

**LAPORAN PRAKTIKUM
AGORITMA PEMROGRAMAN
“PEMOGRAMAN GUI”**

DISUSUN OLEH:

NOFRI ILHAM

2511531013

DOSEN PENGAMPU:

Dr. WAHYUDI, S.T, M.T

ASISTEN PRAKTIKUM:

MUHAMMAD ZAKI AL HAFIZ



**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur kehadiran Allah SWT karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum dengan judul “Pemrograman GUI” tepat pada waktunya. Laporan ini disusun sebagai salah satu syarat untuk memenuhi tugas praktikum pada mata kuliah Algoritma dan Pemrograman / Pemrograman Berbasis Objek.

Dalam penyusunan laporan ini, penulis mempelajari mengenai konsep dasar Graphical User Interface (GUI) dalam bahasa pemrograman Java, penggunaan komponen-komponen GUI seperti JButton, JLabel, JTextField, JComboBox, serta penerapan ActionListener untuk mengatur interaksi antara pengguna dan program.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta semua pihak yang telah membantu penyusunan laporan ini. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi pembaca.

Padang, 21 November 2025

Nofri Ilham

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI	iii
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 Komponen pada GUI.....	3
2.3 Aplikasi Penggunaan pemograman GUI pada Java	3
2.3.1 Desain GUI pada Java	4
2.3.2 Output Pemograman GUI	5
2.3.3 Kode Program Pertama.....	6
2.3.4 Kode Program Kedua	7
BAB III.....	8
3.1 Kesimpulan	9
3.2 Saran.....	9
DAFTAR PUSTAKA.....	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman modern, pembuatan aplikasi tidak hanya berfokus pada proses logika dan perhitungan di dalam kode program, tetapi juga pada bagaimana program tersebut berinteraksi dengan pengguna. Kebanyakan aplikasi saat ini dituntut memiliki tampilan visual yang mudah digunakan, informatif, dan interaktif, bukan sekadar berbasis teks (console based). Oleh karena itu, pemahaman mengenai pemrograman GUI (Graphical User Interface) menjadi salah satu kompetensi penting bagi seorang programmer dalam mengembangkan aplikasi yang memiliki pengalaman penggunaan (user experience) lebih baik dan lebih efisien.

Pada praktikum Minggu ke-8 mata kuliah Pemrograman, mahasiswa mempelajari cara mengimplementasikan antarmuka grafis menggunakan bahasa pemrograman Java dengan memanfaatkan library Swing. Praktikum ini melatih mahasiswa untuk menggunakan berbagai komponen GUI seperti JButton, JLabel, JTextField, JPanel, dan JComboBox, serta mempelajari konsep event handling menggunakan ActionListener untuk menangani aksi yang dilakukan pengguna pada aplikasi. Selain itu, praktikum ini juga bertujuan agar mahasiswa memahami struktur kode pada aplikasi GUI, mulai dari proses inisialisasi komponen, pengaturan layout, hingga eksekusi aksi program berdasarkan input pengguna. Dengan menguasai pemrograman GUI, mahasiswa diharapkan mampu membuat aplikasi desktop yang profesional, fungsional, dan dapat diterapkan pada kebutuhan dunia industri maupun pengembangan software ke depannya.

1.2 Tujuan

1. Mengetahui dan memahami konsep dasar pembuatan aplikasi berbasis GUI (Graphical User Interface) menggunakan bahasa pemrograman Java.
2. Mampu mengimplementasikan komponen-komponen GUI seperti JButton, JLabel, JTextField, JPanel, JComboBox, serta komponen lainnya ke dalam tampilan aplikasi.

3. Melatih kemampuan mahasiswa dalam mengatur layout, menginisialisasi objek komponen GUI, dan menempatkannya ke dalam sebuah window aplikasi.
4. Memahami dan mengaplikasikan konsep event handling menggunakan ActionListener untuk menangani aksi pengguna (misalnya klik tombol).
5. Mengembangkan aplikasi sederhana menggunakan GUI sebagai dasar untuk pengembangan aplikasi desktop yang lebih kompleks.

1.3 Manfaat

1. Mahasiswa memperoleh pengalaman langsung dalam membangun aplikasi yang interaktif dan user-friendly, bukan hanya aplikasi berbasis teks.
2. Meningkatkan pemahaman mahasiswa mengenai cara kerja komponen GUI dan bagaimana proses interaksi antara pengguna dan program melalui event.
3. Menambah keterampilan mahasiswa dalam membuat aplikasi berbasis desktop yang dapat digunakan untuk kebutuhan akademik maupun industri.
4. Melatih kemampuan logika dan kreativitas dalam merancang tampilan serta fungsi dari sebuah aplikasi.
5. Menjadi dasar pembelajaran untuk pengembangan aplikasi lanjutan seperti sistem informasi, aplikasi kasir, form data input, dan project desktop berbasis Java.

BAB II

PEMBAHASAN

2.1 Pengertian GUI

Graphic User Interface (GUI) adalah pemrograman dengan bahasa Java yang dibuat menggunakan aplikasi yang berbasis GUI. Tujuannya adalah menambahkan beberapa komponen yang tidak bisa dibuat dalam basis text. Komponen-komponen tersebut bisa berupa tombol, gambar, dll. Tujuannya adalah untuk memudahkan user menggunakan program yang dibuat tersebut. Kalau dilihat pengertian tentang GUI secara umum adalah Interaksi yang dapat dilaksanakan oleh user melalui menu dan icon yang diperlihatkan dalam modus grafik. Contoh implementasi GUI-based shell ini adalah pada sistem operasi Microsoft Windows.

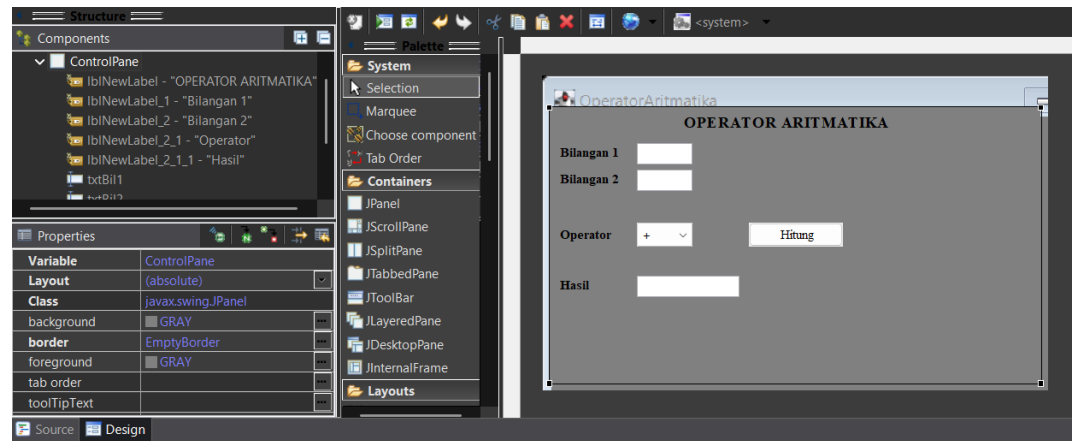
2.2 Komponen pada GUI

1. JLabel, berfungsi sebagai penunjuk tulisan/angka pada GUI.
2. JButton, berfungsi sebagai tombol pada GUI, komponen ini paling sering digunakan karena dibutuhkan user untuk menklik tombol tersebut
3. JRadioButton, berfungsi sebagai selection, komponen ini juga paling sering digunakan.
4. JCheckBox, berfungsi sebagai tanda centang, komponen ini juga paling sering digunakan. Biasanya digunakan pada saat user melakukan setting options.
5. JTextField, berfungsi sebagai kolom tempat untuk menulis.

2.3 Aplikasi Penggunaan pemrograman GUI pada Java

Pada Praktikum pekan ke 8, mahasiswa diinstruksikan untuk mendesain Kalkulator sederhana pada GUI. Berikut adalah penerapan aplikasi atau penerapan secara langsung tentang pemrograman GUI.

2.3.1 Desain GUI pada Java



Gambar 2.1

Gambar di atas merupakan tampilan antarmuka aplikasi (GUI) yang dibuat menggunakan Java Swing dengan bantuan NetBeans GUI Builder. Aplikasi ini diberi judul “Operator Aritmatika” dan digunakan untuk melakukan perhitungan aritmatika dasar.

Terdapat beberapa komponen GUI,:

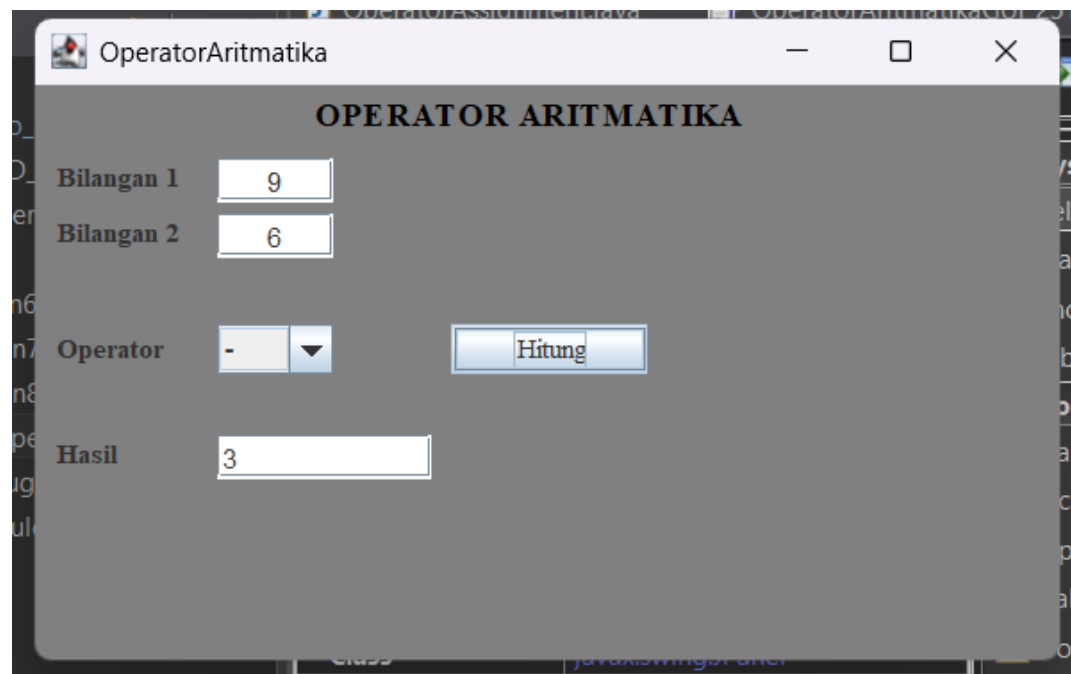
1. Label, digunakan sebagai keterangan untuk setiap bagian, seperti:
 - Bilangan 1 dan Bilangan 2 sebagai penanda input angka
 - Operator untuk memilih jenis operasi
 - Hasil untuk menampilkan output perhitungan
 - Judul OPERATOR ARITMATIKA di bagian atas
2. TextField, txtBill sebagai tempat menginput bilangan pertama, txtBil2 sebagai tempat menginput bilangan kedua, terakhir TextField hasil digunakan untuk menampilkan hasil perhitungan
3. ComboBox, digunakan untuk memilih operator aritmatika, misalnya +, -, *, /, atau %.
4. Tombol “Hitung”, tombol ini berfungsi menjalankan proses perhitungan setelah semua input dimasukkan.

Pengguna memasukkan nilai pada TextField Bilangan 1 dan Bilangan 2, kemudian memilih operator pada ComboBox. Setelah

tombol Hitung ditekan, program akan membaca input yang dimasukkan, lalu menjalankan proses perhitungan menggunakan ActionListener.

Sebelum menghitung, program terlebih dahulu melakukan validasi data, seperti mengecek apakah input kosong atau bukan angka. Jika terjadi kesalahan, program akan menampilkan pesan peringatan menggunakan JOptionPane. Apabila input benar, hasil perhitungan akan ditampilkan pada TextField hasil.

2.3.2 Output Pemograman GUI



Gambar 2.1

Gambar di atas merupakan hasil tampilan dari program Operator Aritmatika setelah dijalankan. Pada contoh tersebut, user telah mengisi input pada field Bilangan 1 dengan nilai 9 dan Bilangan 2 dengan nilai 6. Kemudian user memilih operator “-” (pengurangan) pada ComboBox, lalu menekan tombol Hitung.

Ketika tombol Hitung ditekan, event ActionListener akan memproses input yang dimasukkan dan menjalankan operasi aritmatika sesuai operator yang dipilih. Proses perhitungan yang terjadi adalah $(9 - 6 = 3)$. Hasil dari perhitungan tersebut kemudian ditampilkan pada TextField Hasil, yaitu angka 3. Hal ini menunjukkan bahwa program berjalan sesuai dengan logika yang telah dibuat, yaitu mengambil input dari TextField, melakukan perhitungan berdasarkan operator ComboBox, dan mengembalikan output ke field hasil.

2.3.3 Kode Program Pertama

```
JButton btnNewButton = new JButton("Hitung");
btnNewButton.setFont(new Font("Times New Roman", Font.PLAIN, 12));
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 di input");
        } 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 Bilangan 2 harus angka");
            }
        }
    }
});
```

Kode program 2.1

Pada program ini tombol Hitung berfungsi sebagai pemicu utama untuk menjalankan proses perhitungan pada aplikasi GUI.

Saat tombol ditekan, sistem terlebih dahulu melakukan pengecekan apakah input pada kolom Bilangan 1 dan Bilangan 2 sudah terisi. Jika salah satu kolom masih kosong, maka akan muncul pesan peringatan agar pengguna mengisi data terlebih dahulu. Selain itu, terdapat penanganan kesalahan (try-catch) untuk memastikan bahwa input yang diberikan adalah angka, sehingga apabila pengguna memasukkan karakter selain angka, program akan menampilkan pesan kesalahan yang sesuai.

Setelah input dinyatakan valid, program membaca operator aritmatika yang dipilih dari ComboBox, seperti penjumlahan, pengurangan, perkalian, pembagian, atau modulus. Berdasarkan operator tersebut, perhitungan dilakukan dan hasil akhirnya akan ditampilkan pada kolom Hasil.

2.3.4 Kode Program Kedua

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

Kode Program 2.2

Pada bagian kode ini terdapat dua fungsi yaitu pesanPeringatan dan pesanError. Kedua fungsi ini dipakai untuk menampilkan kotak pesan kepada pengguna ketika terjadi kesalahan saat mengisi data di aplikasi. Fungsi pesanPeringatan digunakan ketika pengguna belum mengisi salah satu kolom input, jadi program akan memberikan peringatan supaya data diisi terlebih dahulu. Pesan ini muncul menggunakan JOptionPane.showMessageDialog dengan tampilan pesan peringatan.

Sedangkan fungsi pesanError dipakai ketika pengguna salah memasukkan format data, seperti mengetikkan huruf ketika program

membutuhkan angka. Dengan memisahkan fungsi untuk peringatan dan error, program menjadi lebih rapi dan mudah dipahami. Selain itu, adanya pesan seperti ini membuat aplikasi lebih nyaman digunakan karena dapat membantu pengguna mengetahui kesalahan dan memperbaikinya tanpa membuat program berhenti atau crash.

BAB III

3.1 Kesimpulan

Berdasarkan praktikum Minggu ke-8 mengenai Pemrograman GUI menggunakan Java Swing, dapat disimpulkan bahwa mahasiswa telah mampu memahami dan mengimplementasikan konsep dasar pembuatan antarmuka grafis pada aplikasi desktop. Melalui pembuatan aplikasi kalkulator sederhana, mahasiswa mempelajari penggunaan berbagai komponen GUI seperti JLabel, JTextField, JButton, JComboBox, serta mekanisme event handling menggunakan ActionListener.

Praktikum ini memberikan pemahaman mengenai cara kerja komponen, proses validasi input, hingga bagaimana program merespons aksi pengguna dan menampilkan hasil perhitungan. Dengan praktik langsung, mahasiswa dapat melihat hubungan antara desain antarmuka, alur logika, serta fungsi kode program sehingga aplikasi yang dibuat dapat berjalan sesuai tujuan. Secara keseluruhan, praktikum ini meningkatkan kemampuan mahasiswa dalam membangun aplikasi interaktif, terstruktur, dan user-friendly menggunakan Java.

3.2 Saran

1. Perbanyak contoh implementasi GUI dalam berbagai kasus, seperti form input data, aplikasi kasir, atau sistem login, agar mahasiswa semakin terbiasa menggunakan komponen-komponen Swing.
2. Gunakan layout manager yang lebih bervariasi, misalnya GridLayout atau BorderLayout, untuk memperdalam pemahaman tentang pengaturan tata letak antarmuka.
3. Tambahkan validasi input yang lebih lengkap, seperti pembatasan jenis karakter, penanganan pembagian dengan nol, atau fitur reset, agar aplikasi lebih aman dan nyaman digunakan.

4. Tingkatkan pemahaman mengenai pemisahan kode (misalnya menerapkan konsep OOP dengan class terpisah untuk logika dan antarmuka) agar program lebih rapi dan mudah dikembangkan.

DAFTAR PUSTAKA

- [1] asbianto.wordpress.com, “Pengertian GUI dan Layout Manager,”. [Daring]. Tersedia pada <https://asbianto.wordpress.com/2015/12/12/pengertian-gui-dan-layout-manager/>
[Diakses: 22-Nov-2025].
- [2] BINUS – HIMSISFO, “Komponen-Komponen GUI pada Java,”. [Daring]. Tersedia pada <https://student-activity.binus.ac.id/himsisfo/2017/05/komponen-komponen-gui-pada-java/>
[Diakses: 22-Nov-2025].