

**LAPORAN PRAKTIKUM ALGORITMA DAN  
PEMROGRAMAN**

**PEKAN 9 PEMROGRAMAN GUI 2**



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, *Frame*, *label*, *text field*, *JButton*, *Absolute Layout*, 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

<b>KATA PENGANTAR.....</b>	.ii
<b>DAFTAR ISI.....</b>	.iii
<b>BAB 1 PENDAHULUAN.....</b>	.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
<b>BAB 2 PEMBAHASAN PRAKTIKUM.....</b>	.3
2.1 Pengertian Pemrograman GUI.....	.3
2.3 Langkah Langkah Pengerjaan Praktikum Pekan 9.....	.5
<b>BAB 3 PENUTUP.....</b>	.19
3.1 Kesimpulan.....	.19
3.2 Saran.....	.19
<b>DAFTAR PUSTAKA.....</b>	.20

# **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.

Komponen yang kita pakai pada projek pekan ini:

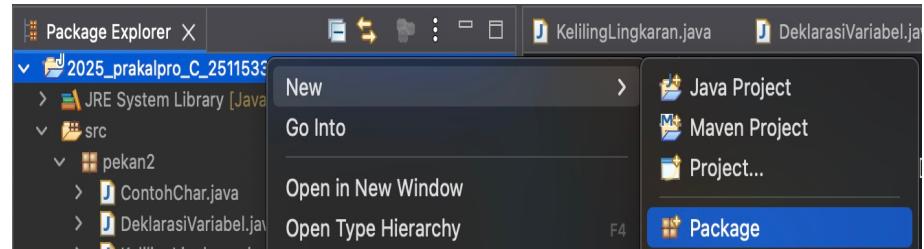
- *WindowBuilder* adalah plugin untuk IDE seperti *Eclipse* yang memungkinkan pengembang Java membuat antarmuka pengguna grafis (*GUI*) secara visual menggunakan *drag-and-drop*, tanpa harus menulis banyak kode secara manual.
- *Swing Designer* adalah alat atau perangkat lunak yang membantu pengembang Java membuat antarmuka pengguna grafis (*GUI*) untuk aplikasi mereka menggunakan *toolkit Swing*, tanpa harus menulis banyak kode secara manual.

- *Application Window* merupakan antarmuka visual yang menampilkan sebuah program atau aplikasi, berfungsi sebagai area interaktif tempat pengguna berinteraksi dengan aplikasi tersebut. Fungsi utamanya adalah menyediakan ruang untuk menampilkan konten, data, dan kontrol aplikasi, serta memungkinkan pengguna untuk memanipulasi, seperti meminimalkan, memaksimalkan, menutup, dan mengubah ukuran jendela.
- *Frame* sebagai wadah utama untuk membangun antarmuka pengguna grafis (*GUI*).
- *TextField* berfungsi untuk memungkinkan pengguna memasukkan teks, angka, atau simbol dalam sebuah aplikasi melalui papan ketik.
- *JButton* berfungsi untuk membuat komponen tombol tekan yang bisa diklik dalam antarmuka pengguna grafis (*GUI*) aplikasi *Java Swing*. Ketika pengguna mengklik tombol, ia akan menghasilkan *ActionEvent* yang dapat digunakan untuk menjalankan tindakan tertentu, seperti menampilkan pesan, membuka jendela lain, atau memperbarui data. Tombol ini juga bisa diberi ikon atau teks, dan dapat dikustomisasi tampilannya lebih lanjut.
- *Absolute layout* untuk memungkinkan penempatan komponen antarmuka pengguna (*UI*) pada koordinat X dan Y tertentu secara *eksplisit*.

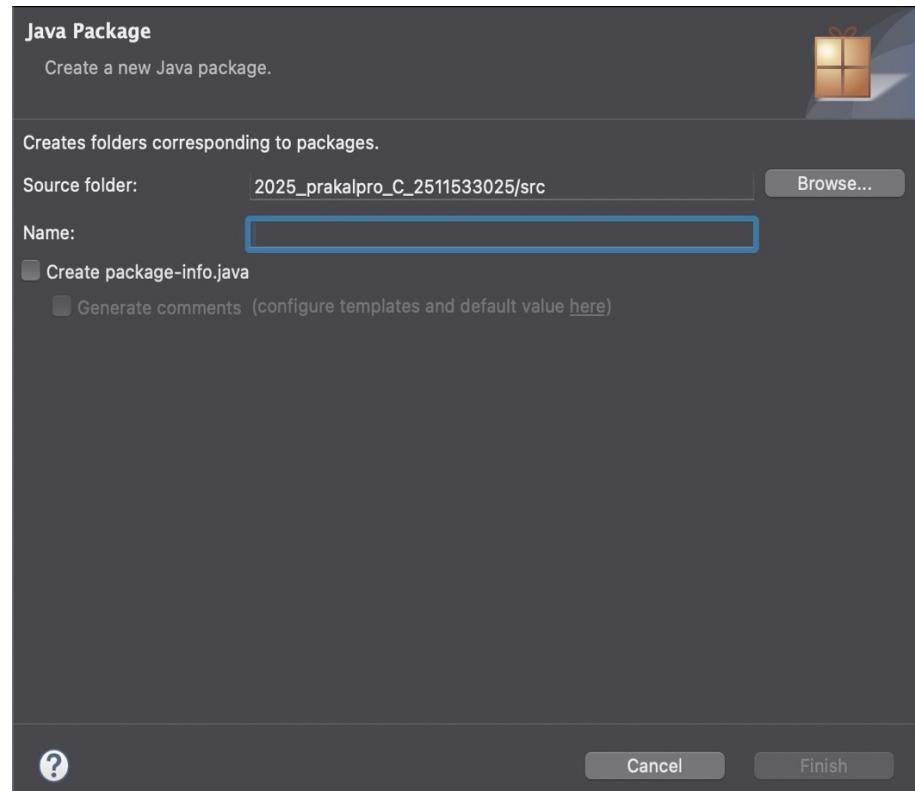
## 2.2 Langkah Langkah Pengerjaan Praktikum Pekan 9

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

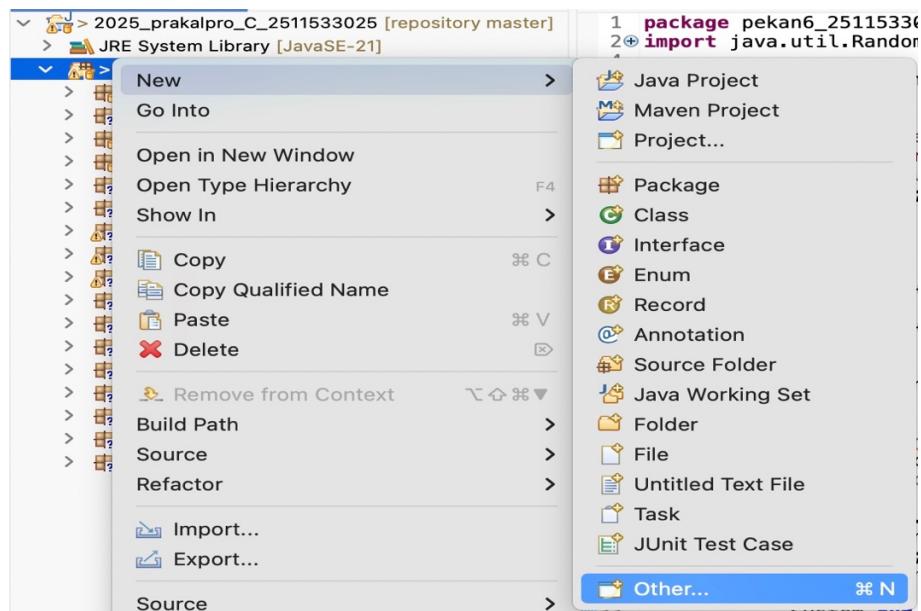
1. Setelah kalian masuk ke eclipse java dan membuat folder file projek java, click kanan pada mouse kalian , kemudian pilih *new* terus pilih *package*.



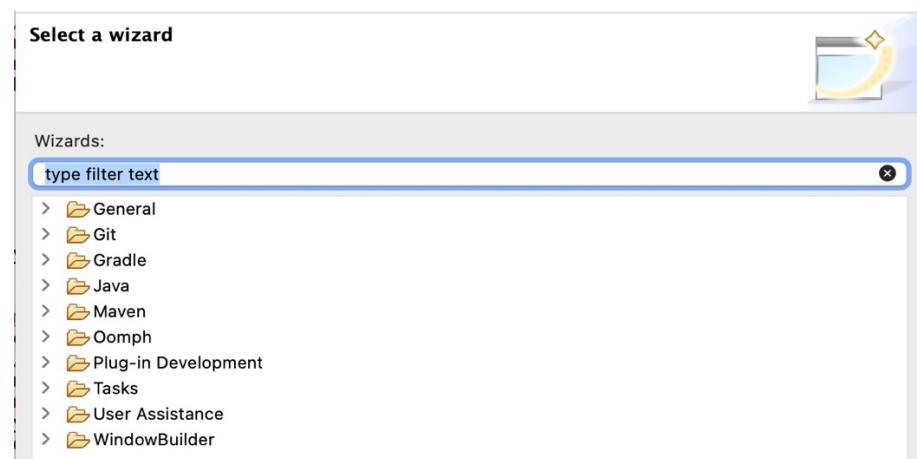
2. Setelah di click akan muncul halaman java *package*, dimana kita disuruh bikin nama untuk folder *package* nya, lalu pilih finish.



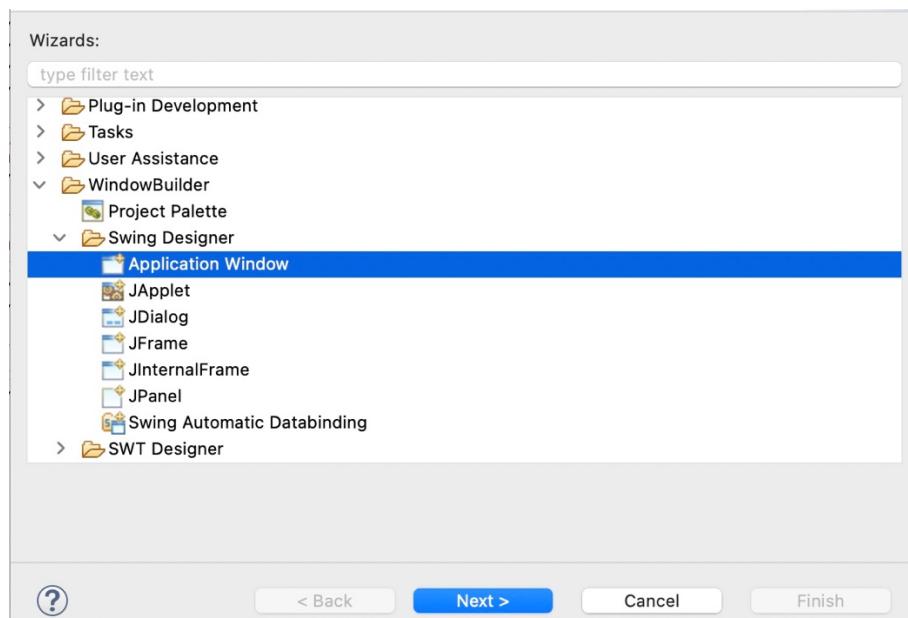
3. Jika folder *packagenya* sudah jadi, click kanan *mouse* pada folder *package* nya, lalu akan muncul pilihan, kalian pilih *new* kemudian pilih *other*.



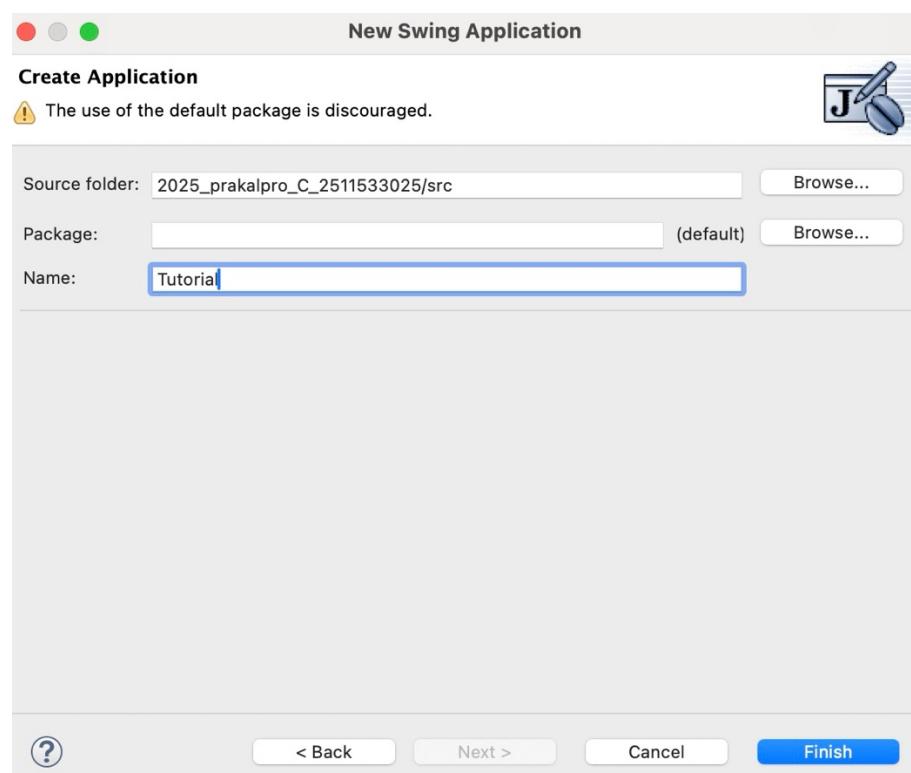
4. Setelah di click akan muncul halaman *Select a wizard*, lalu pilih *WindowBuilder*.



5. Kemudian akan muncul beberapa pilihan, kalian pilih yang Namanya *Application Window*, terus *click Next*.



6. Lalu akan muncul halaman *New Swing Application* seperti gambar dibawah ini dimana kalian diminta memasukan nama sesuai dengan projek yang ingin dibuat. Jika sudah, *click finish*.



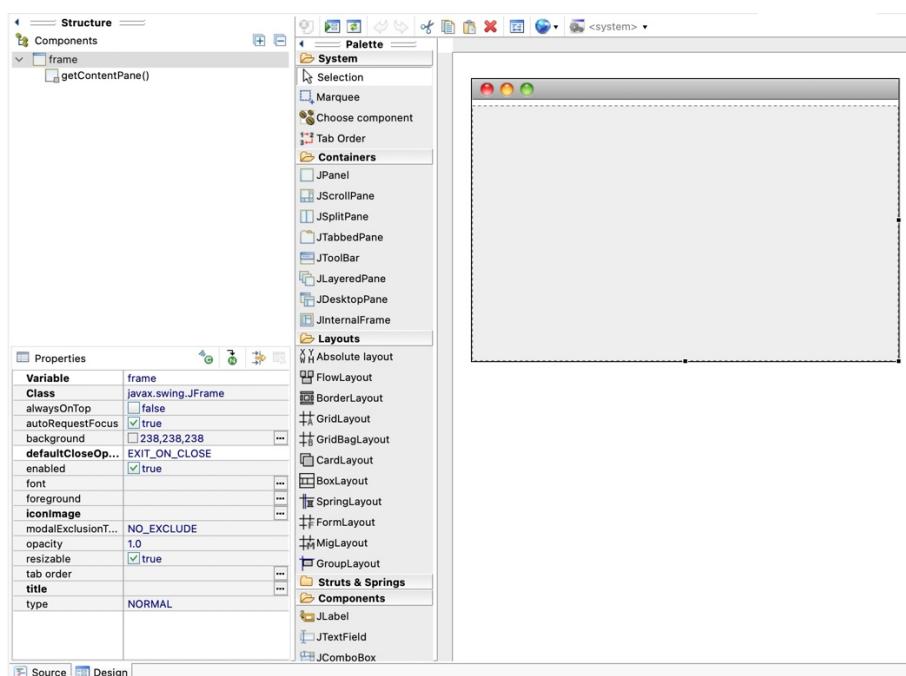
5. Setelah selesai pembuatan folder projeknya maka akan langsung muncul halaman projek seperti gambar dibawah ini.

```

1 package example;
2
3+ import java.awt.EventQueue;[]
4
5 public class Calculator_2511533025 {
6
7     private JFrame frame;
8
9     /**
10      * Launch the application.
11     */
12     public static void main(String[] args) {
13         EventQueue.invokeLater(new Runnable() {
14             public void run() {
15                 try {
16                     Calculator_2511533025 window = new Calculator_2511533025();
17                     window.frame.setVisible(true);
18                 } catch (Exception e) {
19                     e.printStackTrace();
20                 }
21             }
22         });
23     }
24 }
25
26
27 /**
28  * Create the application.
29 */
30 public Calculator_2511533025() {
31     initialize();
32 }
33
34 /**
35  * Initialize the contents of the frame.
36 */
37 private void initialize() {
38     frame = new JFrame();
39     frame.setBounds(100, 100, 450, 300);
40     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
41 }
42
43 }
44

```

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



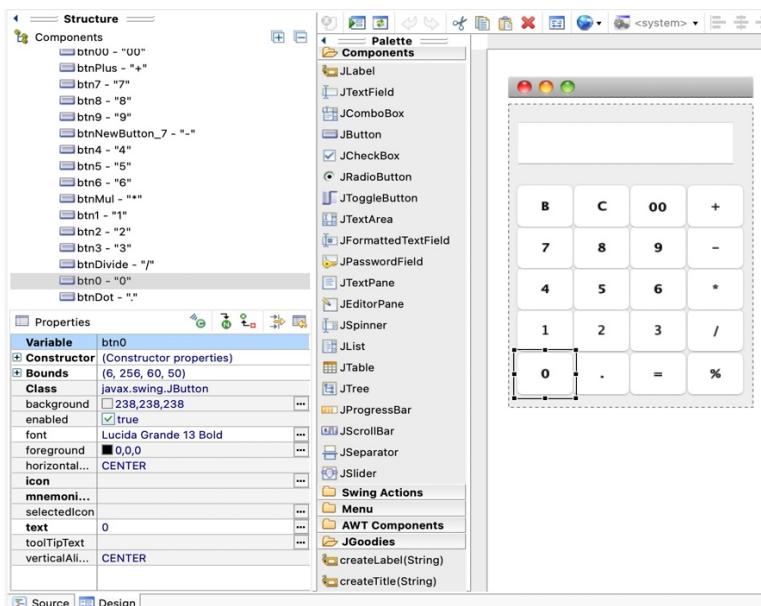
Catatan :

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 penggeraan projek yang berjudul Calculator:

1. Tahap awal

- Atur dulu ukuran *Frame* sesuai dengan yang kalian inginkan
- Tekan pada bagian yang bernama *Absolute Layout* seret kursor ke *frame* lalu tekan satu kali.
- Scrol ke bawah pada bagian components, pilih *JTextField* terus seret kursor ke *frame*, atur posisi dan ukuranya sesuai dengan projek yang dibuat. Terus ubah *font* menjadi *bold*.
- Kemudian pilih JButton arahkan kursor ke frame, tekan satu kali, atur posisi dan ukuranya, terus ubah *font* menjadi *bold*. Karena yang kita butuhkan ada 20 buah kelas yaitu dengan pola 4 x 5, dimana 4 baris dan 5 kolom, maka kalian cukup salin tempel kelas sebelumnya. Setelah selesai menambahkan kelas atur lagi posisi *frame* apabila posisinya kurang pas.
- Kemudian ubah nama *text* dan *variable* pada bagian *properties*. Pada bagian Variable ubah namanya dengan diawali btn dilanjutkan dengan nama *variable*, untuk angka 0-9 namanya menjadi *btn+(angka)*, contoh *btn1* dan seterusnya. Kemudian untuk yang lain agak berbeda yaitu seperti ini simbol (.): *btnDot*, (=): *btnEqual*, (%): *btnPercent*, (/): *btnDivide*, (\*): *btnMul*, (-): *btnSub*, (+): *btnPlus*, (00): *btn00*, (C): *btnClear*, (B): *btnBackSpace*. Kemudian nama pada bagian text juga disesuaikan dengan *variable* nya. Seperti gambar dibawah ini:



- Selanjutnya coba di *run* terlebih dahulu apakah keluar atau tidak sesuai dengan *frame* yang sudah dibuat.

## 2. Tahap Kedua

Setelah selesai mengatur bagian *Design* kita akan mengatur kode program atau lebih tepatnya menambahkan kode program di bagian menu *Source* agar *Frame* yang sudah dibuat dapat dijalankan. Tapi cara menambahkan kode program pada projek ini berbeda dimana kita harus bolak balik dari *design* ke *source* dengan cara menekan dua kali pada simbol/kelas yang sudah dibuat secara satu persatu. Berikut ini adalah kode program tambahan:

- Untuk kode program pada format angka seperti (.,00,0-9), menggunakan kode seperti ini: *String number=textField.getText()+btn1.getText();textField.setText(number);*, dimana pada bagian kode *getText()*+ setelah kode + kita tambah dengan nama *variable* sesuai dengan nama *variable* kelas yang kita pilih atau yang kita ubah atau lebih tepatnya kita sesuaikan dengan barisan awal setelah kode *JButton*. Berikut adalah gambar kode program yang kita buat dengan menambahkan dan mengubah beberapa kode program:

```

131
132 JButton btn1 = new JButton("1");
133△ btn1.addActionListener(new ActionListener() {
134△     public void actionPerformed(ActionEvent e) {
135         String number=textField.getText()+btn1.getText();
136         textField.setText(number);
137     }
}

```

- Berikut adalah penjelasan mengenai kode yang ditambahkan yaitu untuk kode *String number=textField.getText()+btn1.getText();* berfungsi untuk menggabungkan teks dari dua komponen (*textField* dan *btn1*) antarmuka pengguna yang berbeda dan menyimpannya ke dalam satu variabel string. Secara singkat, baris kode ini mengambil teks dari kotak teks dan tombol, lalu menyatukannya menjadi satu *string* baru. Selanjutnya untuk kode *textField.setText(number);* berfungsi untuk mengatur teks pada komponen antarmuka pengguna (*UI*) seperti kotak teks atau label. Secara keseluruhan, baris kode tersebut mengambil nilai yang disimpan dalam variabel *number* dan menampilkannya di dalam komponen *visual textField* pada layar aplikasi.
- Selanjutnya untuk kode program pada format operator seperti (+,-,\*,/,%), menggunakan kode seperti ini:

*first=Double.parseDouble(textField.getText());textField.setText("");operation="+"*. Dimana setelah kode *operation="+"* lebih tepatnya di tengah-tengah tanda petik dua diisi dengan kode operator sesuai dengan yang dipilih atau sesuaikan namanya dengan barisan awal setelah kode *JButton*. Seperti gambar dibawah ini:

```
252     JButton btnPlus = new JButton("+");
253     btnPlus.addActionListener(new ActionListener() {
254         public void actionPerformed(ActionEvent e) {
255             first=Double.parseDouble(textField.getText());
256             textField.setText("");
257             operation="+";
258         }
259     }
```

- Berikut adalah penjelasan mengenai kode yang di tambahkan yaitu untuk kode *first=Double.parseDouble(textField.getText());* berfungsi untuk mengambil teks dari sebuah komponen antarmuka pengguna (*UI*) yang disebut *textField* dan mengubahnya menjadi nilai angka desimal (*double*), kemudian menyimpan nilai tersebut ke dalam variabel bernama *first*. Terus untuk kode *textField.setText("")*; berfungsi untuk mengatur konten teks dari suatu komponen antarmuka pengguna (*UI*) yang bernama *textField* serta mengosongkan atau menghapus teks dari komponen *textField*. Terus untuk kode *operation="(+,-,\*,/,%)"* berfungsi untuk menyimpan simbol operasi pengurangan atau tanda hubung untuk digunakan nanti dalam program atau fungsi tertentu.
- Selanjutnya untuk kode program pada format (C), menggunakan kode seperti ini: *textField.setText(null)*; tapi pada kode kali ini tidak perlu menyamakan namanya dengan barisan awal setelah kode *JButton*. Seperti gambar dibawah ini:

```
78     JButton btnClear = new JButton("C");
79     btnClear.addActionListener(new ActionListener() {
80         public void actionPerformed(ActionEvent e) {
81             textField.setText(null);
82         }
83     }
```

- Berikut adalah penjelasan mengenai kode yang di tambahkan yaitu untuk kode *textField.setText(null)*; untuk mengosongkan atau menghapus teks yang ada di dalam kolom teks (*text field*).
- Selanjutnya untuk kode program pada format (B) akan kita ubah menjadi (*←*) dengan menambahkan kode ini: *String backSpace=null;if(textField.getText().length()>0)* dan *StringBuilder str=newStringBuilder(textField.getText());str.deleteCharAt(textField.getText().length()-1);textField.setText(str.toString());*

*1);backSpace=str.toString();textField.setText(backSpace);*  
Seperti gambar dibawah ini:

```
58     String backSpace=null;
59     if(textField.getText().length()>0)
60     {
61         StringBuilder str=new StringBuilder(textField.getText());
62         str.deleteCharAt(textField.getText().length()-1);
63         backSpace=str.toString();
64         textField.setText(backSpace);
65     }
66 }
```

- Berikut adalah penjelasan mengenai kode yang di tambahkan yaitu untuk kode *String backSpace=null;* berfungsi untuk membuat sebuah variabel bernama *backSpace* yang siap digunakan untuk menyimpan teks, namun pada awalnya diatur agar tidak berisi apa-apa (*null*). Terus untuk kode *if(textField.getText().length()>0)* berfungsi untuk memeriksa apakah kolom teks (*textField*) tidak kosong. Untuk kode *StringBuilder str=newStringBuilder(textField.getText());* berfungsi untuk membuat objek *StringBuilder* baru yang berisi teks yang saat ini ada di dalam komponen antarmuka pengguna *textField*. Untuk kode *str.deleteCharAt(textField.getText().length()-1);* berfungsi untuk menghapus karakter terakhir dari teks yang ada di dalam *textField* dan menyimpannya kembali ke objek *str*. Untuk kode *backSpace=str.toString();* berfungsi untuk menyalin *representasi string* dari *variabel str* ke *variabel backSpace*. Untuk kode *textField.setText(backSpace);* berfungsi untuk mengatur teks dari komponen antarmuka pengguna *textField* menjadi nilai yang disimpan dalam *variabel backSpace*.
- Selanjutnya untuk kode program pada format (=) menggunakan kode seperti gambar dibawah ini, karena operator merupakan kunci utama supaya projek yang dibuat berjalan sesuai *outputnya*, jadi cukup menyalin ualng kode yang sama karena setiap operator perhitungan terikat sama operator =. Satu lagi bagi yang keyboardnya bukan keyboard Inggris jadi perlu tambahkan kode *locale.US* supaya kode program dapat berjalan sesuai keinginan pengguna.

```

223     String answer;
224     second = Double.parseDouble(textField.getText());
225     if (operation == "+") {
226         result = first + second;
227         answer = String.format(Locale.US, "%.2f", result);
228         textField.setText(answer);
229     } else if (operation == "-") {
230         result = first - second;
231         answer = String.format(Locale.US, "%.2f", result);
232         textField.setText(answer);
233     } else if (operation == "*") {
234         result = first * second;
235         answer = String.format(Locale.US, "%.2f", result);
236         textField.setText(answer);
237     } else if (operation == "/") {
238         result = first / second;
239         answer = String.format(Locale.US, "%.2f", result);
240         textField.setText(answer);
241     } else if (operation == "%") {
242         result = first % second;
243         answer = String.format(Locale.US, "%.2f", result);
244         textField.setText(answer);
245     }
246 }
```

- Berikut adalah penjelasan mengenai kode yang di tambahkan yaitu untuk kode *String answer;* berfungsi untuk mempersiapkan tempat penyimpanan di memori untuk data teks yang akan digunakan nanti dalam program. Untuk kode program *second=Double.parseDouble(textField.getText())* berfungsi untuk mengonversi teks dari sebuah *text field* menjadi angka desimal (*double*). Untuk kode *if(operation=="+", -, \*, /, %") {* berfungsi untuk memulai blok kondisional yang akan dieksekusi jika variabel '*operation*' bernilai sama dengan tanda tambah, kurang, kali, bagi, dan persen (+, -, \*, /, %). Untuk kode *result=first+second;* berfungsi untuk menjumlahkan dua nilai dan menyimpan hasilnya dalam variabel baru. Untuk kode *answer=String.format("%.2f", result);* berfungsi untuk memformat nilai variabel '*result*' menjadi *string* dengan dua angka desimal, lalu menyimpannya ke dalam variabel '*answer*'. Untuk kode *textField.setText(answer);* berfungsi untuk mengambil nilai yang tersimpan dalam *answer* dan menampilkannya di layar pada area *textField*.

### 3. Tahap ketiga

Selanjutnya kita masukan kode supaya operasi perhitungan bisa berjalan. Berikut adalah gambar tambahan kode programnya dan penjelesannya:

```

17     double first;
18     double second;
19     double result;
20     String operation;
21     String answer;
22 }
```

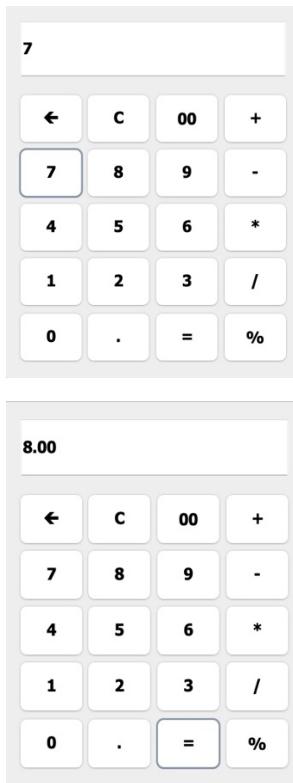
- Untuk kode *double first;* berfungsi untuk memberi tahu *kompiler* bahwa ada variabel bernama *first* yang akan

digunakan untuk menyimpan nilai *desimal*. Untuk kode *double second*; berfungsi untuk mendeklarasikan sebuah *variabel* bernama *second* dengan tipe data *double*. Untuk kode *double result*; berfungsi untuk memberitahu *kompiler* untuk menyisihkan ruang memori yang cukup untuk menyimpan nilai *desimal* presisi ganda dan memberinya nama "*result*" untuk diakses nanti dalam program. Untuk kode *String operation*; berfungsi untuk mendeklarasikan yang menunjukkan bagian dari program yang menangani manipulasi teks. Untuk kode *String answer*; berfungsi untuk menyimpan nilai *string*, dan bagian memori tersebut dapat diakses atau dirujuk menggunakan nama "*answer*".

#### 4. Hasil projek Calculator.

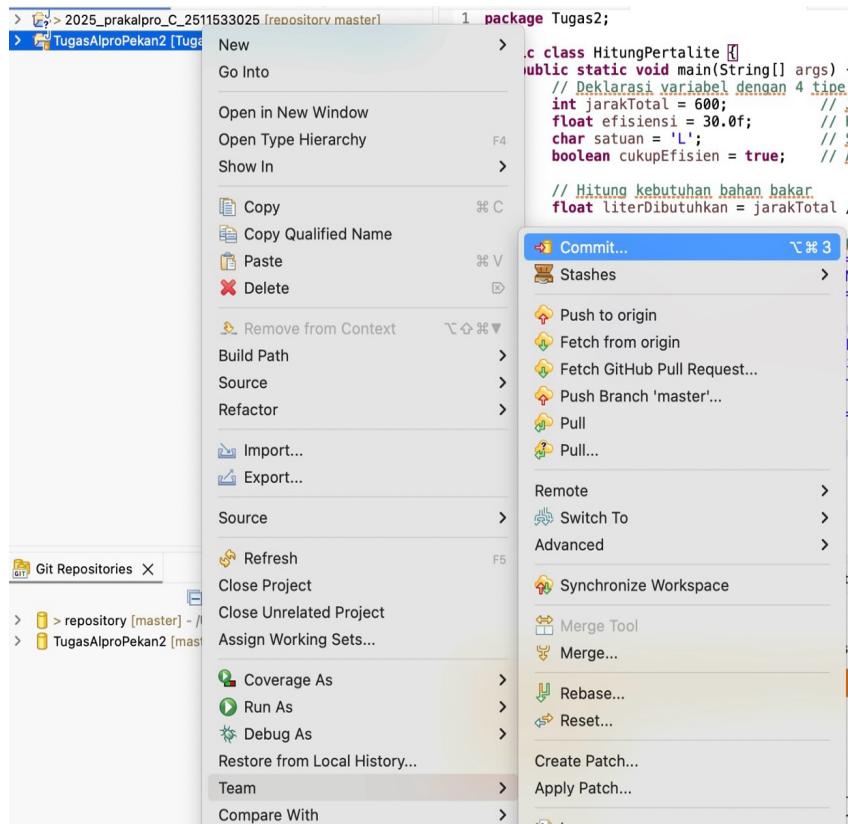
Tapi pada projek kali ini berbeda dengan sebelumnya, karena ketika di *run* akan muncul *Frame* pada layar laptop dimana kita memilih angka pada bagian kolom yang berisi angka , kemudian kita pilih operator yang diinginkan, terus tekan sama dengan, supaya hasilnya keluar pada kolom hasil. Intinya *outputnya* kita jalankan secara manual. Tapi karena setiap kita masukan operator, angka sebelumnya akan hilang , jadi tidak bisa membuktikan , kalau ingin lihat buktinya bisa salin kode di *github* dan coba sendiri , atau pakai cara satu lagi dengan memvideokan. Seperti gambar dibawah ini:





## 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*.

#### Commit Message

**i** Unborn branch: this commit will create the branch 'master'.

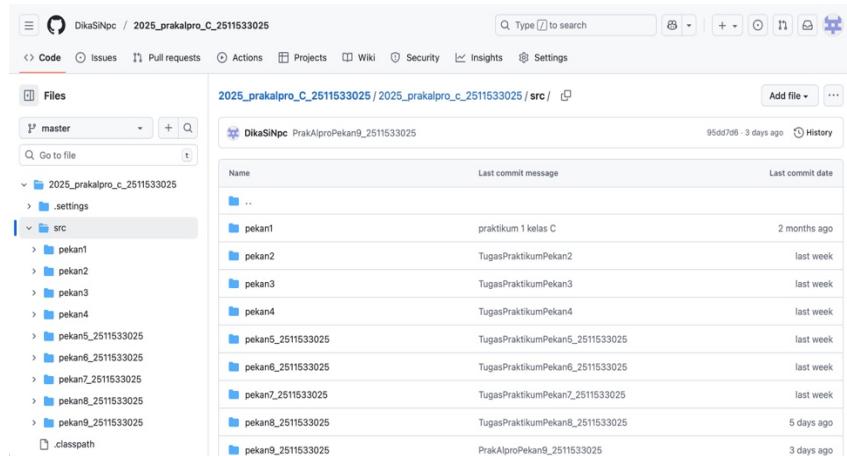
4. Setelah itu tekan *commit* and *pus*, lalu ikutin arahan umtuk memasukan nama dan pw akun git hub kalian, maka projek kalian sudah tersimpan di akun github kalian.



5. Projek yang sudah disimpan di akun projek.

## DikaSiNpc/2025\_prakalpro\_C\_251153 3025

6. Gambar projek apabila sudah masuk di akun github.



The screenshot shows a GitHub repository page for 'DikaSiNpc/2025\_prakalpro\_C\_251153'. The repository name is '2025\_prakalpro\_C\_251153'. The 'src' folder contains several files named 'pekan1' through 'pekan9'. A table lists these files with their last commit message and date:

Name	Last commit message	Last commit date
..	praktikum 1 kelas C	2 months ago
pekan1	TugasPraktikumPekan2	last week
pekan2	TugasPraktikumPekan3	last week
pekan3	TugasPraktikumPekan4	last week
pekan4	TugasPraktikumPekan5	last week
pekan5_251153025	TugasPraktikumPekan6_251153025	last week
pekan6_251153025	TugasPraktikumPekan7_251153025	last week
pekan7_251153025	TugasPraktikumPekan8_251153025	5 days ago
pekan8_251153025	PrakAlproPekan9_251153025	3 days ago
pekan9_251153025		

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 memilih angka sesuai keinginan kita pada halaman *Frame*, 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, *text field*, *Frame*, *label*, *JButton*, *Absolute Layout*, 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*. 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-javagui-untuk-desktop/#>. [Diakses: 19-Nov-2025].
- [5] FormDev Software GmbH, "JFormDesigner," 2025. [Daring]. Tersedia pada: <https://www.formdev.com/jformdesigner/#:~:text=Harga-JFormDesigner>. [Diakses: 28-Nov-2025].
- [6] Eclipse Foundation, "WindowBuilder," 2025. [Daring]. Tersedia pada: <https://eclipse.dev/windowbuilder/#:~:text=WindowBuilder>. [Diakses: 28-Nov-2025].