

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
PENERAPAN GRAPHICAL USER INTERFACE (GUI) DALAM JAVA



disusun Oleh:
Annisa Layli Ramadhani
2511532024
Informatika Kelas B

Dosen Pengampu: Wahyudi. Dr.. S.T.M.T
Asisten Pratikum: Rahmad Dwirizki Olders

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena dengan rahmat-Nya laporan praktikum Algoritma dan Pemrograman tanggal 24 November 2025 dapat diselesaikan. Laporan ini membahas materi tentang Penerapan Graphical User Interface (GUI) dalam Java sebagai dasar penting dalam memahami pemrograman. Penulis mengucapkan terima kasih kepada dosen, asisten, dan rekan-rekan yang telah membantu selama praktikum. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran sangat diharapkan. Semoga laporan ini bermanfaat bagi pembaca.

Padang, 24 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang	4
1.2 Tujuan	4
1.3 Manfaat	5
BAB II PEMBAHASAN.....	6
2.1 Dasar Teori	6
2.1.1 Window Builder	6
2.1.2 Java Swing	7
2.1.3 Application Window	7
2.1.4 Komponen JFrame	8
2.1.5 Event Handling.....	9
2.1.6 Validasi Input	10
2.2 Langkah Kerja.....	10
2.2.1 Pembuatan <i>Package</i> dan <i>Class</i> Pekan 9	10
2.2.2 <i>Commit</i> dan <i>Push</i> ke GitHub.....	17
2.2.3 Analisis Hasil Program	18
BAB III PENUTUP.....	20
3.1 Kesimpulan	20
3.2 Saran.....	20
DAFTAR PUSTAKA	21

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital saat ini, kemampuan pemrograman menjadi salah satu kompetensi penting, baik di dunia akademik maupun industri. Salah satu bahasa pemrograman yang banyak digunakan adalah Java karena sifatnya yang platform-independen, berorientasi objek, dan didukung oleh berbagai lingkungan pengembangan seperti Eclipse IDE. Pembuatan antarmuka grafis (GUI – Graphical User Interface) menjadi aspek penting dalam praktik pemrograman, karena memungkinkan interaksi visual antara program dan pengguna. Pembuatan kalkulator GUI sederhana menggunakan Java dan Eclipse IDE menjadi proyek yang tepat bagi pemula untuk memahami dasar-dasar pemrograman dan pengembangan antarmuka. Proyek ini memiliki kompleksitas yang terjangkau, karena mencakup operasi dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian, sekaligus memungkinkan penerapan elemen GUI seperti tombol, input/output, layout, dan event handling. Selain itu, proyek ini juga memfasilitasi penerapan konsep java secara langsung serta pemanfaatan Eclipse sebagai lingkungan pengembangan, sehingga relevan untuk pembelajaran mandiri bagi pelajar, mahasiswa, maupun programmer pemula. Pembuatan kalkulator GUI ini juga menyiapkan pengguna untuk memahami dasar-dasar pengembangan aplikasi yang lebih kompleks di masa depan.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.2.1 Menjelaskan proses perancangan dan pengembangan kalkulator GUI menggunakan Java dan Eclipse IDE.
- 1.2.2 Memberikan pemahaman tentang penerapan konsep pemrograman berorientasi objek.

- 1.2.3 Menunjukkan penggunaan komponen GUI seperti tombol, field input, dan event handling.

1.3 Manfaat

Manfaat dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.3.1 Menjadi dokumentasi proses pengembangan aplikasi sebagai referensi bagi pelajar, mahasiswa, atau programmer pemula.
- 1.3.2 Meningkatkan pemahaman tentang pemrograman java dan pengembangan antarmuka grafis.
- 1.3.3 Melatih kemampuan analisis, perencanaan, dan penataan program yang terstruktur.

BAB II

PEMBAHASAN

2.1 Dasar Teori

Pada pembuatan aplikasi berbasis GUI menggunakan java, diperlukan pemahaman yang jelas mengenai konsep dan teknologi yang mendukung proses perancangannya. Dasar teori menjadi landasan penting yang menjelaskan prinsip, alat, dan mekanisme yang digunakan dalam pengembangan aplikasi. Dengan memahami teori tersebut, proses pembuatan program dapat dilakukan secara lebih terarah dan sesuai dengan kaidah pemrograman yang baik. Oleh karena itu, bagian ini membahas beberapa konsep utama yang berkaitan dengan penggunaan Window Builder, Java Swing, Application Window beserta komponennya, serta mekanisme event handling, validasi input, dan exception handling yang menjadi dasar dalam pembuatan aplikasi GUI pada program ini.

2.1.1 Window Builder

Window builder merupakan sebuah plugin pada Eclipse IDE yang digunakan untuk merancang antarmuka grafis (GUI) secara visual pada bahasa pemrograman Java. Dengan tool ini, pengembang dapat membuat tampilan aplikasi secara drag and drop tanpa harus menuliskan seluruh kode secara manual. Window builder mendukung berbagai komponen GUI seperti label, tombol, text field, panel, serta komponen swing lainnya.

Penggunaan window builder mempermudah proses desain karena setiap perubahan yang dilakukan pada tampilan akan otomatis menghasilkan kode java secara terstruktur. Hal ini membantu pengembang bekerja lebih cepat, mengurangi kesalahan penulisan kode, dan mempermudah pemahaman terhadap struktur GUI. Window builder juga menyediakan fitur pengaturan tata letak (layout), pengelolaan properti komponen,

serta integrasi langsung dengan logika program yang ditambahkan didalam kode.

2.1.2 Java Swing

Java Swing adalah library bawaan java yang digunakan untuk membuat antarmuka grafis atau GUI (Graphical User Interface). Swing menyediakan berbagai komponen visual yang dapat digunakan untuk membangun aplikasi desktop secara interaktif. Library ini bersifat lightweight, sehingga komponen yang disediakan tidak bergantung pada tampilan sistem operasi dan dapat berfungsi secara konsisten di berbagai platform.

Swing juga mendukung pengaturan tata letak, pengolahan event, serta kustomisasi tampilan komponen, sehingga memungkinkan pengembang membuat aplikasi yang lebih fleksibel dan mudah digunakan. Dengan kemampuannya yang luas, Java Swing menjadi salah satu pilihan utama dalam pengembangan GUI berbasis Java.

2.1.3 Application Window

Application window merupakan jendela utama yang menjadi wadah bagi seluruh komponen antarmuka pada aplikasi kalkulator. Pada pengembangan menggunakan Java, application window biasanya dibuat dengan memanfaatkan kelas seperti JFrame dari library Swing. Window ini berfungsi sebagai area kerja utama tempat menampilkan tombol-tombol operasi, field input, dan label hasil perhitungan. Desain window harus memperhatikan aspek kenyamanan pengguna, seperti ukuran yang proposional, tata letak komponen yang rapi, serta warna dan font yang mudah dibaca. Selain itu, application window juga bertanggung jawab dalam menangani interaksi pengguna, misalnya menanggapi event klik tombol dan memperbarui tampilan hasil perhitungan secara real-time. Dengan begitu,

window ini tidak hanya berperan sebagai penampung visual, tetapi juga sebagai pusat kontrol dari seluruh fungsi aplikasi, sehingga memastikan aplikasi berjalan secara efisien dan interaktif.

2.1.4 Komponen JFrame

Dalam pembangunan antarmuka grafis pada java, JFrame menjadi elemen utama yang berfungsi sebagai jendela utama tempat seluruh komponen aplikasi ditempatkan. Untuk memahami struktur tampilan program, penting untuk mengetahui komponen-komponen yang digunakan di dalam JFrame.

1. JLabel

JLabel adalah komponen yang digunakan untuk menampilkan teks statis pada antarmuka. Fungsinya sebagai penanda atau keterangan bagi pengguna, seperti label untuk input “Bilangan 1”, “Bilangan 2”, “Operator”, dan “Hasil”. JLabel tidak bisa diedit oleh pengguna dan hanya berfungsi sebagai informasi visual.

2. JTextField

JTextField merupakan komponen tempat pengguna memasukkan data dalam bentuk teks. Pada program, JTextField digunakan untuk menginput bilangan pertama dan kedua, serta menampilkan hasil operasi. Komponen ini dapat dibaca oleh program untuk mengambil nilai input dari pengguna.

3. JButton

JButton adalah tombol yang dapat diklik pengguna untuk menjalankan suatu aksi. Pada program, tombol “Proses” digunakan untuk memicu perhitungan berdasarkan input yang diberikan. Event klik tombol ini ditangani melalui ActionListener.

4. JPanel

JPanel adalah wadah atau kontainer yang menampung berbagai komponen GUI lain. Dalam program ini, JPanel berfungsi sebagai area utama tempat penempatan semua komponen seperti label, tombol, text field, dan combo box. JPanel juga mendukung pengaturan warna latar dan tata letak.

5. JComboBox

JComboBox merupakan komponen dropdown yang menyediakan daftar pilihan kepada pengguna. Pada program ini, JComboBox digunakan untuk memilih operator aritmatika seperti +, −, *, /, dan %. Pilihan yang dipilih akan digunakan dalam proses perhitungan.

2.1.5 Event Handling

Event handling adalah mekanisme dalam Java Swing yang digunakan untuk menangani aksi atau interaksi yang dilakukan oleh pengguna terhadap komponen antarmuka. event handling diterapkan saat pengguna menekan tombol Proses. Aksi tersebut ditangkap oleh ActionListener, yaitu antarmuka yang bertugas merespons suatu event. Ketika tombol diklik, metode actionPerformed() akan dijalankan untuk melakukan serangkaian proses, mulai dari membaca input pengguna melalui JTextField, memeriksa operator yang dipilih pada JComboBox, hingga melakukan perhitungan aritmatika. Selain itu, event handling juga digunakan untuk menampilkan pesan peringatan atau kesalahan melalui JOptionPane apabila terjadi input tidak valid. Dengan mekanisme ini, aplikasi dapat merespons tindakan pengguna secara interaktif dan menjalankan fungsi program sesuai kebutuhan.

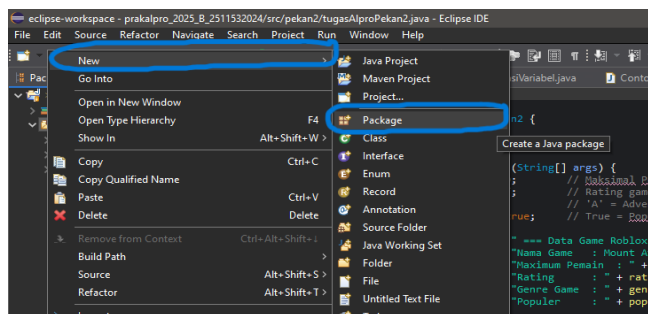
2.1.6 Validasi Input

Validasi input pada program dilakukan untuk memastikan bahwa data yang dimasukkan pengguna sudah benar sebelum dihitung. Program terlebih dahulu memeriksa apakah kolom Bilangan 1 dan Bilangan 2 kosong, dan jika kosong akan menampilkan peringatan melalui pesanPeringatan(). Selain itu, validasi juga dilakukan saat mengonversi input menjadi angka menggunakan Integer.parseInt(). Jika pengguna memasukkan karakter non-angka, program menangani kesalahan tersebut dengan menampilkan pesan melalui pesanError(). Dengan validasi ini, program dapat mencegah kesalahan perhitungan dan memastikan input yang digunakan sudah valid.

2.2 Langkah Kerja

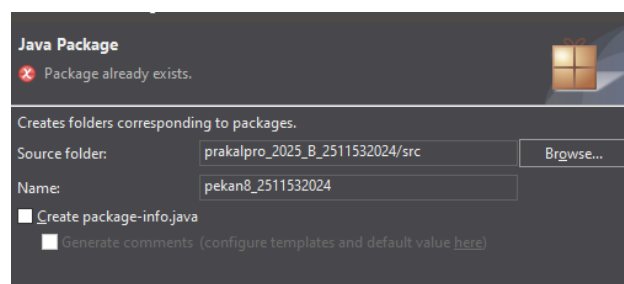
2.2.1 Pembuatan *Package* dan *Class* Pekan 9

1. Buka aplikasi eclipse, lalu buat package baru pada src klik kanan, lalu pilih “New” dan klik tulisan Package.



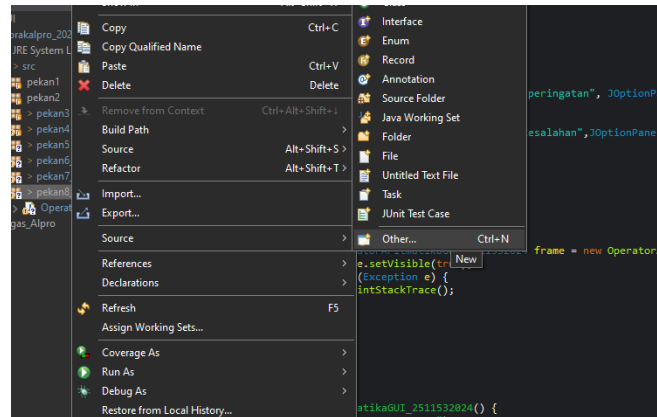
2.2.1 pembuatan package

2. Kemudian buat nama packagenya tanpa spasi, huruf kapital, karakter khusus. Beri nama “pekan9_2511532024”. Lalu klik finish.



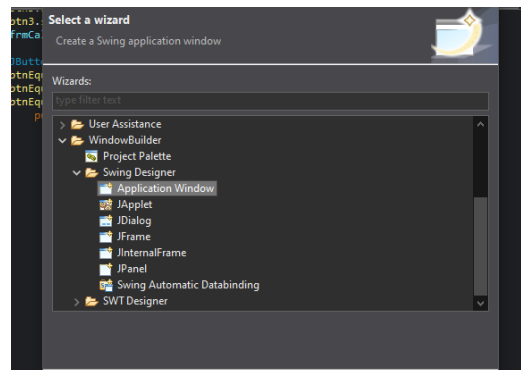
2.2.2 pembuatan package

3. Ketika package sudah jadi. Klik kanan pada bagian “pekan9_2511532024”. Lalu klik “new” dan pilih bagian “other” untuk memulai membuat program.



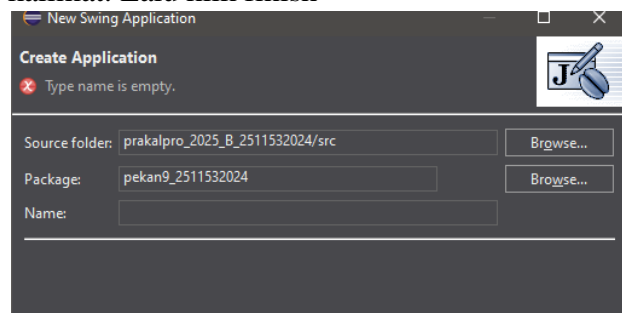
2.2.3 pembuatan file design

4. Kemudian pilih bagian “Application Window”. Dan setelah itu klik next.



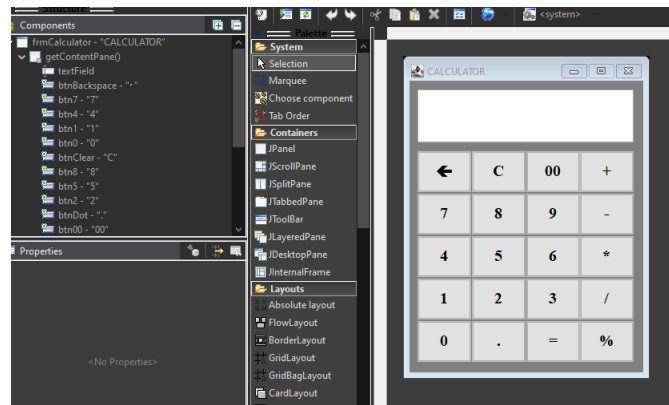
2.2.4 pembuatan file design

5. Setelah itu muncul Create Application Window, buat nama file dengan ketentuan nama harus capslock diawal kalimat. Lalu klik finish



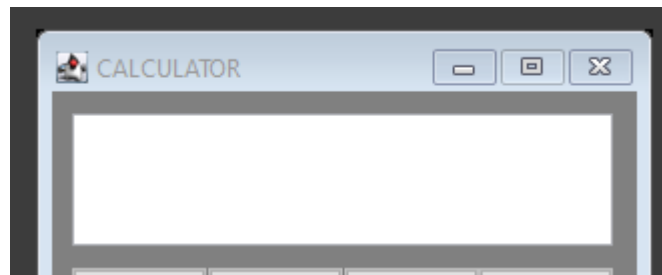
2.2.5 pembuatan file design

8. Setelah mengubah titlenya kita bisa langsung mengatur ukuran, background atau layout pada desain program.



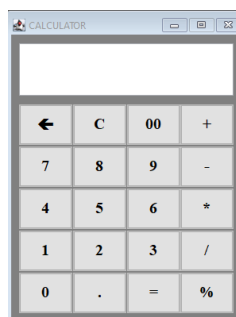
2.2.9 program kalkulator

9. Kemudian kita bisa mulai memasukkan komponen-komponennya seperti JTextField. Dan kita bisa mengedit JTextFieldnya sesuai yang kita inginkan.



2.2.10 program kalkulator

10. Setelah kita menambah komponen JTextField, kita menambahkan komponen JButton sebanyak 20 buah yang diatur menjadi 4 x 5. Kemudian beri label pada tombol tersebut.



2.2.11 program kalkulator

11. Setelah menambahkan komponen tombol tadi, langsung kita mengklik 2x pada bagian tombol angka 0-9 dan angka 00 untuk di beri kode agar bisa digerakkan oleh Action Listener.

```

JButton btn1 = new JButton("1");
btn1.setForeground(Color.BLACK);
btn1.setBackground(Color.LIGHT_GRAY);
btn1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

2.2.12 program kalkulator

12. kemudian masukkan kode berikut agar angka keluar pada bagian textfield ketika tombol ditekan. Lakukan pada tombol angka 0-9 dan angka 00.

```

    public void actionPerformed(ActionEvent e) {
        String number=textField.getText()+btn1.getText();
        textField.setText(number);
    }
});

```

2.2.13 program kalkulator

13. Kemudia klik 2x pada bagian tombol “C” untuk diberi kode agar bisa digunakan oleh Action Listener.

```

btnClear.setBackground(Color.LIGHT_GRAY);
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText(null);
    }
});

```

2.2.14 program kalkulator

14. Setelah itu klik 2x pada bagian tombol “←” untuk diberi kode agar tombol bisa digunakan.

```

btnBackspace.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String backSpace=null;
        if(textField.getText().length()>0) {
            StringBuilder str=new StringBuilder(textField.getText());
            str.deleteCharAt(textField.getText().length()-1);
            backSpace=str.toString();
            textField.setText(backSpace);
        }
    }
});

```

2.2.15 program kalkulator

15. Kemudian untuk bagian operator “+ - * / %”. Tambahkan variabel baru.

```
double first;
double second;
double result;
String operation;
String answer;
```

2.2.17 program kalkulator

16. Setelah itu tambahkan kode pada bagian tombol +, -, *, / , % agar bisa dijalankan dengan cara klik 2x pada setiap tombol operator.

```
public void actionPerformed(ActionEvent e) {
    first=Double.parseDouble(textField.getText());
    textField.setText("");
    operation="+";
}
});
```

2.2.18 program kalkulator

```
public void actionPerformed(ActionEvent e) {
    first=Double.parseDouble(textField.getText());
    textField.setText("");
    operation="-";
}
});
```

2.2.19 program kalkulator

```
public void actionPerformed(ActionEvent e) {
    first=Double.parseDouble(textField.getText());
    textField.setText("");
    operation="*";
}
});
```

2.2.20 program kalkulator

```
public void actionPerformed(ActionEvent e) {
    first=Double.parseDouble(textField.getText());
    textField.setText("");
    operation="/";
}
});
```

2.2.21 program kalkulator

```
public void actionPerformed(ActionEvent e) {
    first=Double.parseDouble(textField.getText());
    textField.setText("");
    operation="%";
}
});
```

2.2.22 program kalkulator

17. Kemudian pada tombol “.” Tambahkan kode berikut. Agar tombol bisa bekerja.

```
public void actionPerformed(ActionEvent e) {
    String number=textField.getText()+btnDot.getText();
    textField.setText(number);
}
});
```

2.2.23 program kalkulator

18. Kemudian pada tombol “=” tambahkan kode berikut agar bisa bekerja.

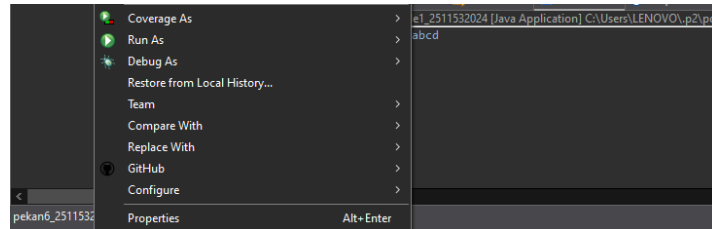
```
public void actionPerformed(ActionEvent e) {
    String answer;
    second=Double.parseDouble(textField.getText());
    if(operation=="+") {
        result=first+second;
        answer=String.format("%.2f", result);
        textField.setText(answer);
    }
    else if(operation=="-") {
        result=first-second;
        answer=String.format("%.2f", result);
        textField.setText(answer);
    }
    else if(operation=="*") {
        result=first*second;
        answer=String.format("%.2f", result);
        textField.setText(answer);
    }
    else if(operation=="/") {
        result=first/second;
        answer=String.format("%.2f", result);
        textField.setText(answer);
    }
    else if(operation=="%") {
        result=first%second;
        answer=String.format("%.2f", result);
        textField.setText(answer);
    }
}
});
```

2.2.24 program kalkulaator

19. Setelah semua tombol diberik kode, jalankan program tersebut dan cek semua tombol apakah bekerja dengan baik atau tidak.

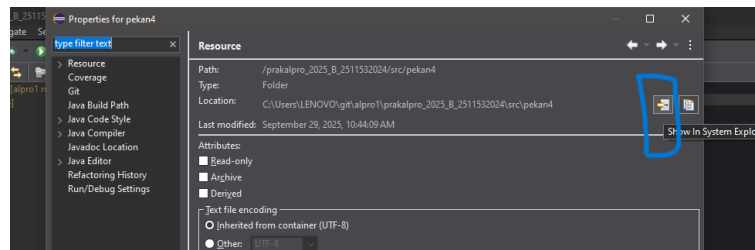
2.2.2 Commit dan Push ke GitHub

1. Klik kanan pada bagian “pekan9_2511532024”.
Kemudian pilih bagian “properties”.



2.2.25 proses commit dan push ke GitHub

2. Setelah itu keluar opsi properties. Kemudian klik “Show In System Explorer”.



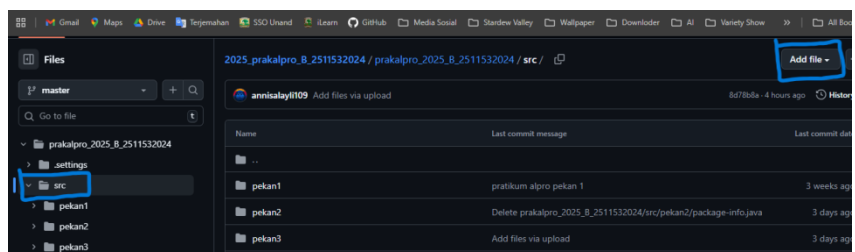
2.2.26 proses commit dan push ke github

3. Setelah itu akan terbuka folder file tersebut di file explorer.



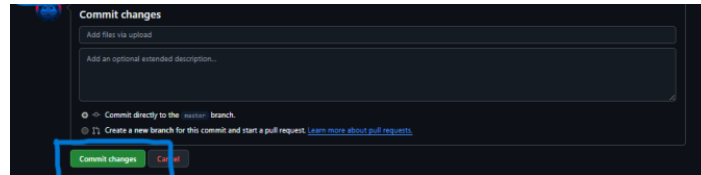
2.2.27 proses commit dan push ke GitHub

4. Kemudian buka akun GitHub dan buka pada bagian src. Dan klik pada bagian “add file”. Dan pilih “upload file”.



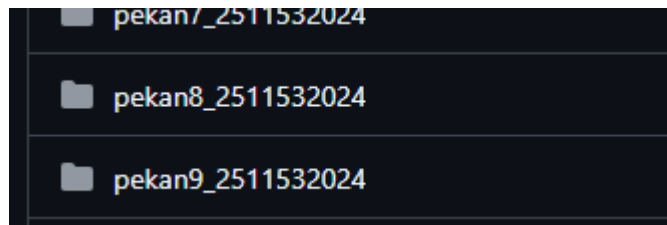
2.2.28 proses commit dan push ke GitHub

- Setelah itu tarik file “pekan9_2511532024” dari file explorer ke bagian upload file dan kemudian tunggu loading file diunggah ke GitHub. Dan setelah itu klik “Commit changes”.



2.2.29 proses commit dan push ke GitHub

- Kemudian cek akun GitHub untuk melihat apakah file yang diupload sudah masuk atau belum. Jika sudah maka upload sudah selesai.



2.2.30 proses commit dan push ke GitHub

2.2.3 Analisis Hasil Program

- Ketika pengguna memasukkan nilai $5 + 2$. Maka hasil dari program adalah 12.00



2.2.31 analisis hasil program

- Ketika pengguna memasukkan nilai $22 - 9$. Maka hasil dari program adalah 13.00



2.2.32 analisis hasil program

3. Ketika pengguna menginputkan nilai $4 * 11$. Maka hasil dari program adalah 44.00

A screenshot of a program's output window. It features a single line of text, "44.00", displayed in a blue font. The text is positioned on the left side of a white rectangular area, which is enclosed by a thin blue border.

2.2.33 analisis hasil program

4. Ketika pengguna menginputkan nilai $29 / 3$. Maka hasil dari program adalah 9.67

A screenshot of a program's output window. It features a single line of text, "9.67", displayed in a blue font. The text is positioned on the left side of a white rectangular area, which is enclosed by a thin blue border.

2.2.34 analisis hasil program

5. Ketika pengguna menginputkan nilai $91 \% 6$. Maka hasil dari program adalah 1.00

A screenshot of a program's output window. It features a single line of text, "1.00", displayed in a blue font. The text is positioned on the left side of a white rectangular area, which is enclosed by a thin blue border.

2.2.35 analisis hasil program

BAB III

PENUTUP

3.1 Kesimpulan

Pembuatan aplikasi kalkulator berbasis GUI dengan menggunakan Java dan Eclipse IDE menunjukkan bahwa konsep pemrograman berorientasi objek serta pengelolaan komponen antarmuka grafis dapat diterapkan secara efektif untuk menghasilkan aplikasi yang interaktif dan mudah digunakan. Melalui pemanfaatan komponen seperti jendela utama, tombol operasi, dan field input, aplikasi mampu menjalankan fungsi perhitungan dasar dengan respon yang baik terhadap tindakan pengguna. Proses pengembangan ini juga memberikan pemahaman mengenai struktur program yang terorganisasi, penggunaan event handling untuk menangani interaksi, serta pentingnya perancangan antarmuka yang sederhana namun fungsional. Secara keseluruhan, proyek ini menjadi dasar pembelajaran yang baik bagi pemula untuk memahami praktik pengembangan aplikasi GUI dalam Java.

3.2 Saran

Untuk pengembangan lebih lanjut, aplikasi dapat ditingkatkan dengan menambahkan fitur lanjutan seperti operasi scientific, validasi input yang lebih ketat, serta desain antarmuka yang lebih modern dan responsif. Selain itu, penggunaan layout manager yang lebih fleksibel dapat membantu menghasilkan tampilan yang lebih rapi dan adaptif. Dokumentasi kode juga sebaiknya diperjelas agar memudahkan proses pemeliharaan dan memungkinkan pengembangan lanjutan oleh pengguna lainnya. Melalui perbaikan dan pengembangan berkelanjutan, aplikasi kalkulator ini dapat menjadi proyek pembelajaran yang lebih komprehensif dan bermanfaat.

DAFTAR PUSTAKA

- [1] Oracle, “*Trail: Creating a GUI With JFC/Swing (The Java™ Tutorials)*,” Oracle Documentation. <https://docs.oracle.com/javase/tutorial/uiswing/>
- [2] Oracle, “*How to Write an Action Listener*,” Java Tutorials. <https://docs.oracle.com/javase/tutorial/uiswing/events/actionlistener.html>
- [3] Oracle, “*Class JOptionPane*,” Java Platform SE Documentation. <https://docs.oracle.com/javase/8/docs/api/javax/swing/JOptionPane.html>
- [4] Oracle, “*Class JPasswordField*,” Java Platform SE Documentation. <https://docs.oracle.com/javase/8/docs/api/javax/swing/JPasswordField.html>