dengan angka. Jika tidak ada kesalahan, maka program akan berlanjut ke proses berikutnya.

Program akan memeriksa jenis Operator apa yang dipilih oleh pengguna. Karena terdapat 5 jenis Operator, masing-masing akan bernilai mulai dari 0 hingga 4. Sebagai contoh, jika pengguna memilih Operasi Penjumlahan, maka variabel **c** akan bernilai **0**. Begitu juga dengan pilihan lainnya.

Setelah pengguna menekan tombol “Proses”, Program akan menjalankan Operasi Aritmatika sesuai dengan nilai variabel **c**. Jika bernilai 0, maka program akan menjumlahkan variabel **a** dan **b**. Begitu pula dengan nilai lainnya dengan Operasi yang berbeda. Pada akhirnya, program akan menyimpan hasil dari Operasi yang dilakukan ke variabel **hasil**. Setelah variabel **hasil** di-Konversi-kan menjadi variabel **String**, Hasil Operasi akan ditampilkan di kolom **Hasil**.

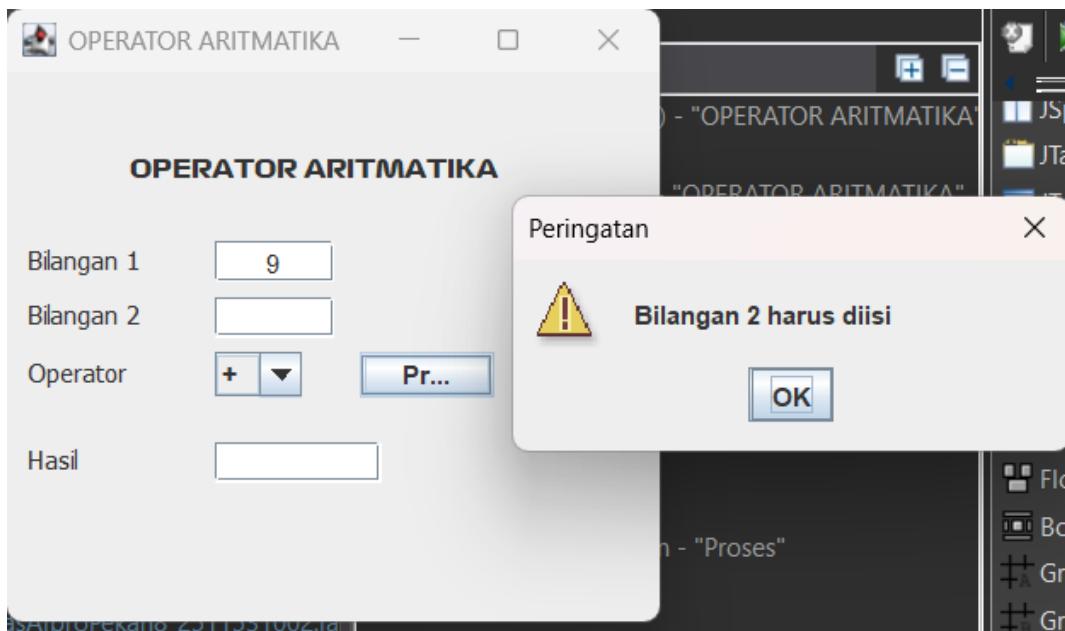
## 2.4 Hasil Output dari Aplikasi



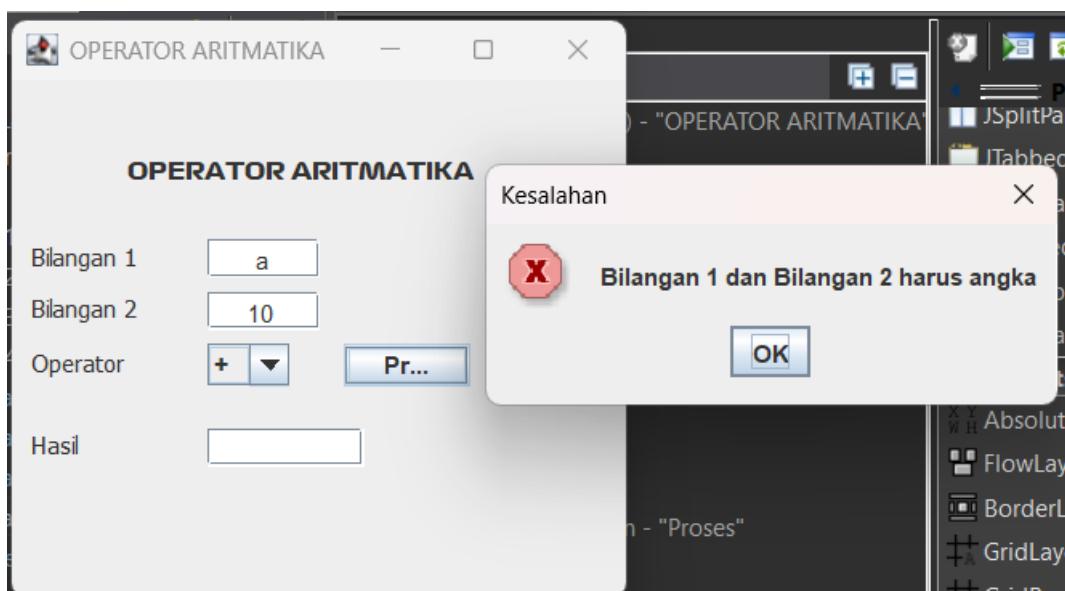
Gambar 2. 9 - Tampilan Aplikasi yang Sudah Selesai dan Dijalankan

Saat Program dijalankan, Program akan menampilkan Aplikasi seperti di atas. Terdapat nama Aplikasi, Keterangan tiap bagian, Kolom yang menerima *input* dari pengguna, Pemilihan Operasi, Tombol untuk menjalankan Operasi, dan Kolom yang menampilkan hasil perhitungan.

Pengguna perlu memasukkan 2 Bilangan yang diinginkan, lalu memilih jenis Operasi, dan Menekan tombol “Proses” untuk menjalankan dan menampilkan hasilnya. Jika terdapat kesalahan, maka akan muncul Pesan Peringatan atau Error, tergantung jenis kesalahan yang dibuat.



Gambar 2. 10 - Pesan Peringatan



Gambar 2. 11 - Pesan Error

## **BAB III**

### **KESIMPULAN**

#### 3.1 Kesimpulan

Praktikum ini memberikan pemahaman mengenai pembuatan antarmuka grafis menggunakan **WindowBuilder** pada Java. Kegiatan meliputi penyusunan elemen tampilan seperti label, bidang teks, tombol, serta pengaturan layout melalui mode desain. Selain itu, praktikum juga mengenalkan penerapan event handling sehingga antarmuka dapat merespons interaksi pengguna. Dengan praktikum ini, konsep dasar pembuatan GUI interaktif berbasis Java dapat dipahami dengan lebih terarah dan mudah.

#### 3.2 Saran Pengembangan

Untuk pengembangan selanjutnya, praktikum dapat dikembangkan dengan menambahkan fitur validasi input, tampilan yang lebih *user-friendly* (ramah pengguna), serta penerapan logika yang lebih kompleks. Pemanfaatan konsep pemisahan antara tampilan dan proses juga dapat ditingkatkan agar struktur program lebih baik. Selain itu, eksplorasi komponen-komponen *GUI* tambahan di **WindowBuilder** dapat memperluas pemahaman dan keterampilan pembuatan aplikasi berbasis antarmuka grafis.

## **DAFTAR PUSTAKA**

- [1] Oracle, “The Java™ Tutorials,” Oracle Docs, 2024. Accessed: Oct. 20, 2024.  
[Online]. Available: <https://docs.oracle.com/javase/>
- [2] Oracle Corporation, “Creating a GUI With Swing,” *Oracle Java Documentation*. Available: <https://docs.oracle.com/javase/tutorial/uiswing/>.  
Accessed: November 23, 2025.