

**LAPORAN PRATIKUM**  
**PEMROGRAMAN ALGORITMA PEMROGRAMAN**  
**OPERATOR ARITMATIKA GUI**



**Disusun Oleh:**

Gina Ramadhani

Nim: 2511533014

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

**Asisten Pratikum:** Rahmad Dwi Rizki Olders

**DAPERTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**

## **KATA PENGANTAR**

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum yang berjudul “Operator Aritmatika GUI” ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban atas kegiatan praktikum yang bertujuan untuk memahami penerapan operator aritmatika dalam pemrograman serta mengimplementasikannya melalui antarmuka grafis (Graphical User Interface).

Dalam penyusunan laporan ini, kami memperoleh banyak bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini kami menyampaikan terima kasih kepada dosen pembimbing, asisten praktikum, serta semua pihak yang turut memberikan arahan dan fasilitas sehingga pelaksanaan praktikum dapat berjalan dengan lancar.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh sebab itu, kritik dan saran yang membangun sangat kami harapkan demi penyempurnaan laporan di masa yang akan datang. Akhir kata, semoga laporan praktikum ini dapat memberikan manfaat serta menambah wawasan bagi pembaca mengenai penerapan operator aritmatika pada pemrograman berbasis GUI.

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>I</b>
<b>DAFTAR ISI.....</b>	<b>II</b>
<b>BAB I.....</b>	<b>1</b>
<b>PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan Praktikum .....	2
1.3 Manfaat Praktikum .....	2
<b>BAB II .....</b>	<b>3</b>
<b>PEMBAHASAN .....</b>	<b>3</b>
2.1 Praktikum “OperatorAritmatikaGUI” .....	3
<b>BAB III.....</b>	<b>13</b>
<b>PENUTUP.....</b>	<b>13</b>
3.1 Kesimpulan.....	13
3.2 Saran .....	13
<b>DAFTAR PUSTAKA.....</b>	<b>14</b>

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Perkembangan teknologi informasi mendorong peningkatan kebutuhan terhadap aplikasi yang tidak hanya berfungsi dengan baik, tetapi juga mudah digunakan oleh berbagai kalangan. Dalam pemrograman modern, keberadaan *Graphical User Interface* (GUI) memiliki peran penting karena memberikan pengalaman interaksi yang lebih intuitif dibandingkan antarmuka berbasis teks. Java sebagai bahasa pemrograman yang bersifat *multiplatform*, berorientasi objek, dan didukung oleh pustaka GUI seperti Swing dan JavaFX, menjadi pilihan yang tepat untuk mempelajari dasar-dasar pembuatan aplikasi interaktif. Salah satu konsep fundamental dalam pemrograman adalah operator aritmatika, yang meliputi operasi dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian. Penerapan operator aritmatika dalam aplikasi GUI tidak hanya membantu memahami logika pemrograman, tetapi juga memperkenalkan cara menghubungkan proses perhitungan dengan komponen antarmuka seperti tombol, label, dan *text field*. Dengan mengimplementasikan operator aritmatika dalam bentuk aplikasi GUI, mahasiswa dapat belajar bagaimana suatu input diproses, bagaimana validasi data dilakukan, serta bagaimana hasil perhitungan ditampilkan secara dinamis kepada pengguna.

Praktikum ini dilakukan untuk memberikan pemahaman yang lebih mendalam mengenai cara kerja operator aritmatika dalam Java serta penerapannya dalam pembuatan aplikasi GUI. Melalui praktikum ini, mahasiswa diharapkan mampu merancang antarmuka sederhana, mengintegrasikan logika perhitungan ke dalam komponen GUI, dan mengembangkan kemampuan dasar dalam membangun aplikasi interaktif berbasis Java. Penguasaan konsep ini menjadi langkah awal menuju pengembangan aplikasi yang lebih kompleks di masa mendatang.

## 1.2 Tujuan Praktikum

1. Mahasiswa dapat memahami dan menerapkan operator aritmatika (seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus) di dalam program Java.
2. Mahasiswa mampu merancang antarmuka grafis (GUI) sederhana menggunakan Java (misalnya dengan Swing atau JavaFX) yang menerima input angka dan operator.
3. Mahasiswa dapat memproses input dari pengguna, melakukan perhitungan aritmatika berdasarkan operator yang dipilih, serta menampilkan hasilnya di GUI.
4. Mahasiswa belajar melakukan validasi input (misalnya memeriksa apakah input berupa angka) agar aplikasi GUI berjalan dengan benar.
5. Mahasiswa mengasah kemampuan debugging dengan mengevaluasi dan memperbaiki kesalahan logika atau tampilan dalam aplikasi hitung-aritmatika berbasis GUI.

## 1.3 Manfaat Praktikum

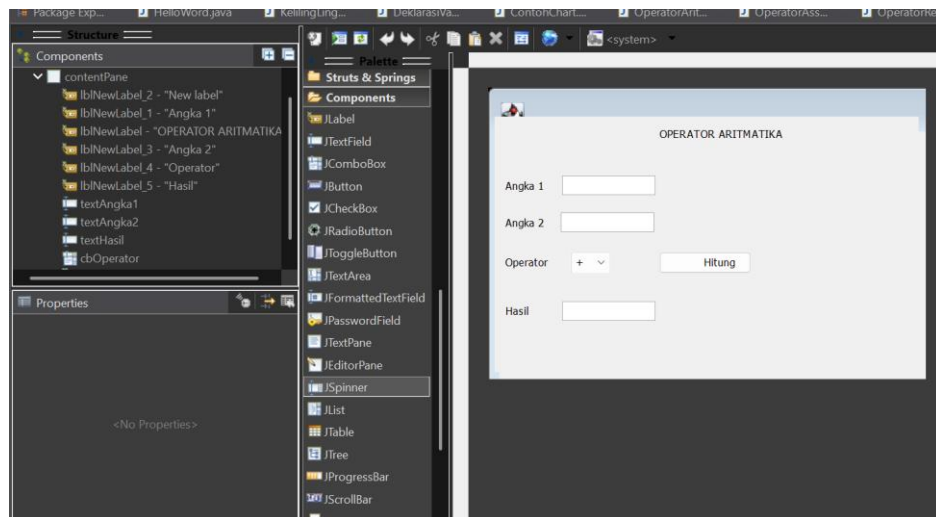
1. Mahasiswa memperkuat pemahaman operator aritmatika di Java, termasuk penggunaan  $+$ ,  $-$ ,  $\times$ ,  $\div$ , dan modulus.
2. Praktikum meningkatkan keterampilan desain GUI dengan Java Swing atau JavaFX, khususnya dalam menangani event dan pengaturan layout.
3. Mahasiswa belajar memvalidasi input numerik, menangani kasus kesalahan seperti pembagian nol, dan memastikan aplikasi berjalan stabil.
4. Praktikum mengembangkan kemampuan berpikir algoritmik dan pemecahan masalah saat menyusun logika perhitungan yang digabung dengan antarmuka pengguna.
5. Mahasiswa memperoleh pengalaman membuat aplikasi interaktif, yang dapat dikembangkan menjadi proyek lebih kompleks seperti kalkulator ilmiah atau sistem edukatif.
6. Kegiatan ini menyiapkan dasar untuk pengembangan aplikasi Java lanjutan, sekaligus memperkuat kolaborasi tim dan dokumentasi kode jika dilakukan secara kelompok.

## BAB II

### PEMBAHASAN

#### 2.1 Praktikum “OperatorAritmatikaGUI”

##### 2.1.1 Langkah Kerja



**Gambar 2.1** Design dari OperatorAritmatikaGUI

Gambar di atas menunjukkan sebuah desain antarmuka aplikasi Java *Swing* yang dibuat menggunakan *NetBeans* GUI Builder untuk menghitung operasi aritmatika sederhana. Pada tampilan tersebut terdapat beberapa komponen utama seperti *JLabel* yang digunakan untuk memberikan keterangan seperti “Angka 1”, “Angka 2”, “Operator”, dan “Hasil”. Dua buah *JTextField* disediakan untuk memasukkan nilai angka pertama dan kedua, sementara satu *JTextField* lain digunakan untuk menampilkan hasil perhitungan. Sebuah *JComboBox* berisi pilihan operator aritmatika seperti +, -, \*, dan / yang dapat dipilih pengguna. Selain itu terdapat *JButton* bertuliskan “Hitung” yang berfungsi menjalankan perhitungan ketika ditekan. Semua komponen tersebut ditata secara rapi dalam sebuah *JPanel* sehingga membentuk sebuah form kalkulator sederhana yang mudah digunakan.

Berikut penjelasan langkah kerjanya:

#### Langkah–Langkah Pembuatan Desain Aplikasi Operator Aritmatika

##### 1. Membuat JFrame Form

1. Buka NetBeans, pilih File → New File.

2. Pilih Swing GUI Forms → JFrame Form.
3. Beri nama file, misalnya OperatorAritmatika.java.
4. Klik Finish, maka NetBeans akan menampilkan area desain GUI kosong.

## 2. Menambahkan JLabel

JLabel digunakan sebagai teks keterangan untuk setiap komponen pada form.

Langkah-langkah:

1. Buka panel Palette, pilih JLabel.
2. Drag and drop JLabel ke form desain.
3. Pada panel Properties, atur:

- text → “Angka 1”
- font → sesuaikan
- name → lblAngka1

Ulangi untuk label: “Angka 2”, “Operator”, “Hasil”, dan judul “OPERATOR ARITMATIKA”.

## 3. Menambahkan JTextField

JTextField digunakan untuk memasukkan angka dan menampilkan hasil.

Langkah-langkah:

1. Dari panel Palette, pilih JTextField.
2. Drag and drop di samping label “Angka 1”.
3. Pada Properties, atur:

- name → textAngka1
- kosongkan bagian text

Lakukan hal yang sama untuk textAngka2 dan textHasil (textHasil dibuat tidak dapat diedit dengan mengubah editable menjadi false).

## 4. Menambahkan JComboBox sebagai Pilihan Operator

ComboBox digunakan untuk memilih operator aritmatika seperti +, -, \*, dan /, %.

Langkah-langkah:

1. Pilih JComboBox dari Palette.

2. Letakkan di samping label “Operator”.
3. Pada Properties, atur:
  - name → cbOperator
  - model → diisi dengan pilihan operator: +, -, \*, /,%
5. Menambahkan JButton “Hitung”

Tombol ini digunakan untuk menjalankan proses perhitungan.

Langkah-langkah:

1. Pilih JButton dari Palette.
2. Letakkan di bagian kanan ComboBox.
3. Pada Properties, atur:
  - text → “Hitung”
  - name → btnHitung
6. Menata Tata Letak (Layout Management)

Penataan tata letak dilakukan agar komponen tertata rapi.

NetBeans menggunakan Absolute Layout. Penataan dapat dilakukan dengan menggeser komponen, mengatur ukuran, serta memanfaatkan garis biru (auto snap) sebagai panduan.

Berikut source code dari design OperatorAritmatikaGUI:

```

1 package pekan8_2511533014;
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 import javax.swing.SwingConstants;
11 import javax.swing.JTextField;
12 import javax.swing.JButton;
13 import javax.swing.JComboBox;
14 import javax.swing.DefaultComboBoxModel;
15 import java.awt.event.ActionListener;
16 import java.awt.event.ActionEvent;
17
18 public class OperatorAritmatikaGUI_2511533014 extends JFrame {
19
20     private static final long serialVersionUID = 1L;
21     private JPanel contentPane;
22     private JTextField textAngka1;
23     private JTextField textAngka2;
24     private JTextField textHasil;
25
26     /**
27      * Launch the application.
28      */
29     private void pesanPeringatan(String pesan) {
30         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
31     }
32     private void pesanError(String pesan) {
33         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
34     }
35 }

```

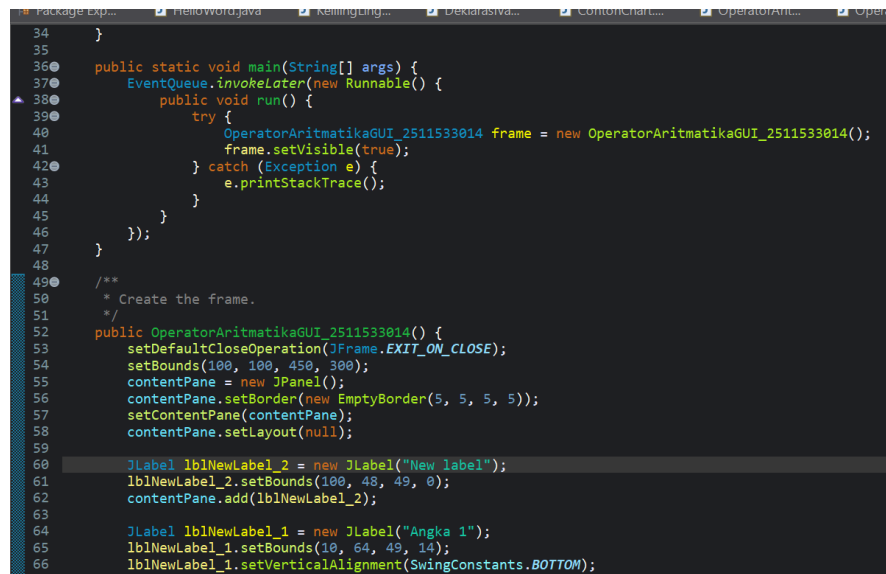
**Gambar 2.2** Kode program OperatorAritmatikaGUI



Kode di atas merupakan bagian awal dari program Java Swing yang digunakan untuk membuat aplikasi perhitungan aritmatika. Pada bagian tersebut terdapat deklarasi package, impor berbagai library Swing seperti JFrame, JPanel, JLabel, JTextField, JButton, dan JComboBox yang digunakan untuk membangun elemen GUI. Kode juga mengimpor ActionListener dan(ActionEvent) untuk menangani aksi pengguna. Kelas `OperatorAritmatikaGUI_2511533014` memperluas JFrame sebagai jendela utama aplikasi. Di dalamnya terdapat deklarasi beberapa komponen seperti panel utama (`contentPane`), text field untuk angka pertama (`textAngka1`), angka kedua (`textAngka2`), dan hasil (`textHasil`). Selain itu, terdapat dua metode penting yaitu `pesanPeringatan()` dan `pesanError()` yang berfungsi menampilkan pemberitahuan menggunakan JOptionPane apabila terjadi kesalahan atau input tidak valid.

Berikut penjelasan langkah-langkah kerjanya:

1. Program memulai dengan mendefinisikan package dan melakukan impor library yang diperlukan untuk membangun komponen GUI serta menangani event.
2. Kelas `OperatorAritmatikaGUI_2511533014` dibuat sebagai turunan dari JFrame, sehingga menjadi jendela utama dari aplikasi.
3. Variabel–variabel GUI seperti `contentPane`, `textAngka1`, `textAngka2`, dan `textHasil` dideklarasikan untuk menyimpan komponen visual yang akan ditampilkan.
4. Program menyediakan metode `pesanPeringatan()` untuk menampilkan pesan peringatan jika pengguna memasukkan input yang tidak sesuai.
5. Program menyediakan metode `pesanError()` untuk menampilkan pesan kesalahan melalui JOptionPane jika terjadi error dalam proses perhitungan atau validasi.
6. Selanjutnya (pada bagian kode yang tidak ditampilkan), komponen–komponen GUI tersebut akan diinisialisasi, ditata pada frame, dan diberikan event listener untuk menjalankan operasi aritmatika ketika tombol ditekan.



```

34     }
35
36     public static void main(String[] args) {
37         EventQueue.invokeLater(new Runnable() {
38             public void run() {
39                 try {
40                     OperatorAritmatikaGUI_2511533014 frame = new OperatorAritmatikaGUI_2511533014();
41                     frame.setVisible(true);
42                 } catch (Exception e) {
43                     e.printStackTrace();
44                 }
45             }
46         });
47     }
48
49     /**
50      * Create the frame.
51      */
52     public OperatorAritmatikaGUI_2511533014() {
53         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
54         setBounds(100, 100, 450, 300);
55         contentPane = new JPanel();
56         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
57         setContentPane(contentPane);
58         contentPane.setLayout(null);
59
60         JLabel lblNewLabel_2 = new JLabel("New label");
61         lblNewLabel_2.setBounds(100, 48, 49, 0);
62         contentPane.add(lblNewLabel_2);
63
64         JLabel lblNewLabel_1 = new JLabel("Angka 1");
65         lblNewLabel_1.setBounds(10, 64, 49, 14);
66         lblNewLabel_1.setVerticalAlignment(SwingConstants.BOTTOM);

```

**Gambar 2.3** Lanjutan kode program OperatorAritmatikaGUI

Kode di atas merupakan lanjutan dari kode program diatas dari program Java Swing yang berfungsi untuk menampilkan dan menginisialisasi jendela aplikasi Operator Aritmatika. Pada metode main, program menggunakan `EventQueue.invokeLater()` untuk memastikan bahwa pembuatan dan penampilan GUI dilakukan pada Event Dispatch Thread, yaitu thread khusus untuk GUI Swing. Di dalamnya, objek `OperatorAritmatikaGUI_2511533014` dibuat dan ditampilkan dengan `setVisible(true)`. Selanjutnya, konstruktor kelas ini bertanggung jawab membangun tampilan GUI, mulai dari mengatur operasi penutupan jendela, ukuran frame, membuat panel utama (`contentPane`), mengatur layout, serta menambahkan komponen seperti `JLabel` ke dalam frame menggunakan koordinat absolut melalui `setBounds()`.

Berikut penjelasan langkah-langkah keernjanya:

1. Program dijalankan melalui metode main, yang memanggil `EventQueue.invokeLater()` agar pembuatan GUI berjalan pada thread GUI Swing yang aman.
2. Di dalam `invokeLater`, program membuat objek baru dari kelas `OperatorAritmatikaGUI_2511533014` dan menampilkannya menggunakan `setVisible(true)`.

3. Konstruktor `OperatorAritmatikaGUI_2511533014()` mulai membuat frame dengan menentukan operasi penutupan jendela menggunakan `setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)`.
4. Frame diberi ukuran menggunakan `setBounds(100, 100, 450, 300)` untuk menentukan posisi awal dan ukurannya.
5. Panel utama (`contentPane`) dibuat menggunakan `new JPanel()`, kemudian diberi border dan ditambahkan ke frame sebagai kontainer komponen GUI.
6. Layout pada `contentPane` diset ke null, artinya program menggunakan Absolute Layout, sehingga setiap komponen harus ditentukan posisi dan ukurannya secara manual.
7. `JLabel` pertama dibuat menggunakan `new JLabel("New label")`, kemudian diberi posisi dengan `setBounds()` dan ditambahkan ke panel menggunakan `contentPane.add()`.
8. Komponen `JLabel` lainnya juga dibuat dengan cara yang sama, seperti `JLabel lblNewLabel_1 = new JLabel("Angka 1")`, lalu diletakkan sesuai koordinat yang ditentukan.

```

67     contentPane.add(lblNewLabel_1);
68
69     JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
70     lblNewLabel.setBounds(169, 11, 127, 14);
71     contentPane.add(lblNewLabel);
72
73     JLabel lblNewLabel_3 = new JLabel("Angka 2");
74     lblNewLabel_3.setBounds(10, 86, 48, 46);
75     contentPane.add(lblNewLabel_3);
76
77     JLabel lblNewLabel_4 = new JLabel("Operator");
78     lblNewLabel_4.setBounds(10, 143, 48, 14);
79     contentPane.add(lblNewLabel_4);
80
81     JLabel lblNewLabel_5 = new JLabel("Hasil");
82     lblNewLabel_5.setBounds(11, 193, 48, 14);
83     contentPane.add(lblNewLabel_5);
84
85     textAngka1 = new JTextField();
86     textAngka1.setBounds(69, 61, 96, 20);
87     contentPane.add(textAngka1);
88     textAngka1.setColumns(10);
89
90     textAngka2 = new JTextField();
91     textAngka2.setBounds(68, 99, 96, 20);
92     contentPane.add(textAngka2);
93     textAngka2.setColumns(10);
94
95     textHasil = new JTextField();
96     textHasil.setBounds(69, 190, 96, 20);
97     contentPane.add(textHasil);
98     textHasil.setColumns(10);
99

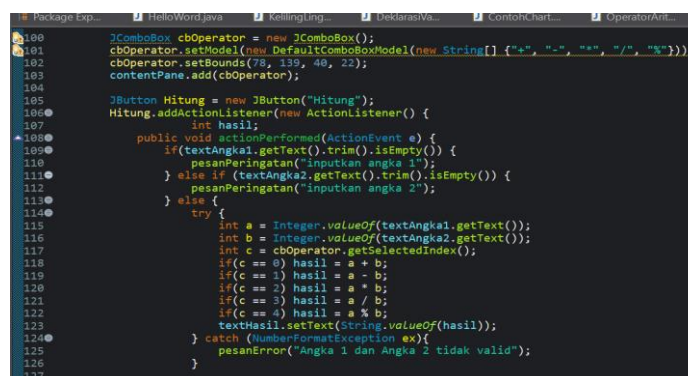
```

**Gambar 2.4** Lanjutan kode program `OperatorAritmatikaGUI`

Kode program tersebut digunakan untuk membangun tampilan GUI menggunakan Java Swing. Di dalamnya terdapat beberapa komponen seperti JLabel dan JTextField yang ditambahkan ke contentPane. Setiap komponen diatur posisinya secara manual menggunakan setBounds, lalu ditempatkan satu per satu ke panel. Label digunakan untuk memberikan keterangan seperti "Operator Aritmatika", "Angka 2", "Operator", dan "Hasil", sedangkan tiga buah text field digunakan untuk menerima input angka pertama, angka kedua, serta menampilkan hasil perhitungan.

Langkah kerja program:

1. Program membuat JLabel bertuliskan "OPERATOR ARITMATIKA" dan menempatkannya pada posisi tertentu.
2. Program membuat dan menambahkan JLabel untuk keterangan "Angka 2".
3. Program membuat dan menambahkan JLabel untuk keterangan "Operator".
4. Program membuat dan menambahkan JLabel untuk tulisan "Hasil".
5. Program membuat JTextField pertama untuk input angka pertama, mengatur posisinya, dan menambahkannya ke contentPane.
6. Program membuat JTextField kedua untuk input angka kedua dan menambahkannya ke panel pada posisi yang ditentukan.
7. Program membuat JTextField ketiga untuk menampilkan hasil, mengatur posisinya, lalu menambahkannya ke panel.
8. Setiap JTextField diberikan jumlah kolom menggunakan setColumns untuk menentukan lebar tampilan input.



```

100 JComboBox cbOperator = new JComboBox();
101 cbOperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/"}));
102 cbOperator.setBounds(78, 139, 40, 22);
103 contentPane.add(cbOperator);
104
105 JButton Hitung = new JButton("Hitung");
106 Hitung.addActionListener(new ActionListener() {
107     int hasil;
108     public void actionPerformed(ActionEvent e) {
109         if(textAngka1.getText().trim().isEmpty()) {
110             pesanPeringatan("inputkan angka 1");
111         } else if (textAngka2.getText().trim().isEmpty()) {
112             pesanPeringatan("inputkan angka 2");
113         } else {
114             try {
115                 int a = Integer.valueOf(textAngka1.getText());
116                 int b = Integer.valueOf(textAngka2.getText());
117                 int c = cbOperator.getSelectedIndex();
118                 if(c == 0) hasil = a + b;
119                 if(c == 1) hasil = a - b;
120                 if(c == 2) hasil = a * b;
121                 if(c == 3) hasil = a / b;
122                 if(c == 4) hasil = a % b;
123                 textHasil.setText(String.valueOf(hasil));
124             } catch (NumberFormatException ex) {
125                 pesanError("Angka 1 dan Angka 2 tidak valid");
126             }
127         }
128     }
129 });

```

**Gambar 2.5** Lanjutan Kode program OperatorAritmatikaGUI

Kode program ini berfungsi untuk melakukan operasi aritmatika berdasarkan pilihan operator pada *JComboBox*. Ketika tombol “Hitung” ditekan, program mengecek apakah kedua *field* angka sudah terisi. Jika salah satu kosong, program menampilkan pesan peringatan. Jika keduanya terisi, program mencoba mengubah nilai input menjadi angka dan mengambil operator yang dipilih. Berdasarkan operator tersebut, program melakukan perhitungan seperti penjumlahan, pengurangan, perkalian, atau pembagian. Hasilnya kemudian ditampilkan pada *text field* hasil. Jika input bukan angka yang valid, maka akan tampil pesan error.

Langkah kerja program:

1. Program menyiapkan *JComboBox* berisi pilihan operator seperti +, -, ×, ÷.
2. Tombol “Hitung” dibuat dan diberi *ActionListener* untuk menangani proses perhitungan.
3. Ketika tombol ditekan, program mengecek apakah *textAngka1* dan *textAngka2* sudah diisi.
4. Jika ada yang kosong, program memunculkan pesan peringatan sesuai input yang belum terisi.
5. Jika kedua input terisi, program mengambil nilai angka pertama dan kedua, lalu mengubahnya menjadi *integer*.
6. Program mengambil operator yang dipilih dari *JComboBox*.
7. Program melakukan perhitungan sesuai operator:
  - Jika operator +  $\rightarrow$  hasil =  $a + b$
  - Jika operator -  $\rightarrow$  hasil =  $a - b$
  - Jika operator  $\times$   $\rightarrow$  hasil =  $a * b$
  - Jika operator  $\div$   $\rightarrow$  hasil =  $a / b$
8. Hasil perhitungan ditampilkan pada *textHasil* melalui *setText()*.
9. Jika pengguna menginput nilai yang bukan angka, program menampilkan pesan error “Angka 1 dan Angka 2 tidak valid”.

```

128
129     }
130 }
131 });
132 Hitung.setHorizontalAlignment(SwingConstants.RIGHT);
133 Hitung.setBounds(128, 139, 65, 22);
134 contentPane.add(Hitung);
135
136
137
138 }
139 }
140

```

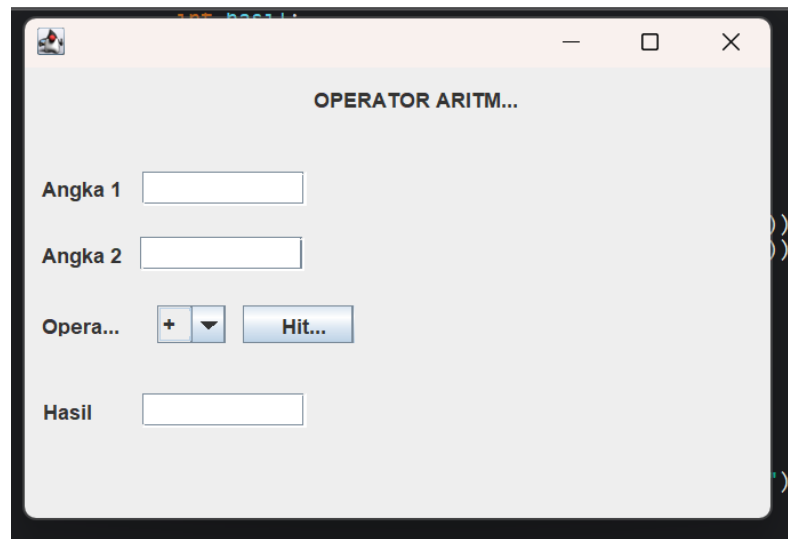
**Gambar 2.6** Lanjutan kode program OperatorAritmatikaGUI

Kode tersebut merupakan bagian akhir dari konfigurasi tombol *Hitung* pada antarmuka *Java Swing*. Setelah *ActionListener* ditutup, tombol diberi pengaturan tampilan, yaitu perataan teks secara horizontal ke kanan menggunakan *setHorizontalAlignment()*. Selanjutnya posisi dan ukuran tombol diatur melalui *setBounds()*, sehingga tombol muncul pada lokasi yang diinginkan di dalam jendela aplikasi. Terakhir, tombol tersebut ditambahkan ke dalam *contentPane* agar tampil sebagai bagian dari *GUI*.

Langkah kerja program:

1. Program menetapkan perataan teks tombol *Hitung* ke arah kanan menggunakan *SwingConstants.RIGHT*.
2. Program mengatur ukuran dan posisi tombol pada layar menggunakan *setBounds(128, 139, 65, 22)*.
3. Program menambahkan tombol *Hitung* ke dalam *contentPane* agar muncul pada tampilan *GUI*.

Dari penjelasan langkah-langkah kerja kode program diatas menghasilkan output seperti ini:



**Gambar 2.7** Output kode program OperatorAritmatikaGUI

Berikut analisis dari output program diatas:

1. Program berhasil menampilkan jendela utama *GUI* dengan judul “Operator Aritmatika”, sesuai label yang telah dibuat dalam kode.
2. Dua kolom input muncul dengan label Angka 1 dan Angka 2, menunjukkan bahwa pengguna harus mengisi dua nilai sebagai *operand*.
3. Komponen *combo box* tampil di bagian operator dan menyediakan pilihan simbol seperti +, -, \*, /, %.
4. Tombol Hitung terlihat di samping *combo box* dan berfungsi untuk menjalankan proses perhitungan saat diklik.
5. Kolom input dengan label Hasil muncul untuk menampilkan hasil proses perhitungan setelah tombol Hitung ditekan.
6. Penataan komponen terlihat rapi menggunakan metode *setBounds()*, sehingga posisi setiap elemen sesuai dengan pengaturan pada kode.
7. Tidak ada *error* pada tampilan, sehingga *GUI* dapat berjalan sebagaimana mestinya.

## BAB III

### PENUTUP

#### 3.1 Kesimpulan

Berdasarkan hasil praktikum, dapat disimpulkan bahwa implementasi operator logika pada *GUI* menggunakan bahasa pemrograman Java dapat berjalan dengan baik apabila komponen antarmuka dan logika program dirancang secara terstruktur. Penggunaan komponen seperti *JLabel*, *JTextField*, *JButton*, dan *JComboBox* memungkinkan pengguna melakukan interaksi secara visual sehingga proses pengolahan data menjadi lebih mudah dipahami. Selain itu, penerapan operator logika pada *event handler* membantu proses validasi input serta pengambilan keputusan dalam program. Secara keseluruhan, praktikum ini memberikan pemahaman mengenai bagaimana logika pemrograman dipadukan dengan antarmuka grafis untuk mendukung fungsi aplikasi yang interaktif.

#### 3.2 Saran

Mahasiswa disarankan untuk lebih teliti saat menyusun tampilan *GUI* agar komponen yang digunakan terlihat rapi dan mudah dipahami oleh pengguna. Selain itu, penting bagi mahasiswa untuk sering berlatih menggunakan operator logika supaya tidak kebingungan ketika menggabungkannya dengan *event* pada tombol atau proses lain di dalam program. Mahasiswa juga sebaiknya menambahkan pengecekan input sederhana, seperti memastikan kolom tidak dibiarkan kosong, agar program yang dibuat lebih aman dan tidak mudah mengalami kesalahan. Untuk membuat tampilan lebih menarik, mahasiswa bisa mencoba menggunakan warna atau memanfaatkan *layout manager* agar tidak perlu selalu mengatur posisi secara manual. Terakhir, mahasiswa dianjurkan membaca dokumentasi Java atau mencoba fitur baru lainnya agar kemampuan dalam membuat aplikasi *GUI* terus berkembang seiring waktu.



## DAFTAR PUSTAKA

- [1] Oracle, “Java Platform, Standard Edition Documentation.”  
<https://docs.oracle.com/javase/> (diakses: 23-Nov-2025).
- [2] Oracle, “Trail: Creating a GUI With Swing.”  
<https://docs.oracle.com/javase/tutorial/uiswing/> (diakses: 22-Nov-2025).
- [3] D. J. Eck, “Introduction to Programming Using Java, Version 8.”  
<http://math.hws.edu/javanotes/> (diakses: 22-Nov-2025).
- [4] Oracle, “Java Tutorials – Event Handling.”  
<https://docs.oracle.com/javase/tutorial/uiswing/events/> (diakses: 22-Nov-2025).
- [5] GeeksforGeeks, “Java Logical Operators.”  
<https://www.geeksforgeeks.org/java-logical-operators/> (diakses: 22-Nov-2025).