

LAPORAN PRAKTIKUM
ALGORITMA PEMOGRAMAN
PEMROGRAMAN GUI 1 PADA BAHASA PEMROGRAMAN JAVA

Disusun Oleh :

Aliifah Felda Mufarrihati Salwaa
2511531011

Dosen Pengampu :

Dr. Wahyudi, S.T., M.T.

Asisten Praktikum :

Aufan Taufiqurrahman



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Laporan praktikum ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum algoritma pemrograman yang membahas tentang pemrograman GUI pada bahasa Java. Melalui laporan ini penulis dapat memahami materi praktikum secara mendalam dan dapat lebih teliti, teratur, serta memiliki kemampuan yang baik dalam penulisan kode pemrograman sesuai kaidah akademik. Sehingga laporan ini dapat menjadi sarana belajar, dokumentasi kegiatan, dan referensi praktikum berikutnya.

Penulis menyadari bahwa laporan ini masih memiliki banyak kekurangan, baik dari isi maupun penyajiannya. Oleh karena itu, saran dan kritik sangat penulis harapkan guna untuk menyempurnakan laporan berikutnya.

Padang, 18 November 2025

Aliifah Felda Mufarrihati Salwaa
2511531011

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Praktikum	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	1
BAB II PEMBAHASAN.....	2
2.1 Pemrograman GUI pada Bahasa Pemrograman Java.....	2
2.2 Praktikum Operator Aritmatika GUI dengan WindowBuilder .	3
2.3 Kode Program Praktikum GUI “Operator Aritmatika”.....	6
2.4 Output Pemrograman GUI “Operator Aritmatika”	11
BAB III KESIMPULAN	14
3.1 Kesimpulan.....	14
3.2 Saran.....	14
DAFTAR PUSTAKA	15

BAB I

PENDAHULUAN

1.1 Latar Belakang Praktikum

Sekarang ini, GUI (*Graphical User Interface*) merupakan hal yang paling penting jika kita hendak mengembangkan sebuah aplikasi ataupun website untuk pengguna. Sebelum adanya GUI, *user interface* yang digunakan adalah berbasis teks dan bekerja melalui input dari *keyboard*. Karena sekarang ini semua orang bisa menggunakan dan membutuhkan web/aplikasi dalam kehidupan sehari-hari, maka GUI sangat dibutuhkan untuk mempermudah interaksi manusia dengan perangkat terutama bagi orang awam.

Sebagai mahasiswa tingkat pertama yang baru memahami tentang pemrograman, hendaknya kita dapat memperlajari tentang GUI. Meskipun tujuan karir kita bukan di bidang *frontend*, tetapi mempelajari pemrograman GUI juga dibutuhkan. Dengan belajar pemrograman GUI kita dapat belajar bagaimana komponen, layout, dan alur interaksi dari sebuah pemrograman. Oleh karena itu, praktikum pekan 8 yang membahas tentang pemrograman GUI ini dilakukan.

1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Mengetahui fungsi dari GUI.
2. Mengetahui langkah-langkah untuk membuat GUI menggunakan bahasa pemrograman Java.
3. Mengetahui hasil dari GUI dengan bahasa pemrograman Java.

1.3 Manfaat Praktikum

Manfaat dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Memahami bagaimana cara pengembangan GUI dan pengaplikasian pemrograman GUI.
2. Mampu mengimplementasikan pemrograman GUI.
3. Meningkatkan kemampuan dalam penggunaan bahasa pemrograman Java.

BAB II

PEMBAHASAN

2.1. Pemrograman GUI pada Bahasa Pemrograman Java

GUI (*Graphical User Interface*) adalah antarmuka sistem operasi berbasis grafis seperti ikon, tombol, menu, dan lainnya untuk mendukung interaksi antara pengguna dan sistem. GUI merupakan cara modern agar pengguna dapat berkomunikasi dengan sistem. Sebelumnya, untuk berinteraksi dengan sistem pengguna menggunakan rangkaian perintah yang diketik secara manual. Dengan adanya GUI kini pengguna lebih dipermudah dengan visualisasi sehingga orang awam juga bisa berinteraksi langsung dengan sistem.

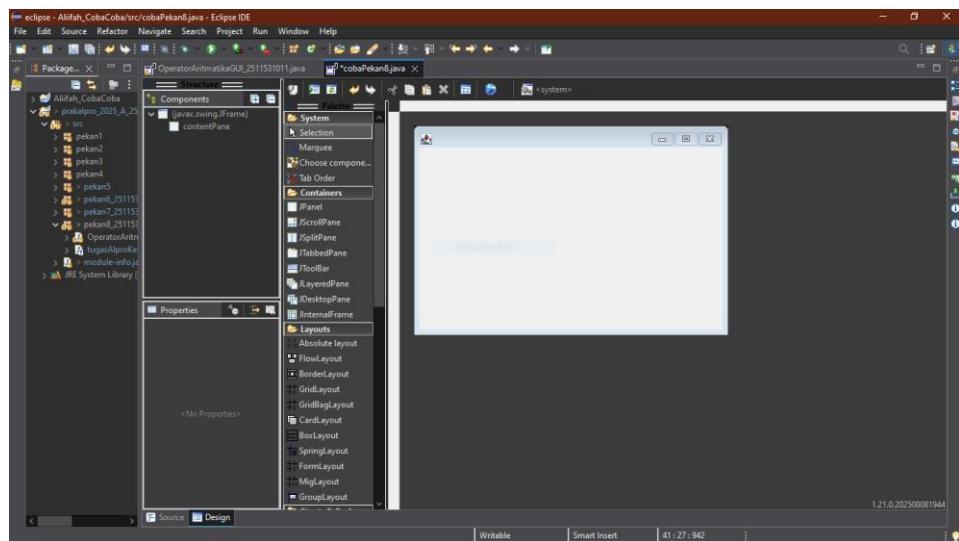
Untuk membuat GUI dengan menggunakan bahasa pemrograman Java, *toolkit* yang banyak digunakan adalah Swing dan JavaFX. Swing adalah *tool* standar yang digunakan oleh pengguna Java, sedangkan JavaFX menyediakan dukungan untuk membuat aplikasi desktop. Swing juga memiliki seperangkat komponen GUI yang lebih canggih, sedangkan JavaFX memiliki komponen yang lebih canggih tetapi lebih sedikit dibandingkan Swing.

Suatu hal yang berbeda, pemrograman GUI bekerja secara *event driven* dimana program tidak berjalan sesuai urutan dari atas ke bawah seperti biasanya, tetapi menunggu user melakukan suatu tindakan. Ketika user klik sebuah tombol, menekan sebuah kunci pada keyboard, maka program yang bersangkutan dengan tindakan itu baru akan berjalan.

2.2. Praktikum Operator Aritmatika GUI dengan WindowBuilder

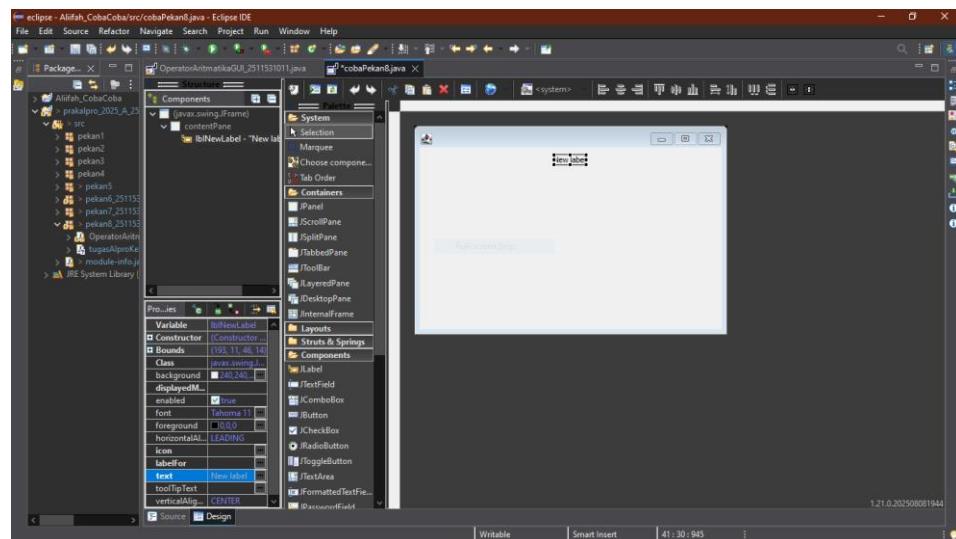
Pada praktikum pekan 8 ini, kita akan membuat pemrograman GUI (*Graphical User Interface*) dengan menggunakan *toolkit* yang ada pada Java yakni Swing. Tetapi kita tidak membuat GUI Swing dengan menuliskan kode secara manual, melainkan dengan menggunakan WindowBuilder dari Eclipse IDE. Dengan menggunakan WindowBuilder, kita hanya perlu mendesain antarmuka dengan cara *drag and drop* dan kode Swing akan dihasilkan. Berikut akan dijelaskan langkah-langkah lengkap beserta kode yang dihasilkan ketika membuat GUI dengan Eclipse WindowBuilder:

1. Membuat class baru dengan Swing JFrame untuk membuat jendela utama dalam GUI. Berikut adalah tampilan dari jendela desain setelah membuat JFrame:



Gambar 2.1: Tampilan pertama WindowBuilder

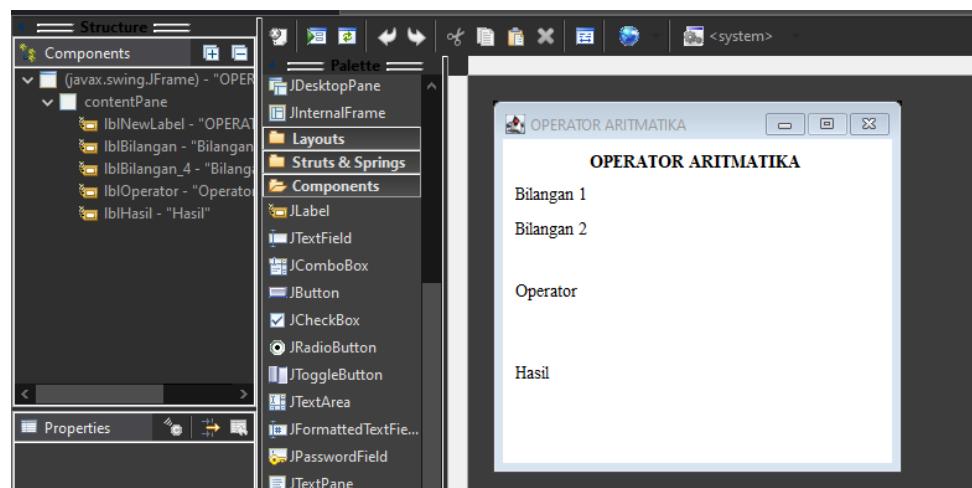
2. Selanjutnya kita akan memasukkan absolute layout ke bagian javax.swing.JFrame dan contentPane ke kanvas atau area desain dengan cara *drag-and-drop*.
3. Kita akan terus melakukan pekerjaan di dalam contentPane. Pertama kita akan memasukkan komponen JLabel sebagai tempat peletakan judul untuk program interface kita. Berikut adalah tampilan yang muncul.



Gambar 2.2: Tampilan ketika memasukkan JLabel

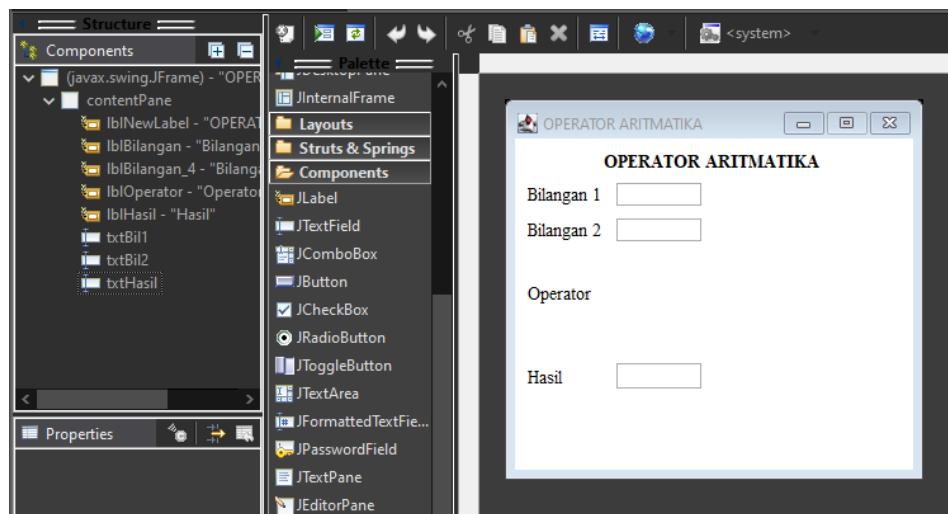
Kita dapat mengatur ukuran dari label sesuai dengan yang kita inginkan. Kita juga dapat mengubah ketentuan label mulai dari nama variabel, isi text, ukuran teks, gaya teks, dan sebagainya pada bagian properties. Dalam praktikum ini, label pertama diletakkan di bagian atas tengah dan isi teks diganti menjadi “OPERATOR ARITMATIKA” sebagai judul dari praktikum.

4. Kemudian kita bisa menambahkan 4 label lagi sebagai keterangan dengan isi teks “Bilangan 1” “Bilangan 2” “Operator” dan “Hasil”. Kita dapat menambahkannya dengan *drag* atau klik dari *palette* ataupun bisa *copy-paste* label pertama lalu kemudian mengubah ukuran dan isi teks. Berikut adalah hasil dari langkah 3 dan 4:



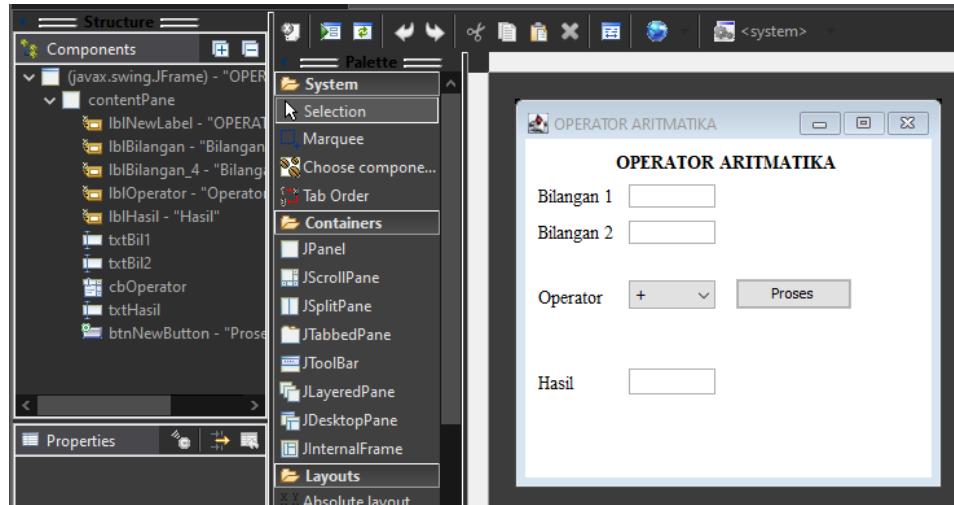
Gambar 2.3: Tampilan setelah memasukkan label lainnya

5. Selanjutnya kita akan membuat kotak untuk input di sebelah label “Bilangan 1” dan “Bilangan 2”. Komponen yang digunakan agar pengguna dapat memasukkan input adalah JTextField. Sama seperti komponen-komponen sebelumnya, kita memasukkan komponen JTextField dengan metode *drag-and-drop* atau klik dan drop di kanvas. Kita juga bisa menyesuaikan ukuran JTextField sesuai dengan yang kita butuhkan. Kita juga akan meletakkan JTextField di samping hasil tetapi tanpa meminta input dari pengguna. Berikut adalah tampilan yang dihasilkan:



Gambar 2.4: Tampilan setelah memasukkan JTextField

6. Untuk bagian operator, kita akan membuat komponen JComboBox agar pengguna dapat memilih operator apa yang ingin mereka gunakan untuk eksekusi bilangan satu dan dua yang telah pengguna input. Kita juga akan memasukkan tombol proses dengan menggunakan JButton. Tombol ini berfungsi sebagai perantara dimana pengguna akan mendapatkan hasil dari operator aritmatika setelah menekan tombol proses. Berikut tampilan yang akan kita dapatkan:



Gambar 2.5: Tampilan akhir GUI

Dengan hasil seperti ini, maka *interface* kita sudah selesai. Langkah selanjutnya adalah untuk membuat kode program sehingga GUI ini dapat menerima input, melakukan proses, dan menampilkan hasil sesuai dengan keinginan pengguna.

2.3. Kode Program Praktikum GUI “Operator Aritmatika”

Pada subbab ini, saya akan membahas lebih lanjut tentang bagaimana kode program dari GUI yang telah kita desain sesuai langkah-langkah diatas agar GUI tersebut dapat berfungsi dengan baik. Berikut adalah penjelasan lengkap dari kode program pada praktikum GUI Operator Aritmatika:

1. Pertama-tama kita akan membuat kode program untuk tombol proses agar input dari pengguna dapat dieksekusi. Lakukan klik dua kali pada tombol proses untuk masuk dalam kode program bagian JButton.

```

117     JButton btnNewButton = new JButton("Proses");
118     btnNewButton.setBackground(new Color(192, 192, 192));
119     btnNewButton.addActionListener(new ActionListener() {
120         int hasil;
121         public void actionPerformed(ActionEvent e) {
122             if (txtBill1.getText().trim().isEmpty()) {
123                 pesanPeringatan ("Silahkan Input Bilangan 1");
124             } else if (txtBill2.getText().trim().isEmpty()) {
125                 pesanPeringatan ("Silahkan Input Bilangan 2");
126             } else if (txtBill2.getText().trim().startsWith("0")) {
127                 pesanPeringatan ("Bilangan 2 tidak boleh 0");
128             }else {
129                 try {
130                     int a = Integer.parseInt(txtBill1.getText());
131                     int b = Integer.parseInt(txtBill2.getText());
132                     int c = cbOperator.getSelectedIndex();
133                     if (c==0) {
134                         hasil = a+b;
135                     }
136                     if (c==1) {
137                         hasil = a-b;
138                     }
139                     if (c==2) {
140                         hasil = a*b;
141                     }
142                     if (c==3) {
143                         hasil = a/b;
144                     }
145                     if (c==4) {
146                         hasil = a%b;
147                     }
148                     txtHasil.setText(String.valueOf(hasil));
149                 } catch (NumberFormatException ex) {
150                     pesanError ("Bilangan 1 dan Bilangan 2 harus angka");
151                 }
152             }
153         }
154     }

```

Gambar 2.6: Tampilan Kode Program pada bagian JButton

Kita akan diarahkan pada baris kode yang mengelola JButton “Proses” yang telah ada karena pembuatan desain (pada program ini dimulai dari baris 118). Disini kita akan membuat logika pemrograman agar GUI dapat bekerja. Berikut penjelasan tentang logika pemrograman yang terdapat pada bagian JButton:

- Mendeklarasikan variabel hasil sebagai keluaran yang akan ditampilkan.
- Membuat logika if dimana jika JTextField Bilangan 1 dan 2 tidak terisi, maka program akan memanggil fungsi pesan peringatan.
- Jika sudah selesai pengecekan dan tidak ada lagi pesan peringatan, maka program akan berjalan.
- Kita akan melakukan deklarasi variabel a dan b sebagai input serta variabel c sebagai index pada JComboBox.
- Kita kembali melakukan logika if dimana jika $c==0$, hasil akan mengandung nilai dari $a+b$, jika $c==1$, hasil akan mengandung nilai dari $a-b$, jika $c==2$, hasil akan mengandung nilai dari $a*b$, jika $c==3$, hasil akan mengandung nilai dari a/b , jika $c==4$, hasil akan mengandung nilai dari $a\%b$.

- f. Kemudian kita akan mencetak hasil dan memasukkannya ke dalam JTextField “Hasil”
 - g. Jika user memasukkan input selain angka, maka program akan mendeteksinya dan akan memanggil pesan error.
2. Untuk lebih memahami tentang GUI dan juga kode Swing yang dihasilkannya, kita akan membahas tentang bagian-bagian kode yang sudah otomatis terisi karena kita menggunakan WindowBuilder. Berikut pengenalan singkat tentang kode program yang ada:
- Deklarasi berbagai komponen yang kita gunakan

```
public class OperatorAritmatikaGUI_2511531011 extends JFrame {

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

Kode Program 2.1: Deklarasi komponen

- Deklarasi fungsi pesanPeringatan dan fungsi pesanError

```
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);
}
```

Kode Program 2.2: Deklarasi pesan peringatan dan pesan error

- Pintah untuk run aplikasi

```
/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                frame = new OperatorAritmatikaGUI_2511531011();
                frame.setVisible(true);
            } catch (Exception e) {
```

```
        e.printStackTrace();  
    }  
});}
```

Kode Program 2.3: Kode untuk run aplikasi

d. Rangkaian kode yang muncul setelah kita membuat tampilan GUI

```
public OperatorAritmatikaGUI_2511531011() {  
    setBackground(new Color(255, 255, 255));  
    setTitle("OPERATOR ARITMATIKA");  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setBounds(100, 100, 330, 300);  
    contentPane = new JPanel();  
    contentPane.setBackground(new Color(255, 255, 255));  
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  
    setContentPane(contentPane);  
    contentPane.setLayout(null)
```

Kode Program 2.4: Kode tampilan awal GUI

e. Rangkaian kode yang muncul setelah membuat label

```
JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
lblNewLabel.setBounds(70, 11, 170, 14);
lblNewLabel.setFont(new Font("Times New Roman",
Font.BOLD, 14));
contentPane.add(lblNewLabel);

JLabel lblBilangan = new JLabel("Bilangan 1");
lblBilangan.setBounds(10, 36, 81, 17);
lblBilangan.setFont(new Font("Times New Roman",
Font.PLAIN, 14));
contentPane.add(lblBilangan);

JLabel lblBilangan_4 = new JLabel("Bilangan 2");
lblBilangan_4.setBounds(10, 64, 81, 17);
lblBilangan_4.setFont(new Font("Times New Roman",
Font.PLAIN, 14));
contentPane.add(lblBilangan_4);

JLabel lblOperator = new JLabel("Operator");
lblOperator.setBounds(10, 114, 81, 17);
lblOperator.setFont(new Font("Times New Roman",
Font.PLAIN, 14));
contentPane.add(lblOperator);

JLabel lblHasil = new JLabel("Hasil");
lblHasil.setBounds(10, 180, 81, 17);
lblHasil.setFont(new Font("Times New Roman",
Font.PLAIN, 14));
contentPane.add(lblHasil);
```

Kode Program 2.5: Kode JLabel

- f. Rangkaian kode yang muncul setelah membuat JTextField

```
txtBill1 = new JTextField();
txtBill1.setBackground(new Color(255, 255, 255));

txtBill1.setHorizontalAlignment(SwingConstants.CENTER);
txtBill1.setBounds(80, 35, 67, 18);
contentPane.add(txtBill1);
txtBill1.setColumns(10);

txtBill2 = new JTextField();

txtBill2.setHorizontalAlignment(SwingConstants.CENTER);
txtBill2.setColumns(10);
txtBill2.setBounds(80, 63, 67, 18);
contentPane.add(txtBill2);
```

Kode Program 2.6: Kode JTextField

- g. Rangkaian kode yang muncul setelah memasukkan JComboBox

```
JComboBox cbOperator = new JComboBox();
cbOperator.setModel(new DefaultComboBoxModel(new
String[] {"+", "-", "*", "/", "%"}));
cbOperator.setBounds(80, 109, 67, 22);
contentPane.add(cbOperator);
```

Kode Program 2.7: Kode JComboBox

- h. Rangkaian kode yang muncul setelah membuat JTextField untuk hasil

```
txtHasil = new JTextField();
txtHasil.setBackground(new Color(255, 255, 255));

txtHasil.setHorizontalAlignment(SwingConstants.CENTER);
txtHasil.setEditable(false);
txtHasil.setBounds(80, 177, 67, 20);
contentPane.add(txtHasil);
txtHasil.setColumns(10);
```

Kode Program 2.1: Kode JTextField Hasil

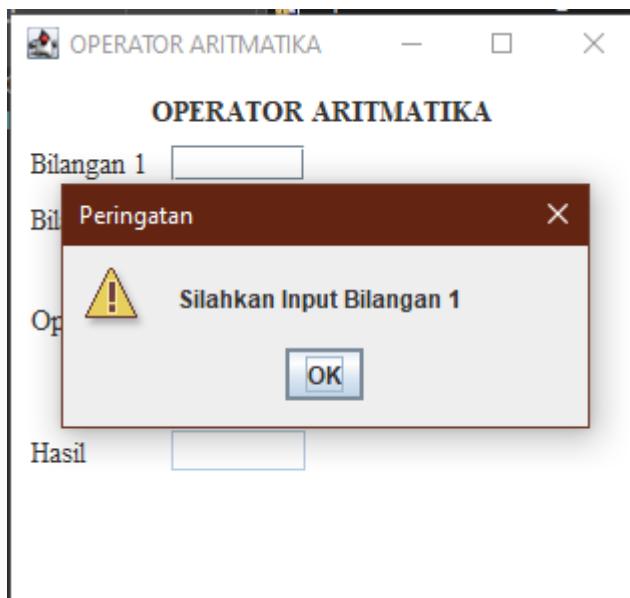
- i. Kemudian kode akan dilanjutkan dengan kode JButton sebagaimana yang telah dijelaskan diatas.

Masing-masing rangkaian kode diatas menggambarkan variabel, nama, warna, format, font, dan lain sebagainya yang berhubungan dengan desain yang telah kita kembangkan. Semua kode program ini kecuali bagian logika JButton, sudah otomatis dibuat saat kita melakukan desain antarmuka diawal. Jika tidak menggunakan WindowBuilder, kita harus menulis kode ini satu-persatu. WindowBuilder tentunya sangat mempermudah kita untuk melakukan pemrograman GUI menggunakan Java Swing.

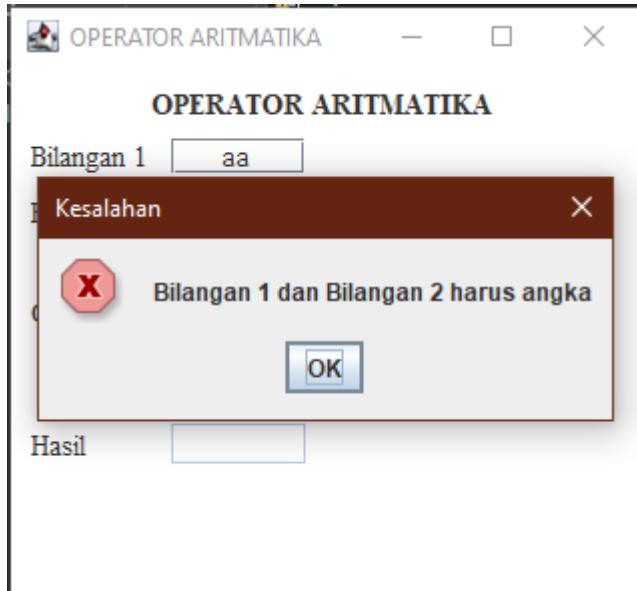
2.4 Output Pemrograman GUI “Operator Aritmatika”

Berikut adalah beberapa contoh output dari pemrograman GUI yang telah kita lakukan:

1. Ketika memanggil pesan error dan pesan peringatan

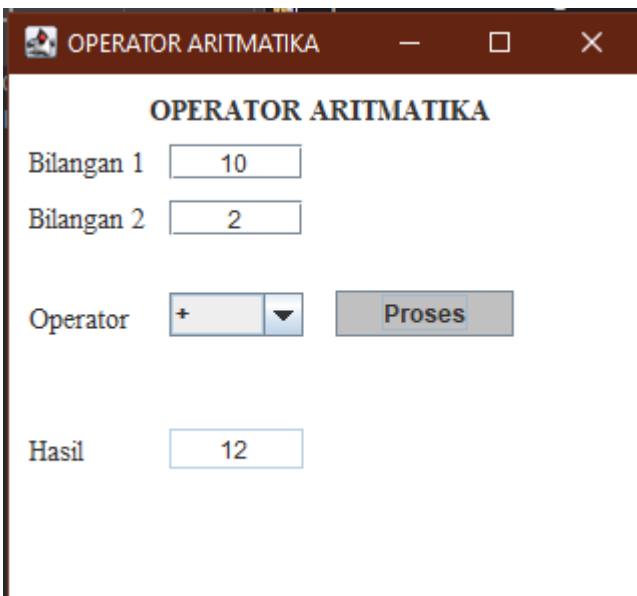


Gambar 2.7: Terjadi peringatan

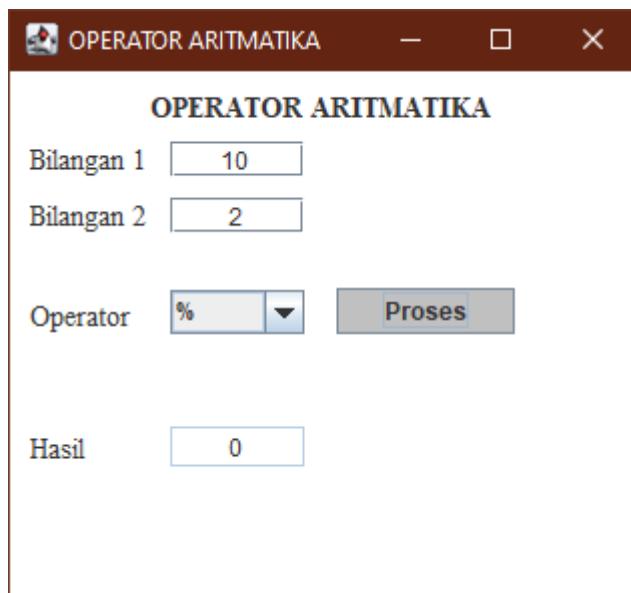


Gambar 2.8: Terjadi error

2. Ketika melakukan operasi aritmatika



Gambar 2.9: Operasi penjumlahan



Gambar 2.10: Operasi modulus

BAB III

KESIMPULAN

3.1. Kesimpulan

Setelah melakukan praktikum dan memahami teori tentang GUI (*Graphical User Interface*) saya merasa bahwasanya GUI sangat amat dibutuhkan dalam perkembangan teknologi saat ini. Jika kita ingin mengembangkan GUI dengan Java, maka kita bisa menggunakan Java Swing atau JavaFX. Tetapi dengan menggunakan Swing dan JavaFX secara langsung, itu berarti kita harus menulis kode secara manual. Oleh karena itu, kita bisa menggunakan WindowBuilder yang ada pada Eclipse IDE untuk membuat desain dengan metode *drag-and-drop* dan tetap bisa mengedit *source* kode sesuai dengan keinginan.

3.2. Saran

Agar lebih mahir dan lebih memahami tentang pemrograman GUI, tentunya kita harus banyak berlatih. Praktikum kali hanyalah dasar dari pemrograman GUI. Tentunya dengan perkembangan teknologi dan *interface* yang semakin maju, kita tidak bisa hanya belajar dasar saja. Jika sudah berhadapan dengan klien, kita akan membuat sebuah *interface* yang sesuai dengan keinginan klien yang tentunya lebih rumit. Pengulangan dan percobaan secara berulang adalah cara yang paling ampuh agar dapat mahir dalam membuat sebuah program dan belajar pemrograman.

DAFTAR PUSTAKA

- [1] H. Schildt, *Java: A Beginner's Guide*, 6th ed. New York, NY: McGraw-Hill Education, 2014
- [2] J. David, Introduction to Programming Using Java, 9th ed. New York, NY, 2022
- [3] "GUI," Revou. [Daring]. Tersedia pada: <https://www.revou.co/id/kosakata/gui>. [Diakses: 19 Nov 2025].
- [4] "Introduction to Java Swing," GeeksforGeeks, 30 Juli 2024. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/introduction-to-java-swing/>. [Diakses: 19 Nov 2025].
- [5] C. Yahampath, "Key Differences Between Java Swing and Java FX," Medium, 11 Januari 2024. [Daring]. Tersedia pada: <https://medium.com/@chirathyahampath99/key-differences-between-java-swing-and-java-fx-1290a407bf0f>. [Diakses: 19 Nov 2025].
- [6] Oracle, "Object-Oriented Programming Concepts," The Java Tutorials. [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/concepts/object.html>. [Diakses: 19 Nov 2025].