

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
OPERATOR ARITMATIKA GUI**



Disusun oleh:

Ghinada Fathanawafa Algma
2511533008

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

Asisten Praktikum:
Rahmad Dwirizki Olders

**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum dengan judul “*Operator Aritmatika GUI*” tepat pada waktunya. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum Algoritma dan Pemrograman pada pekan ke delapan, khususnya dalam memahami dan mengimplementasikan konsep operator aritmatika melalui antarmuka grafis (*Graphical User Interface*) pada Bahasa Java dengan menggunakan eclipse windows builder.

Praktikum ini bertujuan untuk melatih mahasiswa agar memahami suatu aplikasi antar muka dengan menggunakan bahasa pemrograman Java, juga memberikan pengalaman langsung dalam merancang tampilan interaktif yang memudahkan pengguna melakukan operasi perhitungan, dan mengintegrasikan teori pemrograman dengan praktik nyata sehingga menghasilkan pemahaman yang lebih komprehensif.

Dalam penyusunan laporan ini, penulis menyadari masih terdapat kekurangan baik dari segi isi maupun penyajian. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pembimbing, asisten praktikum, serta rekan-rekan yang telah memberikan dukungan dan bantuan selama proses praktikum berlangsung. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi dalam pengembangan ilmu pengetahuan di bidang pemrograman.

Padang, November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan.....	1
1.3 Manfaat	2
BAB II PEMBAHASAN	3
2.1 Dasar Teori	3
2.1.1 Operator Aritmatika	3
2.1.2 GUI (<i>Graphical User Interface</i>)	3
2.1.3 Integrasi Operator Aritmatika dengan GUI.....	4
2.1.4 Relevansi Dalam Pemrograman	4
2.2 Langkah Kerja.....	4
2.2.1 Membuat <i>package</i>	4
2.2.2 Membuat <i>frame</i>	5
2.2.3 Program Operator Aritmatika GUI.....	7
2.2.4 Kode Program Keseluruhan	15
2.3 Output Program.....	17
2.3.1 Output 1 (Jika tidak ada kesalahan)	17
2.3.2 Output 2 (Jika mengosongkan salah satu bilangan).....	17
2.3.3 Output 3 (Jika yang di input bukan angka)	18
BAB III KESIMPULAN.....	19

3.1 Kesimpulan	19
DAFTAR PUSTAKA	20

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah membawa perubahan besar dalam cara manusia berinteraksi dengan komputer. Salah satu aspek penting dalam pemrograman adalah pemahaman terhadap operator aritmatika, yang menjadi dasar dalam melakukan berbagai perhitungan matematis. Operator ini tidak hanya digunakan dalam perhitungan sederhana, tetapi juga menjadi fondasi bagi algoritma yang lebih kompleks dalam bidang komputasi.

Dalam praktik pemrograman modern, penggunaan Graphical User Interface (GUI) menjadi semakin penting karena memberikan kemudahan bagi pengguna untuk berinteraksi dengan aplikasi secara visual dan intuitif. Dengan adanya GUI, operasi aritmatika dapat dilakukan melalui tombol, kotak teks, dan elemen grafis lainnya, sehingga pengguna tidak perlu berhadapan langsung dengan kode program. Hal ini meningkatkan aksesibilitas dan kenyamanan dalam penggunaan aplikasi.

Dengan adanya praktikum ini, diharapkan mahasiswa mampu memahami keterkaitan antara logika perhitungan aritmatika dengan desain antarmuka grafis, sehingga dapat mengembangkan aplikasi yang tidak hanya fungsional tetapi juga mudah digunakan.

1.2 Tujuan

1. Memahami konsep dasar operator aritmatika dalam pemrograman, seperti penjumlahan, pengurangan, perkalian, pembagian dan sisa hasil bagi.
2. Mengimplementasikan operator aritmatika ke dalam program berbasis GUI, sehingga pengguna dapat melakukan perhitungan melalui antar muka grafis yang interaktif.

3. Melatih keterampilan mahasiswa dalam merancang antar muka grafis yang sederhana, fungsional, dan ramah pengguna.
4. Menghubungkan teori dengan praktik, khususnya bagaimana logika perhitungan aritmatika dapat diterapkan dalam aplikasi nyata.
5. Meningkatkan kemampuan *problem solving* melalui pembuatan aplikasi sederhana yang mengintegrasikan logika aritmatika dengan desain GUI.

1.3 Manfaat

1. Mahasiswa lebih memahami cara kerja operator aritmatika dalam bahasa pemrograman Java.
2. Menghubungkan logika perhitungan aritmatika dengan implementasi dalam aplikasi berbasis GUI.
3. Melatih kemampuan membuat tampilan program yang interaktif dan ramah pengguna.
4. Membiasakan mahasiswa untuk menyelesaikan masalah melalui perancangan aplikasi sederhana.
5. Memberikan bekal awal memahami integrasi logika perhitungan dengan desain aplikasi yang lebih kompleks.
6. Mendorong mahasiswa untuk mengembangkan aplikasi yang tidak hanya fungsional tetapi juga menarik secara visual.

BAB II

PEMBAHASAN

2.1 Dasar Teori

2.1.1 Operator Aritmatika

Operator aritmatika adalah simbol atau instruksi dalam bahasa pemrograman yang digunakan untuk melakukan operasi matematis dasar. Operator ini meliputi:

1. Penjumlahan (+) → digunakan untuk menambahkan dua nilai.
2. Pengurangan (-) → digunakan untuk mengurangi nilai pertama dengan nilai kedua.
3. Perkalian (*) → digunakan untuk mengalikan dua nilai.
4. Pembagian (/) → digunakan untuk membagi nilai pertama dengan nilai kedua.
5. Modulus (%) → digunakan untuk mendapatkan sisa hasil pembagian.

2.1.2 GUI (*Graphical User Interface*)

GUI adalah antarmuka pengguna berbasis grafis yang memungkinkan interaksi dengan program melalui elemen visual seperti tombol, kotak teks, label, dan menu. GUI mempermudah pengguna dalam menjalankan aplikasi tanpa harus berhadapan langsung dengan kode program. Karakteristik GUI:

1. Interaktif → pengguna dapat memberikan input melalui klik, ketikan, atau pilihan menu.
2. Visual → menampilkan informasi dalam bentuk grafis yang mudah dipahami.
3. User-friendly → dirancang agar mudah digunakan oleh berbagai kalangan.

2.1.3 Integrasi Operator Aritmatika dengan GUI

Dalam praktikum ini, operator aritmatika diimplementasikan melalui GUI sehingga pengguna dapat melakukan perhitungan dengan cara memasukkan angka ke dalam kotak teks dan menekan tombol operasi. Program kemudian akan menampilkan hasil perhitungan secara otomatis. Konsep ini menghubungkan logika perhitungan matematis dengan desain antarmuka grafis, sehingga aplikasi menjadi lebih fungsional dan mudah digunakan.

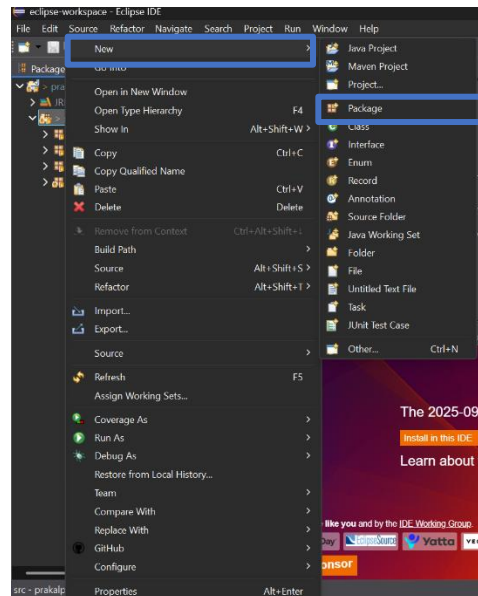
2.1.4 Relevansi Dalam Pemrograman

- Membiasakan mahasiswa memahami dasar logika perhitungan.
- Melatih keterampilan merancang aplikasi sederhana berbasis GUI.
- Menjadi bekal untuk pengembangan aplikasi yang lebih kompleks.

2.2 Langkah Kerja

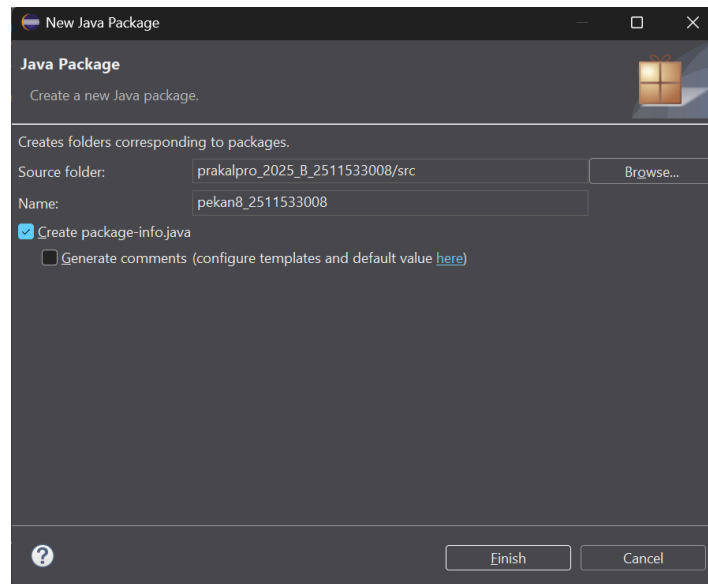
2.2.1 Membuat *package*

1. Pertama tama kita buat *package* terlebih dahulu caranya klik kanan dan plih *new* lalu klik *package*.



Gambar 2.1 membuat *package*

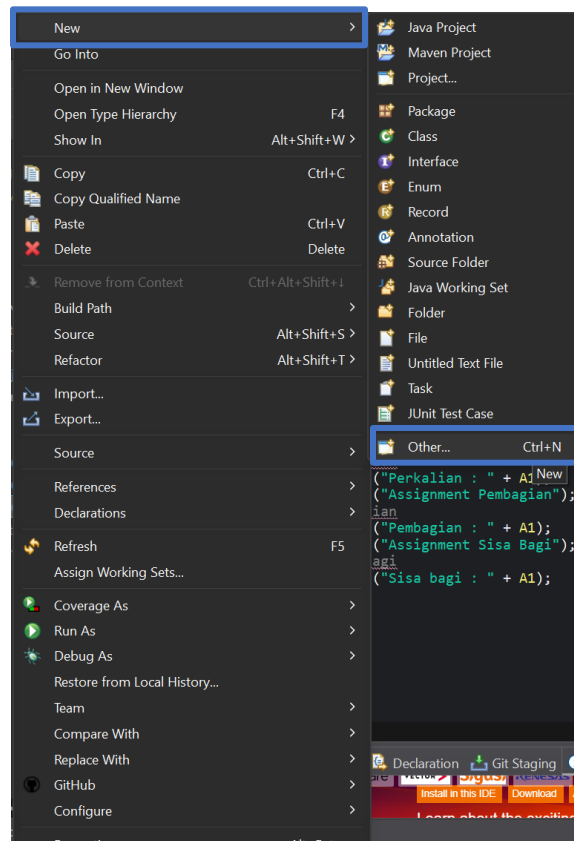
2. Setelah itu kita akan diminta untuk mengisi nama *package*.



Gambar 2.2 nama package

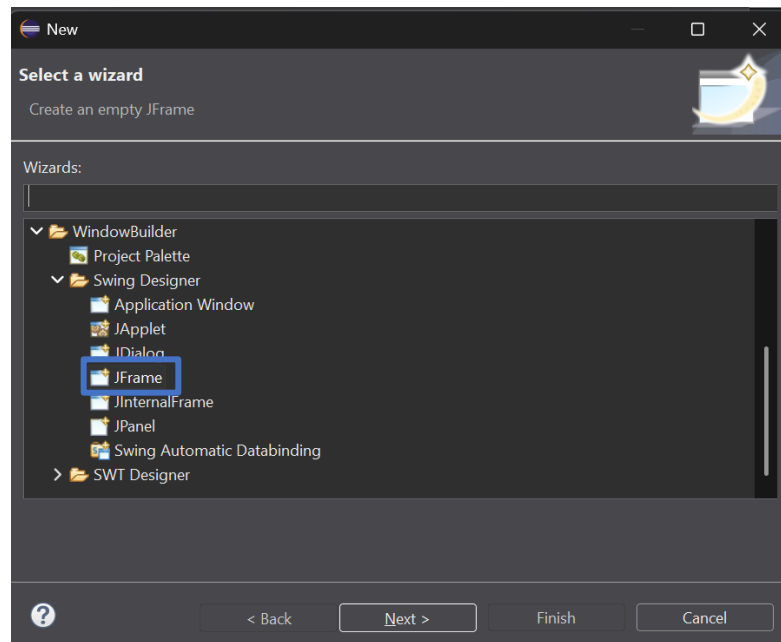
2.2.2 Membuat *frame*

1. Pertama tama klik kanan pada *package* dan pilih *new* lalu klik *other*.



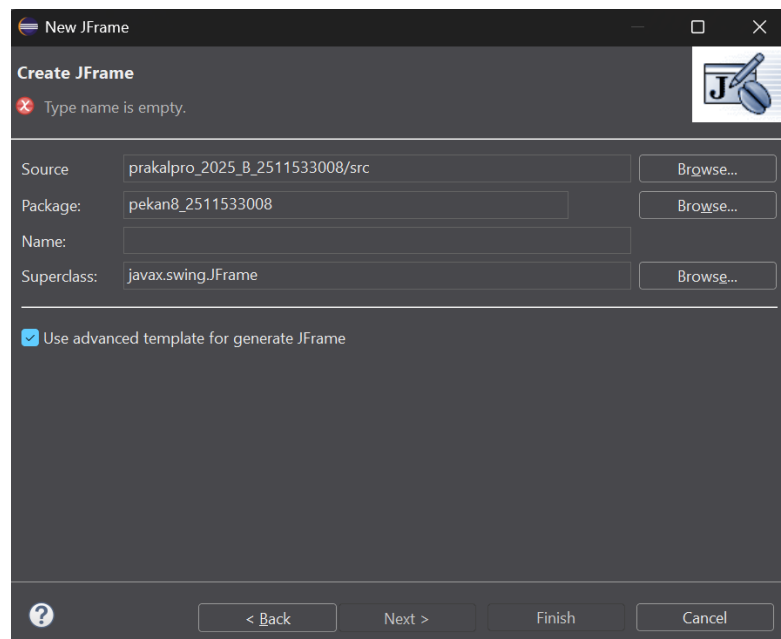
Gambar 2.3 langkah 1 membuat *frame*

2. Setelah itu akan muncul tampilan seperti ini dan klik *JFrame* yang ada di menu *WindowBuilder*, dan klik *next*.



Gambar 2.4 langkah 2 membuat frame

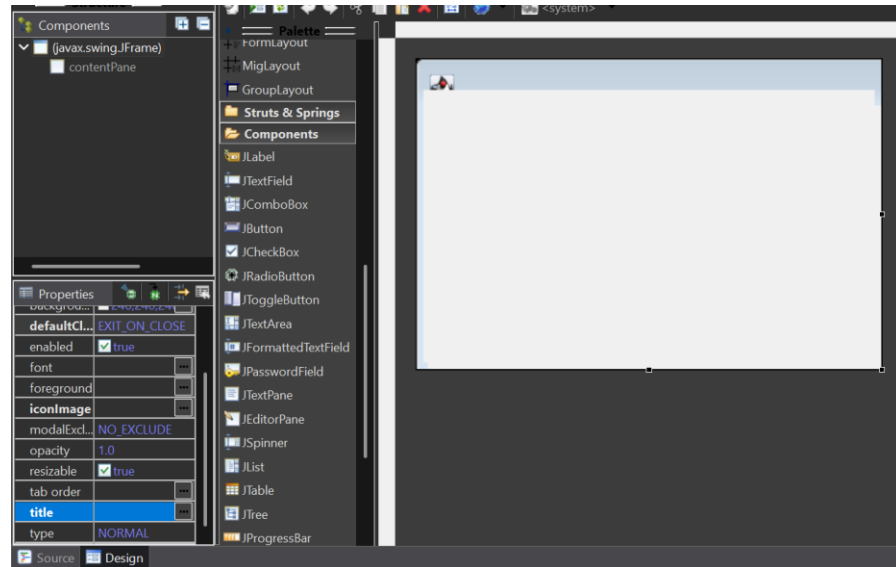
3. Setelah itu kita akan diminta untuk memberi nama *JFrame* yang dibuat.



Gambar 2.5 memberi nama frame

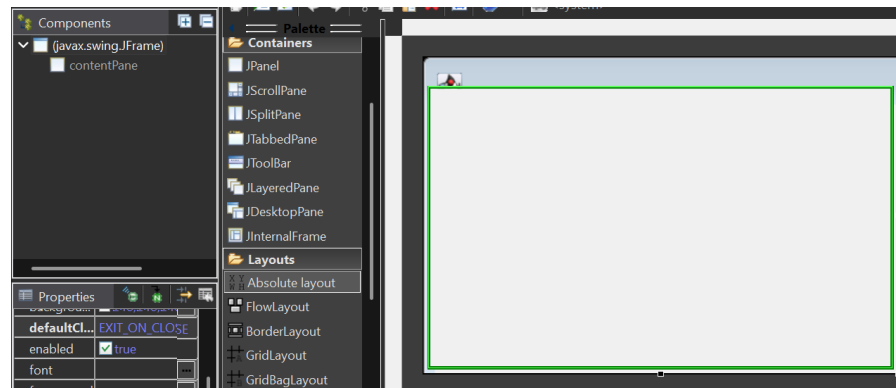
2.2.3 Program Operator Aritmatika GUI

1. Setelah berhasil membuat *Jframe* kita akan mulai mendesain tampilan dari bentuk aplikasinya pada menu *design*.



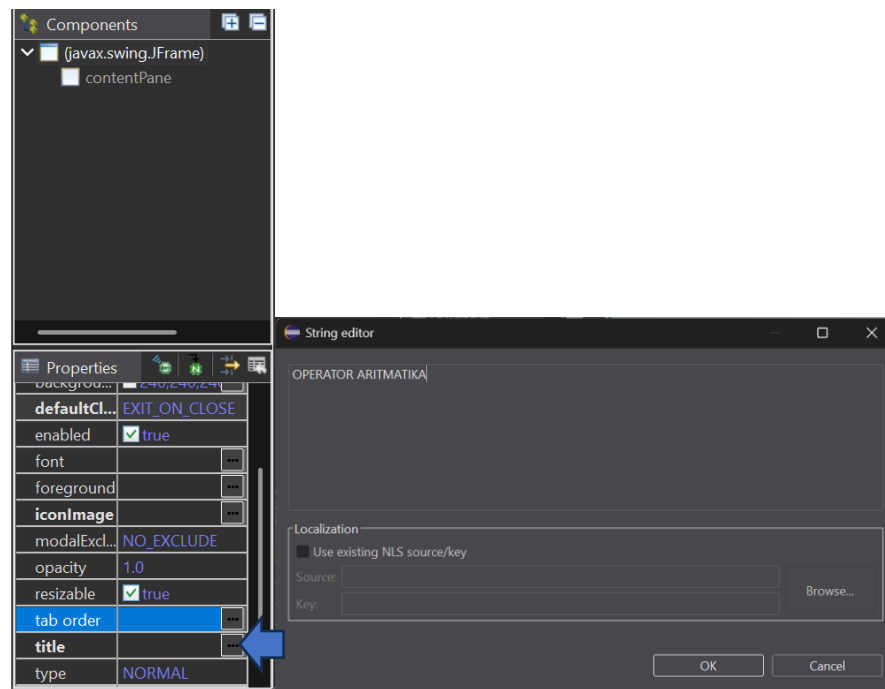
Gambar 2.6 tampilan design

2. Klik (*javax.swing.JFrame*) ubah layout nya menjadi *absolute layout*.



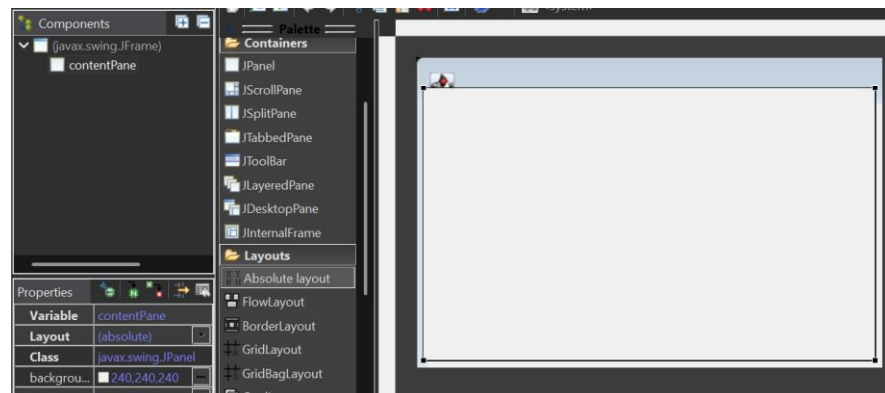
Gambar 2.7 ubah layout

3. Pada *title* klik titik 3 dan ketik OPERATOR ARITMATIKA.



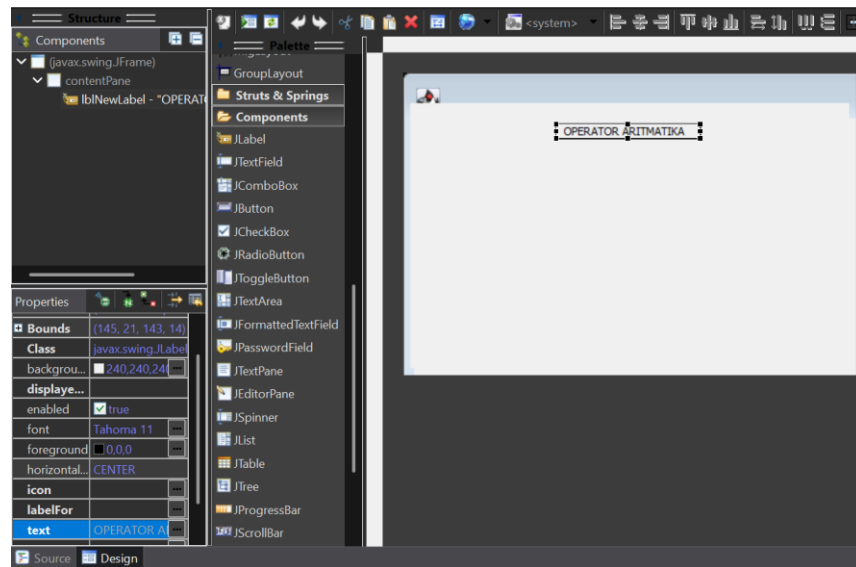
Gambar 2.8 menambahkan title

4. Setelah itu klik *contentPane* lalu ubah *layoutnya* jadi *absolute layout*.



Gambar 2.9 ubah layout contentPane

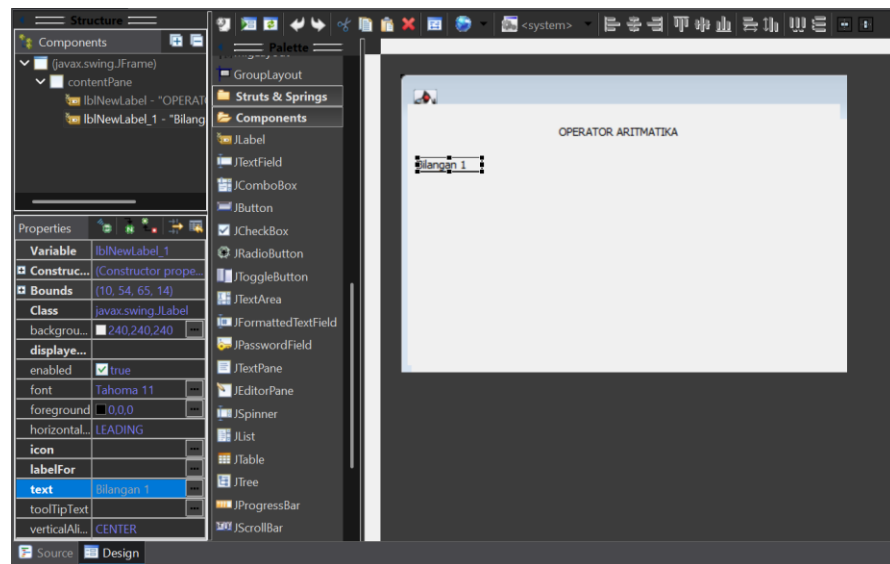
5. Tambahkan *JLabel* untuk judul program. Ubah *horizontalAlignment* ke *CENTER*, dan di *text* ketikkan *OPERATOR ARITMATIKA*.



Gambar 2.10 tambahkan JLabel untuk judul program

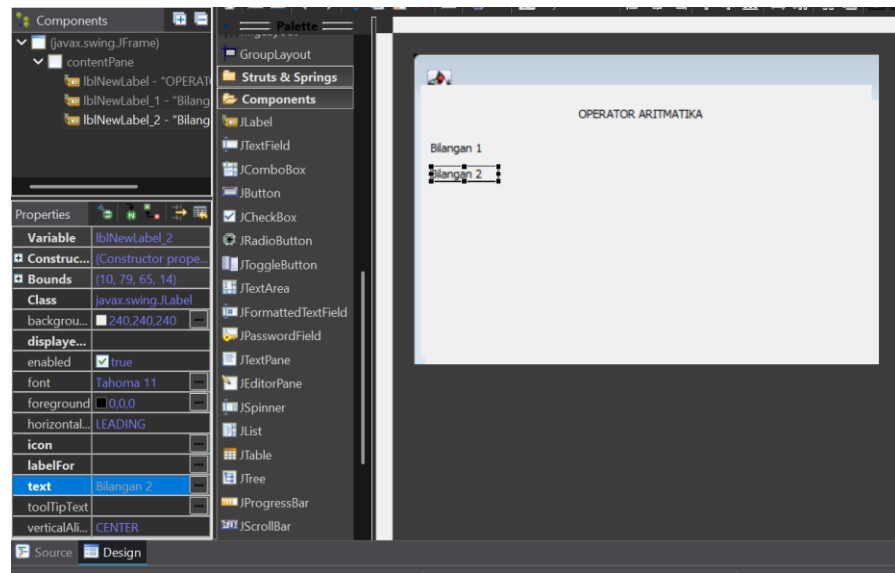
JLabel digunakan untuk menambahkan suatu label atau teks ke dalam design aplikasi.

6. Tambahkan *JLabel* lagi untuk Bilangan 1, ubah *horizontalAlignment* menjadi LEADING dan ketik Bilangan 1 pada *text*.



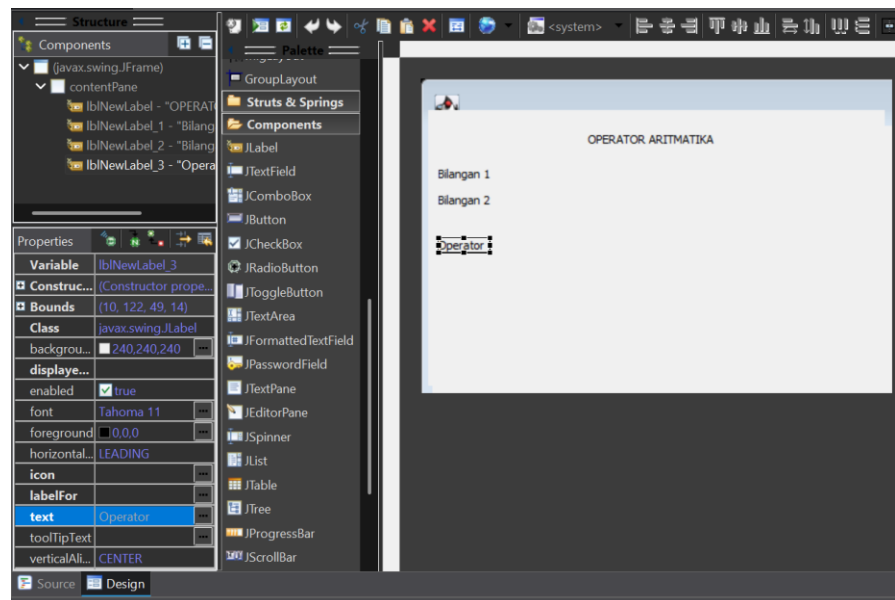
Gambar 2.11 tambahkan JLabel untuk bilangan 1

7. Tambahkan *JLabel* lagi untuk Bilangan 2, ubah *horizontalAlignment* menjadi LEADING dan ketik Bilangan 2 pada *text*.



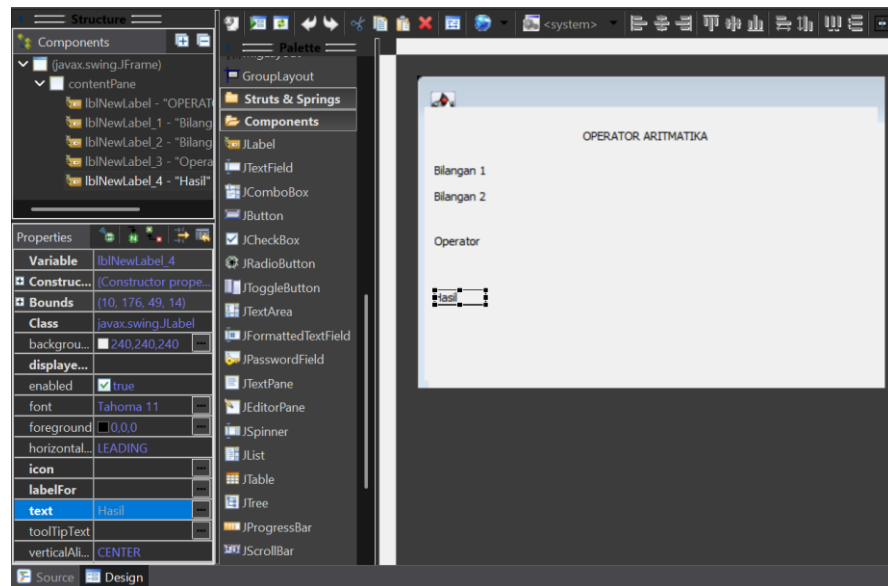
Gambar 2.12 tambahkan JLabel untuk bilangan 2

8. Tambahkan *JLabel* lagi untuk Operator, ubah *horizontalAlignment* menjadi LEADING dan ketik Operator pada *text*.



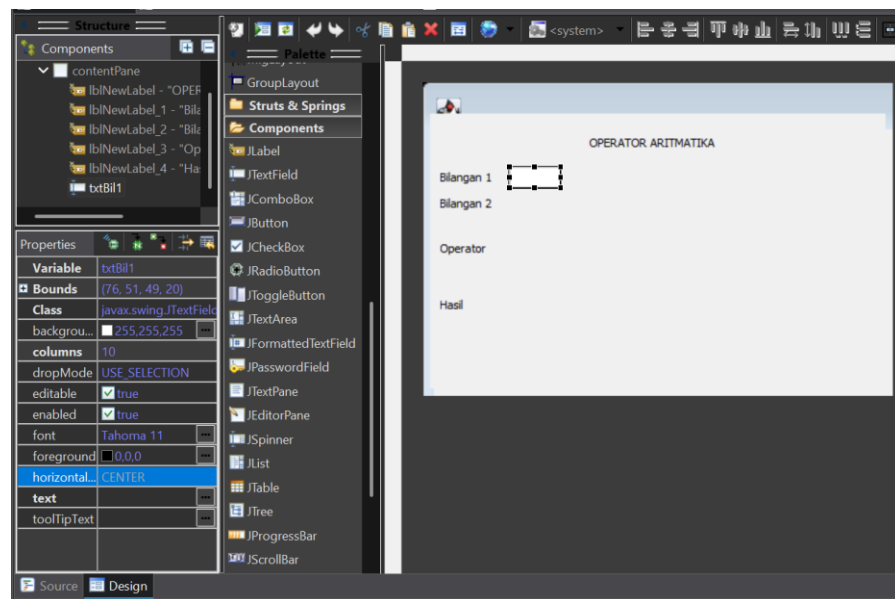
Gambar 2.13 tambahkan JLabel untuk Operator

9. Tambahkan *JLabel* lagi untuk Hasil, ubah *horizontalAlignment* menjadi LEADING dan ketik Hasil pada *text*.



Gambar 2.14 tambahkan JLabel untuk hasil

10. Tambahkan *JTextField* agar untuk bilangan 1, ubah *variable* menjadi *txtBil1* dan ubah *horizontalAlignment* jadi *CENTER*.

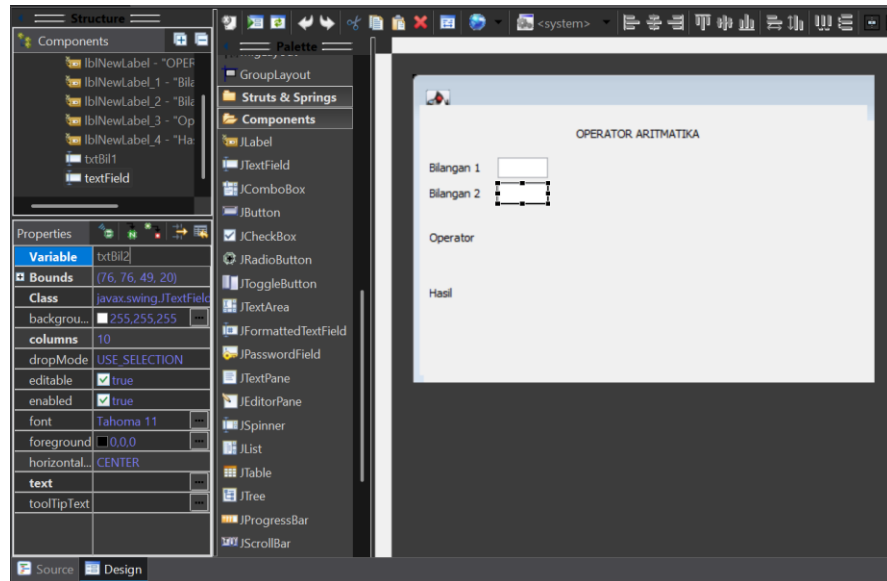


Gambar 2.15 tambahkan JTextField untuk bilangan 1

JTextField digunakan agar pengguna bisa menginputkan sesuatu, seperti sekarang digunakan untuk menginputkan bilangan.

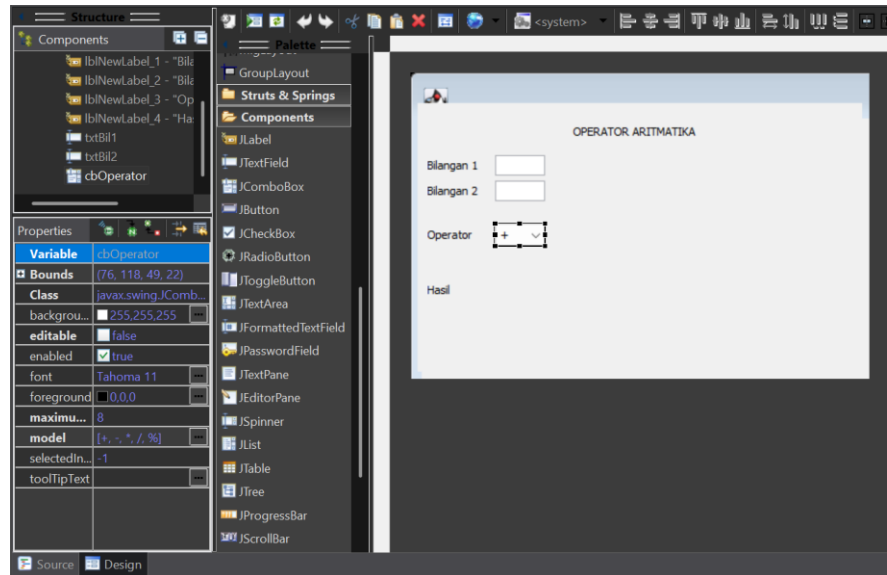
Mengubah *variable* tujuannya adalah untuk memudahkan pemanggilan dan pengaturan program pada kode program.

11. Tambahkan *TextField* agar untuk bilangan 1, ubah *variable* menjadi *txtBil1* dan ubah *horizontalAlignment* jadi CENTER.



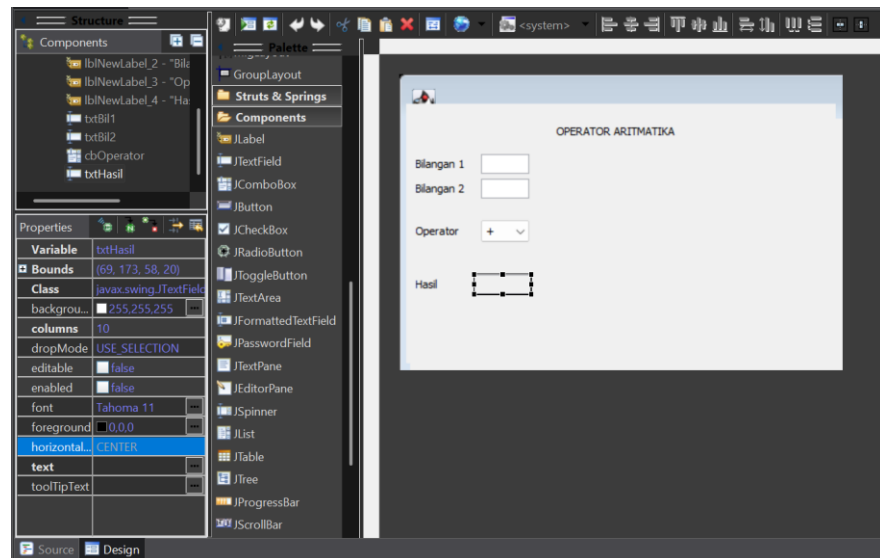
Gambar 2.16 *TextField* untuk bilangan 2

12. Tambahkan *JComboBox* untuk menambahkan operator operator aritmatika pada *model* dan ubah *variable* jadi *cbOperator*.



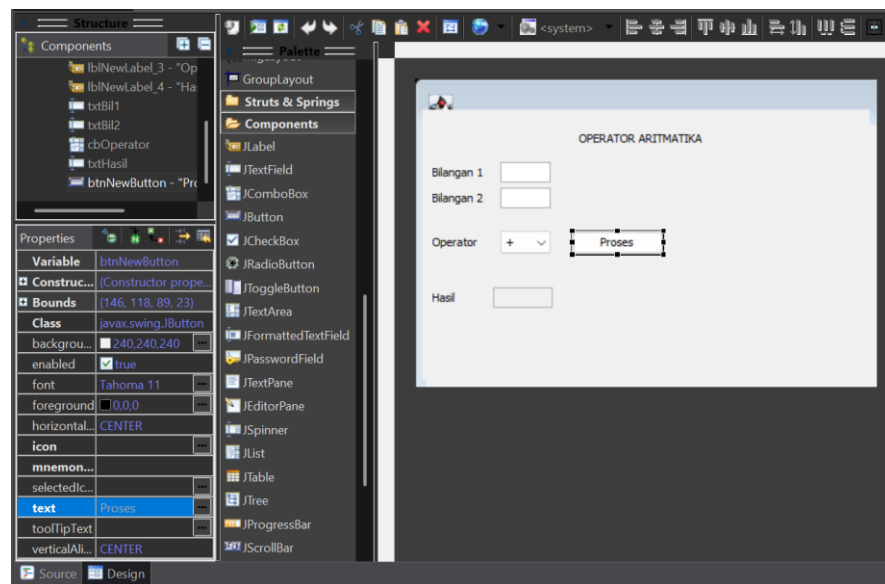
Gambar 2.17 *JComboBox* untuk operator

13. Tambahkan *TextField* agar untuk hasil, ubah *variable* menjadi *txtHasil*, ubah *horizontalAlignment* jadi CENTER, dan hapus centang pada *editable* dan *enable* sehingga nanti menjadi false.



Gambar 2.18 JTextField untuk hasil

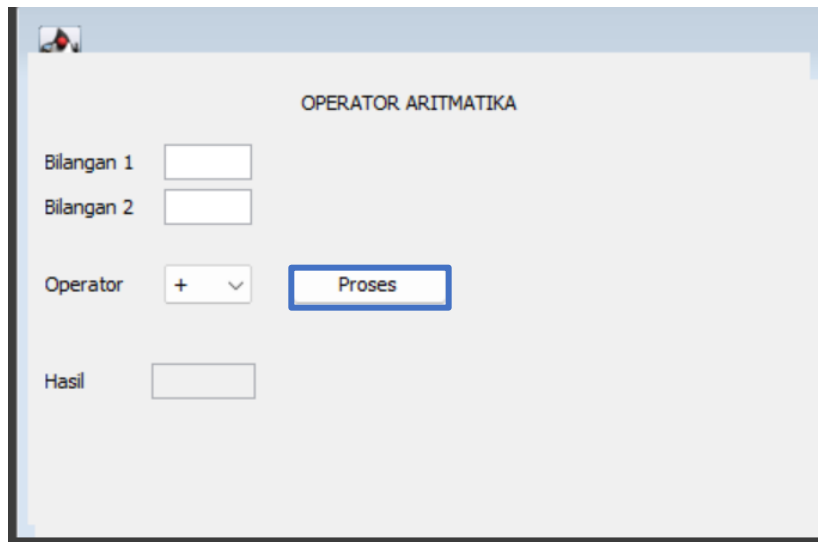
14. Tambahkan *JButton* di sebelah operator untuk proses dan ketik pada *text* Proses.



Gambar 2.19 JButton untuk proses

JButton digunakan untuk menambahkan tombol yang nantinya bisa di klik oleh pengguna.

15. Klik 2 kali pada proses agar bisa menampilkan kode program lalu tambahkan kode program untuk menjalankan aplikasi operator aritmatika yang di buat.



Gambar 2.20 klik 2 kali pada proses

```

99      JButton btnNewButton = new JButton("Proses");
100     btnNewButton.addActionListener(new ActionListener() {
101         int hasil;
102         public void actionPerformed(ActionEvent e) {
103             int a = Integer.parseInt(txtBil1.getText());
104             int b = Integer.parseInt(txtBil2.getText());
105             int c = cbOperator.getSelectedIndex(); //operator
106             if (c==0) {
107                 hasil = a+b;
108             }
109             if (c==1) {
110                 hasil = a-b;
111             }
112             if (c==2) {
113                 hasil = a*b;
114             }
115             if (c==3) {
116                 hasil = a/b;
117             }
118             if (c==4) {
119                 hasil = a%b;
120             }

```

Kode program 2.1 kode program agar operator aritmatika bisa berjalan.

16. Tapi program diatas belum lengkap karena belum ada peringatan untuk itu kita perlu menambahkan kode berikut ke dalam program.

```

27     private void pesanPeringatan(String pesan) {
28         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
29     }
30     private void pesanError(String pesan) {
31         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
32     }

```

Kode program 2.2 kode untuk program peringatan

17. Agar tampil peringatan saat kita melakukan kesalahan input maka kita akan ubah lagi kode program 2.1 diatas menjadi seperti ini.

```

108     btnNewButton.addActionListener(new ActionListener() {
109         int hasil;
110         public void actionPerformed(ActionEvent e) {
111             if(txtBil1.getText().trim().isEmpty()) {
112                 pesanPeringatan("Bilangan 1 harus diisi");
113             } else if (txtBil2.getText().trim().isEmpty()) {
114                 pesanPeringatan("Bilangan 2 harus diisi");
115             } else {
116                 try {
117                     int a = Integer.parseInt(txtBil1.getText());
118                     int b = Integer.parseInt(txtBil2.getText());
119                     int c = cbOperator.getSelectedIndex(); //operator
120                     if (c==0) {
121                         hasil = a+b;
122                     }
123                     if (c==1) {
124                         hasil = a-b;
125                     }
126                     if (c==2) {
127                         hasil = a*b;
128                     }
129                     if (c==3) {
130                         hasil = a/b;
131                     }
132                     if (c==4) {
133                         hasil = a%b;
134                     }
135                 } catch (NumberFormatException ex) {
136                     pesanError("Bilangan 1 dan Bilangan 2 harus angka");
137                 }
138             }
139             txtHasil.setText(String.valueOf(hasil));
140         }
141     }

```

Kode program 2.3 kode program operator aritmatika

2.2.4 Kode Program Keseluruhan

```

1 package pekan8_2511533008;
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
11 import java.awt.Font;
12 import javax.swing.SwingConstants;
13 import javax.swing.JTextField;
14 import javax.swing.JComboBox;
15 import javax.swing.DefaultComboBoxModel;
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_2511533008 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     private void pesanPeringatan(String pesan) {
30         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
31     }
32
33     private void pesanError(String pesan) {
34         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
35     }
36
37     /**
38      * Launch the application.
39      */
40     public static void main(String[] args) {
41         EventQueue.invokeLater(new Runnable() {
42             public void run() {
43                 try {
44                     OperatorAritmatikaGUI_2511533008 frame = new OperatorAritmatikaGUI_2511533008();
45                     frame.setVisible(true);
46                 } catch (Exception e) {
47                     e.printStackTrace();
48                 }
49             }
50         });
51     }
52
53     /**
54      * Create the frame.
55      */
56     public OperatorAritmatikaGUI_2511533008() {
57         setTitle("OPERATOR ARITMATIKA");
58         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
59         setBounds(100, 100, 450, 300);
60         contentPane = new JPanel();
61         contentPane.setBackground(new Color(255, 255, 255));
62         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
63         setContentPane(contentPane);
64         contentPane.setLayout(null);
65

```

```

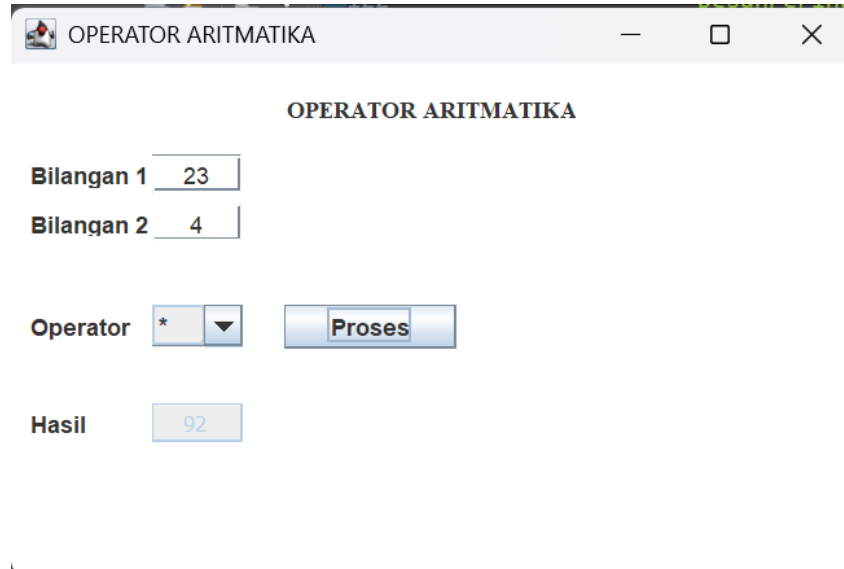
65
66 JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
67 lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
68 lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 12));
69 lblNewLabel.setBounds(142, 11, 149, 27);
70 contentPane.add(lblNewLabel);
71
72 JLabel lblNewLabel_1 = new JLabel("Bilangan 1");
73 lblNewLabel_1.setBounds(10, 50, 63, 14);
74 contentPane.add(lblNewLabel_1);
75
76 JLabel lblNewLabel_1_1 = new JLabel("Bilangan 2");
77 lblNewLabel_1_1.setBounds(10, 75, 63, 14);
78 contentPane.add(lblNewLabel_1_1);
79
80 JLabel lblNewLabel_1_2 = new JLabel("Operator");
81 lblNewLabel_1_2.setBounds(10, 128, 63, 14);
82 contentPane.add(lblNewLabel_1_2);
83
84 JLabel lblNewLabel_1_3 = new JLabel("Hasil");
85 lblNewLabel_1_3.setBounds(10, 178, 63, 14);
86 contentPane.add(lblNewLabel_1_3);
87
88 txtBil1 = new JTextField();
89 txtBil1.setHorizontalAlignment(SwingConstants.CENTER);
90 txtBil1.setBounds(73, 47, 47, 20);
91 contentPane.add(txtBil1);
92 txtBil1.setColumns(10);
93
94 txtBil2 = new JTextField();
95 txtBil2.setHorizontalAlignment(SwingConstants.CENTER);
96 txtBil2.setBounds(73, 72, 47, 20);
97 contentPane.add(txtBil2);
98 txtBil2.setColumns(10);
99
100
101 JComboBox cbOperator = new JComboBox();
102 cbOperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "N"}));
103 cbOperator.setBounds(73, 124, 47, 22);
104 contentPane.add(cbOperator);
105
106 txtHasil = new JTextField();
107 txtHasil.setForeground(new Color(0, 0, 0));
108 txtHasil.setFont(new Font("Tahoma", Font.PLAIN, 11));
109 txtHasil.setHorizontalAlignment(SwingConstants.CENTER);
110 txtHasil.setEnabled(false);
111 txtHasil.setEditable(false);
112 txtHasil.setBounds(73, 175, 47, 20);
113 contentPane.add(txtHasil);
114 txtHasil.setColumns(10);
115
116 JButton btnNewButton = new JButton("Proses");
117 btnNewButton.addActionListener(new ActionListener() {
118     int hasil;
119     public void actionPerformed(ActionEvent e) {
120         if (txtBil1.getText().trim().isEmpty()) {
121             pesanPeringatan("Bilangan 1 harus diisi");
122         } else if (txtBil2.getText().trim().isEmpty()) {
123             pesanPeringatan("Bilangan 2 harus diisi");
124         } else {
125             try {
126                 int a = Integer.parseInt(txtBil1.getText());
127                 int b = Integer.parseInt(txtBil2.getText());
128                 int c = cbOperator.getSelectedIndex(); //operator
129                 if (c==0) {
130                     hasil = a+b;
131                 }
132                 if (c==1) {
133                     hasil = a-b;
134                 }
135                 if (c==2) {
136                     hasil = a*b;
137                 }
138                 if (c==3) {
139                     hasil = a/b;
140                 }
141                 if (c==4) {
142                     hasil = a%b;
143                 }
144             } catch (NumberFormatException ex) {
145                 pesanError("Bilangan 1 dan Bilangan 2 harus angka");
146             }
147             txtHasil.setText(String.valueOf(hasil));
148         }
149     }
150 });
151 btnNewButton.setBounds(141, 124, 49, 23);
152 contentPane.add(btnNewButton);
153
154 }
155 }
156

```

Kode program 2.4 kode program keseluruhan

2.3 Output Program

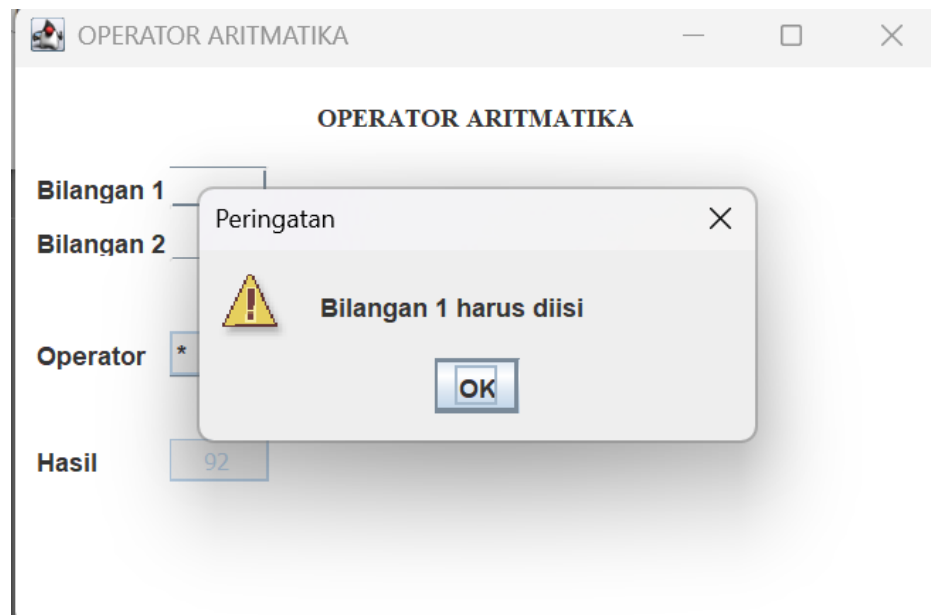
2.3.1 Output 1 (Jika tidak ada kesalahan)



The screenshot shows a window titled "OPERATOR ARITMATIKA". Inside, there are two input fields: "Bilangan 1" with the value "23" and "Bilangan 2" with the value "4". Below these is an "Operator" dropdown menu showing a multiplication symbol (*). To the right of the dropdown is a "Proses" button. At the bottom, there is a "Hasil" label and a text box displaying the result "92".

Gambar 2.21 output tanpa kesalahan

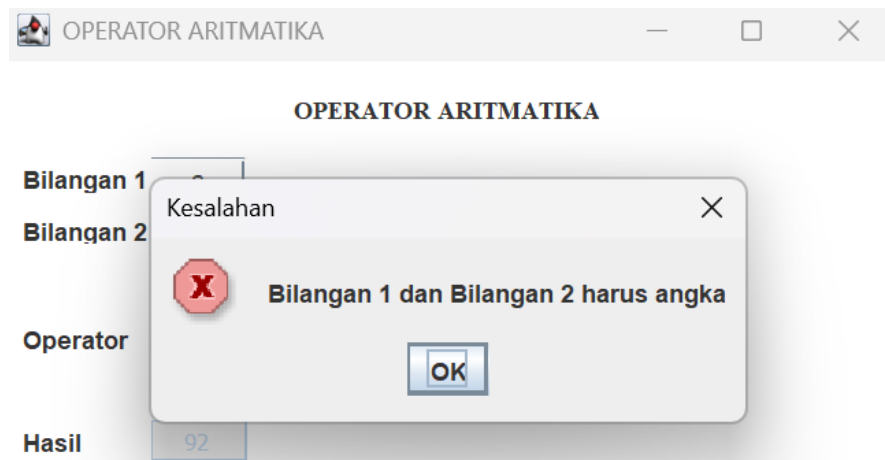
2.3.2 Output 2 (Jika mengosongkan salah satu bilangan)



The screenshot shows the same "OPERATOR ARITMATIKA" window as in Gambar 2.21, but with an error dialog box overlaid. The dialog box is titled "Peringatan" and contains a yellow warning triangle icon. The text inside the dialog says "Bilangan 1 harus diisi". There is an "OK" button at the bottom right of the dialog box. The background application window is slightly dimmed.

Gambar 2.22 output jika salah satu bilangan kosong

2.3.3 Output 3 (Jika yang di input bukan angka)



Gambar 2.23 output jika yang diinputkan bukan angka

BAB III

KESIMPULAN

3.1 Kesimpulan

Berdasarkan hasil praktikum Operator Aritmatika GUI, dapat disimpulkan bahwa operator aritmatika merupakan komponen dasar dalam pemrograman yang berfungsi untuk melakukan perhitungan matematis sederhana maupun kompleks.

Melalui penerapan GUI, pengguna operator aritmatika menjadi lebih interaktif dan mudah dipahami oleh pengguna karena disajikan dalam bentuk visual.

Praktikum ini berhasil menghubungkan teori pemrograman dengan implementasi praktis, sehingga mahasiswa memperoleh pengalaman langsung dalam merancang aplikasi sederhana berbasis GUI.

Mahasiswa mampu meningkatkan keterampilan dalam *problem solving* serta memahami keterkaitan antara logika perhitungan dengan desain antarmuka grafis.

Praktikum ini memberikan bekal awal untuk pengembangan aplikasi yang lebih kompleks, sekaligus menumbuhkan kreativitas dalam merancang program yang fungsional dan ramah pengguna.

DAFTAR PUSTAKA

- [1] P. J. Deitel and H. M. Deitel, *Java: How to Program (Early Objects)*, 11th ed. Pearson Education, 2017.
- [2] H. Schildt, *Java: The Complete Reference*, 10th ed. McGraw-Hill Education, 2018.
- [3] Oracle, *Java Platform, Standard Edition Documentation*. [Online]. Available: <https://docs.oracle.com/javase/>
- [4] Sutarman, *Pengantar Teknologi Informasi*. Jakarta: Bumi Aksara, 2019.
- [5] A. Nugroho, *Pemrograman Java untuk Pemula*. Yogyakarta: Andi Offset, 2020.