

LAPORAN PRAKTIKUM
PEMROGRAMAN GUI 1

Disusun Oleh:

MUHAMAD FAJRI AULIA
NIM 2511532009

DOSEN PENGAMPU : DR. WAHYUDI, S.T, M.T
ASISTEN PRAKTIKUM : AUFAN TAUFIQURRAHMAN



FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
PADANG , 19 NOVEMBER 2025

DAFTAR ISI

DAFTAR ISI	i
KATA PENGANTAR	ii
DAFTAR GAMBAR	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	1
1.4 Manfaat	2
1.5 Persyaratan Praktikum.....	2
BAB II PEMBAHASAN	3
2.1 Pengertian GUI dalam Java	3
2.2 Langkah Pengerjaan	3
2.2.1 Operator Aritmatika GUI	3
BAB III KESIMPULAN	9
3.1 KESIMPULAN	9
3.2 SARAN	9
DAFTAR PUSTAKA	11

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan praktikum mata kuliah Algoritma Pemrograman dengan baik dan tepat waktu.

Laporan ini disusun untuk memenuhi salah satu tugas praktikum pada pertemuan kelima dengan judul “**PEMROGRAMAN GUI 1**”. Melalui penyusunan laporan ini, penulis berharap dapat memberikan gambaran mengenai penggunaan String dalam sebuah program java.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan laporan maupun pemahaman di masa yang akan datang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu serta semua pihak yang telah memberikan bimbingan, arahan, dan bantuan hingga laporan ini dapat terselesaikan. Semoga laporan ini dapat bermanfaat bagi pembaca.

Padang, 10 November 2025

Penulis

DAFTAR GAMBAR

Gambar 1	3
Gambar 2	4
Gambar 3	4
Gambar 4	5
Gambar 5	5
Gambar 6	6

BAB I

PENDAHULUAN

1.1 Latar Belakang

Java merupakan salah satu bahasa pemrograman yang banyak digunakan untuk membangun aplikasi lintas platform karena sifatnya yang *write once, run anywhere*. Dalam pengembangan aplikasi modern, kebutuhan antarmuka yang interaktif dan mudah digunakan semakin meningkat. Oleh karena itu, kemampuan membuat **Graphical User Interface (GUI)** menjadi kompetensi penting bagi mahasiswa informatika.

Pada praktikum **Pemrograman GUI 1**, mahasiswa diperkenalkan pada pembuatan antarmuka berbasis Java. Melalui komponen-komponen Java GUI seperti JButton, JLabel, dan JTextField, mahasiswa dapat belajar merancang tampilan aplikasi serta memproses interaksi pengguna melalui mekanisme *event handling*.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, masalah utama yang akan adalah:

- Bagaimana cara membuat antarmuka GUI menggunakan Java (Swing/AWT)?
- Apa saja komponen dasar GUI dalam Java dan bagaimana cara menggunakannya?
- Bagaimana menghubungkan aksi pengguna (klik tombol, input teks, dll.) dengan logika program melalui event handling?
- Bagaimana merancang aplikasi Java sederhana berbasis GUI yang berfungsi dengan baik?

1.3 Tujuan

Adapun tujuan dari pelaksanaan praktikum ini adalah:

- Memahami konsep GUI dalam pemrograman Java.
- Menggunakan komponen GUI dasar seperti frame, label, tombol, dan text field.
- Mengimplementasikan event handling menggunakan listener.

- Merancang dan membuat aplikasi GUI Java sederhana.
- Melatih kemampuan logika dan kreativitas dalam membangun antarmuka aplikasi.

1.4 Manfaat

Manfaat yang diperoleh mahasiswa melalui praktikum ini antara lain:

- Memiliki kemampuan membuat aplikasi Java berbasis GUI.
- Mampu memahami struktur program Java GUI beserta komponennya.
- Mampu mengelola interaksi pengguna melalui mekanisme event.

1.5 Persyaratan Praktikum

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

- Telah mengikuti perkuliahan teori Pemrograman Java sebagai dasar pemahaman.
- Membawa perlengkapan yang diperlukan, antara lain laptop atau computer yang sudah terpasang Java Development Kit (JDK) dan Integrated Development Enviroment (IDE) yang direkomendasikan. Serta telah menginstal extention Window Builder didalam Eclipse
- Mengikuti setiap sesi praktikum sesuai jadwal yang ditetapkan dan hadir minimal sesuai ketentuan program studi.
- Mematuhi tata tertib laboratorium, termasuk menjaga keamanan data, perangkat, serta lingkungan kerja.
- Menyusun laporan praktikum dengan format dan aturan yang telah ditetapkan dalam pedoman ini.

BAB II

PEMBAHASAN

2.1 Pengertian GUI dalam Java

Graphical User Interface (GUI) adalah antarmuka berbasis grafis yang memungkinkan pengguna berinteraksi dengan aplikasi melalui elemen visual seperti tombol, jendela, ikon, dan menu. Dalam Java, GUI dapat dibuat menggunakan dua pustaka utama, yaitu Abstract Window Toolkit (AWT) dan Swing. Swing merupakan versi yang lebih modern dan lengkap dibandingkan AWT karena menyediakan lebih banyak komponen, tampilan yang konsisten di berbagai platform, serta kemudahan dalam pengembangan. GUI pada Java bekerja dengan memadukan komponen visual dan logika program yang saling terhubung melalui mekanisme *event-driven programming*.

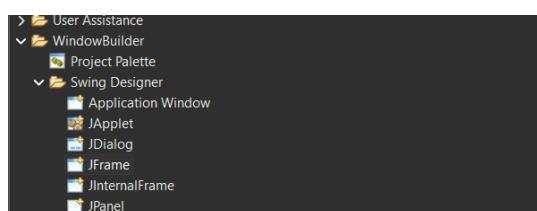
2.2 Langkah Pengerjaan

Untuk lebih memahami pengembangan GUI kita dapat melakukan praktik dengan menggunakan ekstension Window Builder :

2.2.1 Operator Aritmatika GUI

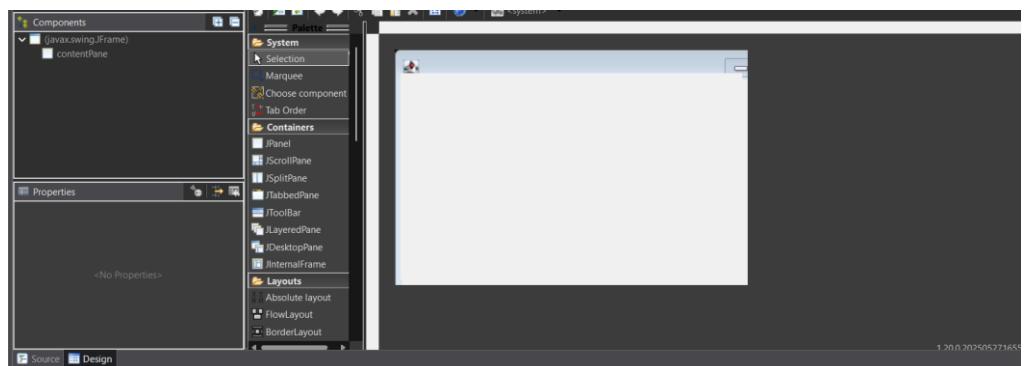
Pada program ini kita akan membuat sebuah operator aritmatika untuk penjumlahan, pengurangan, perkalian, dan pembagian dengan GUI menggunakan Window Builder.

- Pilih new pada bagian package kemudian pilih bagian other disana kita menemukan bagian window builder dan pada bagian Swing designer kita memilih Jframe.



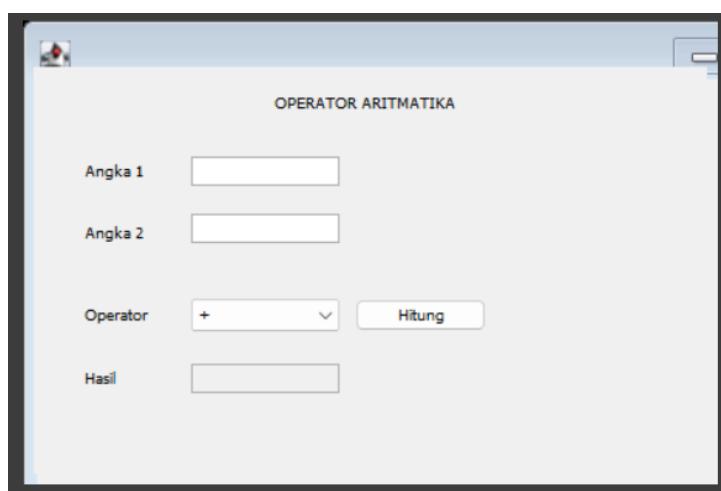
Gambar 1

- Setelah memulai program kita berada di bagian sintax yang mana sudah berisi beberapa .
- Pada bagian bawah kanan kita dapat merubah tampilan untuk melihat bagaimana tampilan program GUI dengan memilih bagian design dan didapatkan bagian dimana layout JFramenya masih belum berisi data data.



Gambar 2

- Selanjutnya disini kita mulai Menyusun isi dari JFrame. Kita gunakan beberapa macam komponen diantaranya textfield sebagai tempat user menginputkan sebuah nilai atau data. Combobox sebagai tempat peletakkan beberapa pilihan untuk user.
- JButton sebagai tombol pemulai aksi dari sebuah program yang kita buat. FormattedTextfield. Dan JLabel sebagai penulisan bagian judul atau penanda perintah bagi user. Sehingga didapat tampilan sebagai berikut.



Gambar 3

- Selanjutnya komponen komponen tadi kita ubah Namanya yang mana nanti akan memudahkan kita dalam membuat syntax nya.
- Untuk combobox kita dapat memasukkan opsi opsinya didalam bagian model.
- Setelah membuat bentuk dari JFrame nya selanjutnya kita menambah pada bagian syntaxnya.
- Dalam program kita menyatakan 3 variabel pertama variable a untuk Jtext field 1 dan b untuk Jtext field 2 dan c untuk hasil dari aritmatika program.

```

int a = Integer.valueOf(txtAngka1.getText());
int b = Integer.parseInt(txtAngka2.getText());
int c = cdOperator.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;

```

Gambar 4

- Buat sebuah statement if sebagai penanda perintah dalam program.
- Untuk sentuhan terakhir kita menambahkan indikasi atau pemberitahuan eror jika sebuah program mengalami eror. Karena jika kita tidak membuat sebuah program pemberitahu eror didapatkan sebuah eror seperti ini

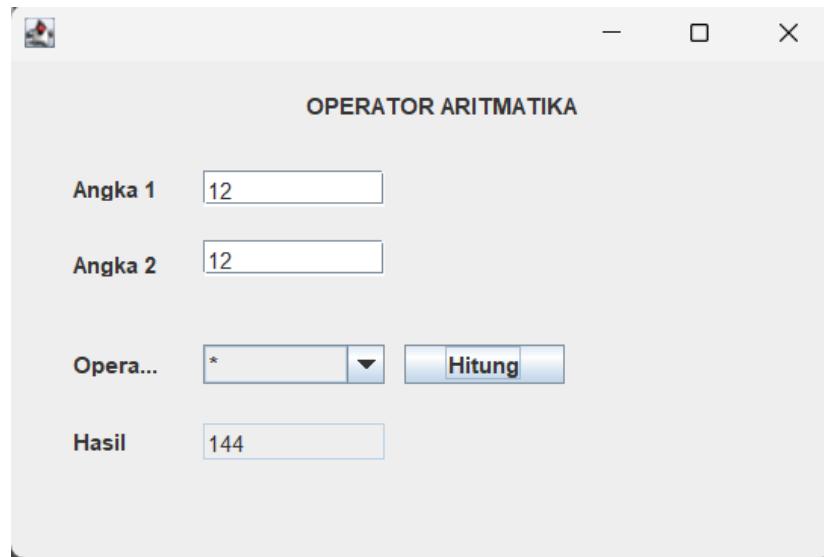
```

Exception in thread "AWT-EventQueue-0" java.lang.ArithmaticException / by zero
at PEKNB/PekanB_2511532012.OperasiAritmatikaGUI_2511532012.actionPerformed(OperatorAritmatikaGUI_2511532012.java:118)
at java.desktop/java.awt.AbstractButton.fireActionPerformed(AbstractButton.java:1972)
at java.desktop/java.awt.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2314)
at java.desktop/java.awt.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:407)
at java.desktop/java.awt.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:282)
at java.desktop/java.awt.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:279)
at java.desktop/java.awt.Component.processMouseEvent(Component.java:6621)
at java.desktop/java.awt.swing.JComponent.processMouseEvent(JComponent.java:3398)
at java.desktop/java.awt.Component.dispatchEvent(Component.java:6386)
at java.desktop/java.awt.Container.dispatchEvent(Container.java:2266)
at java.desktop/java.awt.Component.dispatchEventImpl(Component.java:4996)

```

Gambar 5

- Kita membuat sebuah pemberitahu eror untuk meminimalisir tampilan seperti ini. Jadi makin complex program kita. Maka makin jarang kita menemukan si merah merah ini
- Dari program didapatkan output sebagai berikut.



Gambar 6

- Untuk syntaxnya sebagai berikut

```
• package Pekan8_2511532012;  
•  
• import java.awt.BorderLayout;  
• import java.awt.EventQueue;  
•  
• import javax.swing.JFrame;  
• import javax.swing.JPanel;  
• import javax.swing.border.EmptyBorder;  
• import javax.swing.JLabel;  
• import javax.swing.JOptionPane;  
• import javax.swing.JTextField;  
• import javax.swing.JComboBox;  
• import javax.swing.DefaultComboBoxModel;  
• import javax.swing.JButton;  
• import java.awt.event.ActionListener;  
• import java.awt.event.ActionEvent;  
•  
• public class OperatorAritmatikaGUI_2511532012 extends JFrame {  
•  
•     private static final long serialVersionUID = 1L;  
•     private JPanel contentPane;  
•     private JTextField txtAngka1;  
•     private JTextField txtAngka2;  
•     private JTextField txtHasil;  
•  
•     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);  
•     }  
• }
```

```

•     }
•     /**
•      * Launch the application.
•      */
•     public static void main(String[] args) {
•         EventQueue.invokeLater(new Runnable() {
•             public void run() {
•                 try {
•                     OperatorAritmatikaGUI_2511532012 frame =
• new OperatorAritmatikaGUI_2511532012();
•                     frame.setVisible(true);
•                 } catch (Exception e) {
•                     e.printStackTrace();
•                 }
•             }
•         });
•     }
•
•     /**
•      * Create the frame.
•      */
•     public OperatorAritmatikaGUI_2511532012() {
•         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
•         setBounds(100, 100, 450, 300);
•         contentPane = new JPanel();
•         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
•         setContentPane(contentPane);
•         contentPane.setLayout(null);
•
•         JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
•         lblNewLabel.setBounds(156, 10, 161, 27);
•         contentPane.add(lblNewLabel);
•
•         JLabel lblNewLabel_1 = new JLabel("Angka 1");
•         lblNewLabel_1.setBounds(33, 61, 45, 13);
•         contentPane.add(lblNewLabel_1);
•
•         JLabel lblNewLabel_2 = new JLabel("Angka 2");
•         lblNewLabel_2.setBounds(33, 101, 45, 13);
•         contentPane.add(lblNewLabel_2);
•
•         JLabel lblNewLabel_3 = new JLabel("Operator");
•         lblNewLabel_3.setBounds(33, 154, 45, 13);
•         contentPane.add(lblNewLabel_3);
•
•         JLabel lblNewLabel_4 = new JLabel("Hasil");
•         lblNewLabel_4.setBounds(33, 195, 45, 13);
•         contentPane.add(lblNewLabel_4);
•
•         txtAngka1 =
•             new JTextField();
•         txtAngka1.setBounds(102, 58, 96, 19);
•         contentPane.add(txtAngka1);
•         txtAngka1.setColumns(10);

```

```

•
•
•     txtAngka2 = new JTextField();
•     txtAngka2.setColumns(10);
•     txtAngka2.setBounds(102, 95, 96, 19);
•     contentPane.add(txtAngka2);
•
•
•     txtHasil = new JTextField();
•     txtHasil.setEditable(false);
•     txtHasil.setColumns(10);
•     txtHasil.setBounds(102, 192, 96, 19);
•     contentPane.add(txtHasil);
•
•
•     JComboBox cdOperator = new JComboBox();
•     cdOperator.setModel(new DefaultComboBoxModel(new String[]
•     {"+", "-",
•      "*",
•      ":",
•      "%" }));
•     cdOperator.setBounds(102, 150, 96, 21);
•     contentPane.add(cdOperator);
•
•
•     JButton btnNewButton = new JButton("Hitung");
•     btnNewButton.addActionListener(new ActionListener() {
•         int Hasil;
•         public void actionPerformed(ActionEvent e) {
•             if(txtAngka1.getText().trim().isEmpty()) {
•                 pesanPeringatan("Inputkan angka1");
•             }else if(txtAngka2.getText().trim().isEmpty()) {
•                 pesanPeringatan("Inputkan Angka2");
•             }else {
•                 try {
•
•                     int a = Integer.valueOf(txtAngka1.getText());
•                     int b = Integer.parseInt(txtAngka2.getText());
•                     int c = cdOperator.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;
•                     txtHasil.setText (String.valueOf(Hasil));
•                 } catch (NumberFormatException ex) {
•                     pesanError("Angka1 dan Angka 2 tidak
• valid");
•                 }
•             }
•         }
•     });
•     btnNewButton.setBounds(208, 150, 85, 21);
•     contentPane.add(btnNewButton);
•
•
• }
• }
```

BAB III

KESIMPULAN

3.1 KESIMPULAN

Berdasarkan hasil praktikum Pemrograman GUI 1, dapat disimpulkan bahwa penggunaan Java Swing menjadi dasar penting dalam pembuatan antarmuka aplikasi berbasis desktop. Mahasiswa dapat memahami struktur dasar GUI seperti JFrame, JPanel, JButton, JLabel, dan komponen lainnya yang berfungsi sebagai elemen visual. Melalui penerapan event handling, program dapat merespon berbagai aksi pengguna sehingga aplikasi menjadi interaktif dan mudah digunakan. Pemahaman ini memberikan gambaran awal mengenai bagaimana sebuah aplikasi desktop bekerja melalui konsep event-driven programming.

Dengan bantuan WindowBuilder, proses perancangan antarmuka menjadi lebih mudah karena dapat dilakukan secara visual dengan metode drag-and-drop. Hal ini mempermudah penempatan komponen, pengaturan layout, serta mempercepat proses pembuatan tampilan aplikasi. Secara keseluruhan, praktikum ini membantu mahasiswa memahami dasar pengembangan GUI secara lebih praktis dan terstruktur, sehingga dapat menjadi bekal untuk membangun aplikasi Java yang lebih kompleks pada tahap selanjutnya.

3.2 SARAN

- Diharapkan mahasiswa dapat lebih banyak berlatih membuat desain GUI sendiri untuk meningkatkan kreativitas dan pemahaman terhadap komponen Swing.
- Penggunaan WindowBuilder sebaiknya diimbangi dengan pemahaman kode manual agar mahasiswa memahami struktur program Java GUI dengan lebih baik.
- Pada praktikum selanjutnya, disarankan untuk mempelajari integrasi GUI dengan logika program yang lebih kompleks, serta memahami konsep layout manager yang lebih profesional.

- Asisten atau pengajar dapat menambahkan latihan yang lebih beragam agar mahasiswa dapat terbiasa mengatasi berbagai masalah yang muncul dalam pengembangan GUI.

DAFTAR PUSTAKA

- [1] Oracle. *Java Swing Tutorial*. Oracle Documentation.
- [2] Eclipse Foundation. *WindowBuilder Pro Documentation*.
- [3] Deitel, Paul & Deitel, Harvey. *Java: How to Program*. Pearson Education, 2017.
- [4] Tutorialspoint. *Java Swing Tutorial*.