

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
PEMROGRAMAN GUI 1

Disusun Oleh :

Nama : Nabila Khairunnisa
Nim : 2511531003
Dosen Pengampu : Dr. Wahyudi, S.T, M.T
Asisten Praktikum : Aufan Taufiqurrahman



FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji Syukur atas kehadirat Allah SWT yang telah melimpahkan Rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan praktikum algoritma dan pemrograman dengan judul “Pemrograman GUI 1” dengan baik dan tepat waktu. Dalam menyelesaikan laporan ini saya banyak mendapat arahan dan bimbingan, oleh karena itu saya ingin mengucapkan terimakasih kepada

1. Bapak Dr. Wahyudi, S.T, M.T selaku dosen pengampu
2. Uda Aufan Taufiqurrahman selaku asisten labor
3. Orang tua yang senantiasa mendoakan
4. Teman teman yang selalu memotivasi

Penulis menyadari bahwa laporan ini jauh dari kata sempurna. Oleh sebab itu, penulis sangat membuka diri apabila ada yang ingin memberikan kritikan dan saran yang sifatnya membantu, penulis akan sangat senang menerima. Tujuannya agar untuk kedepannya bisa menyempurnakan laporan.

Padang, 17 November 2025

Nabila Khairunnisa

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum.....	2
BAB II PEMBAHASAN.....	3
2.1 Deskripsi Praktikum.....	3
2.2 Langkah Langkah praktikum	3
2.2.1 Operator Aritmatika GUI.....	3
BAB III KESIMPULAN.....	10
3.1 Kesimpulan	10
3.2 Saran	10
DAFTAR PUSTAKA	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Graphical User Interface (GUI) atau antarmuka grafis merupakan sistem komponen visual interaktif yang digunakan dalam perangkat lunak komputer, smartphone, dan berbagai perangkat elektronik lainnya. GUI menampilkan objek yang menyampaikan informasi dan mewakili tindakan yang dapat diambil oleh pengguna. Objek objek tersebut berubah warna, ukuran atau visibilitas saat pengguna melakukan interaksi. GUI pertama kali dikembangkan pada tahun 1981 oleh Alan Kay, Douglas Engelbart, dan sekelompok peneliti adi xerox PARC. Kemudian pada 1983, Apple memperkenalkan komputer Lisa yang menggunakan GUI.

GUI di representasikan dengan elemen elemen seperti Window, icon, dan menu. Kemudian komputer dapat menjalankan perintah seperti membuka, menghapus dan memindahkan file. [1] Antarmuka Pengguna Grafis (GUI) memberikan sejumlah manfaat signifikan bagi pengguna, yang secara kolektif meningkatkan pengalaman pengguna (user experience) secara keseluruhan. Manfaat utamanya adalah kemudahan penggunaan; GUI memungkinkan siapa saja, bahkan tanpa keahlian teknis atau penguasaan bahasa pemrograman, untuk mengoperasikan komputer dan perangkat mobile karena elemen visual yang informatif. Ini secara signifikan mempermudah penggunaan komputer dan perangkat mobile karena pengguna hanya perlu mengarahkan mouse atau mengetuk layar, alih-alih mengetikkan perintah (command) secara manual seperti pada Antarmuka Pengguna Baris Perintah (Command-Line User Interface atau CUI). Kemudahan dan kecepatan interaksi ini pada gilirannya meningkatkan produktivitas, memungkinkan pekerjaan diselesaikan lebih cepat dan waktu dimanfaatkan secara lebih efektif. Terakhir, GUI juga meningkatkan aksesibilitas; tampilan grafis dan interaksi berbasis visual yang intuitif membuatnya lebih mudah digunakan oleh orang-orang yang mungkin memiliki keterbatasan fisik atau disabilitas tertentu. [2]

Antarmuka Pengguna Grafis (GUI) menawarkan keunggulan utama berupa sifatnya yang intuitif dan mudah digunakan, berkat penggunaan elemen visual yang mudah dikenali seperti ikon, jendela, dan menu, yang memungkinkan interaksi pengguna menjadi lebih alami. Selain itu, GUI sangat unggul dalam mendukung multitasking, karena pengguna dapat membuka dan mengelola beberapa jendela atau aplikasi secara simultan, yang pada akhirnya meningkatkan efisiensi kerja. Namun, GUI juga memiliki beberapa kelemahan. Yang paling utama adalah kebutuhan sumber daya yang lebih tinggi (memori dan daya prosesor) dibandingkan Antarmuka Baris Perintah (CLI), yang dapat membebani perangkat dengan spesifikasi rendah. Selain itu,

kompleksitas interaksi visualnya membuat GUI berpotensi memiliki kerentanan yang lebih tinggi terhadap malware. Terakhir, untuk tugas-tugas spesifik yang memerlukan serangkaian perintah cepat atau scripting, CLI seringkali dinilai lebih efisien dan fleksibel daripada GUI. [3]

Cara kerja GUI didukung oleh empat mekanisme utama. Pertama, Window Manager bertanggung jawab untuk mengatur dan mengelola tampilan jendela aplikasi di layar, memastikan setiap jendela dapat ditempatkan, diubah ukurannya, dipindahkan, dan ditutup oleh pengguna dengan mudah. Kedua, Widget Toolkit (atau GUI framework) menyediakan berbagai elemen visual interaktif siap pakai seperti tombol, menu, kotak centang, dan bilah gulir yang membentuk antarmuka, sekaligus mengelola respons sistem terhadap interaksi yang dilakukan pengguna pada elemen-elemen tersebut. Ketiga, Graphics System bertugas merender dan menampilkan semua komponen visual mulai dari ikon, jendela, hingga teks ke layar pengguna dengan mengubah deskripsi elemen menjadi piksel yang jelas dan responsif. Terakhir, seluruh interaksi pengguna dimungkinkan melalui Input Devices, seperti keyboard, mouse, stylus, atau layar sentuh, yang mengirimkan sinyal perintah yang kemudian diterjemahkan oleh sistem GUI untuk menjalankan aksi yang diinginkan pengguna. [4]

1.2 Tujuan

1. Mampu membuat frame aplikasi menggunakan Java Swing (JFrame)
2. Mampu menggunakan dan menempatkan komponen dasar GUI (JLabel, JTextField, JComboBox, dan JButton)
3. Mampu menerapkan mekanisme pemilihan operator aritmatika melalui JComboBox
4. Mampu melakukan validasi input dan pesan error untuk mencegah *crash* pada program

1.3 Manfaat praktikum

1. Mahasiswa dapat membuat frame aplikasi menggunakan Java Swing (JFrame)
2. Mahasiswa dapat menggunakan dan menempatkan komponen dasar GUI (JLabel, JTextField, JComboBox, dan JButton)
3. Mahasiswa dapat menerapkan mekanisme pemilihan operator aritmatika melalui JComboBox
4. Mahasiswa dapat melakukan validasi input dan pesan error untuk mencegah *crash* pada program

BAB II

PEMBAHASAN

2.1 Deskripsi Praktikum

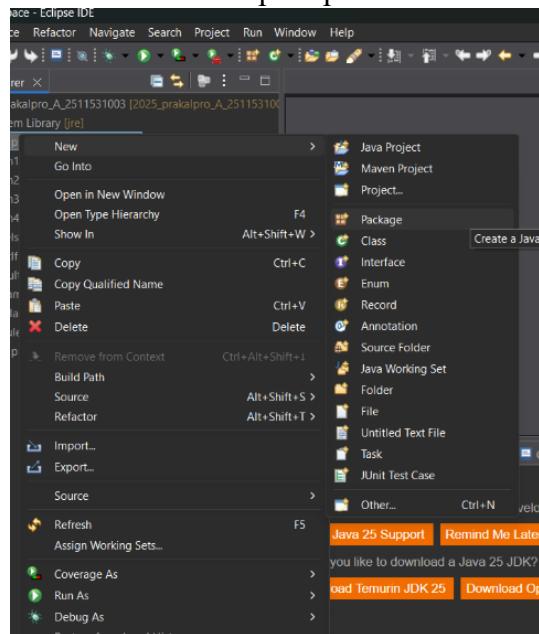
Praktikum Algoritma dan Pemrograman pada pekan ini bertujuan untuk membuat suatu aplikasi kalkulator sederhana berbasis Graphical User Interface (GUI) menggunakan kerangka kerja Java Swing dengan bantuan WindowBuilder. Setelah memulai proyek java dan membuat JFrame baru, langkah awal dalam mendesain tampilan dengan menambahkan komponen JLabel, JTextField, JComboBox, dan JButton. Setelah desain selesai, fokus praktikum beralih ke kode *source*. Pada source, di implementasikan *ActionListener* pada tombol “process” untuk menjalankan logika program. Logika ini mencakup beberapa hal yaitu validasi untuk memastikan kedua *text field* tidak kosong, menggunakan blok *try-catch* dengan *NumberFormatException* untuk memastikan input adalah angka sehingga mencegah *crash* program serta menentukan operasi aritmatika yang sesuai (if-else if) sebelum menampilkan hasil akhir ke *txtHasil*.

2.2 Langkah Langkah Praktikum

2.2.1 Operator Aritmatika GUI

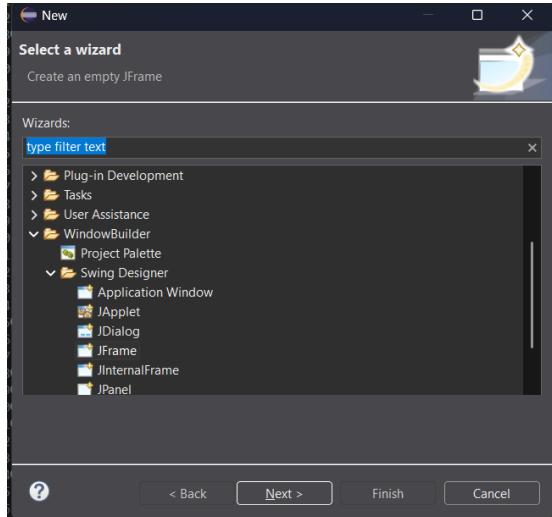
Langkah praktikum :

1. Buatlah package terlebih dahulu dengan mengklik kanan di folder A_prakalpro_2025_2511531003/src , pilih new dan klik package. Setelah itu beri nama pada package tanpa huruf kapital, karakter khusus serta tanpa “space”. lalu “finish”.



Gambar 2.2.1 membuat package

2. Klik kanan pada package pekan 8_2511531003 yang sudah dibuat sebelumnya. Arahkan cursor ke New, kemudian pilih other. Klik WindowBuilder, lalu pilih Swing Designer. Terakhir pilih JFrame dan klik tombol Next.



Gambar 2.2.2 membuat JFrame

3. Buat nama “OperatorAritmatikaGUI_2511531003” dengan ketentuan nama harus Uppercase pada awal kalimat dan tanpa “spasi” lalu “finish”.
4. Maka akan muncul tampilan seperti berikut dan klik tab *Design* pada bagian bawah editor

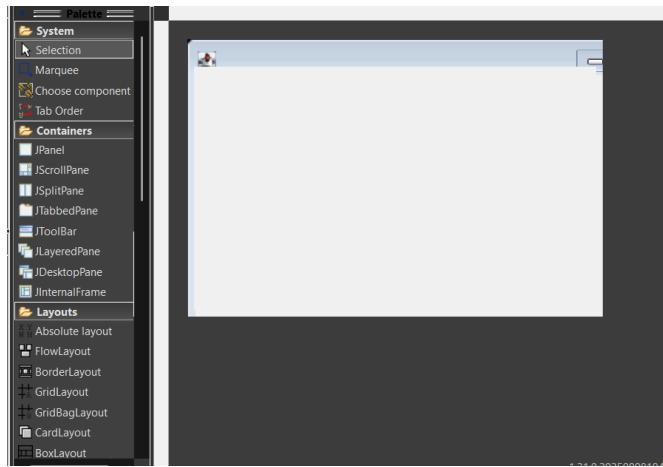
```

10
11  private static final long serialVersionUID = 1L;
12  private JPanel contentPane;
13
14 /**
15  * Launch the application.
16  */
17 public static void main(String[] args) {
18     EventQueue.invokeLater(new Runnable() {
19         public void run() {
20             try {
21                 OperatorAritmatika frame = new OperatorAritmatika();
22                 frame.setVisible(true);
23             } catch (Exception e) {
24                 e.printStackTrace();
25             }
26         }
27     });
28 }
29 /**
30  * Create the frame.
31  */
32 public OperatorAritmatika() {
33     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34     setBounds(100, 100, 450, 300);
35     contentPane = new JPanel();
36     contentPane.setBorder(new EmptyBorder(5, 5, 5));
37     setContentPane(contentPane);
38 }
39
40 }
41

```

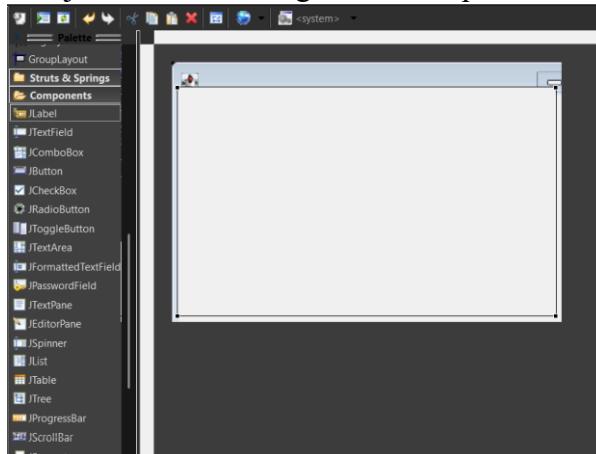
Gambar 2.2.4 tampilan program

5. Pada bagian Pallete, pilih absolute layout lalu klik pada JFrame



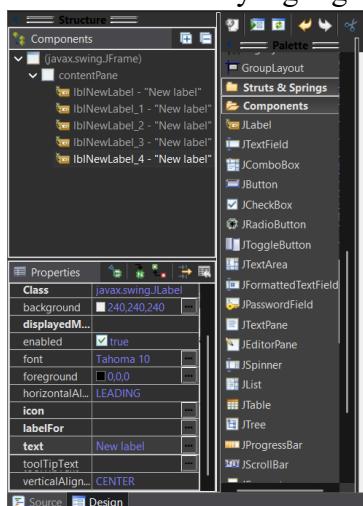
Gambar 2.2.5 tampilan pallete

6. Gulir ke bawah pada panel pallete, lalu pilih JLabel untuk membuat judul atau keterangan dan klik pada JFrame



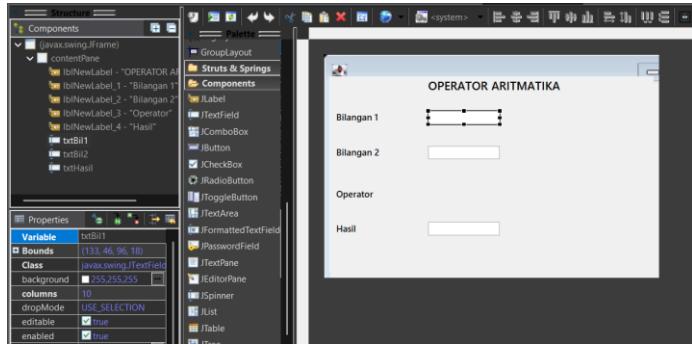
Gambar 2.2.6 pilih JLabel

7. Pada panel Components, klik JLabel yang teks nya ingin diganti. kemudian lihat pada panel properties yang bernama text. Klik dan ketikkan teks baru yang ingin dibuat.



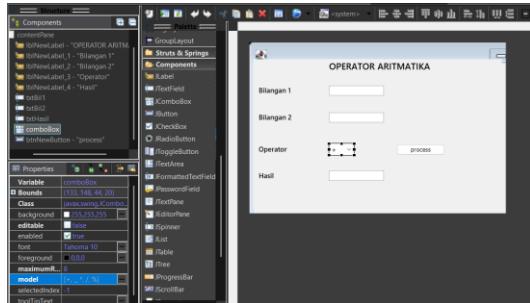
Gambar 2.2.7 mengubah teks JLabel

- Pada panel pallete, pilih JTextField untuk menerima input dari pengguna. Lalu klik pada JFrame. Untuk mengubah variabel, lihat pada panel properties yang bernama Variable. Klik dan ketikkan variabel yang ingin dibuat



Gambar 2.2.8 membuat JTextField

- Untuk bagian operator, menggunakan JComboBox yang berfungsi untuk membuat daftar pilihan yang dapat dilihat pengguna dengan mengklik panah tarik-turun. Daftar pilihan dapat dibuat dengan klik bagian model pada panel properties dan membuat pilihan yang ingin dibuat. Untuk memproses hasil operator, tambahkan JButton untuk tombol yang dapat di klik oleh pengguna untuk memulai suatu aksi.



Gambar 2.2.9 membuat JComboBox dan JButton

- Kembali ke tab *source* di bagian bawah editor dan akan muncul seperti berikut

```
1 package pekan8_2511531003;
2
3 import java.awt.EventQueue;
4
5 public class OPERATOR extends JFrame {
6
7     private static final long serialVersionUID = 1L;
8     JPanel contentPane;
9     JTextField txtBill;
10    JTextField txtBill2;
11    JTextField txtHasil;
12
13    /**
14     * Launch the application.
15     */
16
17    public static void main(String[] args) {
18        EventQueue.invokeLater(new Runnable() {
19            public void run() {
20                try {
21                    OPERATOR frame = new OPERATOR();
22                    frame.setVisible(true);
23                } catch (Exception e) {
24                    e.printStackTrace();
25                }
26            }
27        });
28    }
29
30    /**
31     * Create the frame.
32     */
33
34    public OPERATOR() {
35        initialize();
36    }
37
38    private void initialize() {
39        // ...
40
41        lblNewLabel_4.setBounds(10, 197, 56, 12);
42        contentPane.add(lblNewLabel_4);
43
44        txtBill = new JTextField();
45        txtBill.setBounds(133, 46, 96, 18);
46        contentPane.add(txtBill);
47        txtBill.setColumns(10);
48
49        txtBill2 = new JTextField();
50        txtBill2.setBounds(133, 93, 96, 18);
51        contentPane.add(txtBill2);
52        txtBill2.setColumns(10);
53
54        txtHasil = new JTextField();
55        txtHasil.setBounds(133, 195, 96, 18);
56        contentPane.add(txtHasil);
57        txtHasil.setColumns(10);
58
59        JComboBox cbOperator = new JComboBox();
60        cbOperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}));
61        cbOperator.setBounds(133, 148, 44, 20);
62        contentPane.add(cbOperator);
63
64        JButton btnNewButton = new JButton("process");
65        btnNewButton.setBounds(251, 148, 84, 20);
66        contentPane.add(btnNewButton);
67
68    }
69
70    /**
71     * This method is called from within the constructor to
72     * initialize the form.
73     * WARNING: Do NOT modify this code. The content of this method is
74     * always regenerated by the Form Editor.
75     */
76    // JAVADOC: overridden
77    @Override
78    protected void validate() {
79        // ...
80    }
81
82    /**
83     * Creates the contents of this frame.
84     */
85    // JAVADOC: protected
86    @Override
87    protected void createContents() {
88        // ...
89    }
90
91    /**
92     * This method is called from within the constructor to
93     * set the visibility of the window.
94     */
95    // JAVADOC: protected
96    @Override
97    protected void setVisible(boolean b) {
98        // ...
99    }
100 }
```

Gambar 2.2.10 kembali ke program

11. Input *private void pesanPeringatan (String pesan)* dan *private void pesanError (String pesan)* untuk memberikan peringatan dan memberi tahu pengguna tentang kesalahan.

```
 1 package pekan8_2511531003;
 2 
 3 import java.awt.EventQueue;■
 4 
 5 public class OPERATOR extends JFrame {
 6 
 7     private static final long serialVersionUID = 1L;
 8     private JPanel contentPane;
 9     private JTextField txtBill;
10     private JTextField txtBill2;
11     private JTextField txtHasil;
12 
13     /**
14      * Launch the application.
15      */
16 
17     public static void main(String[] args) {
18         EventQueue.invokeLater(new Runnable() {
19             public void run() {
20                 try {
21                     OPERATOR frame = new OPERATOR();
22                     frame.setVisible(true);
23                 } catch (Exception e) {
24                     e.printStackTrace();
25                 }
26             }
27         });
28     }
29 
30     /**
31      * Create the frame.
32      */
33 }
```

Gambar 2.2.11 input pesanPeringatan dan pesanError

12. Setelah `JButton btnNewButton = new JButton("process")` , daftarkan `addActionListener` ke `btnNewButton` untuk di eksekusi setiap kali tombol di klik

```

108     JButton btnNewButton = new JButton("process");
109     btnNewButton.addActionListener(new ActionListener() {
110         int hasil;
111
112         public void actionPerformed(ActionEvent e) {
113             if (txtBill1.getText().trim().isEmpty()) {
114                 pesanPeringatan ("Silahkan input Bilangan 1");
115             } else if (txtBill2.getText().trim().isEmpty()) {
116                 pesanPeringatan ("Silahkan input bilangan 2");
117             } else if (txtBill2.getText().trim().startsWith("0")) {
118                 pesanPeringatan ("Bilangan 2 tidak boleh angka 0");
119             } else {
120                 try {
121                     int a = Integer.parseInt(txtBill1.getText());
122                     int b = Integer.parseInt(txtBill2.getText());
123                     int c = cbOperator.getSelectedIndex();
124
125                     if (c==0) {hasil= a+b;}
126                     if (c==1) {hasil= a-b;}
127                     if (c==2) {hasil= a*b;};
128                     if (c==3) {hasil= a/b;};
129                     if (c==4) {hasil= a%b;};
130
131                     txtHasil.setText (String.valueOf(hasil));
132                 } catch (NumberFormatException ex) {
133                     pesanError ("Bilangan 1 dan Bilangan 2 harus Angka");
134                 }
135             }
}

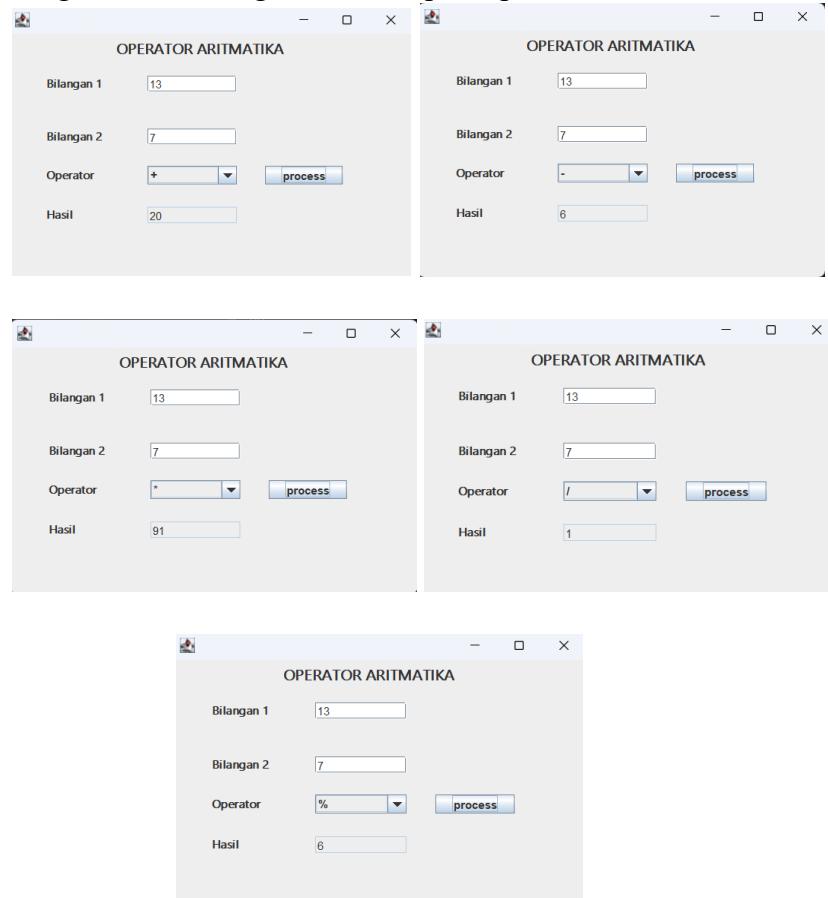
```

Gambar 2.2.12 kode program

13. Deklarasikan variabel bilangan (int) untuk menyimpan hasil perhitungan aritmatika
14. Input method `actionPerformed(ActionEvent e)` sebagai tempat semua logika yang harus dijalankan saat tombol proses diklik.
15. Pengecekan input apakah `text Field` untuk bilangan 1 kosong, `trim` untuk menghapus spasi di awak dan di akhir, dan `isEmpty` untuk mengecek kekosongan.
16. Jika `txtBill1` kosong, panggil pesan peringatan untuk memberitahu pengguna
17. Pengecekan input apakah `text field` untuk bilangan 2 kosong. Jika kosong, tampilkan pesan peingatan
18. Tambahkan `try` untuk memulai blok kode yang mungkin menimbulkan exception (kesalahan)
19. Konversi teks dari `txtBill1` menjadi bilangan bulat (int) dan menyimpannya di variabel a. Konversi teks dari `txtBill2` menjadi bilangan bulat (int) dan menyimpannya di variabel b
20. Menginput indeks yang dipilih dari JComboBox dan menyimpannya di variabel c
21. Jika indeks operator adalah 0, lakukan penjumlahan. Jika indeks operator adalah 1, lakukan operasi pengurangan. Jika indeks operator adalah 2, lakukan operasi perkalian. Jika indeks operator adalah 3, lakukan operasi pembagian. Jika indeks operator adalah 4, lakukan operasi modulo (sisa pembagian).
22. Menginput `txtHasil.setText (String.valueOf(hasil))` untuk mengkonversi nilai hasil (berupa int) kembali menjadi string dan

menampilkannya di *txtHasil*

23. Bagian *catch (NumberFormatException ex)* dijalankan jika terjadi exception di dalam blok *try*
24. Jika terjadi *NumberFormatException*, panggil method *pesanError* untuk memberitahu pengguna bahwa input harus berupa angka.
25. Program akan menghasilkan output seperti berikut



Gambar 2.2.25 ouput program

BAB III KESIMPULAN

3.1 Kesimpulan

Praktikum ini berhasil memberikan pemahaman dan keterampilan dasar dalam merancang, membangun dan mengimplementasikan aplikasi berbasis Graphical User Interface (GUI). Berdasarkan hasil praktikum, dapat disimpulkan bahwa JFrame berfungsi sebagai wadah utama untuk menampung berbagai komponen Swing seperti JLabel (untuk *display pasif*), JTextField (untuk input dan output), JComboBox (untuk pilihan yang efisien) dan JButton (untuk mengeksekusi). Aplikasi GUI ini menghubungkan desain antarmuka dengan perhitungan aritmatika.

3.2 Saran

Untuk meningkatkan pemahaman dan keterampilan praktikum di masa mendatang disarankan untuk melakukan beberapa langkah pengembangan lanjutan yaitu memisahkan kode perhitungan aritmatika dari kode *event handling* GUI dengan membuat method terpisah yang menangani kalkulasi, sehingga kode utama lebih mudah diuji. Selain itu, dapat meningkatkan akurasi aplikasi dengan mengganti tipe data input dan perhitungan dari int ke double agar aplikasi dapat memproses dan menampilkan bilangan desimal.

DAFTAR PUSTAKA

- [1] S. Ranti, "Pengertian GUI, Cara Kerja, dan Contohnya," kompas.com, 30 November 2022. [Online]. Available: <https://tekno.kompas.com/read/2022/11/30/03150087/pengertian-gui-cara-kerja-dan-contohnya>. [Accessed 19 November 2025].
- [2] "Apa itu GUI bagaimana cara kerjanya dan apa saja komponennya dengan contoh?," jobstreet, 23 October 2025. [Online]. Available: <https://id.jobstreet.com/id/career-advice/article/apa-itu-gui-fungsi-contoh-komponen>. [Accessed 11 November 2025].
- [3] L. B. Viola, "Apa Itu GUI? Pengertian, Fungsi, Kelebihan dan Kekurangan," Jakarta Studio, 24 Juni 2024. [Online]. Available: <https://www.jakartastudio.com/apa-itu-gui/>. [Accessed 20 November 2025].
- [4] M. I. Ansar, "Apa itu GUI? Cara Kerja, Mekanisme, dan Contohnya," Digital Solusi, 20 April 2025. [Online]. Available: <https://digitalsolusigrup.co.id/gui-adalah/>. [Accessed 20 November 2025].