

**LAPORAN PRATIKUM PEMROGRAMAN ALGORITMA DAN
PEMROGRAMAN**

PEMAKAIAN GUI

disusun Oleh:

Habib Al Faruq

2511532010

Dosen Pengampu: Dr. Wahyudi S.T.M.T

Asisten Praktikum: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur atas kehadirat Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum algoritma pemrograman mengenai *string* di Java. Penulisan laporan ini bertujuan untuk membuat sebuah laporan tentang materi pemrograman mengenai penggunaan *string*.

Penyusunan laporan praktikum ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada dosen pengampu mata kuliah praktikum pemrograman algoritma dan pemrograman, Bapak Dr. Wahyudi. S.T.M.T dan asisten praktikum, Uda Muhammad Zaki Al Hafiz yang telah memberikan arahan dan masukan dalam proses penyusunan laporan praktikum ini. Penulis berharap laporan praktikum ini dapat memberikan manfaat bagi pembaca dalam mempelajari konsep penggunaan *string*.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Praktikum	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum	2
 BAB II PEMBAHASAN.....	 4
2.1 Perulangan While.....	4
2.1.1 Penjelasan Kode Program.....	4
2.1.2 Contoh Output	6
 BAB III KESIMPULAN.....	 9
3.1 Ringkasan Hasil Praktikum.....	9
DAFTAR PUSTAKA.....	10

BAB I

PENDAHULUAN

1.1 Latar Belakang Praktikum

Perkembangan teknologi perangkat lunak saat ini mendorong programmer untuk tidak hanya bergantung pada program berbasis teks, tetapi juga mampu membangun aplikasi dengan antarmuka yang lebih interaktif dan mudah diakses oleh pengguna. Penggunaan Graphical User Interface (GUI) menjadi sangat penting karena memberikan pengalaman yang lebih intuitif dan ramah pengguna dibandingkan aplikasi yang sepenuhnya berjalan pada mode console.

Java sendiri menyediakan pustaka Swing, yang menawarkan berbagai komponen GUI seperti tombol, kotak teks, label, hingga dialog peringatan. Melalui praktikum ini, mahasiswa diajak mengenal proses pembuatan aplikasi GUI sederhana menggunakan Java Swing di Eclipse—mulai dari merancang tampilan hingga menghubungkan setiap komponen dengan logika program melalui mekanisme event handling.

Program operator aritmatika yang dikembangkan dalam praktikum ini berfungsi sebagai contoh dasar bagaimana sebuah aplikasi GUI memproses input pengguna. Melalui program tersebut, mahasiswa dapat melihat langsung bagaimana validasi data dilakukan serta bagaimana sistem memberikan pesan kesalahan yang tepat agar aplikasi tetap berjalan dengan benar dan responsif.

1.2 Tujuan Praktikum

Tujuan utama praktikum ini adalah memberikan pemahaman yang lebih komprehensif mengenai pembuatan antarmuka grafis menggunakan Java Swing di Eclipse. Melalui kegiatan ini, mahasiswa belajar menambahkan berbagai komponen

GUI seperti label, text field, combo box, dan tombol sekaligus memahami bagaimana sebuah antarmuka disusun secara sistematis. Praktikum ini juga diarahkan agar mahasiswa mampu mengaitkan desain tampilan dengan logika program melalui pemanfaatan event handling, sehingga setiap tindakan pengguna dapat direspon secara tepat oleh aplikasi.

Di samping itu, praktikum ini bertujuan melatih mahasiswa dalam melakukan validasi input dan memproses data sesuai operator aritmatika yang dipilih pengguna. Mahasiswa diharapkan mampu menerapkan pengecekan input kosong, memastikan jenis data yang dimasukkan sudah benar, serta mengelola pesan kesalahan dengan bantuan JOptionPane. Dengan pendekatan tersebut, praktikum ini memberikan bekal penting dalam memahami alur kerja aplikasi GUI sederhana, mulai dari perancangan tampilan hingga penerapan logika yang bekerja di balik antarmuka.

1.3 Manfaat Praktikum

Praktikum ini memberikan manfaat penting bagi mahasiswa karena membantu mereka memahami cara merancang aplikasi berbasis GUI yang interaktif dan mudah digunakan. Dengan berlatih menempatkan komponen antarmuka serta menghubungkannya dengan logika program, mahasiswa dapat melihat secara langsung bagaimana sebuah aplikasi desktop dibangun secara runut. Pengalaman ini menjadi fondasi awal yang sangat berguna ketika mereka mengembangkan aplikasi yang lebih kompleks di masa mendatang.

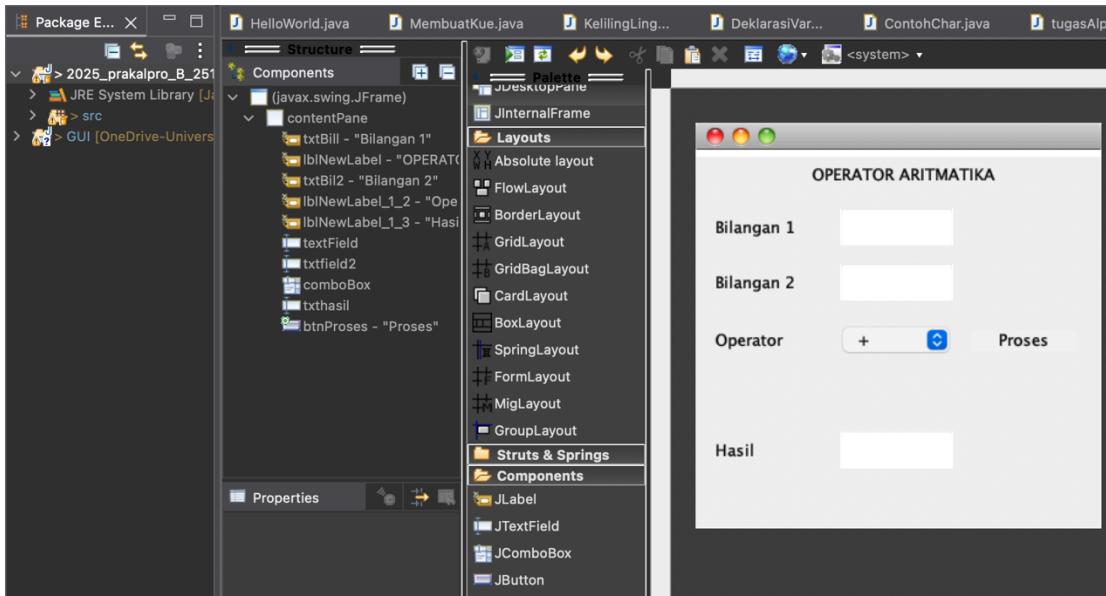
Selain itu, praktikum ini meningkatkan kemampuan mahasiswa dalam menangani proses validasi data dan penanganan kesalahan. Mahasiswa belajar memastikan input pengguna sesuai ketentuan, mencegah terjadinya error selama program berjalan, dan memberikan pesan peringatan yang informatif melalui JOptionPane. Keterampilan ini sangat relevan dalam pengembangan perangkat lunak

profesional, karena memastikan aplikasi tetap stabil dan memberikan pengalaman penggunaan yang lebih baik.

BAB II

PEMBAHASAN

2.1. Operasi Aritmatika Menggunakan GUI



Gambar 2.1

2.1.1 Langkah-langkah Membuat program

1. Klik kanan pada src => new => other => JFrame => buat namanya => finish
2. Klik JLabel pada palette, tarik ke kanvas dan buat text nya "OPERATOR ARITMATIKA", "Bilangan 1", "Bilangan 2", "Operator" dan "Hasil"
3. Klik JTextField pada palette, tarik ke kanvas. Kotak di samping Bilangan 1 diganti nama variabelnya sebagai "textField". Kotak di samping Bilangan 2 diganti nama variabelnya sebagai "txtfield2". Sedangkan kotak di samping Hasil diganti nama variabelnya dengan "txthasil"
4. Klik JComboBox, tarik ke kanvas disamping Operator dan diganti nama variabelnya dengan comboBox. Klik kotak combo box, pilih menu model pada properties dan buat simbol "+", "-", "*", "/", "%" .

5. Klik JButton, tarik ke kanvas di samping combo box, buat text nya "Proses" dan ubah nama variabelnya sebagai "btnProses"

```
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 2.2

6. Pada kode program kita buat program pada gambar 2.2 agar program hanya bisa berjalan jika Bilangan 1 dan Bilangan 2 dan berisi angka.

```
JButton btnProses = new JButton("Proses");  
btnProses.setBounds(225, 142, 85, 16);  
btnProses.addActionListener(new ActionListener() {  
    int hasil;  
    public void actionPerformed(ActionEvent e) {  
        if(textField.getText () .trim() .isEmpty())  
            {pesanPeringatan ("Bilangan 1 harus diisi");}  
        else if (txtfield2.getText () .trim() .isEmpty () ) { pesanPeringatan ("Bilangan 2 harus diisi") ;}  
        else {  
            try {  
                int a = Integer.parseInt(textField.getText());  
                int b = Integer.parseInt(txtfield2.getText());  
                int c = comboBox.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;}  
            catch (NumberFormatException ex){  
                pesanError ("Bilangan 1 dan Bilangan 2 harus angka");}  
            txthasil.setText(String.valueOf(hasil));  
        }  
    }  
});
```

Gambar 2.3

7. Pada kode program selanjutnya kita buat program yang apabila Bilangan 1 tidak diisi maka akan otomatis keluar output "Bilangan 1 harus diisi". Lanjut kita buat program yang apabila Bilangan 2 tidak diisi maka akan otomatis keluar output "Bilangan 2 harus diisi"
8. Jika kedua bilangan sudah diisi dengan angka, maka kita deklarasikan variabel a sebagai integer dengan mengambil variabel textField pada kanvas sebelumnya

9. Kita deklarasikan variabel b sebagai integer dengan mengambil variabel txtfield2 pada kanvas sebelumnya
10. Kita deklarasikan variabel c sebagai integer dengan mengambil nilai variabel comboBox pada kanvas dengan memilih model yang telah kita masukkan sebelumnya
11. Lalu kita buat program if untuk operator aritmatika seperti gambar 2.3, yang mana angka 0 sampai 4 yang kita pilih merepresentasikan operator yang kita pilih. Program akan menjalankan perhitungan sesuai dengan operator yang kita pilih
12. Hasil dari operator aritmatika akan otomatis keluar dibagian txthasil
13. Jika salah satu atau bahkan kedua bilangan 1 dan 2 tidak berupa angka, maka akan otomatis keluar output "Bilangan 1 dan Bilangan 2 Harus Angka"

2.1.2 Contoh Output



Gambar 2.4

OPERATOR ARITMATIKA

Bilangan 1	20
Bilangan 2	4
Operator	-
Hasil	16

Proses

Gambar 2.5

OPERATOR ARITMATIKA

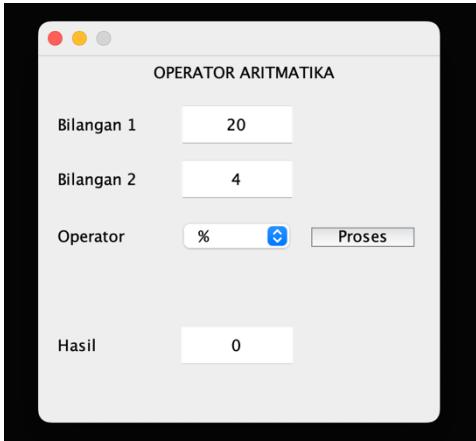
Bilangan 1	20
Bilangan 2	4
Operator	*
Hasil	80

Gambar 2.6

OPERATOR ARITMATIKA

Bilangan 1	20
Bilangan 2	4
Operator	/
Hasil	5

Gambar 2.7



Gambar 2.8

BAB III

KESIMPULAN

3.1 Ringkasan Hasil Praktikum

Praktikum ini memberikan pemahaman yang komprehensif mengenai proses pembuatan aplikasi berbasis Graphical User Interface (GUI) dengan menggunakan Java Swing di Eclipse. Melalui tahapan yang dilakukan, mahasiswa belajar merancang antarmuka, menempatkan komponen seperti label, text field, combo box, dan tombol, serta memahami bagaimana setiap elemen tersebut berperan dalam membangun interaksi antara pengguna dan program. Pengetahuan ini menjadi fondasi penting bagi mahasiswa untuk mengembangkan aplikasi desktop yang lebih kompleks di tahap berikutnya.

Selain fokus pada desain tampilan, praktikum ini juga menekankan keterkaitan antara antarmuka dan logika program melalui penggunaan event handling. Mahasiswa mempelajari bagaimana input yang diberikan pengguna diproses, bagaimana operasi aritmatika dijalankan sesuai operator yang dipilih, dan bagaimana hasil perhitungan ditampilkan kembali pada antarmuka. Proses tersebut membantu mahasiswa memahami alur kerja aplikasi secara menyeluruh, mulai dari penerimaan data, pemrosesan, hingga keluaran yang ditampilkan.

Secara keseluruhan, praktikum ini berhasil meningkatkan kemampuan mahasiswa dalam melakukan validasi input, mencegah kesalahan data, dan memberikan pesan peringatan yang jelas melalui JOptionPane. Keterampilan ini sangat penting dalam pengembangan perangkat lunak karena mendukung terciptanya aplikasi yang stabil, akurat, dan mudah digunakan. Dengan pemahaman mengenai konsep GUI, logika program, dan error handling, mahasiswa memiliki landasan yang kuat untuk melanjutkan pengembangan aplikasi di tingkat yang lebih lanjut.

DAFTAR PUSTAKA

- [1] Oracle, *The Java® Tutorials — Primitive Data Types*. [Online]. Available:
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- [2] GeeksforGeeks, *Java Data Types – GeeksforGeeks*. [Online]. Available:
<https://www.geeksforgeeks.org/data-types-in-java>