

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN

Pembuatan Aplikasi Operator Aritmatika Menggunakan Desain GUI Java Eclipse

Disusun Oleh:

Dinda Amelia

(2511531020)

Dosen Pengampu:

Dr. Wahyudi, S.T, M.T

Asisten Praktikum:

Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG 2025

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan kesempatan serta kemudahan kepada penulis untuk dapat menyelesaikan Laporan Praktikum pada mata kuliah Algoritma Pemrograman, sehingga Laporan Praktikum ini dapat dikumpulkan dengan tepat waktu. Atas rahmat dan karunianya Laporan Praktikum dapat terselesaikan dengan baik. Sholawat serta salam juga penulis sampaikan kepada baginda tercinta yaitu Nabi Muhammad SAW, yang kita nantikan syafa'atnya di akhirat nanti. Laporan Praktikum ini bertujuan untuk menambah wawasan para pembaca untuk lebih memperdalam ilmu yang ada pada makalah ini.

Laporan praktikum ini disusun sebagai bentuk penanggung jawaban atas pelaksanaan kegiatan praktikum mata kuliah algoritma pemograman yang membahas tentang Pembuatan Aplikasi Operator Aritmatika Menggunakan Desain GUI Java Eclipse. Dalam praktikum ini, penulis tidak hanya belajar tentang operasi aritmatika seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus, tetapi juga memahami bagaimana membangun sebuah aplikasi yang interaktif dan *user-friendly*. Proses pembuatan aplikasi ini meliputi pembuatan project, desain form, sampai implementasi *event* tombol menggunakan bahasa Java. Diharapkan laporan ini dapat memberikan pemahaman kepada pembaca mengenai langkah-langkah membangun aplikasi GUI sederhana berbasis operator aritmatika.

Dalam penyusunan Laporan Praktikum ini, penulis mengalami banyak kesulitan dan penulis menyadari bahwa Laporan Praktikum ini masih jauh dari kesempurnaan. Untuk itu, penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan pada Laporan Praktikum ini.

Padang, 19 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 Praktikum	1
BAB II: PEMBAHASAN	2
2.1 Dasar Teori.....	2
2.2 Langkah kerja.....	2
2.2.1 Buka eclipse.....	2
2.2.2 Membuat Package	3
2.2.3 Membuat JFrame Form	3
2.2.4 Mendesain GUI.....	4
2.2.5 Source Code.....	6
2.2.6 Output Program (GUI)	11
BAB III: KESIMPULAN.....	12
DAFTAR PUSTAKA.....	13

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam praktikum ini, penulis tidak hanya belajar tentang operasi aritmatika seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus, tetapi juga memahami bagaimana membangun sebuah aplikasi yang interaktif dan *user-friendly*. Proses pembuatan aplikasi ini meliputi pembuatan project, desain form, sampai implementasi event tombol menggunakan bahasa Java. Diharapkan laporan ini dapat memberikan pemahaman kepada pembaca mengenai langkah-langkah membangun aplikasi *GUI* sederhana berbasis operator aritmatika. Pembuatan aplikasi dengan antarmuka grafis (*GUI*) menjadi salah satu cara efektif untuk menghubungkan teori dan penerapannya. Dengan memanfaatkan *WindowBuilder* pada Eclipse, mahasiswa dapat membuat tampilan program yang interaktif tanpa harus menulis kode *layout* secara manual. Hal ini memberikan pengalaman baru dalam membangun aplikasi sederhana yang dapat melakukan operasi aritmatika melalui input pengguna dan tombol perhitungan.

1.2 Tujuan

- 1.2.1 Mengetahui dan memahami penggunaan operator aritmatika dalam pemrograman Java.
- 1.2.2 Mampu membuat aplikasi *GUI* sederhana menggunakan *WindowBuilder* pada Eclipse.
- 1.2.3 Melatih keterampilan dalam mengelola input dari pengguna dan memprosesnya menggunakan logika aritmatika.

1.3 Manfaat Praktikum

- 1.3.1 Mahasiswa mampu memahami cara kerja operator aritmatika dalam program Java.
- 1.3.2 Mahasiswa dapat membuat antarmuka grafis (*GUI*) yang interaktif dan menarik.
- 1.3.3 Meningkatkan kemampuan dalam menggunakan *WindowBuilder* untuk desain form secara visual.

BAB II

PEMBAHASAN

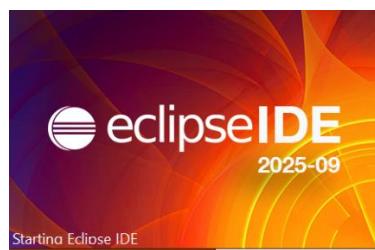
2.1 Dasar Teori

Pada praktikum ini, dibuat sebuah aplikasi sederhana yang menggunakan operator aritmatika sebagai fungsi utama. Aplikasi ini dibangun menggunakan Java dan didesain melalui *WindowBuilder* agar tampilan menjadi lebih mudah dibuat dan dipahami. Langkah pertama dimulai dari pembuatan project dan package baru di Eclipse, kemudian dilanjutkan dengan membuat sebuah *JFrame Form* sebagai media tampilan *GUI*. Di dalam form tersebut, ditambahkan beberapa komponen seperti *JTextField* untuk input angka, *JButton* untuk melakukan operasi aritmatika, dan *JLabel* untuk menampilkan hasil perhitungan. Setiap tombol diberi fungsi melalui event *actionPerformed*, di mana Java akan membaca nilai dari *text field*, mengonversinya menjadi tipe data numerik, kemudian memproses operasi yang dipilih oleh pengguna. Misalnya, tombol “+” melakukan operasi penjumlahan, tombol “-” untuk pengurangan, tombol “*” untuk perkalian, tombol “/” untuk pembagian, dan tombol “%” untuk modulus atau sisa bagi.

Selanjutnya, hasil perhitungan ditampilkan pada label hasil, sehingga pengguna dapat melihat *output* secara langsung. Penggunaan *GUI* memudahkan pengguna dalam menjalankan aplikasi tanpa harus mengetikkan perintah di *console*. Dari proses ini, mahasiswa dapat memahami bagaimana membuat tampilan aplikasi, bagaimana event tombol bekerja, dan bagaimana operator aritmatika diterapkan dalam sebuah aplikasi. Program berjalan sesuai harapan dan memberikan hasil perhitungan yang tepat berdasarkan input pengguna.

2.2 Langkah kerja

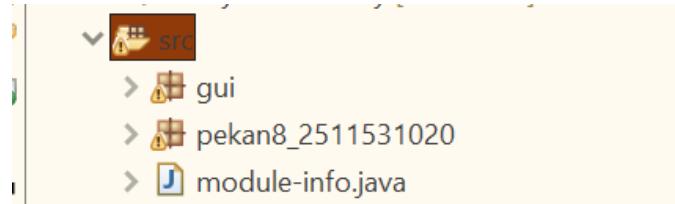
2.2.1 Buka eclipse



Gambar 2.1 Buka eclipse jalankan

2.2.2 Membuat Package

- Klik kanan folder src
- Pilih New → Package
- Nama: pekan8_2511531020



Gambar 2.2 Membuat Package

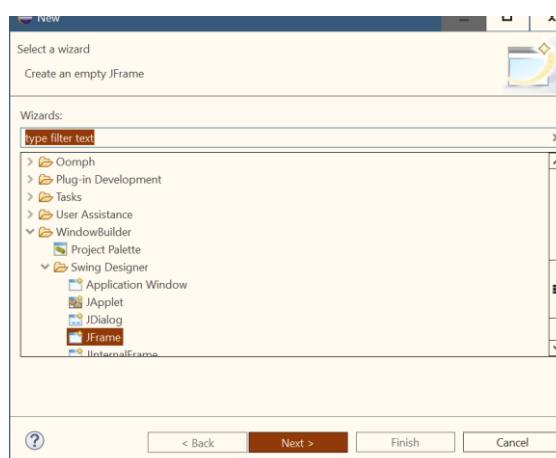
2.2.3 Membuat JFrame Form

- Klik kanan package
- Pilih New → Other



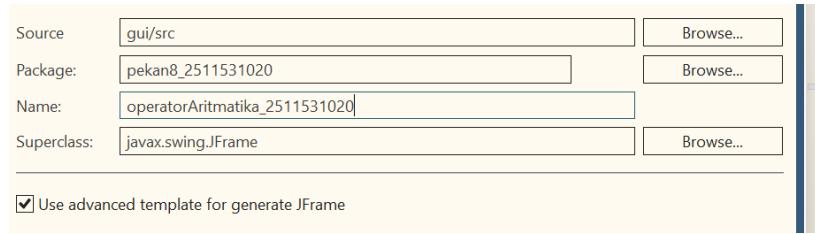
Gambar 2.3 Membuat JFrame Form

- Cari: WindowBuilder → Swing Designer → JFrame



Gambar 2.4 Mencari JFrame

- Nama file: operatorAritmatika_2511531020

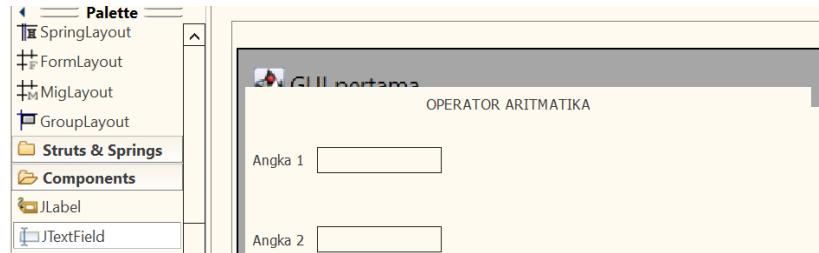


Gambar 2.5 Membuat Nama file

- Klik Finish
- File form akan terbuka dengan dua mode: Source → melihat kode
Design → mendesain GUI secara visual
- Pilih desain

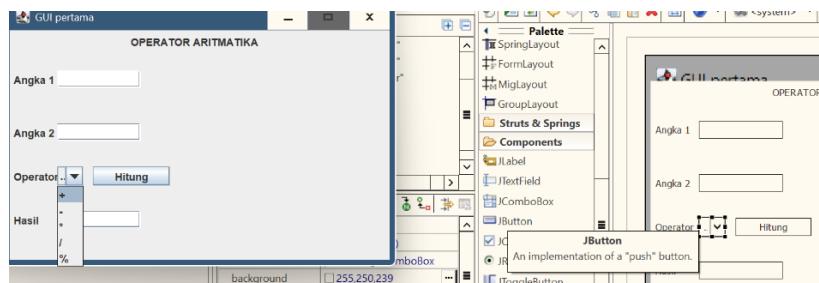
2.2.4 Mendesain GUI

- JTextField → input angka 1 dan angka 2



Gambar 2.6 Menginput Angka 1 dan Angka 2

- 5 JButton → Tambah, Kurang, Kali, Bagi, Modulus



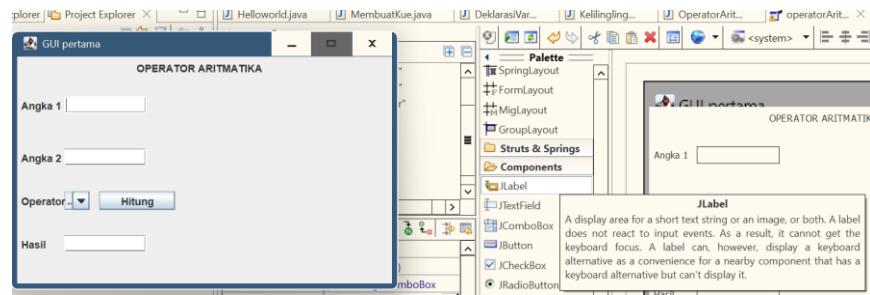
Gambar 2.7 Menambah (+,-,*,/,%)) di JButton

- JLabel → Ubah text menjadi:

“Angka 1:”

“Angka 2:”

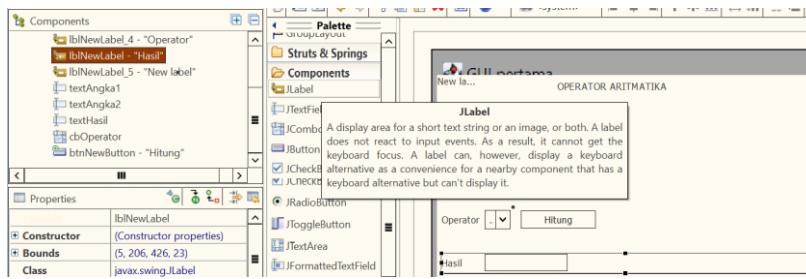
“Hasil:”



Gambar 2.8 Menggunakan JLabel untuk text Angka 1, Angka 2, dan Hasil

- Menambahkan Label Hasil Dengan JLabel

Ubah text menjadi→Hasil:-



Gambar 2.9 Menambahkan Label Hasil Dengan JLabel

- Jalankan program dengan cara → Klik kanan file FormAritmatika.java → Pilih Run As → Java Application : GUI aplikasi akan muncul

2.2.5 Source Code

```
1 package pekan8_2511531020;
2
3④ import java.awt.EventQueue;□
18
19 public class operatorAritmatika_2511531020 extends JFrame {
20
21     private static final long serialVersionUID = 1L;
22     private JPanel contentPane;
23     private JTextField textAngka1;
24     private JTextField textAngka2;
25     private JTextField textHasil;
26
27④     private void pesanPeringatan(String pesan) {
28         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
29     }
30④     private void pesanError(String pesan) {
31         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
32     }
33
34④     /**
35      * Launch the application.
36      */
37④     public static void main(String[] args) {
38         EventQueue.invokeLater(new Runnable() {
39④             public void run() {
40④                 try {
41                     operatorAritmatika_2511531020 frame = new operatorAritmatika_2511531020();
42                     frame.setVisible(true);
43④                 } catch (Exception e) {
44                     e.printStackTrace();
45                 }
46             }
47         });
48     }
49
50④     /**
51      * Create the frame.
52      */
53④     public operatorAritmatika_2511531020() {
54         setResizable(false);
55         setTitle("GUI pertama");
56         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
57         setBounds(100, 100, 450, 300);
58         contentPane = new JPanel();
59         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
60         setContentPane(contentPane);
61         contentPane.setLayout(null);
62
63         JLabel lblNewLabel_2 = new JLabel("OPERATOR ARITMATIKA");
64         lblNewLabel_2.setBounds(139, 6, 292, 14);
65         contentPane.add(lblNewLabel_2);
66
67         JLabel lblNewLabel_1 = new JLabel("Angka 1");
68         lblNewLabel_1.setBounds(5, 31, 144, 50);
69         contentPane.add(lblNewLabel_1);
70
71         JLabel lblNewLabel_3 = new JLabel("Angka 2");
72         lblNewLabel_3.setBounds(5, 106, 426, 23);
73         contentPane.add(lblNewLabel_3);
74
75         JLabel lblNewLabel_4 = new JLabel("Operator");
76         lblNewLabel_4.setBounds(5, 156, 54, 23);
```

Gambar 2.10 Source Code

```
46
47     });
48 }
49
50④ /**
51      * Create the frame.
52      */
53④ public operatorAritmatika_2511531020() {
54     setResizable(false);
55     setTitle("GUI pertama");
56     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
57     setBounds(100, 100, 450, 300);
58     contentPane = new JPanel();
59     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
60     setContentPane(contentPane);
61     contentPane.setLayout(null);
62
63     JLabel lblNewLabel_2 = new JLabel("OPERATOR ARITMATIKA");
64     lblNewLabel_2.setBounds(139, 6, 292, 14);
65     contentPane.add(lblNewLabel_2);
66
67     JLabel lblNewLabel_1 = new JLabel("Angka 1");
68     lblNewLabel_1.setBounds(5, 31, 144, 50);
69     contentPane.add(lblNewLabel_1);
70
71     JLabel lblNewLabel_3 = new JLabel("Angka 2");
72     lblNewLabel_3.setBounds(5, 106, 426, 23);
73     contentPane.add(lblNewLabel_3);
74
75     JLabel lblNewLabel_4 = new JLabel("Operator");
76     lblNewLabel_4.setBounds(5, 156, 54, 23);
```

Gambar 2.11 Source Code

```

77     contentPane.add(lblNewLabel_4);
78
79     JLabel lblNewLabel = new JLabel("Hasil");
80     lblNewLabel.setBounds(5, 206, 426, 23);
81     contentPane.add(lblNewLabel);
82
83     JLabel lblNewLabel_5 = new JLabel("New label");
84     lblNewLabel_5.setBounds(0, 0, 48, 14);
85     contentPane.add(lblNewLabel_5);
86
87     JTextField textAngka1 = new JTextField();
88     textAngka1.setBounds(55, 46, 96, 20);
89     contentPane.add(textAngka1);
90     textAngka1.setColumns(10);
91
92     JTextField textAngka2 = new JTextField();
93     textAngka2.setBounds(55, 107, 96, 20);
94     contentPane.add(textAngka2);
95     textAngka2.setColumns(10);
96
97     JTextField textHasil = new JTextField();
98     textHasil.setBounds(55, 207, 96, 20);
99     textHasil.setToolTipText("txtHasil");
100    contentPane.add(textHasil);
101    textHasil.setColumns(10);
102
103    JComboBox cbOperator = new JComboBox();
104    cbOperator.setBounds(55, 156, 30, 22);
105    cbOperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}));
106    contentPane.add(cbOperator);

```

Gambar 2.12 Source Code

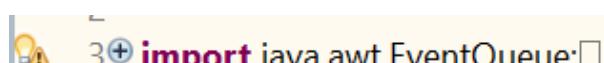
```

107
108    JButton btnNewButton = new JButton("Hitung");
109    btnNewButton.addActionListener(new ActionListener() {
110        int hasil;
111        public void actionPerformed(ActionEvent e) {
112            if(textAngka1.getText().trim().isEmpty()) {
113                pesanPeringatan("Inputkan angka 1");
114            } else if (textAngka2.getText().trim().isEmpty()) {
115                pesanPeringatan("Inputkan angka 2");
116            } else {
117                try {
118                    int a= Integer.valueOf(textAngka1.getText());
119                    int b= Integer.parseInt(textAngka2.getText());
120                    int c= cbOperator.getSelectedIndex();
121                    if(c==0) {hasil= a+b; }
122                    if(c==1) {hasil= a-b; }
123                    if(c==2) {hasil= a*b; }
124                    if(c==3) {hasil= a/b; }
125                    if(c==4) {hasil= a%b; }
126                    textHasil.setText(String.valueOf(hasil));
127                }catch (NumberFormatException ex) {
128                    pesanError("Angka 1 dan Angka 2 tidak valid");
129                }
130            }
131        }
132    });
133    btnNewButton.setBounds(96, 156, 88, 22);
134    contentPane.add(btnNewButton);
135
136}
137

```

Gambar 2.13 Source Code

- Bagian Import Library



Gambar 2.14 Bagian Import Library

import java.awt.EventQueue; → digunakan untuk menjalankan GUI secara thread-safe.

- Deklarasi Class Utama

```
public class operatorAritmatika_2511531020 extends JFrame {
```

Gambar 2.15 Deklarasi Class Utama

Program ini membuat sebuah class yang sekaligus menjadi *JFrame*, yaitu jendela utama dari aplikasi.

- Deklarasi Variabel

```
21  private static final long serialVersionUID = 1L;
22  private JPanel contentPane;
23  private JTextField textAngka1;
24  private JTextField textAngka2;
25  private JTextField textHasil;
```

Gambar 2.16 Deklarasi Variabel

Digunakan untuk menampung: Panel utama (contentPane), Inputan angka 1& angka 2, Output hasil

- Method Pesan Peringatan & Kesalahan

```
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  }
```

Gambar 2.17 Method Pesan Peringatan & Kesalahan

Berfungsi untuk: Menampilkan peringatan ketika input kosong dan menampilkan error ketika input bukan angka valid.

- Method main()

```
37  public static void main(String[] args) {
38      EventQueue.invokeLater(new Runnable() {
39          public void run() {
40              try {
41                  operatorAritmatika_2511531020 frame = new operatorAritmatika_2511531020(
42                      frame.setVisible(true);
43              } catch (Exception e) {
44                  e.printStackTrace();
45              }
46          }
47      });
48  }
```

Gambar 2.18 Method main()

Penjelasan: Program dijalankan dengan memastikan GUI berada di thread khusus GUI dan Program dijalankan dengan memastikan GUI berada di thread khusus GUI.

- Constructor: Membuat Frame GUI

Bagian ini menjalankan seluruh desain aplikasi:

```
54     setResizable(false);
55     setTitle("GUI pertama");
56     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
57     setBounds(100, 100, 450, 300);
```

Gambar 2.19 Membuat Frame GUI

Menentukan: Ukuran jendela, posisi jendela, judul jendela, dan tidak bisa di-resize

- Membuat Content Pane

```
58     contentPane = new JPanel();
59     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
60     setContentPane(contentPane);
61     contentPane.setLayout(null);
```

Gambar 2.20 Membuat Content Pane

Panel utama memakai layout null, sehingga posisi komponen ditentukan manual dengan setBounds().

- Membuat Label

```
63     JLabel lblNewLabel_2 = new JLabel("OPERATOR ARITMATIKA");
64     lblNewLabel_2.setBounds(139, 6, 292, 14);
65     contentPane.add(lblNewLabel_2);
66
67     JLabel lblNewLabel_1 = new JLabel("Angka 1");
68     lblNewLabel_1.setBounds(5, 31, 144, 50);
69     contentPane.add(lblNewLabel_1);
70
71     JLabel lblNewLabel_3 = new JLabel("Angka 2");
72     lblNewLabel_3.setBounds(5, 106, 426, 23);
73     contentPane.add(lblNewLabel_3);
74
75     JLabel lblNewLabel_4 = new JLabel("Operator");
76     lblNewLabel_4.setBounds(5, 156, 54, 23);
77     contentPane.add(lblNewLabel_4);
78
79     JLabel lblNewLabel = new JLabel("Hasil");
80     lblNewLabel.setBounds(5, 206, 426, 23);
81     contentPane.add(lblNewLabel);
```

Gambar 2.21 Membuat Label

Semua label ditambahkan dengan setBounds() untuk mengatur posisi.

- Input Text Field

```
87     textAngka1 = new JTextField();
88     textAngka1.setBounds(55, 46, 96, 20);
89     contentPane.add(textAngka1);
90     textAngka1.setColumns(10);

91
92     textAngka2 = new JTextField();
93     textAngka2.setBounds(55, 107, 96, 20);
94     contentPane.add(textAngka2);
95     textAngka2.setColumns(10);

96
97     textHasil = new JTextField();
98     textHasil.setBounds(55, 207, 96, 20);
99     textHasil.setToolTipText("txtHasil");
100    contentPane.add(textHasil);
101    textHasil.setColumns(10);
```

Gambar 2.22 Input Text Field

Digunakan untuk: mengisi angka pertama, Mengisi angka kedua, dan menampilkan hasil

- ComboBox Operator

```
103     JComboBox cbOperator = new JComboBox();
104     cbOperator.setBounds(55, 156, 30, 22);
105     cbOperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}));
106     contentPane.add(cbOperator);
```

Gambar 2.23 ComboBox Operator

Isi Operator: { "+", "-", "*", "/", "%" } → User memilih operator dari combo box ini.

- Tombol Hitung dan Aksinya

Bagian paling penting:

```
109     btnNewButton.addActionListener(new ActionListener() {
110         Search
111         int hasil;
112         public void actionPerformed(ActionEvent e) {
113             if(textAngka1.getText().trim().isEmpty()) {
114                 pesanPeringatan("Inputkan angka 1");
115             } else if (textAngka2.getText().trim().isEmpty()) {
116                 pesanPeringatan("Inputkan angka 2");
117             } else {
118                 try {
119                     int a= Integer.valueOf(textAngka1.getText());
120                     int b= Integer.parseInt(textAngka2.getText());
121                     int c= cbOperator.getSelectedIndex();
```

Gambar 2.24 Tombol Hitung dan Aksinya

Validasi Input: Jika angka 1 kosong → tampilkan peringatan, Jika angka 2 kosong → tampilkan peringatan, dan Jika bukan angka → tampilkan error.

- Logika Perhitungan

```
121      if(c==0) {hasil= a+b; }
122      if(c==1) {hasil= a-b; }
123      if(c==2) {hasil= a*b; }
124      if(c==3) {hasil= a/b; }
125      if(c==4) {hasil= a%b; }
```

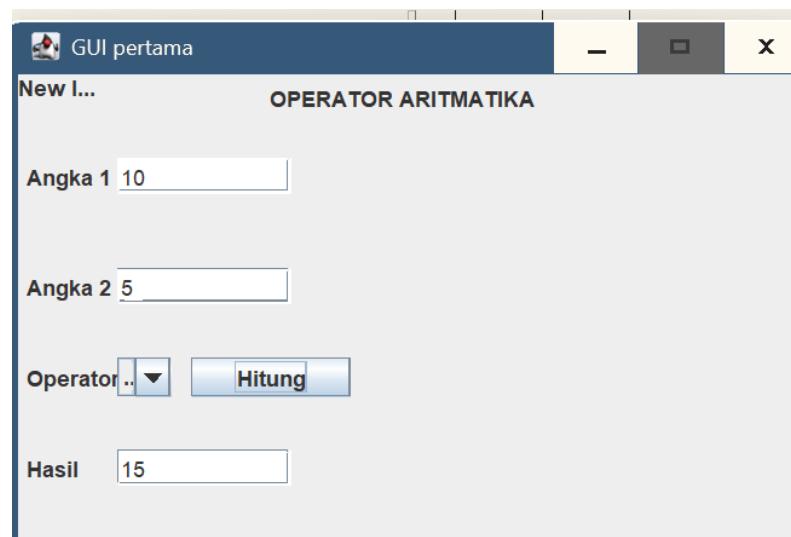
Gambar 2.25 Logika Perhitungan

- Menampilkan Hasil

```
textHasil.setText(String.valueOf(hasil));
```

Gambar 2.26 Logika Perhitungan

2.2.6 Output Program (GUI)



Gambar 2.27 Output Program (GUI)

Output yang dihasilkan program adalah nilai hasil perhitungan dari dua angka yang dimasukkan pengguna. Setelah user mengisi Angka 1, Angka 2, lalu memilih operator seperti +, -, *, /, atau %. Operator yang user pilih adalah “+” program akan menghitung sesuai pilihan. Nilai akhirnya akan muncul pada kotak Hasil.

BAB III

KESIMPULAN

Berdasarkan praktikum pembuatan aplikasi Operator Aritmatika memakai Java Eclipse dan WindowBuilder dapat di simpulkan bahwa pada pratikum kali ini mahasiswa bisa membuat aplikasi “Operator Aritmatika ” dengan sukses dan bisa mengaplikasikan penggunaan komponen-komponen dasar GUI seperti JLabel, JTextField, JComboBox, dan JButton, serta cara memproses event dengan ActionListener. Program yang dikembangkan mampu melaksanakan operasi aritmatika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus berdasarkan data yang dimasukkan pengguna. Output yang ditampilkan sesuai dengan proses perhitungan yang dipilih, sehingga menunjukkan bahwa logika program, validasi input, dan pemanggilan fungsi-fungsi Swing berfungsi dengan baik. Dari keseluruhan laporan dan percobaan, dapat disimpulkan bahwa pengembangan aplikasi GUI ini berkontribusi dalam memperdalam pemahaman terhadap konsep antarmuka pengguna, penanganan kesalahan, dan cara mengaitkan desain dengan logika perhitungan dalam bahasa pemrograman Java.

DAFTAR PUSTAKA

- [1] Nita, S. (2021). JOB SHEET PEMROGRAMAN BERORIENTASI OBJEK WITH “JAVA”(Teori & Implementasi Java).
- [2] Dewanta, F., & Nuha, H. H. (2021). *Pemrograman Java Untuk Aplikasi Berbasis Jaringan*. Coins Research.
- [3] Juharsa, J. (2012). Materi Iii.
- [4] Scarpino, M., Mihalkovic, L., Ng, S., & Holder, S. (2004). *SWT/JFace in Action: GUI Design with Eclipse 3.0*. Simon and Schuster.
- [5] AMURU, I., & Dzikrillah, A. R. (2024). Database-Based Gui System To Increase The Effectiveness Of Student Data Management In The Fkip Uhamka Dormitory. *Jurnal Teknik Informatika (Jutif)*, 5(4), 21-31.