

**LAPORAN PRATIKUM
PEMROGRAMAN ALGORITMA PEMROGRAMAN**

**LAPORAN PRAKTIKUM PEMROGRAMAN JAVA:
PEMBUATAN GUI (Graphical User Interface) MENGGUNAKAN
WindowBuilder PADA JAVA**

disusun oleh:

Khairunnisa M.

NIM 2511532005

Dosen Pengampu:

DR. Wahyudi, S. T., M. T.

Asisten Pratikum:

Aufan Taufiqurrahman



**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Puji syukur penulis kehadirat Allah SWT karena berkat rahmat, taufik, dan hidayah-Nya, laporan praktikum yang berjudul “Pembuatan GUI Menggunakan WindowBuilder pada Java” ini dapat diselesaikan dengan baik. Laporan ini disusun untuk memenuhi tugas praktikum pada mata kuliah Pemrograman berbasis objek serta sebagai dokumentasi proses pembelajaran mengenai pembuatan antarmuka grafis menggunakan Java Swing dan WindowBuilder.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan, baik dari segi penulisan maupun isi. Oleh karena itu, kritik dan saran yang membangun sangat penulisan harapkan demi kesempurnaan laporan di masa mendatang. Penulis juga mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta semua pihak yang telah memberikan bimbingan dan dukungan selama proses praktikum hingga penyusunan laporan ini.

Padang, 18 November 2025

Khairunnisa M.

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum	1
1.3 Manfaat Praktikum	1
BAB II	3
PEMBAHASAN	3
2.1. Pembahasan Materi	3
2.2. operatorAritmatikaGUI_2511532005.java	3
2.3. Analisa pemrograman	8
BAB III	10
KESIMPULAN DAN SARAN	10
DAFTAR PUSTAKA	11
Bibliography	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi menuntut para pengembang perangkat lunak untuk menghasilkan aplikasi yang tidak hanya berfungsi dengan baik, tetapi juga memiliki tampilan antarmuka yang mudah digunakan dan menarik. Salah satu teknologi yang digunakan dalam pengembangan aplikasi desktop adalah Java dengan library Swing, yang menyediakan berbagai komponen untuk membuat Graphical User Interface (GUI). Namun, pembuatan GUI secara manual melalui penulisan kode sering kali memakan waktu dan membutuhkan ketelitian tinggi.

Untuk mempermudah proses tersebut, hadir plugin WindowBuilder pada lingkungan pengembangan Eclipse. WindowBuilder memungkinkan pembuatan GUI secara visual menggunakan metode *drag and drop*, sehingga proses desain menjadi lebih cepat, efisien, dan mudah dipaham oleh pemula. Oleh karena itu, praktikum ini dilakukan agar mahasiswa mampu memahami konsep dasar pembuatan GUI serta memanfaatkan WindowBuilder dalam membangun aplikasi berbasis Java.

1.2 Tujuan Praktikum

1. Memahami konsep dasar Graphical User Interface (GUI) pada Java.
2. Mampu menggunakan WindowBuilder untuk mendesain tampilan GUI secara visual.
3. Mengetahui fungsi dan penggunaan komponen-komponen dasar Swing seperti JFrame, JLabel, JTextField, JButton, dan JPanel.
4. Mengimplementasikan event handling untuk menangani interaksi pengguna pada aplikasi GUI.
5. Menghasilkan program sederhana berbasis GUI sesuai langkah-langkah praktikum.

1.3 Manfaat Praktikum

1. Mahasiswa memperoleh pemahaman praktis tentang cara membuat aplikasi desktop dengan tampilan antarmuka menggunakan Java.
2. Mahasiswa dapat menguasai penggunaan WindowBuilder sebagai alat bantu dalam perancangan GUI secara efektif dan efisien.
3. Praktikum ini meningkatkan keterampilan pemrograman mahasiswa, khususnya dalam menghubungkan tampilan GUI dengan logika program.

4. Mahasiswa mampu mengembangkan aplikasi sederhana yang interaktif dan mudah digunakan.
5. Praktikum menjadi dasar untuk mempelajari pengembangan aplikasi lebih lanjut seperti validas input, manajemen event, hingga pengembangan aplikasi skala besar.

BAB II

PEMBAHASAN

2.1. Pembahasan Materi

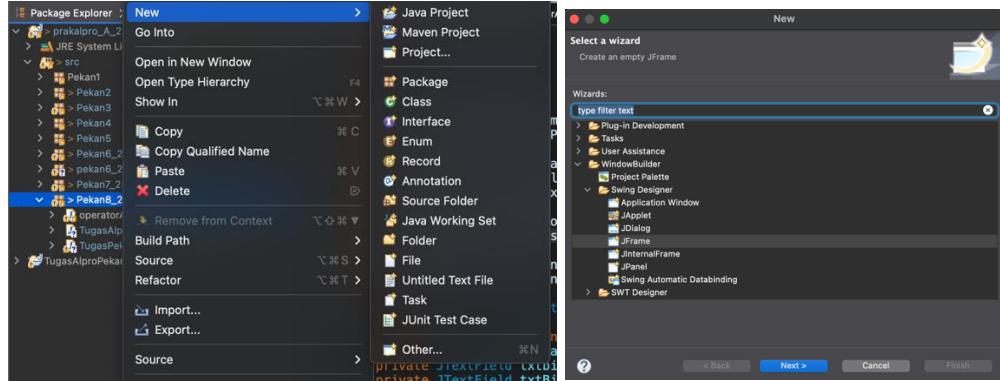
Operator aritmatika merupakan simbol-simbol yang digunakan dalam bahasa pemrograman untuk menjalankan operasi matematika dasar, seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus (sisa bagi). Dalam Java, operator aritmatika yang sering digunakan yaitu: (+) penjumlahan, (-) pengurangan, (*) perkalian, (/) pembagian, dan (%) modulus untuk menghitung sisa hasil pembagian. Operator-operator ini menjadi dasar penting dalam membangun logika perhitungan pada berbagai aplikasi.

Dalam konteks pembuatan GUI (*Graphical User Interface*) menggunakan WindowBuilder pada Java, operator aritmatika dapat diintegrasikan ke dalam aplikasi berbasis Swing untuk memberikan pengalaman interaktif kepada pengguna. Dengan memanfaatkan fitur *drag and drop* yang disediakan WindowBuilder, mahasiswa dapat menambahkan komponen seperti JTextField, JComboBox, JButton, dan JLabel untuk merancang antarmuka secara lebih intuitif. Ketika pengguna memasukkan dua angka dan memilih jenis operasi melalui JComboBox, program akan memproses perhitungan setelah tombol “Proses” ditekan. Proses ini melibatkan penerapan event handling menggunakan *ActionListener* pada tombol, serta penambahan validasi input untuk memastikan data yang dimasukkan benar, misalnya memastikan input berupa angka dan mencegah kesalahan seperti pembagian oleh nol. Jika terjadi kesalahan, aplikasi dapat memberikan peringatan melalui JOptionPane, sehingga penggunaan aplikasi tetap aman dan mudah dipahami.

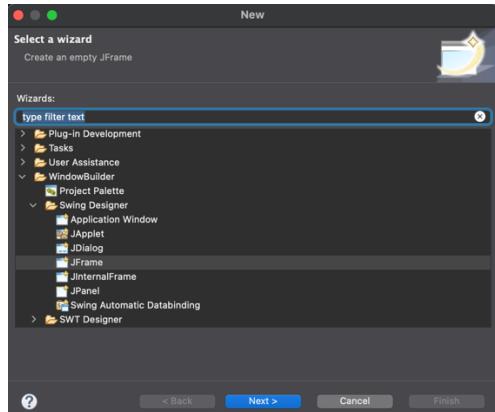
2.2. operatorAritmatikaGUI_2511532005.java

Dilakukanlah praktikum penggunaan GUI di operator aritmatika untuk memperdalam pemahaman dan mengetahui pengimplementasian nya. Berikut langkah langkah pembuatan aplikasi GUI “operatorAritmatikaGUI” menggunakan WindowBuilder pada Eclipse.

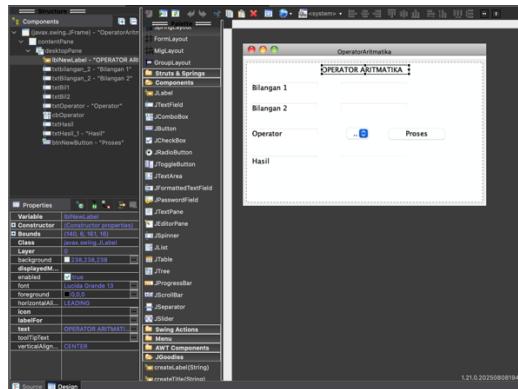
1. Buat kelas JFrame Baru:
 - a. Buka Eclipse dan buat proyek Java baru.
 - b. Klik kanan pada proyek, pilih “New” > “Other” > “WindowBuilder” > “Swing Designer” > “JFrame”.



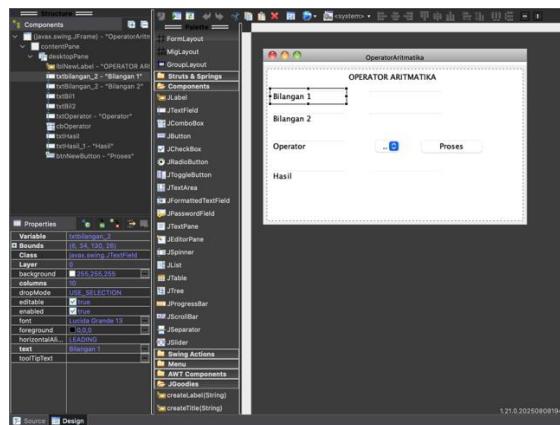
- c. Beri nama kelas, “operatorAritmatikaGUI_NIM”
2. Desain Layout GUI:
- Klik “Design” di bar bawah.
 - Lalu klik “Layouts” > “Absolute layout” > klik kearah template di bagian kanan.



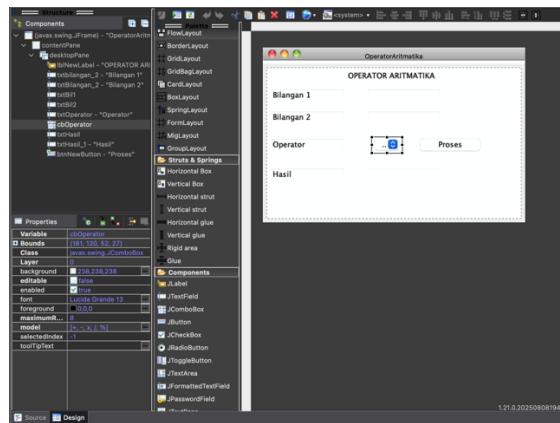
- Klik “Components” > “JLabel”. Lalu ke “Properties” > tulis judulnya Operasi Aritmatika” > “Variabel” > isi dengan “lblNewLabel” > “VerticalAlignment” > isi dengan “CENTER”.



- d. Lanjut ke bagian terpenting untuk input dan outputnya. Klik “JTextField” > arahkan ke template dan klik” > “text” isi dengan “Bilangan 1” > ke “Properties” > ganti “Variabel” sesuai dengan di layout .



- e. Lalkukan hal yang sama kepada Bilangan 2, Operator, kolom input Bilangan 1, kolom input Bilangan 2, Hasil dan kolom output Hasil.
- f. Lanjut ke “Components” > “Jcombobox” > klik kearah layouts > ubah “Variabel” > dan terakhir isi “model” dengan (+,-,*,/,%). Ini di buat untuk memilih operasi apa yang diinginkan pengguna untuk dilakukan.



- g. Lanjut klik “components” > “JButton” > klik kea rah layouts > “text” isi dengan Proses. Hal ini dilakukan untuk membuat tombol memulai operasi aritmatika.
- h. Kembali lagi ke Source
- i. Masukkan kode pemrograman dimana berisi pesanPeringatan mengenai kesalahan input.

```

22     private static final long serialVersionUID = 1L;
23     private JPanel contentPane;
24     private JTextField txtbilangan_2;
25     private JTextField txtbilangan_2;
26     private JTextField txtBil1;
27     private JTextField txtBil2;
28     private JTextField txtOperator;
29     private JTextField txtHasil;
30
31     private void pesanPeringatan(String pesan){
32         JOptionPane.showMessageDialog (this, pesan,"Peringatan", JOptionPane.WARNING_MESSAGE);
33     }
34     private void pesanError(String pesan) {
35         JOptionPane.showMessageDialog(this, pesan,"Kesalahan", JOptionPane.WARNING_MESSAGE);
36     }
37
38     private JTextField txtHasil_1;
39     private JButton btnNewButton;

```

- j. Lalu lanjut masukkan logika aritmatika atau rumus aritmatika, agar pemrograman dapat memproses input dari pengguna.

```

126      btnNewButton = new JButton("Proses");
127      btnNewButton.setBounds(257, 119, 117, 29);
128      btnNewButton.addActionListener(new ActionListener() {
129          int hasil;
130          public void actionPerformed(ActionEvent e) {
131              if (txtBil1.getText().trim().isEmpty()) {
132                  pesanPeringatan ("Silahkan Isi Bilangan 1");
133              } else if (txtBil2.getText().trim().isEmpty()) {
134                  pesanPeringatan ("Silahkan Isi Bilangan 2");
135              } else if (txtBil2.getText().trim().startsWith("0")) {
136                  pesanPeringatan ("Bilangan 2 tidak boleh angka 0");
137              } else {
138                  try {
139                      int b = Integer.parseInt(txtBil2.getText());
140                      int a = Integer.parseInt(txtBil1.getText());
141                      int cbOperator.getSelectedIndex();
142                      if (c==1) {hasil= a+b;};
143                      if (c==2) {hasil= a-b;};
144                      if (c==3) {hasil= a*b;};
145                      if (c==4) {hasil= a/b;};
146                      if (c==5) {hasil= a%b;};
147                  } catch (NumberFormatException ex) {
148                      pesanError ("Bilangan 1 dan bilangan 2 harus Angka");
149                  }
150                  int a = Integer.parseInt(txtBil1.getText());
151                  int b = Integer.parseInt(txtBil2.getText());
152                  int c = cbOperator.getSelectedIndex();
153
154                  if(c==1) {
155                      hasil= a+b;
156                  }
157                  if(c==2) {
158                      hasil= a-b;
159                  }
160                  if(c==3) {
161                      hasil= a*b;
162                  }
163                  if(c==4) {
164                      hasil= a/b;
165                  }
166                  if(c==5) {
167                      hasil= a%b;
168                  }
169
170                  txtHasil.setText(String.valueOf(hasil));
171
172              });
173      });
174      desktopPane.add(btnNewButton);
175
176
177

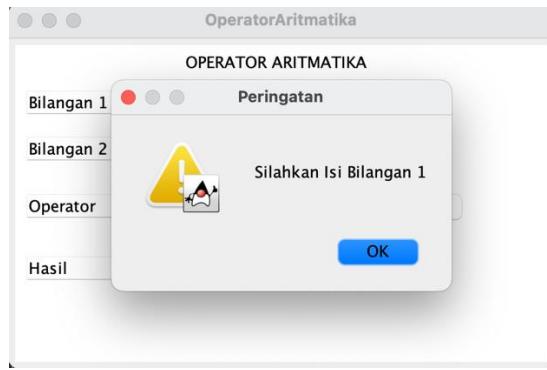
```

k. Pemrograman sudah selesai dan bisa di uji coba.

l. Hasil pemrograman benar:



m. Hasil pemrograman kurang input:



Dengan mengikuti langkah-langkah diatas, mahasiswa sudah bisa bikin aplikasi GUI sederhana untuk operasi aritmatika menggunakan WindowBuilder di Java. Mulai dari desain layout hingga heading, semuanya bisa dilakukan secara visual dan mudah.

2.3. Analisa pemrograman

Implementasi kode program untuk aplikasi GUI Operator Aritmatika menunjukkan penggunaan efektif dari komponen Swing dalam Java, di mana kelas utama ‘operatorAritmatikaGUI_2511532005’ yang mewarisi JFrame berfungsi sebagai wadah utama untuk elemen-elemen antarmuka seperti JTextField, JComboBox, dan JButton. Struktur kode ini memanfaatkan event handling melalui ActionListener pada tombol “Proses” untuk mengeksekusi logika perhitungan aritmatika berdasarkan indeks JComboBox, dengan validasi input yang mencakup pemeriksaan terhadap nilai kosong, pembagian oleh nol, dan format angka. Kelebihan utama dari desain ini terletak pada kemudahan penggunaan WindowBuilder untuk pembuatan layout visual, serta integrasi JOptionPane untuk memberikan umpan balik kepada pengguna, sehingga aplikasi menjadi lebih efektif dan ramah pengguna. Secara keseluruhan, kode ini berhasil mendemonstrasikan konsep aritmatika dalam konteks GUI, memungkinkan mahasiswa memahami bagaimana teori matematika diterapkan secara praktis dalam pemrograman.

Namun, analisis mendalam mengungkap beberapa kekurangan yang perlu diperhatikan, seperti adanya duplikasi kode dalam metode actionPerformed yang dapat menyebabkan inkonsistensi hasil perhitungan, serta kesalahan spesifik pada operator

modulus yang menggunakan simbol & alih-alih %, yang berpotensi menghasilkan output yang salah. Selain itu, penggunaan JTextField sebagai label statis kurang efisien dan rentan terhadap kesalahan input pengguna, sementara layout absolute dengan setBounds kurang responsif terhadap perubahan ukuran jendela. Saran perbaikan meliputi penghapusan kode duplikat, perbaikan operator, serta migrasi ke layout manager yang lebih fleksibel seperti GridBagLayout, agar aplikasi lebih robust dan sesuai standar praktikum. Dengan langkah-langkah ini, kode dapat dioptimalkan untuk memberikan pengalaman pembelajaran yang lebih baik bagi mahasiswa.

BAB III

KESIMPULAN DAN SARAN

Berdasarkan praktikum yang telah dilakukan mengenai pembuatan Graphical User Interface (GUI) menggunakan WindowBuilder pada Java, dapat disimpulkan bahwa proses pembuatan antarmuka menjadi lebih mudah dan efisien melalui pendekatan visual design. WindowBuilder memberikan kemudahan bagi mahasiswa untuk menempatkan komponen GUI seperti JLabel, JTextField, JButton, dan JComboBox secara drag-and-drop tanpa harus menuliskan seluruh kode secara manual. Selain itu, praktikum ini membantu mahasiswa memahami bagaimana komponen GUI terhubung dengan logika program melalui penerapan event handling, seperti penggunaan ActionListener pada tombol. Dengan demikian, praktikum ini berhasil meningkatkan pemahaman mahasiswa baik pada aspek desain antarmuka maupun implementasi logika dasar dalam aplikasi berbasis Java Swing. [1]

Untuk pengembangan pembelajaran di masa mendatang, disarankan agar mahasiswa lebih banyak berlatih membuat variasi desain GUI dan mencoba menggunakan berbagai layout manager agar lebih memahami fleksibilitas yang disediakan Java Swing. Selain itu, akan lebih baik apabila praktikum selanjutnya mencakup topik lanjutan seperti validasi input yang lebih kompleks, penyimpanan data, atau integrasi dengan database, sehingga aplikasi yang dibuat menjadi lebih fungsional. Pemberian contoh kasus yang lebih beragam juga dapat membantu mahasiswa memahami penerapan GUI pada situasi nyata dan meningkatkan kemampuan problem-solving dalam pengembangan perangkat lunak.

DAFTAR PUSTAKA

Bibliography

- [V. Siahaan and R. H. Sianipar, "TUTORIAL JAVA: GUI, GRAFIKA, DAN ANIMASI,"
1 Sparta Publishing, 23 Mei 2019. [Online]. Available:
] [https://books.google.co.id/books?hl=en&lr=&id=hNuZDwAAQBAJ&oi=fnd&pg=PP2&dq=PEMBUATAN+GUI+\(Graphical+User+Interface\)+MENGGUNAKAN+WindowBuilder+PADA+JAVA&ots=MOQf_7AFeG&sig=rpk-AqB2Mxc5GGfzHpwMLoAZfVY&redir_esc=y#v=onepage&q&f=false](https://books.google.co.id/books?hl=en&lr=&id=hNuZDwAAQBAJ&oi=fnd&pg=PP2&dq=PEMBUATAN+GUI+(Graphical+User+Interface)+MENGGUNAKAN+WindowBuilder+PADA+JAVA&ots=MOQf_7AFeG&sig=rpk-AqB2Mxc5GGfzHpwMLoAZfVY&redir_esc=y#v=onepage&q&f=false).