

LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN
“PEMROGRAMAN GUI JAVA”

DISUSUN OLEH:

ALIYATAR RAFI AHMAD

2511533031

DOSEN PENGAMPU:

Dr. WAHYUDI, S.T, M.T

ASISTEN PRAKTIKUM:

JOVANTRI IMMANUEL GULO



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan judul **“Pemrograman GUI pada Java”** ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu bagian dari kegiatan praktikum untuk memahami konsep dasar pembuatan antarmuka grafis (Graphical User Interface) menggunakan bahasa pemrograman Java.

Dalam penyusunan laporan ini, penulis mendapatkan banyak bantuan, baik berupa bimbingan, materi, maupun arahan. Oleh karena itu, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta semua pihak yang telah mendukung proses pengerjaan praktikum ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Kritik dan saran yang membangun sangat diharapkan demi perbaikan laporan di masa mendatang.

Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi pembaca mengenai dasar-dasar pemrograman GUI pada Java.

Padang 21 November 2025

Aliyatar Rafi Ahmad

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 Pengertian GUI	3
2.2 Java Swing	3
2.2.1 JFrame.....	3
2.2.2 JPanel.....	4
2.2.3 JLabel.....	4
2.2.4 JButton.....	4
2.2.5 JTextField.....	4
2.2.6 JComboBox.....	4
2.3 Aplikasi Penggunaan Pemrogram GUI	5
2.3.1 Design GUI	5
2.3.2 Contoh Penggunaan Kalkulator	6
2.3.3 Kode Program	7
BAB III PENUTUP	9
3.1 Kesimpulan.....	9
3.2 Saran.....	9
DAFTAR PUSTAKA.....	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat menuntut adanya aplikasi yang tidak hanya berfungsi dengan baik, tetapi juga memiliki antarmuka yang mudah digunakan. Java sebagai salah satu bahasa pemrograman yang populer menyediakan kemampuan untuk membangun aplikasi dengan tampilan grafis melalui **GUI (*Graphical User Interface*)**. Pemrograman GUI memungkinkan pengguna berinteraksi dengan aplikasi secara lebih intuitif melalui elemen visual seperti tombol, kotak teks, menu, dan sebagainya.

Dalam dunia pengembangan perangkat lunak, kemampuan membuat GUI menjadi keterampilan penting karena sebagian besar aplikasi modern memiliki tampilan grafis. Java menyediakan berbagai library seperti **Swing** dan **AWT** yang digunakan untuk membangun antarmuka tersebut. Melalui praktikum ini, mahasiswa diharapkan memahami cara membuat komponen GUI dasar, mengatur layout, menangani event, serta menghubungkan antarmuka dengan logika program.

Oleh karena itu, laporan praktikum ini disusun untuk menjelaskan proses, langkah kerja, serta hasil dari implementasi pemrograman GUI pada Java. Dengan memahami konsep dasar GUI, mahasiswa dapat mengembangkan aplikasi yang lebih interaktif dan user-friendly di masa mendatang.\

1.2 Tujuan

1. Memahami konsep dasar Graphical User Interface (GUI) dalam pemrograman Java.
2. Mempelajari penggunaan komponen GUI seperti tombol, label, text field, combo box, dan lainnya.
3. Mengetahui cara mengatur layout dan posisi komponen pada jendela aplikasi.

1.3 Manfaat

1. Mahasiswa memperoleh pemahaman nyata tentang bagaimana sebuah antarmuka aplikasi dibangun dalam Java.
2. Menambah kemampuan teknis dalam mengembangkan aplikasi desktop dengan tampilan grafis.
3. Menjadi bekal dasar untuk pengembangan aplikasi yang lebih kompleks di bidang pemrograman Java.

BAB II

PEMBAHASAN

2.1 Pengertian GUI

GUI (Antarmuka Pengguna Grafis) di Java adalah pembuat pengalaman visual yang mudah digunakan Java aplikasi. GUI terutama terdiri dari komponen grafis seperti tombol, label, jendela, dll. yang dapat digunakan oleh pengguna untuk berinteraksi dengan aplikasi. GUI berperan penting dalam membangun antarmuka yang mudah digunakan. Dengan menggunakan bahasa pemrograman Java, kamu bisa membuat aplikasi yang berbasis GUI (berbasis desktop). *Graphical User Interface* (GUI) memungkinkan pengguna berinteraksi dengan aplikasi melalui elemen-elemen visual seperti tombol, kotak teks, *combo box*, *label*, dan sebagainya.

2.2 Java Swing

Swing di Java adalah toolkit *Graphical User Interface* atau GUI yang mencakup komponen GUI. Swing menyediakan serangkaian widget dan paket yang kaya akan komponen GUI yang canggih untuk aplikasi Java. Swing merupakan bagian dari *Java Foundation Classes* yang disingkat JFC, yang merupakan API untuk pemrograman Java GUI yang menyediakan GUI. Swing menyediakan berbagai komponen seperti *JFrame*, *JPanel*, *JButton*, *JLabel*, *TextField*, dan *ComboBox*.

2.2.1 JFrame

JFrame merupakan dasar dari setiap aplikasi Swing. Ia berfungsi sebagai jendela utama tempat komponen-komponen lain ditempatkan. *JFrame* menyediakan fitur-fitur seperti bilah judul, pengaturan ukuran, penutupan, dan opsi minimisasi, sehingga menjadikannya komponen yang esensial untuk setiap aplikasi antarmuka pengguna grafis (GUI) Java.

2.2.2 JPanel

JPanel adalah kontainer sederhana yang menampung dan mengatur beberapa komponen Swing. Ia membantu dalam merancang antarmuka pengguna grafis (GUI) dengan mengelompokkan komponen-komponen yang terkait. Pengembang sering menggunakan JPanel untuk mengatur tombol, label, atau bidang teks secara rapi di dalam JFrame.

2.2.3 JLabel

JLabel digunakan untuk menampilkan teks atau gambar statis pada antarmuka. Komponen ini sering digunakan untuk judul, deskripsi, dan teks petunjuk dalam sebuah aplikasi. Berbeda dengan bidang teks, JLabel tidak memungkinkan masukan dari pengguna.

2.2.4 JButton

JButton adalah komponen dasar yang memungkinkan pengguna untuk melakukan tindakan dengan mengklik. Tombol-tombol dalam Swing dapat memicu peristiwa seperti membuka jendela baru, mengirimkan formulir, atau menjalankan fungsi tertentu dalam program.

2.2.5 JTextField

JTextField digunakan untuk masukan teks satu baris. Komponen ini memungkinkan pengguna untuk memasukkan data, seperti nama, alamat email, atau kueri pencarian. JTextField bekerja dengan pendengar acara (event listeners) untuk menangkap masukan pengguna dan memprosesnya sesuai kebutuhan. Dalam praktikum ini digunakan untuk memasukkan:

- Bilangan pertama
- Bilangan kedua
- Menampilkan hasil perhitungan

2.2.6 JComboBox

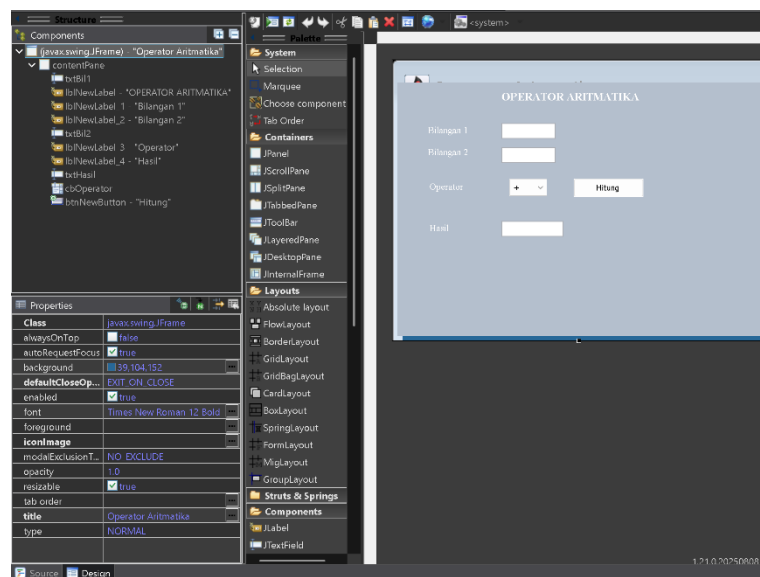
JComboBox menyediakan daftar dropdown di mana pengguna dapat memilih item dari beberapa pilihan. Fitur ini berguna untuk memilih nilai

yang telah ditentukan sebelumnya, seperti nama negara, peran pengguna, atau opsi pengaturan. Pada praktikum JComboBox digunakan untuk operator aritmatika.

2.3 Aplikasi Penggunaan Pemrogram GUI

Pada praktikum pekan 8, mahasiswa diinstruksikan untuk membuat kalkulator sederhana dengan menggunakan pemrograman GUI. Berikut adalah beberapa aplikasi atau penerapan secara langsung tentang pemrograman GUI.

2.3.1 Design GUI



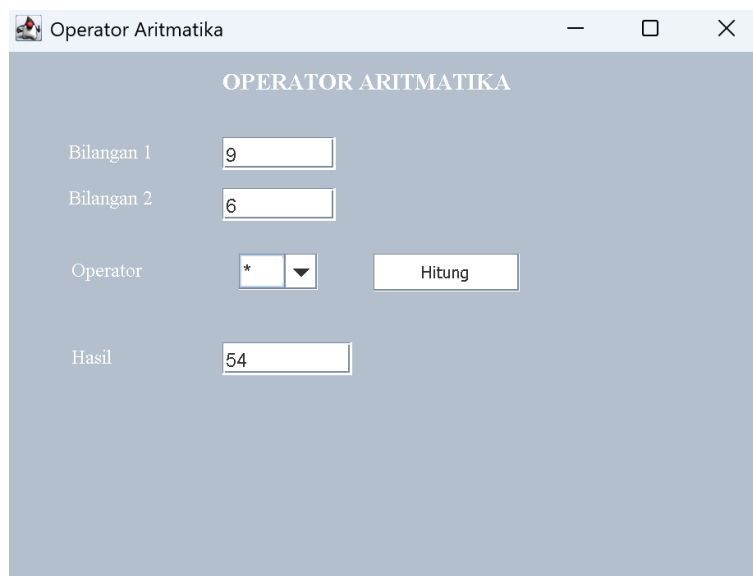
Gambar 2.1

Gambar di atas menunjukkan tampilan antarmuka aplikasi (GUI) yang dibuat menggunakan Java Swing pada NetBeans GUI Builder. Aplikasi ini diberi judul “**Operator Aritmatika**” dan digunakan untuk melakukan operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus.

Program **Operator Aritmatika GUI** yang dibuat pada praktikum ini merupakan aplikasi sederhana berbasis Java Swing yang digunakan untuk melakukan operasi aritmatika dasar. Aplikasi terdiri dari beberapa komponen GUI, yaitu *text field* untuk input bilangan, *combo box* untuk memilih operator, dan tombol Hitung untuk memproses perhitungan.

Ketika pengguna memasukkan Bilangan 1, Bilangan 2, kemudian memilih operator (+, −, ×, ÷, atau %), tombol Hitung akan menjalankan proses perhitungan melalui event ActionListener. Program melakukan validasi terlebih dahulu, yaitu memastikan text field tidak kosong dan input benar-benar berupa angka. Jika ada kesalahan, program menampilkan pesan peringatan menggunakan JOptionPane.

2.3.2 Contoh Penggunaan Kalkulator



Gambar 2.2

Gambar di atas menunjukkan tampilan saat program Operator Aritmatika dijalankan. Pada contoh tersebut, pengguna memasukkan nilai Bilangan 1 = 9 dan Bilangan 2 = 6, lalu memilih operator * (perkalian) dari combo box. Setelah tombol Hitung ditekan, program memproses operasi aritmatika yang dipilih dan menampilkan hasilnya pada field Hasil, yaitu 54.

Tampilan GUI menunjukkan bahwa aplikasi berjalan dengan benar, seluruh komponen berfungsi, dan proses perhitungan dilakukan sesuai input pengguna.

2.3.3 Kode Program

```

JButton btnNewButton = new JButton("Hitung");
btnNewButton.setBounds(225, 123, 89, 23);
btnNewButton.setBackground(new Color(255, 255, 255));
btnNewButton.addActionListener(new ActionListener() {
    int hasil;
    public void actionPerformed(ActionEvent e) {
        if (txtBil1.getText().trim().isEmpty()) {
            pesanPeringatan("Bilangan 1 Harus Diisi");
        } else if (txtBil2.getText().trim().isEmpty()) {
            pesanPeringatan("Bilangan 2 Harus Diisi");
        } else {
            try {
                int a= Integer.valueOf(txtBil1.getText());
                int b= Integer.valueOf(txtBil2.getText());
                int c= cbOperator.getSelectedIndex();
                if (c==0) { hasil=a+b;}
                if (c==1) { hasil=a-b;}
                if (c==2) { hasil=a*b;}
                if (c==3) { hasil=a/b;}
                if (c==4) { hasil=a%b;}
                txtHasil.setText(String.valueOf(hasil));
            } catch (NumberFormatException ex) {
                pesanError("Bilangan 1 dan 2 Harus Angka");
            }
        }
    }
});

```

Gambar 2.3

Kode di atas berfungsi untuk menangani aksi ketika tombol Hitung ditekan. Pertama, program memeriksa apakah kolom Bilangan 1 dan Bilangan 2 sudah terisi. Jika salah satunya kosong, maka akan muncul pesan peringatan. Jika kedua input terisi, program mencoba mengubah nilai teks menjadi angka menggunakan `Integer.valueOf()` di dalam blok `try`. Setelah itu, program mengambil operator yang dipilih dari combo box, lalu melakukan operasi sesuai indeks operator tersebut (penjumlahan, pengurangan, perkalian, pembagian, atau modulus). Hasil perhitungan kemudian ditampilkan pada field hasil. Jika terjadi kesalahan seperti input bukan angka, blok `catch` akan menampilkan pesan error.

```

private void pesanPeringatan(String pesan) {
    JOptionPane.showMessageDialog(this, pesan, "peringatan", JOptionPane.WARNING_MESSAGE);
}
private void pesanError(String pesan) {
    JOptionPane.showMessageDialog(this, pesan, "kesalahan", JOptionPane.ERROR_MESSAGE);
}

```

Gambar 2.4

Pada program GUI sebelumnya, tombol **Hitung** digunakan untuk menjalankan proses validasi dan perhitungan berdasarkan dua input angka serta operator yang dipilih. Saat tombol tersebut ditekan, program pertama-tama mengecek apakah kedua kolom bilangan sudah terisi; jika ada yang kosong, program memanggil metode `pesanPeringatan()` untuk menampilkan dialog peringatan agar pengguna mengisi input yang belum lengkap. Setelah input dianggap valid, program mencoba mengonversi teks menjadi angka dan melakukan operasi sesuai pilihan operator. Jika proses konversi gagal karena pengguna memasukkan karakter yang bukan angka, blok `catch` akan menangkap kesalahan tersebut dan memanggil metode `pesanError()` untuk menampilkan pesan kesalahan dengan ikon error. Dengan demikian, kedua metode ini bekerja sebagai pendukung utama alur validasi dan penanganan error, sehingga program menjadi lebih aman, informatif, dan mudah digunakan.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum pemrograman GUI pada Java, dapat disimpulkan bahwa pembuatan antarmuka menggunakan komponen Swing seperti JLabel, JTextField, JButton, dan JComboBox memungkinkan aplikasi berinteraksi secara langsung dengan pengguna. Melalui event handling, khususnya ActionListener, program dapat merespons aksi pengguna dan menjalankan proses seperti validasi input serta perhitungan aritmatika. Penerapan metode tambahan seperti pesanPeringatan() dan pesanError() juga membantu membuat program lebih informatif dan mudah digunakan. Secara keseluruhan, praktikum ini memperkuat pemahaman mengenai konsep dasar GUI, event handling, dan pembuatan aplikasi interaktif di Java.

3.2 Saran

Untuk pengembangan selanjutnya, aplikasi GUI dapat ditingkatkan dengan menambahkan pengecekan yang lebih lengkap, seperti validasi pembagian dengan nol atau format input tertentu. Tampilan antarmuka juga bisa diperbaiki menggunakan layout manager agar lebih responsif dan rapi. Selain itu, penggunaan fitur lanjutan seperti try-catch yang lebih spesifik, desain modular, atau migrasi ke JavaFX dapat menjadi langkah berikutnya agar aplikasi lebih profesional dan mudah dikembangkan.

DAFTAR PUSTAKA

- [1] J. Hartman, "Java Tutorial Swing: Cara Membuat Aplikasi GUI di Java," GURU99, 12 Agustus 2025. [Online]. Available: <https://www.guru99.com/id/java-swing-gui.html>. [Accessed 21 November 2025].
- [2] T. Tech, "A Beginner's Guide to Java Swing Components," Medium, 30 March 2025. [Online]. Available: <https://medium.com/@neerajrs124/a-beginners-guide-to-java-swing-components-bed7a8dca96f>. [Accessed 21 November 2025].