

LAPORAN PRATIKUM
ALGORITMA PEMROGRAMAN DAN PEMROGRAMAN
PEMROGRAMAN GUI 1
DISUSUN OLEH:
Rayya Syaqinah Putri Hasibuan
2511532007

DOSEN PENGAMPU:

Dr. Wahyudi, S.T. M.T

ASISTEN PRATIKUM:

Aufan Taufiqurrahman



PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
2025

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga laporan praktikum ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban pelaksanaan praktikum Algoritma Pemrograman dengan judul Pemrograman GUI 1. Laporan ini bertujuan untuk mendokumentasikan prosedur, hasil, serta pembahasan dari kegiatan praktikum, sekaligus sebagai sarana pembelajaran bagi penulis untuk lebih memahami materi terkait.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta semua pihak yang telah membantu sehingga laporan ini dapat terselesaikan.

Padang, 18 November 2025

Rayya Syaqinah Putri Hasibuan

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Pratikum.....	1
1.3 Manfaat Pratikum.....	1
BAB II PEMBAHASAN	2
2.1 Deskripsi Materi Pratikum	2
2.2 Pemrograman Aritmatika GUI	2
BAB III KESIMPULAN DAN SARAN	10
3.1 KESIMPULAN.....	10
3.2 SARAN	10
DAFTAR PUSTAKA.....	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Secara sederhana, GUI adalah wajah dari sebuah program atau aplikasi yang kita gunakan. Sebelum adanya GUI, pengguna harus mengingat perintah teks kompleks untuk menjalankan aplikasi. Dengan GUI, pengguna cukup mengklik ikon atau memilih opsi dari menu, yang jauh lebih mudah dimengerti. Praktikum GUI kali ini mencakup dasar-dasar pembuatan antar visual menggunakan komponen seperti *JTextField*, *JLabel*, dan *JButton*.

1.2 Tujuan Praktikum

Tujuan praktikum ini untuk mengenalkan konsep dasar pembuatan antarmuka grafis (GUI) dengan JavaScript. Mahasiswa belajar menggunakan komponen seperti *JLabel*, *JButton*, dan lainnya sebagai elemen utama tampilan aplikasi. Melalui praktikum ini, mahasiswa dilatih memahami struktur dasar pembuatan GUI dan cara menambahkannya pada program. Praktikum ini juga mengarahkan mahasiswa untuk mengetahui bagaimana interaksi pengguna dapat diterjemahkan dalam bentuk kode program.

1.3 Manfaat Praktikum

Melalui praktikum ini, mahasiswa dapat belajar mempermudah bagaimana agar interaksi manusia (pengguna) dan komputer dengan cara visualisasi informasi melalui ikon, tombol, dan menu karena GUI berfungsi untuk menyajikan infomasi secara visual, menerima input dari pengguna (seperti menginput angka pada praktikum ini), mengolah input tersebut dan menampilkannya sebagai output sehingga membuat penggunaan teknologi menjadi lebih intuitif, mudah dan tidak memerlukan pengetahuan teknis mendalam.

BAB II

PEMBAHASAN

2.1 Deskripsi Materi Pratikum

Dalam bahasa Java, terdapat dua paket komponen untuk membuat tampilan GUI, yaitu `java.awt` dan `javax.swing`. Paket AWT adalah paket lama, sedangkan Swing merupakan versi yang lebih baru dan banyak digunakan saat ini. Hampir semua komponen AWT sudah digantikan oleh Swing, misalnya `Button` pada AWT diganti menjadi `JButton`, dan `TextField` diganti menjadi `JTextField` pada Swing. Komponen *Swing* biasanya memiliki huruf “*J*” di awal namanya.

Dalam dunia pengembangan perangkat lunak, antarmuka pengguna (GUI — Graphical User Interface) memainkan peran penting dalam memberikan pengalaman pengguna yang baik. Java Swing adalah sebuah toolkit yang kuat yang digunakan untuk membangun antarmuka pengguna dalam aplikasi Java. Dengan Java Swing, pengembang dapat membuat antarmuka pengguna yang menarik dan responsif dengan mudah.

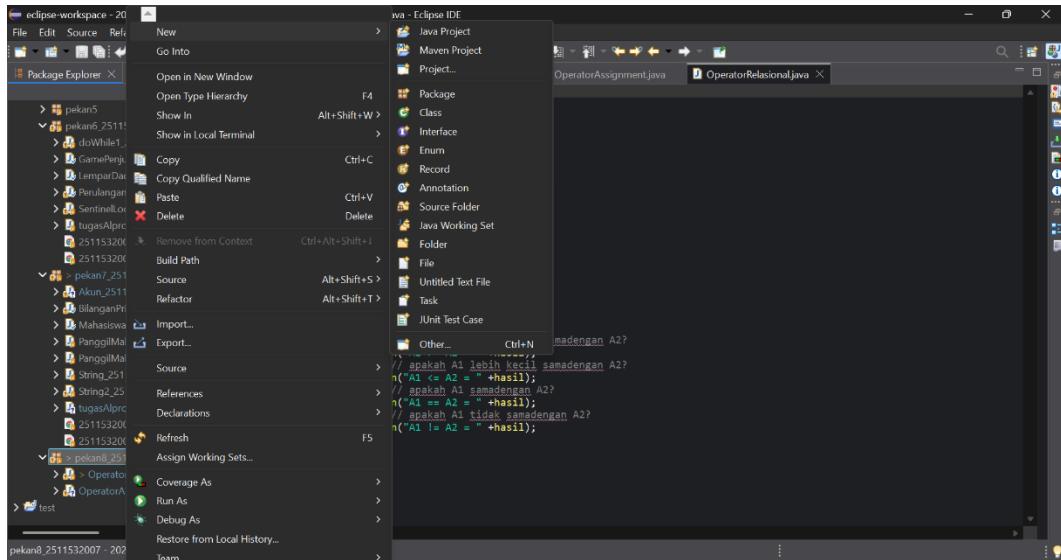
Beberapa contoh komponen GUI dalam Java antara lain `JLabel` untuk menampilkan teks, `JTextField` untuk memasukkan satu baris teks, `JTextArea` untuk menampilkan teks lebih dari satu baris, dan `JButton` sebagai tombol untuk menjalankan perintah ketika diklik. Semua komponen ini ditempatkan di dalam `JFrame`, yaitu jendela utama tempat GUI ditampilkan.

2.2 Pemrograman Aritmatika GUI

Berikut adalah penjelasan langkah demi langkah untuk pemrograman kode java yang diimplementasikan ke GUI dengan memilih operator aritmatika (+,-,*,/,%) dan menampilkan hasil perhitungan.

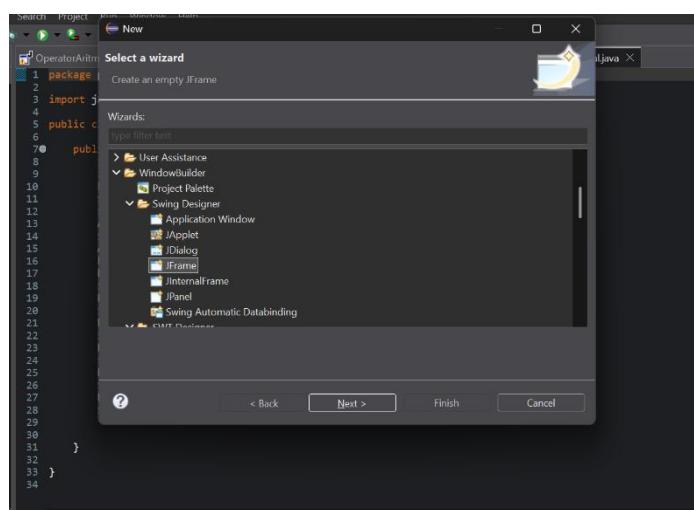
Langkah pengerjaan :

1. Membuat *package* baru dengan nama "pekan8_2511532007".
2. Klik kanan dan pilih *new > other*



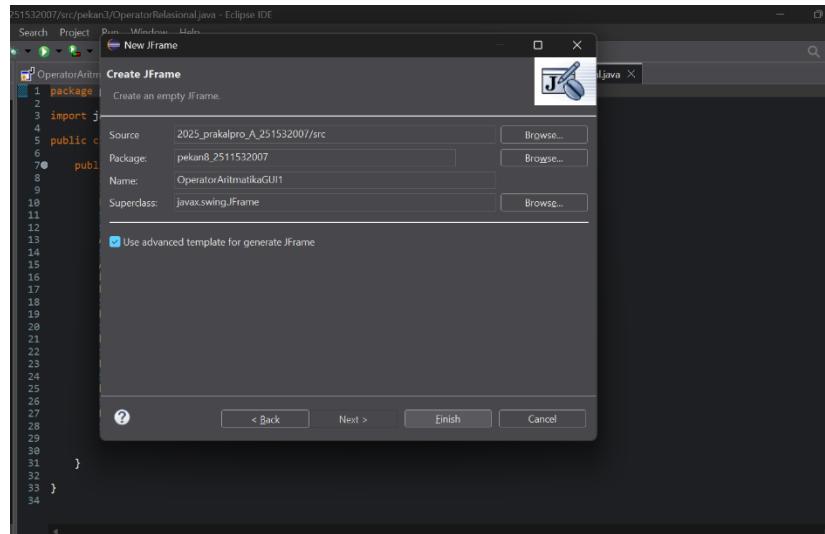
Gambar 2.2.1 posisi *New* dan *Other* pada Eclipse

3. Pilih *Window Builder > Swing Designer > JFrame*



Gambar 2.2.2 posisi *Window Builder*

4. Setelah klik *JFrame*, input nama untuk *JFrame* "OperatorAritmatikaGUI1" lalu klik *finish*.



Gambar 2.2.3 penginputan nama untuk *JFrame*

5. Maka tampilan Eclipse IDE Java akan seperti gambar 2.2.4, kemudian Klik design pada bawah kanan disamping source.

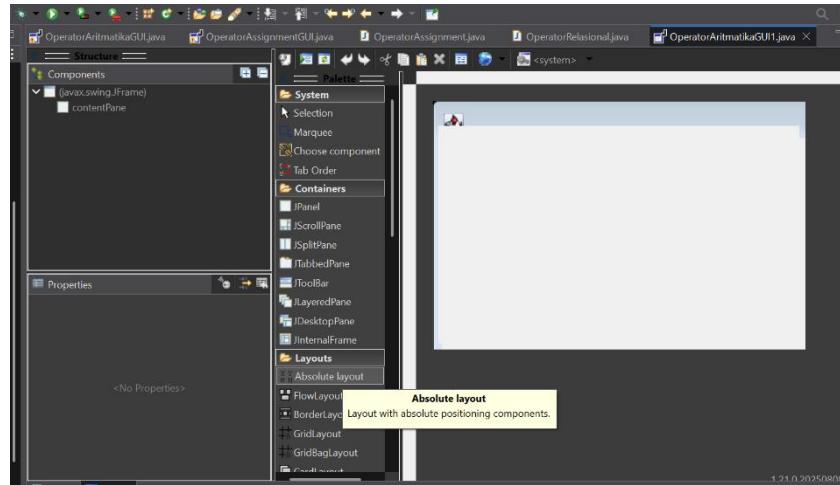
```

251532007/src/pekan8_2511532007/OperatorAritmatikaGUI1.java - Eclipse IDE
Search Project Run Window Help
File Edit View Insert Run Debug Tools Window Help
OperatorAritmatikaGUI1.java OperatorAssignmentGUI.java OperatorAssignment.java OperatorRelasional.java
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                     OperatorAritmatikaGUI1 frame = new OperatorAritmatikaGUI1();
22                     frame.setVisible(true);
23                 } catch (Exception e) {
24                     e.printStackTrace();
25                 }
26             }
27         });
28     }
29
30     /**
31      * Create the frame.
32      */
33     public OperatorAritmatikaGUI1() {
34         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35         setBounds(100, 100, 450, 300);
36         contentPane = new JPanel();
37         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
38         setContentPane(contentPane);
39     }
40
41
42
43

```

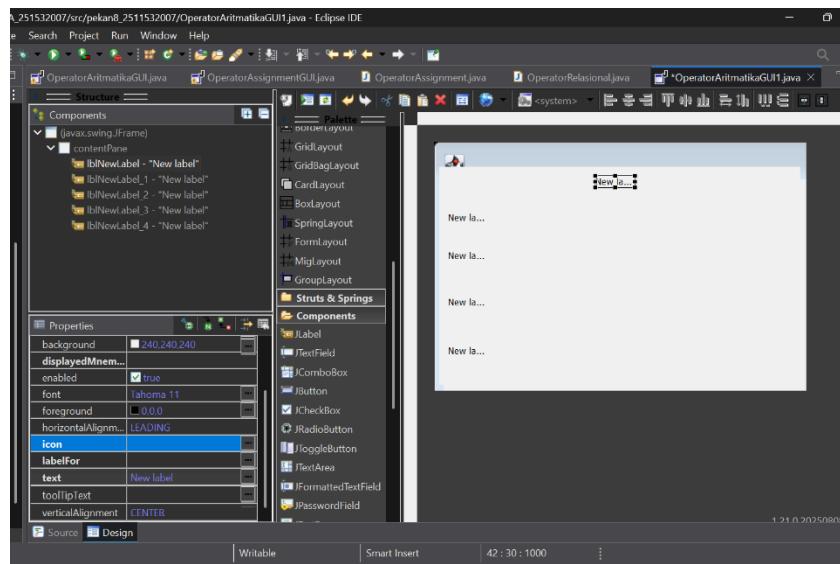
Gambar 2.2.4 tampilan Eclipse IDE Java setelah langkah ke 4

6. Setelah klik design, klik *Absolute Layout* lalu tarik ke frame.



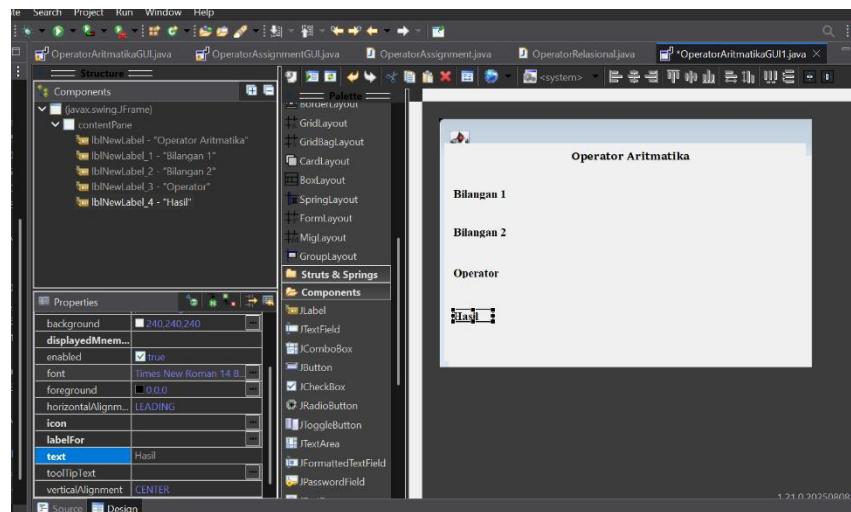
Gambar 2.2.5 letak *absolute layout* pada Eclipse IDE Java

7. Scroll ke bawah dan temukan *components* pada bar, klik *JLabel* untuk *title* dan nama untuk pengguna menginputkan inputan sesuai nama yang *programmer* buat.



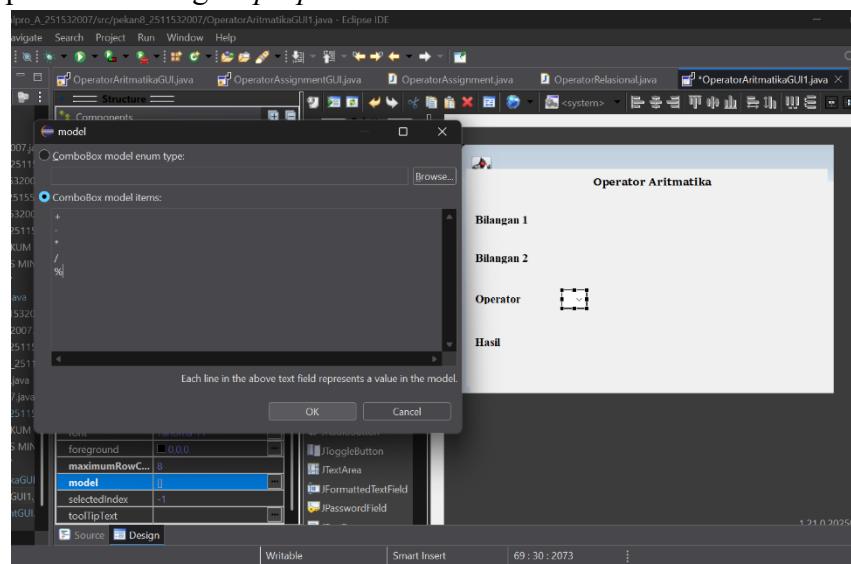
Gambar 2.2.5 *components* dan *properties* untuk *design*

8. Untuk mengubah teks pada "New Label" adalah di bagian *properties* dan isi nama sesuai dengan yang *programmer* inginkan di bagian teks lalu sesuaikan jenis *font* sesuai keinginan programmer dengan estetika.



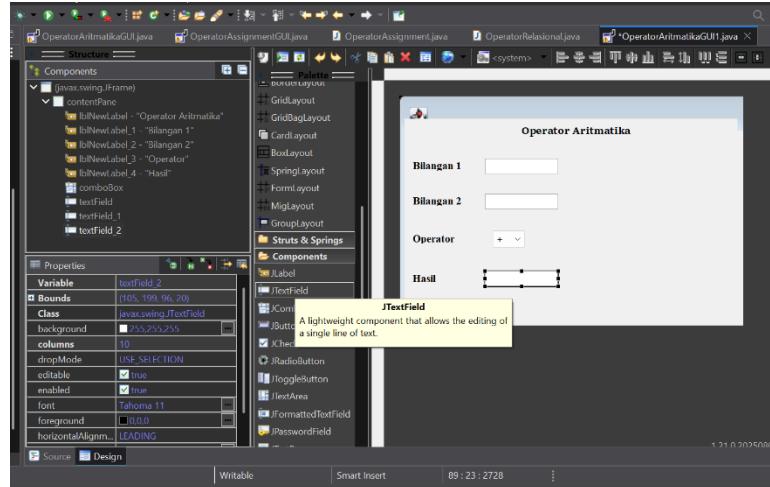
Gambar 2.2.6 kreasi *JLabel* untuk memperindah *design*.

9. Langkah selanjutnya klik *JComboBox* untuk pilihan operator, dan input jenis operator pada model bagian *properties* lalu klik OK.



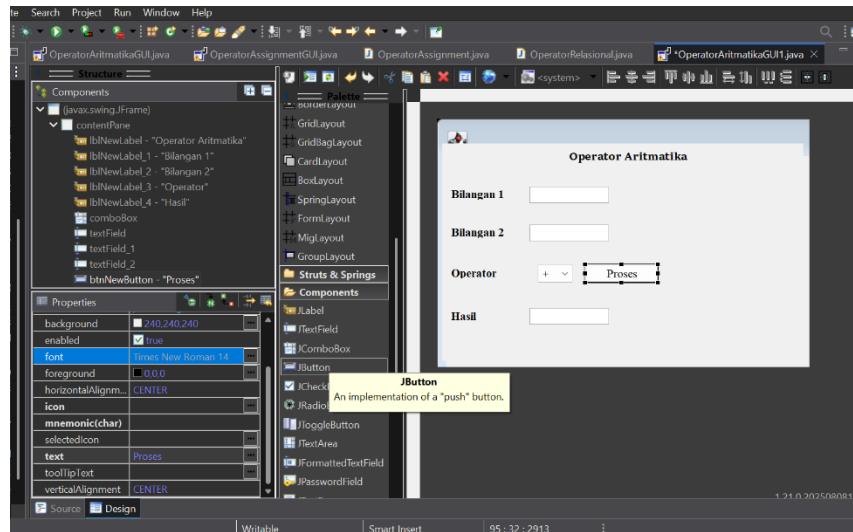
Gambar 2.2.7 menginput *JcomboBox* pada *design*.

10. Selanjutnya klik *JTextField* untuk kolom pengisian.



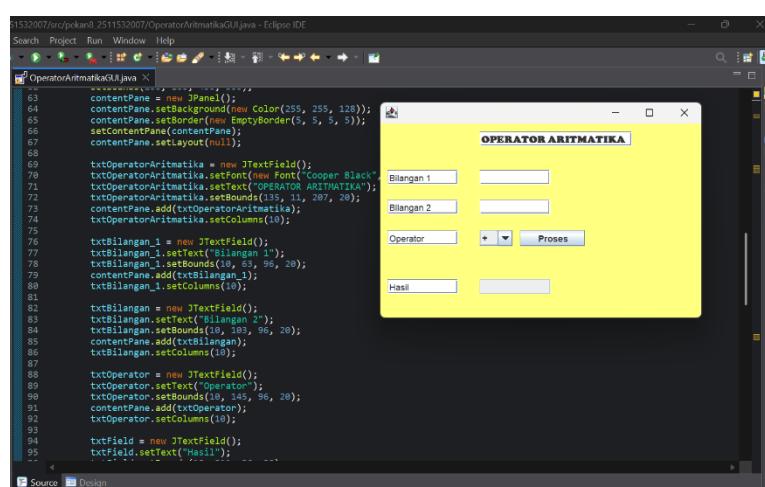
Gambar 2.2.8 menginput *JTextField* untuk kolom pengisian.

11. Klik *JButton* untuk tombol menjalankan perintah lalu ubah nama buttonnya menjadi proses.



Gambar 2.2.9 menginput *JButton* untuk tombol proses

12. Kreasikan dengan mengubah warna *background*, lalu tampilan akan seperti ini.



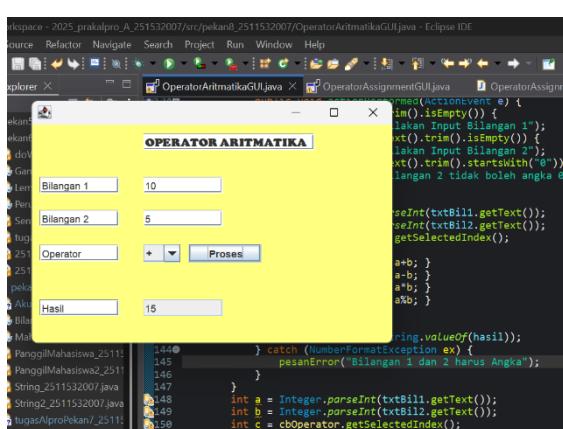
Gambar 2.2.10 mengubah warna tampilan *design*.

13. Input kode program *int hasil*; yang berfungsi menyimpan hasil perhitungan dari operasi aritmatika yang dipilih.
 14. Input kode *public void actionPerformed(ActionEvent e)* untuk pengecekan seluruh logika dan juga proses perhitungan dijalankan pada method ini.
 15. Validasi input *if(txtBil1.getText().trim().isEmpty())* untuk mencegah proses perhitungan angka ketika tidak lengkap. *Trim* berfungsi untuk menghilangkan spasi agar input "" dianggap kosong.
 16. Kode *try { ... } catch (NumberFormatException ex) { ... }* berguna untuk mencoba mengubah input teks menjadi angka (*Integer.parseInt()*).
 17. *int a = Integer.parseInt(txtBil1.getText());* untuk input bilangan pertama dari String ke integer, dan kode ini dijalankan setelah lolos validasi.
 18. Memasukkan logika operasi *if(c==0) { hasil = a+b; }, if(c==1) { hasil = a-b; }, if(c==2) { hasil = a*b; }, if(c==3) { hasil = a%b; }*.
 19. Pada kode program juga menginput kode

```
private void pesanPeringatan(String pesan) {
    JOptionPane.showMessageDialog(this, pesan,
    "Peringatan", JOptionPane.WARNING_MESSAGE);
}

private void pesanError(String pesan) {
    JOptionPane.showMessageDialog(this, pesan,
    "Peringatan", JOptionPane.ERROR_MESSAGE);
}
```

Agar ketika pengguna tidak menginput angka pada salah satu kolom bilangan, maka program akan memberikan peringatan sesuai yang diinputkan oleh programmer.
 20. *Run* program dan pengguna bisa menginput angka dan contoh hasil program pada gambar 2.2.11



Gambar 2.2.11 hasil kode program "OperatorAritmatikaGUI1"

Dari salah satu output pada gambar dapat kita lihat bahwa dari seluruh kode program berhasil melakukan perhitungan aritmatika sesuai operator yang disediakan dan menampilkan desain antarmuka yang interaktif dan mudah digunakan.

BAB III

KESIMPULAN DAN SARAN

3.1 KESIMPULAN

Praktikum pembuatan program aritmatika berbasis GUI pada Java berhasil memperkenalkan cara membangun antarmuka visual menggunakan komponen Swing seperti JFrame, JLabel, JTextField, JButton, dan JComboBox. Melalui praktikum ini, mahasiswa tidak hanya memahami cara menampilkan elemen GUI, tetapi juga menerapkan logika perhitungan aritmatika dalam sebuah aplikasi interaktif. Proses validasi input juga melatih mahasiswa agar program mampu menangani kesalahan pengguna dengan baik. Hasil output menunjukkan bahwa perhitungan dapat berjalan sesuai dengan operator yang dipilih. Secara keseluruhan, praktikum ini memberikan pengalaman langsung dalam menggabungkan konsep pemrograman logika dengan desain antarmuka yang mudah digunakan.

3.2 SARAN

Dalam melakukan praktikum selanjutnya, disarankan untuk lebih teliti dalam menuliskan sintaks dan memahami alur eksekusi program agar tidak terjadi error saat kompilasi.

DAFTAR PUSTAKA

- [1] "Pahami konsep Graphical-User Interface (GUI) secara mendalam," CodePolitan. [Online]. Available: <https://www.codepolitan.com/blog/pahami-konsep-graphical-user-interface-gui-selama-mdwlb/> [Accessed: 19-Nov-2025].
- [2] "Chapter 4: GUI," Runestone Academy, [Online]. Available: <https://runestone.academy/ns/books/published/javajavajava/ch4-gui.html> [Accessed: 19-Nov-2025].
- [3] Sofwan, "Sofwan Java 15," [Online]. Available: https://ilmukomputer.org/wp-content/uploads/2018/07/sofwan_java-15.pdf [Accessed: 19-Nov-2025].