

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

disusun Oleh:

SOFIAN ARBA’I

2511533029

Dosen Pengampu:

Dr. WAHYUDI, S.T, M.T

Asisten Praktikum:

JOVANTRI IMMANUEL GULO



**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
2025**

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur kehadirat Allah SWT karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan Laporan Praktikum Pekan 8 dengan judul "*Pemrograman GUI*" ini dengan baik. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum serta sebagai sarana untuk meningkatkan pemahaman mahasiswa mengenai konsep dan penerapan *Graphical User Interface (GUI)* dalam pemrograman.

Dalam praktikum pekan ini, penulis mempelajari dasar-dasar pembuatan antarmuka grafis menggunakan bahasa pemrograman Java, mulai dari pemahaman komponen GUI, pengaturan layout, hingga penggunaannya dalam mengembangkan aplikasi sederhana. Penyusunan laporan ini diharapkan dapat membantu penulis maupun pembaca dalam memahami materi secara lebih mendalam serta menjadi referensi untuk pembelajaran selanjutnya.

Penyusunan laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan berbagai pihak. Oleh karena itu, saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Dosen pengampu mata kuliah Praktikum Algoritma dan Pemrograman yang telah memberikan ilmu, arahan, serta bimbingannya.
2. Asisten praktikum yang telah membimbing dan memberikan penjelasan selama kegiatan praktikum berlangsung.

Saya meminta maaf jika banyak terdapat kesalahan pada penulisan laporan ini. Oleh karena itu, saya menerima kritik dan saran yang membangun demi perbaikan di masa mendatang.

Padang, 20 November 2025

Sofian Arba'i

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 Fungsi dan Manfaat GUI	3
2.3 Kelebihan dan Kekurangan GUI	4
2.4 Swing pada Pemrograman GUI	4
2.5 Aplikasi Penggunaan Pemrograman GUI.....	5
2.5.1 Design GUI pada Java	5
2.5.2 Kode Program Pemrograman GUI	6
2.5.3 Kode Program Pemrograman GUI	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 mendorong kebutuhan akan aplikasi yang tidak hanya memiliki fungsi yang baik, tetapi juga mudah digunakan oleh pengguna. Salah satu aspek penting dalam pengembangan aplikasi modern adalah *Graphical User Interface* (GUI), yaitu antarmuka berbasis grafis yang memungkinkan pengguna berinteraksi dengan sistem secara lebih intuitif dan efisien. Dibandingkan dengan antarmuka berbasis teks (*Command Line Interface*), GUI memberikan pengalaman yang lebih interaktif melalui penggunaan tombol, menu, form, dan elemen visual lainnya. Dalam pemrograman Java, GUI dapat dibangun menggunakan berbagai pustaka seperti AWT, Swing, maupun JavaFX. Pemahaman mengenai komponen-komponen GUI, cara dan pengaturan layout merupakan kemampuan dasar yang perlu dikuasai oleh mahasiswa informatika ketika mengembangkan aplikasi yang dapat berinteraksi langsung dengan pengguna.

Praktikum Pekan 8 mengenai *Pemrograman GUI* diselenggarakan untuk memberikan pengalaman langsung dalam membuat dan mengimplementasikan aplikasi sederhana dengan antarmuka grafis. Melalui praktikum ini, mahasiswa diharapkan mampu memahami konsep dasar GUI, mengenal berbagai komponen visual, serta dapat membangun aplikasi yang responsif dan mudah digunakan.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum pekan 8 tentang pemrograman GUI adalah sebagai berikut:

1. Memahami konsep dasar GUI dalam pemrograman Java, termasuk struktur, komponen, dan prinsip kerja antarmuka grafis.

2. Mengenal dan menggunakan komponen-komponen GUI seperti label, tombol, text field, combo box, panel, dan layout manager.
3. Mampu membuat aplikasi sederhana berbasis GUI
4. Menjadi dasar pembelajaran untuk pengembangan aplikasi lebih kompleks di materi pertemuan berikutnya.

1.3 Manfaat

Manfaat yang didapatkan setelah melakukan praktikum tentang pemrograman GUI sebagai berikut:

1. Menambah wawasan tentang cara kerja antarmuka grafis pada aplikasi berbasis Java.
2. Meningkatkan kemampuan dalam merancang dan mengimplementasikan antarmuka pengguna yang intuitif dan interaktif.
3. Memberikan pengalaman langsung dalam menghubungkan konsep teori dengan implementasi nyata melalui praktik pemrograman.
4. Melatih keterampilan problem-solving melalui pembuatan aplikasi yang mengintegrasikan logika program dengan komponen visual.

BAB II

PEMBAHASAN

2.1 Pengertian GUI

GUI atau Graphical User Interface merupakan sistem yang memungkinkan para pengguna untuk melakukan interaksi dengan komputer. GUI merupakan sebuah desain antar muka pada komputer, selain GUI ada pula CLI atau Command Line Interface, yang berbasis text dimana para pengguna dapat melakukan interaksi dengan komputer melalui perintah text atau Command Line. GUI merupakan antarmuka pada system yang menggunakan menu grafis. Menu grafis ini maksudnya terdapat tampilan yang lebih ditekankan untuk membuat system yang user-friendly agar para pengguna lebih nyaman menggunakannya. Secara sederhana, GUI adalah suatu media virtual yang dapat membuat pengguna memberikan perintah tertentu pada komputer tanpa mengetik perintah atau Command. Dengan GUI, perintah dapat dikonversi menjadi ikon dalam layar monitor yang dapat diklik untuk memulai fungsinya.

2.2 Fungsi dan Manfaat GUI

Fungsi utama GUI adalah untuk menyederhanakan interaksi antara pengguna dan perangkat lunak. GUI mengkonversi perintah teks yang sebelumnya harus diketik menjadi ikon atau menu yang dapat diklik, sehingga pengguna dapat menjalankan program dengan lebih mudah dan cepat.

GUI memungkinkan pengguna untuk mengoperasikan perangkat lunak tanpa perlu menghafal perintah teks, cukup dengan klik ikon atau menu yang tersedia. Menyediakan visualisasi data yang lebih baik, sehingga informasi dapat dipahami dengan lebih cepat dan jelas. Menawarkan tampilan dan kontrol yang konsisten di berbagai aplikasi, mengurangi kurva pembelajaran pengguna. Memungkinkan pengguna membuka dan mengoperasikan beberapa jendela atau aplikasi sekaligus, meningkatkan efisiensi kerja.

2.3 Kelebihan dan Kekurangan GUI

Intuitif dan Mudah Digunakan: GUI menggunakan elemen visual yang mudah dikenali seperti ikon, jendela, dan menu. Penggunaan elemen-elemen ini memungkinkan pengguna untuk berinteraksi dengan perangkat lunak secara lebih alami dan intuitif. Mendukung Multitasking: GUI memungkinkan pengguna untuk membuka beberapa jendela atau aplikasi secara bersamaan. Hal ini memudahkan pengguna dalam mengelola berbagai tugas sekaligus dan meningkatkan efisiensi kerja (MySkill).

Selain kelebihan, GUI juga memiliki kekurangan diantara adalah, kebutuhan Sumber Daya Lebih Tinggi: GUI memerlukan sumber daya komputer yang lebih besar dibandingkan CLI, seperti memori dan daya prosesor yang lebih tinggi. Hal ini dapat mempengaruhi kinerja sistem, terutama pada perangkat dengan spesifikasi rendah. Kerentanan Terhadap Malware: Karena banyaknya interaksi visual, GUI lebih rentan terhadap serangan malware dan virus. Interaksi yang lebih intens dengan elemen grafis meningkatkan risiko keamanan sistem. Keterbatasan dalam Tugas Spesifik: Untuk tugas yang membutuhkan serangkaian perintah cepat atau skrip, CLI lebih efisien dibandingkan GUI. GUI sering kali memiliki keterbatasan dalam fleksibilitas dan kecepatan untuk tugas-tugas tertentu.

2.4 Swing pada Pemrograman GUI

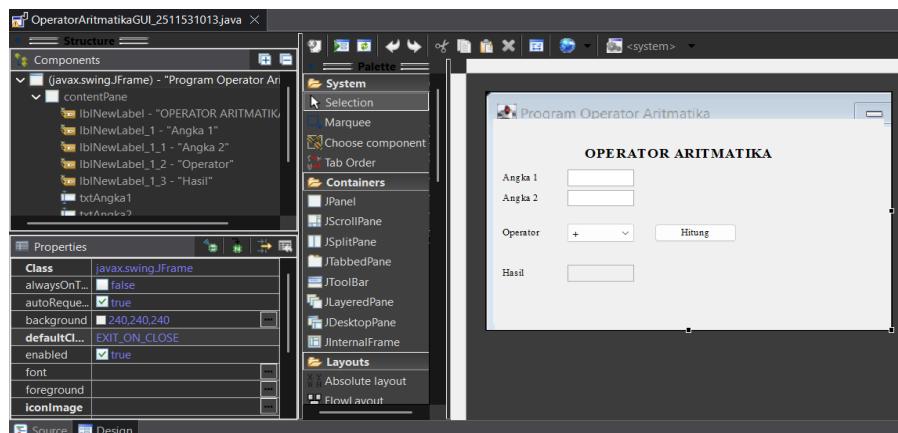
Java Swing adalah toolkit yang kuat untuk membangun antarmuka pengguna dalam aplikasi desktop. Java Swing adalah bagian dari API (Application Programming Interface) Java yang digunakan untuk membangun antarmuka pengguna dalam aplikasi desktop. Swing menawarkan berbagai komponen GUI, termasuk tombol, kotak teks, daftar, tabel, dan banyak lagi. Dengan menggunakan Swing, pengembang dapat membuat tampilan yang menarik dan berinteraksi dengan pengguna. Setiap komponen swing pada GUI memiliki kegunaan atau fungsi tersendiri. Seperti tombola tau *button* yang berfungsi merespon saat ditekan oleh pengguna.

2.5 Aplikasi Penggunaan Pemrograman GUI

Pada pekan 8 tentang pemrograman GUI ini mahasiswa diminta untuk memperhatikan dan mengikuti perintah dari dosen. Mahasiswa diinstruksikan supaya mengikuti, mendesain dan ikut menulis kode program yang dijelaskan oleh dosen. Berikut adalah beberapa aplikasi atau penerapan secara langsung tentang pemrograman GUI ini.

2.5.1 Design GUI pada Java

Pada bagian ini mahasiswa diminta untuk mengikuti dosen dengan mendesain bagian depan dari aplikasi atau lebih tepatnya membuat kalkulator. Berikut adalah contoh dari desain yang dibuat untuk operator aritmatika atau bisa dibilang kalkulator sederhana.



Gambar 2.1

Pada tampilan NetBeans di gambar tersebut, Palette, Structure, dan Properties memiliki fungsi berbeda dalam proses pembuatan GUI. Palette terletak di bagian tengah kiri dan berisi kumpulan komponen yang dapat digunakan dalam desain GUI, seperti label, text field, button, panel, serta berbagai layout. Pengguna cukup melakukan drag-and-drop komponen dari Palette ke form desain untuk membangun tampilan antarmuka. Structure berada di sisi paling kiri dan menampilkan hierarki atau susunan komponen yang ada pada form, mirip seperti tree view. Melalui Structure, pengguna dapat melihat

komponen mana yang berada di dalam container tertentu, memudahkan pengaturan posisi, pengelompokan, dan pemilihan komponen. Sementara itu, Properties berada di bawah Structure dan menampilkan seluruh properti dari komponen yang sedang dipilih, seperti teks, warna, ukuran, font, event, dan pengaturan lainnya.

2.5.2 Kode Program Pemrograman GUI

Pada bagian ini, mahasiswa tetap mengikuti perintah dari dosen dan memahaminya. Untuk menjalankan hasil desain yang telah dibuat tadi tetap dibutuhkan untuk membuat beberapa kode program atau mengubah kode program supaya desain yang telah dibuat bisa dijalankan sesuai dengan keinginan kita. Penjelasan kode program akan akan saya bagi menjadi beberapa bagian dan dimulai pada bagian ini.

```

114     JButton btnNewButton = new JButton("Hitung");
115     btnNewButton.addActionListener(new ActionListener() {
116         int Hasil ;
117         public void actionPerformed(ActionEvent e) {
118             if(txtAngka1.getText().trim().isEmpty()) {
119                 pesanPeringatan("Inputkan Angka 1");
120             } else if(txtAngka2.getText().trim().isEmpty()) {
121                 pesanPeringatan("Inputkan Angka 2");
122             } else {
123                 try {
124                     int a= Integer.valueOf(txtAngka1.getText());
125                     int b= Integer.parseInt(txtAngka2.getText());
126                     int c= cbOperator.getSelectedIndex();
127                     if (c==0) {Hasil= a+b; }
128                     if (c==1) {Hasil= a-b; }
129                     if (c==2) {Hasil= a*b; }
130                     if (c==3) {Hasil= a/b; }
131                     if (c==4) {Hasil= a%b; }
132                     txtHasil.setText(String.valueOf(Hasil));
133                 }
134             } catch (NumberFormatException ex) {
135                 pesanError("Angka 1 dan 2 tidak Valid");
136             }
137         }
138     }
139 
```

Kode Program 2.1

Pada bagian if–else dalam kode tersebut, program melakukan validasi input sebelum perhitungan dijalankan. Pertama, kondisi if(txtAngka1.getText().trim().isEmpty()) digunakan untuk memastikan kolom Angka 1 tidak dibiarkan kosong; jika kosong, program langsung

menampilkan pesan peringatan. Jika Angka 1 terisi, pengecekan dilanjutkan ke else if(txtAngka2.getText().trim().isEmpty()) untuk memastikan kolom Angka 2 juga tidak kosong. Hanya jika kedua input telah terisi, program masuk ke blok else yang berisi proses perhitungan. Di dalam blok ini, nilai teks dikonversi menjadi integer melalui *try-catch*, lalu program menentukan operasi yang dipilih berdasarkan indeks ComboBox menggunakan serangkaian kondisi if(c == 0) hingga if(c == 4), yang masing-masing mewakili penjumlahan, pengurangan, perkalian, pembagian, dan modulo. Struktur if-else tersebut memastikan bahwa perhitungan hanya dilakukan ketika input valid, sementara kesalahan seperti memasukkan karakter non-angka akan ditangani oleh blok *catch* yang menampilkan pesan error.

2.5.3 Kode Program Pemrograman GUI

Kode program bagian kedua adalah kode program yang sangat berhubungan dengan kode program pada bagian pertama. Kode program ini berfungsi untuk memperkenalkan catch yang menghasilkan output ketika user membiarkan salah satu angka kosong. Berikut adalah kode program.

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

Kode Program 2.2

Pada bagian kode program tersebut terdapat dua metode, yaitu pesanPeringatan() dan pesanEror(), yang keduanya berfungsi menampilkan kotak dialog pesan menggunakan JOptionPane.showMessageDialog(). Masing-masing metode menerima parameter berupa teks pesan, lalu menampilkannya pada jendela peringatan dengan tipe WARNING_MESSAGE. Meskipun namanya berbeda, kedua metode sebenarnya melakukan hal yang sama, hanya digunakan untuk membedakan konteks pesan saat dipanggil di bagian

lain program. Hubungannya dengan kode sebelumnya adalah bahwa metode pesanPeringatan() dipanggil ketika kolom Angka 1 atau Angka 2 kosong dalam proses pengecekan if–else, sehingga pengguna diberi tahu bahwa input belum lengkap. Sementara itu, metode pesanEror() dipanggil di dalam blok *catch* ketika terjadi kesalahan konversi, seperti pengguna memasukkan huruf atau karakter yang bukan angka. Dengan demikian, kedua metode ini berfungsi sebagai mekanisme penanganan pesan yang membuat antarmuka lebih interaktif dan membantu pengguna memahami kesalahan yang terjadi saat menjalankan program.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum pekan 8 tentang *Pemrograman GUI*, dapat disimpulkan bahwa Graphical User Interface merupakan komponen penting dalam pengembangan aplikasi modern karena memungkinkan interaksi yang lebih mudah, intuitif, dan efisien antara pengguna dan sistem. Melalui praktikum ini, mahasiswa memperoleh pemahaman mengenai penggunaan komponen-komponen GUI seperti label, tombol, text field, combo box, panel, serta struktur layout dalam Java Swing. Selain itu, mahasiswa juga belajar menghubungkan elemen visual dengan logika program, seperti melakukan validasi input, menampilkan pesan menggunakan *JOptionPane*, hingga melakukan operasi aritmatika berdasarkan pilihan pengguna. Praktikum ini memberikan pengalaman langsung yang membantu mahasiswa memahami konsep antarmuka grafis serta meningkatkan keterampilan pemrograman dalam membangun aplikasi sederhana berbasis GUI.

3.2 Saran

Untuk pengembangan praktikum selanjutnya, disarankan agar mahasiswa diberikan lebih banyak latihan mandiri yang melibatkan variasi komponen GUI agar pemahaman semakin mendalam. Selain itu, materi tentang pengaturan layout dan penanganan event lebih kompleks dapat ditambahkan untuk meningkatkan kemampuan mahasiswa dalam merancang antarmuka yang lebih interaktif dan responsif. Penggunaan studi kasus aplikasi sederhana juga dapat membantu mahasiswa memahami hubungan antara desain GUI dan logika program secara lebih nyata. Terakhir, akan lebih baik jika praktikum dilengkapi dengan pembahasan kesalahan umum (error handling) agar mahasiswa terbiasa membuat aplikasi yang lebih stabil dan ramah pengguna.

DAFTAR PUSTAKA

- [1] BANGWOK.ID, “Jurnal Java AWT dan Swing,”. [Daring]. Tersedia pada: [Jurnal Java AWT dan Swing - Bangwok.id](#) [Diakses: 19-Nov-2025].
- [2] JakartaStudio, “Apa Itu GUI? Pengertian, Fungsi, Kelebihan dan Kekurangan,”. [Daring]. Tersedia pada: [Apa Itu GUI? Pengertian, Fungsi, Kelebihan dan Kekurangan](#) [Diakses: 19-Nov-2025].
- [3] Medium, “Java Swing: Membangun Antarmuka Pengguna yang Menarik dengan GUI,”. [Daring]. Tersedia pada: [Java Swing: Membangun Antarmuka Pengguna yang Menarik dengan GUI | by RICZKY PRATAMA | Medium](#) [Diakses: 20-Nov-2025].