

**LAPORAN PRAKTIKUM ALGORITMA DAN
PEMROGRAMAN**
PEMROGRAMAN GUI 1



Disusun Oleh :
DIKA GIWANDA
NIM 2311533025
Dosen Pengampu :
DR. WAHYUDI, S.T, M.T
Asisten Praktikum :
JOVANTRI IMMANUEL GULO

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
PADANG, NOVEMBER 2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat, hidayah, dan karunianya sehingga laporan praktikum pemrograman *GUI (Graphical User Interface)* ini dapat diselesaikan dengan baik dan tepat waktu. Sholawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW, keluarga, sahabat, dan seluruh pengikutnya hingga akhir zaman.

Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban, refleksi, serta dokumentasi atas kegiatan praktikum yang telah dilaksanakan, dengan fokus pada penerapan konsep pemrograman berbasis antarmuka grafis menggunakan bahasa pemrograman Java. Pemrograman *GUI* merupakan salah satu aspek penting dalam pengembangan perangkat lunak modern, karena memungkinkan pengguna berinteraksi dengan sistem secara lebih intuitif dan visual melalui elemen-elemen seperti tombol, *label*, *text field*, dan jendela. Dalam praktikum ini, penulis mempelajari penggunaan *library Swing* dan atau *JavaFX* untuk membangun aplikasi desktop yang responsif, sesuai dengan prinsip-prinsip desain antarmuka yang baik.

Penyusunan laporan ini tidak mungkin terwujud tanpa bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang tulus kepada dosen pembimbing yang telah memberikan arahan, bimbingan, dan masukan yang berharga selama proses praktikum, serta kepada rekan-rekan mahasiswa yang turut berkontribusi dalam diskusi dan kolaborasi.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi pembaca, khususnya dalam memahami dasar-dasar dan aplikasi pemrograman *GUI*.

Akhir kata, penulis berharap semoga ilmu yang diperoleh melalui praktikum ini dapat diaplikasikan secara positif dalam pengembangan perangkat lunak yang di masa depan.

Padang, 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	.ii
DAFTAR ISI.....	.iii
BAB 1 PENDAHULUAN.....	.1
1.1 Pengertian Praktikum.....	.1
1.2 Tujuan Praktikum.....	.1
1.3 Persyaratan Praktikum.....	.1
1.4 Tempat dan Waktu Praktikum.....	.2
1.5 Manfaat Praktikum.....	.2
BAB 2 PEMBAHASAN PRAKTIKUM.....	.3
2.1 Pengertian Pemrograman GUI.....	.3
2.3 Langkah Langkah Pengerjaan Praktikum Pekan 8.....	.4
BAB 3 PENUTUP.....	.18
3.1 Kesimpulan.....	.18
3.2 Saran.....	.18
DAFTAR PUSTAKA.....	.19

BAB 1

PENDAHULUAN

1.1 Pengertian Praktikum

Praktikum Java adalah kegiatan pembelajaran yang dilakukan di laboratorium komputer untuk mengasah keterampilan mahasiswa dalam memahami serta menerapkan konsep pemrograman Java. Kegiatan ini tidak hanya menekankan pada penguasaan teori,tetapi juga pada latihan penyusunan kode program, pengujian, hingga analisis hasil eksekusi. Praktikum dipandang sebagai wahana latihan yang menjembatani pemahaman konseptual dengan kemampuan teknis pemrograman.

1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini antara lain sebagai berikut :

1. Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik.
4. Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum.
5. Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.
6. Mengetahui dan mengaplikasikan if, else, if else, dan switch.

1.3 Persyaratan Praktikum

Agar praktikum berjalan lancar, mahasiswa perlu memenuhi beberapa persyaratan berikut:

1. Telah mengikuti perkuliahan teori Pemrograman Java sebagai dasar pemahaman.

2. Membawa perlengkapan yang diperlukan, antara lain laptop atau komputer yang sudah terpasang *Java Development Kit* (JDK) dan *Integrated Development Environment* (IDE) yang direkomendasikan.
3. Mengikuti setiap sesi praktikum sesuai jadwal yang ditetapkan dan hadir minimal sesuai ketentuan program studi.
4. Mematuhi tata tertib laboratorium, termasuk menjaga keamanan data, perangkat, serta lingkungan kerja.
5. Menyusun laporan praktikum dengan format dan aturan yang telah ditetapkan dalam pedoman ini.

1.4 Waktu dan Tempat Praktikum

Pelaksanaan praktikum Java mengikuti kalender akademik yang berlaku pada program studi. Setiap sesi praktikum dilaksanakan sesuai jadwal yang ditentukan oleh dosen pengampu. Tempat kegiatan umumnya berlangsung di laboratorium komputer, namun pada kondisi tertentu dapat dilaksanakan secara mandiri dengan perangkat masing-masing, selama memenuhi syarat teknis yang ditetapkan.

1.5 Manfaat Praktikum

Manfaat praktikum bahasa Java meliputi kemampuan memahami dan menerapkan konsep dasar pemrograman seperti tipe data dan kontrol alur, melatih penggunaan kode yang modular dan dapat digunakan kembali, serta membangun pemahaman tentang sintaksis dan struktur *object oriented programming* (OOP) yang lebih mendalam. Selain itu, praktikum ini juga membantu membangun fondasi untuk mengembangkan aplikasi lintas platform yang lebih kompleks di berbagai bidang teknologi.

BAB 2

PEMBAHASAN PRAKTIKUM

2.1 Pengertian Pemrograman GUI

Pemrograman *GUI* Java adalah pengembangan antarmuka pengguna grafis (*Graphical User Interface*) untuk aplikasi Java, yang memungkinkan interaksi visual dengan komponen seperti tombol, menu, dan jendela, bukan hanya perintah teks. Ini dapat dicapai menggunakan toolkit seperti *AWT*, *Swing*, atau *JavaFX*, yang menyediakan serangkaian komponen dan fungsi untuk membuat aplikasi yang mudah digunakan dan menarik.

Konsep Utama:

- Komponen *GUI*: Elemen visual yang membangun antarmuka, seperti *JLabel* (untuk teks), *JButton* (untuk tombol), *JTextField* (untuk input teks), *JCheckBox* (untuk kotak centang), dan *JRadioButton* (untuk pilihan tunggal).
- Event *Handling*: Mekanisme untuk merespons tindakan pengguna (seperti klik tombol). Saat pengguna berinteraksi, sebuah "event" terjadi, dan kode yang disebut "*event handler*" akan dijalankan untuk memproses aksi tersebut.
- *Container*: Wadah untuk menampung komponen *GUI*, seperti *JFrame* untuk jendela utama atau *JPanel* untuk grup komponen.
- *Layout Manager*: Mengatur tata letak dan posisi komponen di dalam container untuk memastikan tampilan yang konsisten di berbagai ukuran layar dan sistem operasi.

Cara kerja:

- Komponen *visual*: Pengguna berinteraksi dengan elemen grafis seperti tombol, menu, dan ikon, yang jauh lebih intuitif daripada baris perintah teks.
- Wadah (Container): Komponen *GUI* diletakkan di dalam wadah seperti *frame/window*.

- Perpustakaan (*Libraries*): Java menyediakan perpustakaan grafis untuk membangun antarmuka:
- *AWT* (Abstract Window Toolkit): Pustaka grafis dasar Java.
- *Swing*: *Toolkit GUI* yang lebih kaya, menawarkan komponen yang lebih luas dan tampilan yang konsisten di berbagai *platform*.
- *JavaFX*: *Framework* modern untuk membangun antarmuka yang dinamis dan kaya media.
- Tata letak (*Layout*): Pengembang menggunakan layout manager untuk mengatur posisi dan ukuran komponen di dalam wadah secara fleksibel.

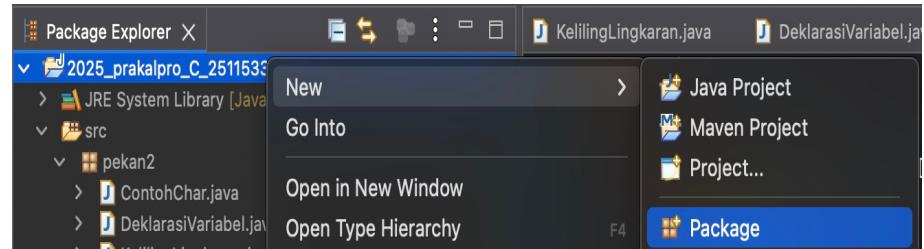
Langkah-langkah dasar:

- Impor paket: Impor paket yang diperlukan, seperti *Swing*.
- Buat jendela utama: Buat *frame* utama untuk aplikasi.
- Tambahkan komponen: Tambahkan komponen *GUI* (misalnya, tombol, label) ke dalam *frame*.
- Atur tata letak: Gunakan layout manager untuk menyusun komponen.
- Tangani kejadian: Tulis kode untuk merespons tindakan pengguna, seperti mengklik tombol.
- Tampilkan antarmuka: Pastikan antarmuka ditampilkan di layar.

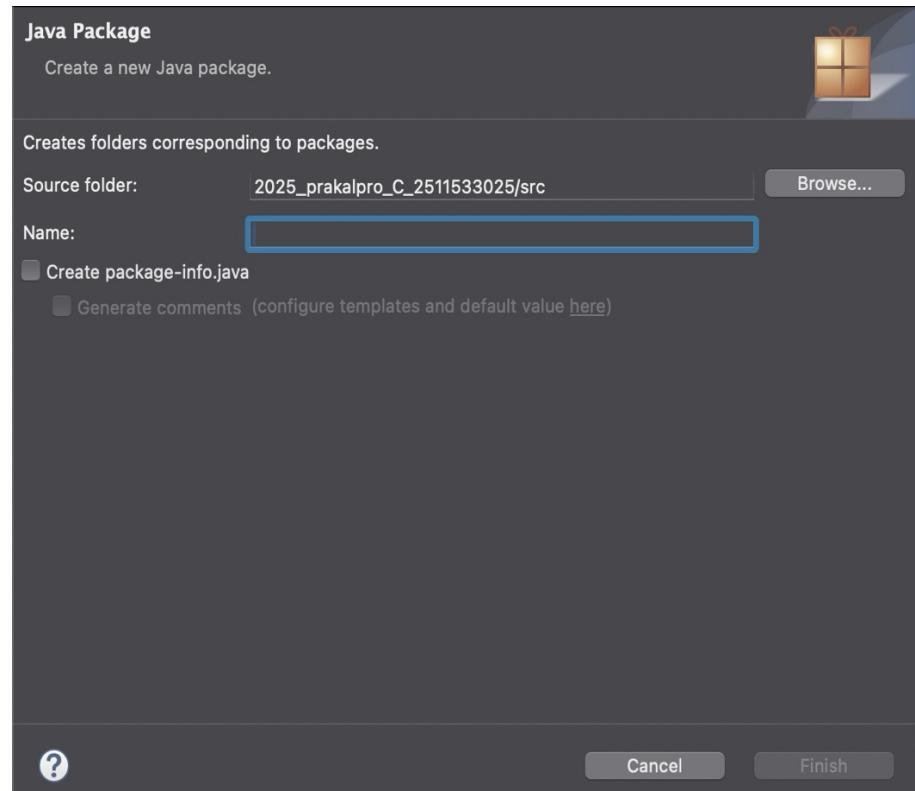
2.2 Langkah Langkah Pengerjaan Praktikum Pekan 7

Sebelum ke langkah Langkah pengerjaan contoh projek, saya akan menjelaskan terlebih dahulu gimana caranya menambahkan projek nya terlebih dahulu :

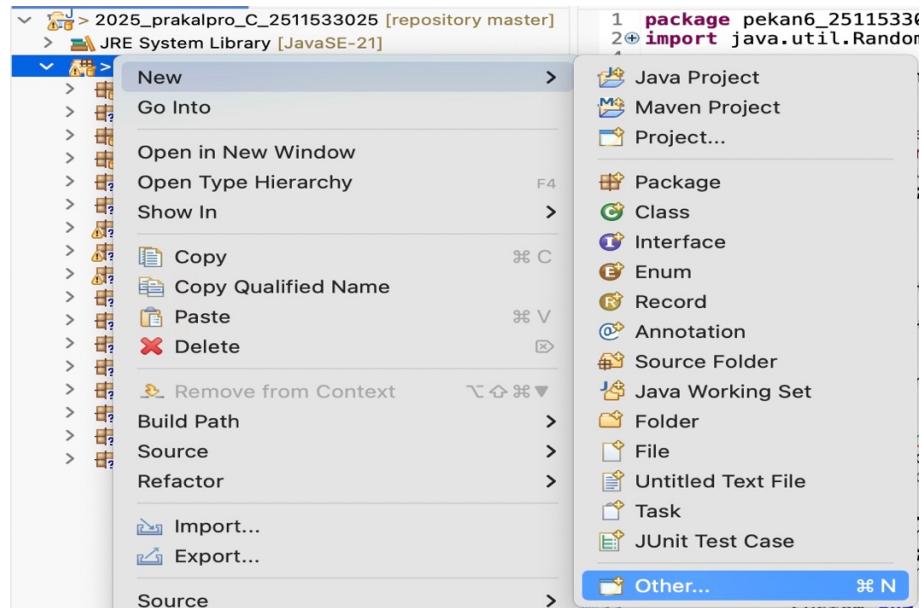
1. Setelah kalian masuk ke eclipse java dan membuat folder file projek java, click kanan pada mouse kalian , kemudian pilih *new* terus pilih *package*.



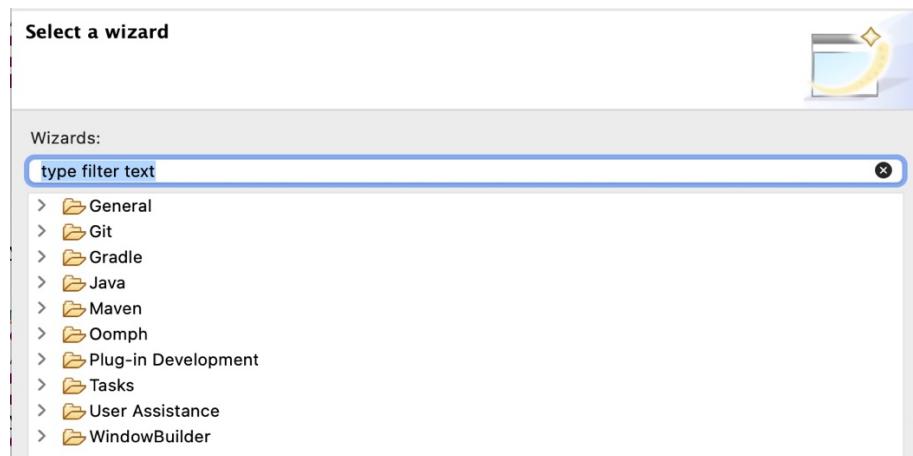
Setelah di click akan muncul halaman java package, dimana kita disuruh bikin nama untuk folder package nya, lalu pilih finish.



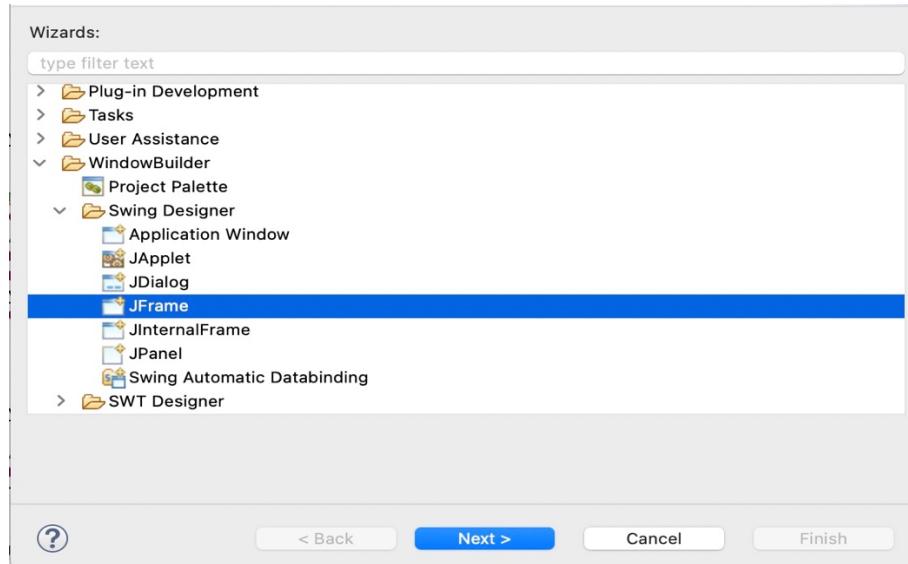
2. Jika folder *packagenya* sudah jadi, click kanan *mouse* pada folder *package* nya, lalu akan muncul pilihan, kalian pilih *new* kemudian pilih *other*.



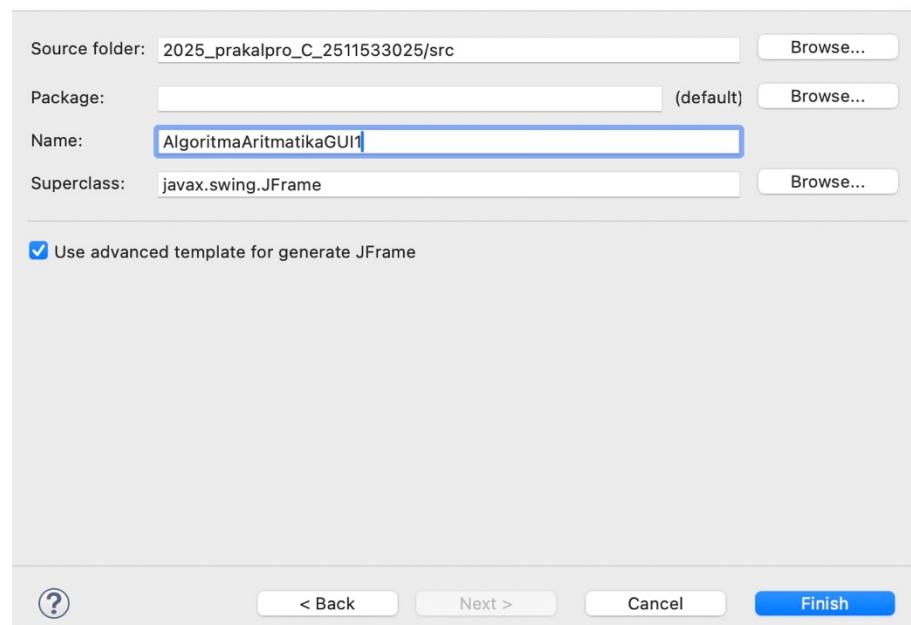
3. Setelah di click akan muncul halaman *Select a wizard*, lalu pilih *WindowBuilder*.



Kemudian akan muncul beberapa pilihan, kalian pilih yang Namanya *JFrame*.



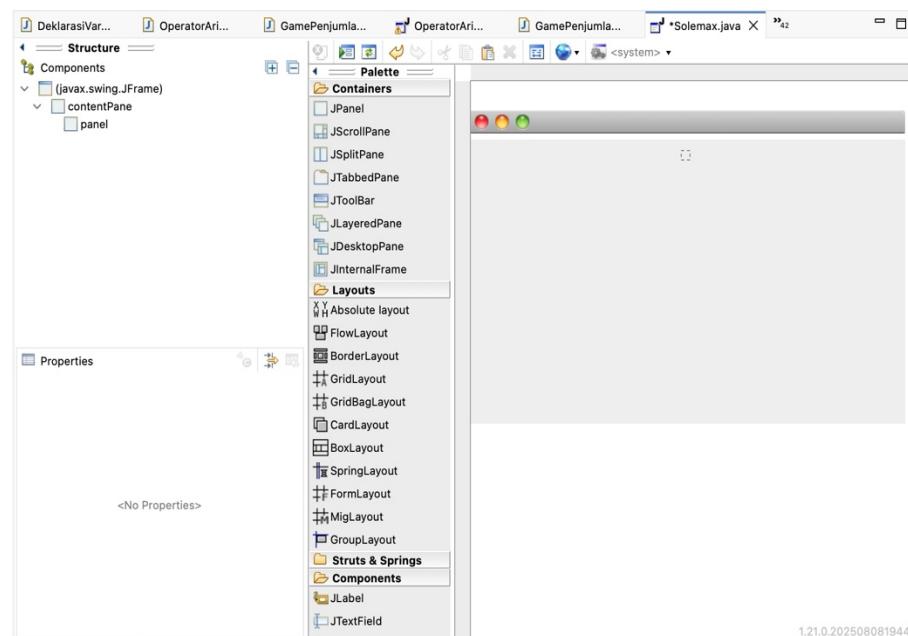
4. Lalu akan muncul halaman seperti gambar dibawah ini dimana kalian diminta memasukan nama sesui dengan projek yang ingin dibuat. Jika sudah, *click finish*.



5. Setelah selesai pembuatan folder projeknya maka akan langsung muncul halaman projek seperti gambar dibawah ini.

```
1 package pekan8_2511533025;
2
3 import java.awt.EventQueue;
4
5
6 public class OperatorAritmatikaGUI_2511533025 extends JFrame {
7
8     private static final long serialVersionUID = 1L;
9
10    JPanel contentPane;
11    JTextField txtBil1;
12    JTextField txtBil2;
13    JTextField txtHasil;
14
15
16    private void pesanPeringatan(String pesan) {
17        JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
18    }
19
20    private void PesanError(String pesan) {
21        JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
22    }
23
24
25    /**
26     * Launch the application.
27     */
28    public static void main(String[] args) {
29        EventQueue.invokeLater(new Runnable() {
30            public void run() {
31                try {
32                    OperatorAritmatikaGUI_2511533025 frame = new OperatorAritmatikaGUI_2511533025();
33                    frame.setVisible(true);
34                } catch (Exception e) {
35                    e.printStackTrace();
36                }
37            }
38        });
39    }
40
41    /**
42     * Create the frame.
43     */
44    public OperatorAritmatikaGUI_2511533025() {
45
46
47
48
49
50
51
52
53
54}
```

6. Kemudian tekan design dan akan langsung muncul halaman seperti gambar dibawah ini.



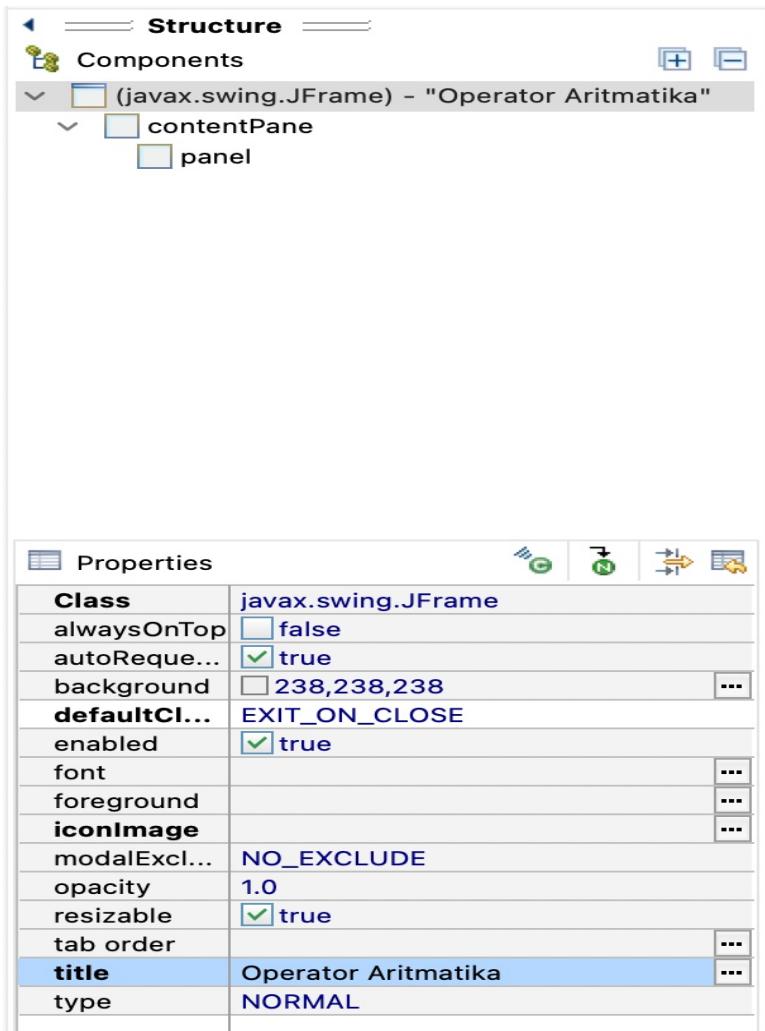
Catatan :

Jika kita ingin membuat projek baru lagi, maka cukup buat folder *class* yang baru.

- A. Baiklah karena proses menambahkan projek sudah dijelaskan, selanjutnya langkah langkah penggerjaan projek yang berjudul Operator Aritmatika:

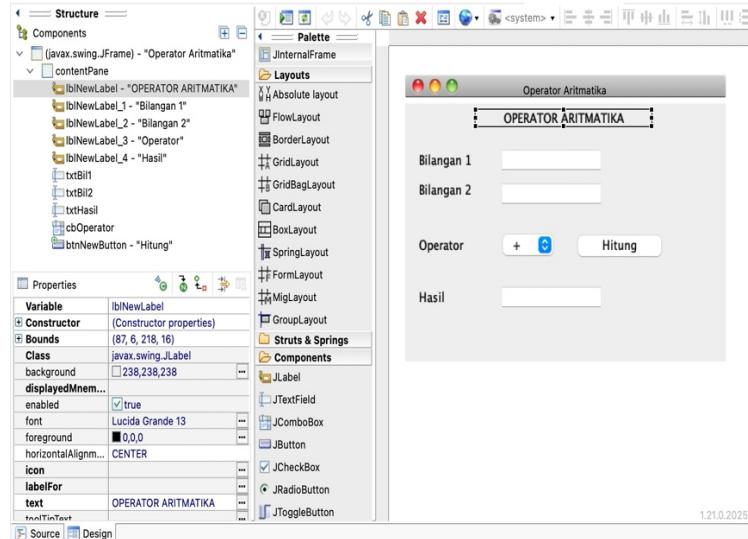
 1. Tahap awal

- Atur dulu ukuran *JFrame* sesuai dengan yang kalian inginkan
- Tekan pada bagian yang bernama (*javax.swing.JFrame*), lalu pilih *title* dan masukan nama atau judul projek ingin kita buat terus tekan *enter/return*. Seperti gambar dibawah ini:



- Berikut ini adalah tahapan lebih lanjut:
 1. Scrol kebawah pilih bagian *components*.
 2. Pilih *JLabel*, seret kursor ke *JFrame* kemudian click 1 kali, kemudian atur posisi nya sesuai dengan projek yang dibuat.
 3. Pada bagian properties pilih *horizontalAlignment*, yang awalnya *leading* ubah menjadi *center*. Jika sudah, selanjutnya pilih text, masukan nama sesuai projek yang dibuat, terus tekan *enter*.
 4. Lalu scrol ke atas cari bagian *layout* , Pilih *Absolute layout* seret kursor ke *JFrame* click 1 kali

5. Lalu scroll ke bawah lagi bagian *components* , pilih *JLabel* lagi
 6. Pada *JLabel* yang pertama kita namakan sebagai judul , sedangkan pada *JLabel* ke 2 , ke 3, ke 4, dan ke 5 sebagai nama dari proses projek yang kita buat. Dan jangan lupa memasukan nama sesuai dengan proses projek yang kita buat , caranya seperti langkah ke 3.
 7. Lalu pilih *JTextField* seret kursor ke *JFrame* kemudian click 1 kali , terus atur posisi sesuai projek yang dibuat. Ulangi sampai 2 kali, karena yang kita butuhkan sebanyak 3.
 8. Lalu cari bagian *properties* , pilih variabel terus ubah namanya sesuai dengan projek yang dibuat. Terus tekan *enter*. Ini untuk semua *JTextfield*.
 9. Selanjutnya pilih *JComboBox* pada bagian *components*, seret kursor ke *JFrame* click 1 kali dan atur posisinya.
 10. Lalu cari bagian *properties*, pilih variable ubah namanya sesuai dengan projek yang dibuat. Selanjutnya pilih model tekan titik 3, isi bagian label bernama *Combobox* model items , lalu isi sesuai dengan projek yang dibuat. Tekan *enter*.
 11. Selanjutnya pilih *JButton* pada bagian *components*, seret kursor ke *JFrame* click 1 kali dan atur posisinya.
 12. Cari bagian *properties* pilih text ubah namanya sesuai dengan projek yang dibuat. Tekan *enter*.
 13. Jika sudah selanjutnya tekan menu *Source* untuk menambahkan kode program supaya *JFrame* yang kita buat bisa jalan.
- Berikut adalah gambar bagian *Design*:



2. Tahap Pertengahan

Setelah selesai mengatur bagian *Design* kita akan mengatur kode program atau lebih tepatnya menambahkan kode program di bagian menu *Source* agar *JFrame* yang dibuat dapat dijalankan. Berikut ini adalah kode program tambahan:

- Tambahkan kode `JComboBox cbOperator = new JComboBox()` untuk menyediakan menu tarik-turun (*dropdown*) yang memungkinkan pengguna memilih satu opsi dari daftar pilihan yang telah ditentukan sebelumnya.
- Tambahkan kode `cboperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}))` untuk mengatur pilihan operator matematika yang tersedia dalam komponen *dropdown* atau *combobox* bernama `cbOperator`.
- Tambahkan kode `cboperator.setBounds(116, 126, 73, 27);` untuk mengatur posisi dan ukuran komponen antarmuka pengguna grafis (*GUI*) di Java.
- Tambahkan kode `contentPane.add(cboperator);` untuk menambahkan komponen `cboperator` ke dalam area konten utama jendela aplikasi, sehingga komponen tersebut dapat ditampilkan dan berinteraksi dengan pengguna.
- Tambahkan kode `JButton btnNewButton = new JButton("Hitung")` untuk menyediakan elemen interaktif yang dapat diklik oleh pengguna untuk memicu suatu tindakan atau perintah tertentu dalam program.
- Tambahkan kode `btnNewButton.addActionListener(new ActionListener() {` Untuk mendaftarkan objek pendengar (*listener*) yang akan merespons kejadian tertentu, seperti saat tombol diklik, item menu dipilih, atau *field* teks ditekan *Enter*.
- Tambahkan kode `int hasil;` untuk mendeklarasikan sebuah variabel dengan nama hasil yang bertipe data *int*.
- Tambahkan kode `public void actionPerformed(ActionEvent e) {` Metode ini secara otomatis dipanggil oleh sistem *Java Development Kit (JDK)* ketika suatu aksi terjadi, seperti saat tombol diklik, item menu dipilih, atau *field* teks ditekan *Enter*.
- Tambahkan kode `if (txtBill.getText().trim().isEmpty()) {` untuk memeriksa apakah bidang teks bernama `txtBill` kosong atau hanya berisi spasi/*whitespace*.
- Tambahkan kode pesan Peringatan ("Bilangan 1 Harus diisi"); untuk menampilkan sebuah pesan peringatan kepada pengguna. Pesan yang ditampilkan adalah "Bilangan 1 Harus diisi", yang mengindikasikan bahwa pengguna perlu memasukkan nilai untuk "Bilangan 1" sebelum melanjutkan proses selanjutnya dalam program tersebut. Fungsi ini biasanya digunakan untuk validasi input data.

- Tambahkan kode `} else if(txtBil2.getText().trim().isEmpty())` untuk memeriksa apakah kolom teks bernama `txtBil2` kosong atau hanya berisi spasi. validasi input dasar untuk memastikan pengguna memasukkan data yang substansial ke dalam bidang teks tertentu sebelum melanjutkan pemrosesan lebih lanjut.
- Tambahkan kode pesan Peringatan ("Bilangan 2 Harus diisi"); untuk menampilkan sebuah pesan peringatan kepada pengguna. Pesan yang ditampilkan adalah "Bilangan 2 Harus diisi", yang mengindikasikan bahwa pengguna perlu memasukkan nilai untuk "Bilangan 2" sebelum melanjutkan proses selanjutnya dalam program tersebut. Fungsi ini biasanya digunakan untuk validasi input data.
- Tambahkan kode `} else` untuk menyediakan jalur alternatif eksekusi ketika kondisi utama tidak terpenuhi.
- Tambahkan kode `try {` sebagai "area pengujian" untuk kode yang berpotensi bermasalah, memastikan stabilitas dan ketahanan aplikasi.
- Tambahkan kode `int a= Integer.valueOf(txtBil1.getText());` untuk mendapatkan teks dari komponen antarmuka pengguna (*UI*) yang bernama `txtBil1`, mengubah teks tersebut menjadi nilai *integer* (bilangan bulat), dan menyimpannya ke dalam variabel *integer* bernama `a`.
- Tambahkan kode `int b= Integer.valueOf(txtBil2.getText());` untuk mendapatkan teks dari komponen antarmuka pengguna (*UI*) yang bernama `txtBil2`, mengubah teks tersebut menjadi nilai *integer* (bilangan bulat), dan menyimpannya ke dalam variabel *integer* bernama `b`.
- Tambahkan kode `int c= cboperator.getSelectedIndex()` untuk mengambil *indeks* (posisi) item yang saat ini dipilih dari komponen antarmuka pengguna bernama `cboperator` dan menyimpannya dalam variabel *integer* bernama `c`.
- Tambahkan kode `if (c==0) {hasil= a+b;}` untuk melakukan penjumlahan jika suatu kondisi terpenuhi.
- Tambahkan kode `if (c==1) {hasil= a-b;}` untuk melakukan pengurangan jika suatu kondisi terpenuhi.
- Tambahkan kode `if (c==2) {hasil= a*b;}` untuk melakukan perkalian jika suatu kondisi terpenuhi.
- Tambahkan kode `if (c==3) {hasil= a/b;}` untuk melakukan pembagian jika suatu kondisi terpenuhi.
- Tambahkan kode `if (c==4) {hasil= a%b;}` untuk menghitung sisa bagi jika suatu kondisi terpenuhi.
- Tambahkan kode `txtHasil.setText(String.valueOf(hasil))`; Untuk menampilkan nilai dari variabel hasil ke dalam komponen teks `txtHasil`.

- Tambahkan kode } *catch* (*NumberFormatException ex*) { sebagai penanganan pengecualian (*exception handling*) dalam bahasa pemrograman Java.
- Tambahkan kode *PesanEror*("Bilangan 1 dan Bilangan 2 harus angka"); untuk menampilkan pesan kesalahan kepada pengguna.
- Tambahkan kode *btnNewButton.setBounds(207, 125, 117, 29)*; untuk menentukan posisi dan ukuran sebuah komponen grafis dalam *layout absolut*.
- Tambahkan kode *contentPane.add(btnNewButton)*; untuk menambahkan objek komponen visual, dalam hal ini sebuah tombol (*btnNewButton*), ke dalam sebuah wadah (*container*) yang disebut *contentPane*.
- Berikut adalah gambar tahapan pertengahan pada projek yang dibuat:

```

99
100 JComboBox cbOperator = new JComboBox();
101 cbOperator.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}));
102 cbOperator.setBounds(116, 126, 73, 27);
103 contentPane.add(cbOperator);
104
105 JButton btnNewButton = new JButton("Hitung");
106 btnNewButton.addActionListener(new ActionListener() {
107     int hasil;
108     public void actionPerformed(ActionEvent e) {
109         if (txtBil1.getText().trim().isEmpty()) {
110             pesanPeringatan("Bilangan 1 Harus diisi");
111         } else if (txtBil2.getText().trim().isEmpty()) {
112             pesanPeringatan("Bilangan 2 Harus diisi");
113         } else {
114             try {
115                 int a= Integer.valueOf(txtBil1.getText());
116                 int b= Integer.valueOf(txtBil2.getText());
117                 int c= cbOperator.getSelectedIndex();
118                 if (c==0) {hasil= a+b;}
119                 if (c==1) {hasil= a-b;}
120                 if (c==2) {hasil= a*b;}
121                 if (c==3) {hasil= a/b;}
122                 if (c==4) {hasil= a%b;}
123                 txtHasil.setText(String.valueOf(hasil));
124             } catch (NumberFormatException ex) {
125                 PesanEror("Bilangan 1 dan Bilangan 2 harus angka");
126             }
127         }
128     }
129 });
130 btnNewButton.setBounds(207, 125, 117, 29);
131 contentPane.add(btnNewButton);
132
133 }
134 }
135

```

3. Tahap akhir

Sebenarnya kode program yang perlu kita tambahkan sudah bisa untuk menjalankan projek yang dibuat, tetapi kita masih perlu menambahkan beberapa kode program lagi supaya hasil projek menjadi sempurna. Kode program yang perlu kita tambahkan yaitu kode yang dimana ketika menjalankan kode program, apabila tidak memasukan perintah yang keluar, *outputnya* adalah pesan eror berupa peringatan sehingga ketika tidak mengisi perintah pada salah satu *output* projek akan keluar peringatan dimana perintah wajib diisi. Berikut adalah tambahan kode programnya:

- Tambahkan kode *private void* pesan Peringatan (*String* pesan) { Untuk mendefinisikan sebuah metode (fungsi) yang menangani pesan peringatan.
- Tambahkan kode *JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE)* untuk menampilkan sebuah kotak dialog pesan kepada pengguna.
- Tambahkan kode *private void* Pesan Eror (*String* pesan) { untuk mendeklarasikan sebuah metode (fungsi) bernama Pesan Eror.
- Tambahkan kode *JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);* untuk menampilkan sebuah kotak dialog peringatan kepada pengguna.
- Berikut gambar kode program tambahan:

```

27
28  private void pesanPeringatan(String pesan) {
29      JOptionPane.showMessageDialog(this,pesan,"Peringatan",JOptionPane.WARNING_MESSAGE);
30  }
31  private void PesanEror(String pesan) {
32      JOptionPane.showMessageDialog(this,pesan,"Kesalahan",JOptionPane.ERROR_MESSAGE);
33  }

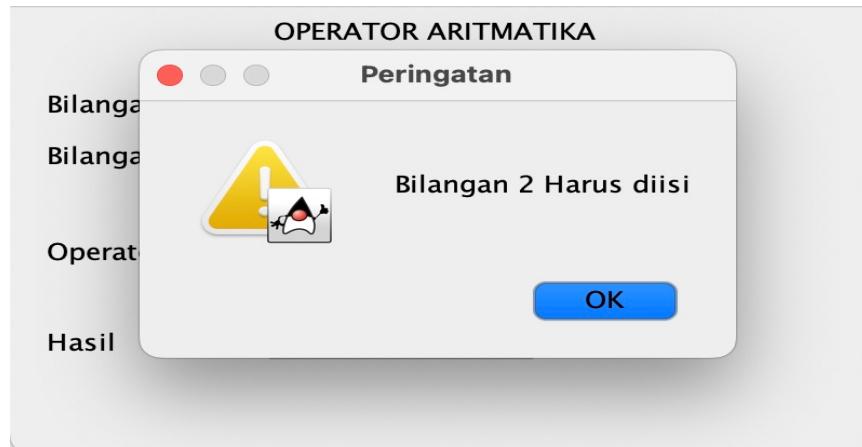
```

4. Hasil projek Operator Aritmatika.

Tapi pada projek kali ini berbeda dengan sebelumnya, karena ketika di *run* akan muncul *JFrame* pada layar laptop dimana kita memasukan angka pada bagian kolom warna putih, kemudian kita pilih operator yang diinginkan, terus tekan hitung, supaya hasilnya keluar pada kolom hasil. Intinya *outputnya* kita jalankan secara manual seperti gambar dibawah ini:

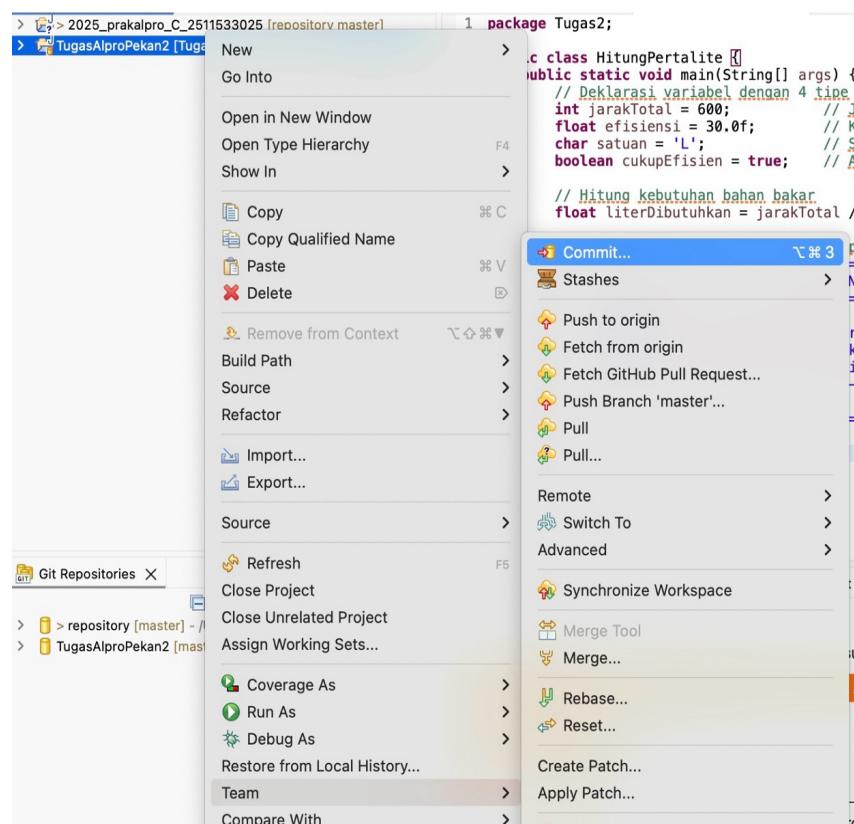
OPERATOR ARITMATIKA		
Bilangan 1	17	
Bilangan 2	7	
Operator	*	Hitung
Hasil	119	

Tapi apabila kita tidak memasukan angka pada salah satu kolom, maka akan muncul peringatan dimana peringatannya sesuai dengan bagian kolom yang tidak diisi. Berikut adalah gambar *outputnya*:



B. Langkah Penyimpanan

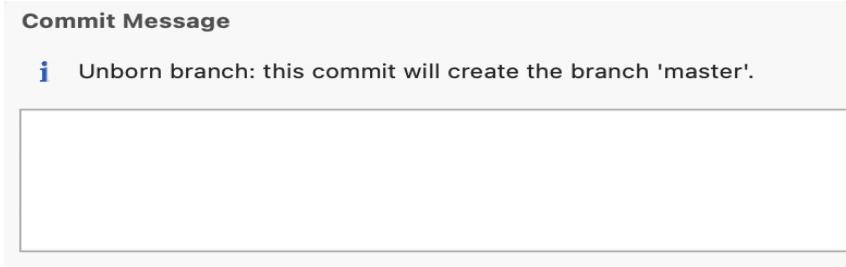
1. Sebelum kita masuk ke langkah, kita harus buat akun github dulu. Jika sudah buat akun github, baru bisa kita simpan di githubnya. Selanjutnya ikutin langkah ini.Tekan kanan mouse pada folder projek, setelah itu pilih team, terus ke *commit*.



2. Lalu tekan tombol plus 2/double plus warna hijau.



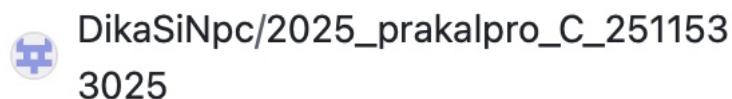
3. Lalu masukan pesan pada kolom *commit message*.



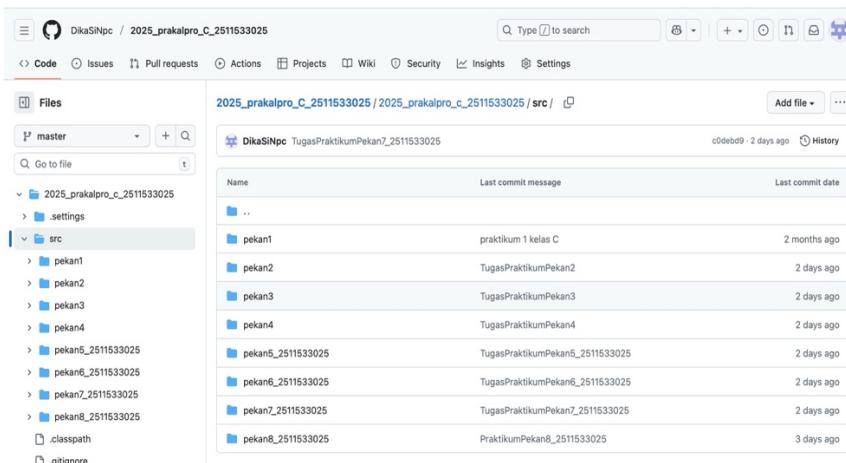
4. Setelah itu tekan *commit* and *pus*, lalu ikutin arahan umtuk memasukan nama dan pw akun git hub kalian, maka projek kalian sudah tersimpan di akun github kalian.



5. Projek yang sudah disimpan di akun projek.



6. Gambar projek apabila sudah masuk di akun github.



Sedikit tambahan jika ingin cek hasil program nya tekan tombol *run* warna hijau.



Catatan :

Jika kita ingin membuat projek baru lagi, maka cukup buat folder *class* yang baru.

Tapi pada program yang sekarang berbeda dari sebelumnya karena pada program ini ada yang inputnya kita masukan angka sesuai keinginan kita pada halaman *JFrame*, tapi ingat harus sesuai perintah kode yang kita masukan.

BAB 3

PENUTUP

3.1 Kesimpulan

Berdasarkan Praktikum yang telah dilakukan, praktikum pemrograman *GUI* pada Java berhasil memperkenalkan dan membekali penulis dalam membangun antarmuka grafis interaktif menggunakan *library Swing*. Penulis mampu mengimplementasikan komponen dasar seperti tombol, label, dan *text field*, serta mengelola event melalui listener untuk merespons interaksi pengguna. Penggunaan *layout manager* juga memperdalam pemahaman tentang tata letak komponen yang responsif. Meskipun *Swing* memiliki keterbatasan dalam tampilan modern, praktikum ini memberikan fondasi kuat dalam pemrograman berbasis *GUI* dan pentingnya desain antarmuka yang *user-friendly*. Untuk pengembangan lebih lanjut, direkomendasikan eksplorasi *JavaFX* sebagai alternatif yang lebih modern. Dan penulis sangat senang bisa mengikuti praktikum ini karena menambah wawasan dan ilmu pengetahuan tentang dunia pemrograman.

3.2 Saran

Untuk meningkatkan kualitas praktikum sebaiknya praktikum pemrograman *GUI* pada Java diperluas dengan memperkenalkan *JavaFX* sebagai alternatif modern yang lebih fleksibel dan menawarkan tampilan *visual* yang lebih menarik dibandingkan *Swing*, serta dilengkapi dengan contoh penerapan desain berbasis *MVC* untuk membangun aplikasi yang lebih terstruktur dan *scalable*. Selain itu, disarankan untuk menyertakan modul dasar pengujian antarmuka (*UI testing*) dan aksesibilitas agar mahasiswa memahami pentingnya membuat aplikasi tidak hanya fungsional, tetapi juga ramah pengguna dari berbagai kebutuhan. Latihan tambahan berupa proyek kecil yang mengintegrasikan *GUI* dengan database dapat memperkuat pemahaman aplikasi nyata dalam konteks pengembangan perangkat lunak secara keseluruhan.

DAFTAR PUSTAKA

- Sumber daring (website):

- [1] F. Amarizuki, "Java Swing: Membangun Antarmuka Pengguna," Medium, 2023. [Daring]. Tersedia pada: <https://medium.com/@furatamarizuki/java-swing-membangun-antarmuka-pengguna>. [Diakses: 19-Nov-2025].
- [2] Dealls, "GUI Adalah: Pengertian, Fungsi, dan Contohnya," 2025. [Daring]. Tersedia pada: <https://dealls.com/pengembangan-karir/gui-adalah>. [Diakses: 19-Nov-2025].
- [3] HIMSISFO BINUS, "Komponen-Komponen GUI pada Java," 2017. [Daring]. Tersedia pada: <https://student-activity.binus.ac.id/himsisfo/2017/05/komponen-komponen-gui-pada-java/>. [Diakses: 19-Nov-2025].
- [4] Jocodev, "Panduan Lengkap Framework Java GUI untuk Desktop," 2025. [Daring]. Tersedia pada: <https://jocodev.id/panduan-lengkap-framework-javascript-untuk-desktop/#>. [Diakses: 19-Nov-2025].