

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGARAMAN
PEMROGRAMAN GUI 1

Disusun Oleh:

Khairun Nisa

2511532015

Dosen Pengampu:

DR. WAHYUDI, S.T, M.T

Asisten Pratikum:

Aufan Taufiqurrahman



FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
2025

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas terselesaikannya laporan praktikum Algoritma dan Pemrograman ini. Laporan ini disusun sebagai dokumentasi hasil kegiatan praktikum yang telah dilaksanakan pada 18 November 2025, dengan pembahasan pemrograman GUI 1.

Penulis mengucapkan terima kasih kepada Bapak Dosen serta asisten laboratorium yang telah membimbing selama praktikum berlangsung. Semoga laporan ini dapat bermanfaat bagi pembaca dan menjadi referensi untuk perkembangan ilmu pengetahuan di bidang pemrograman.

Padang, 21 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Pratikum	1
1.3 Manfaat Pratikum	2
BAB II PEMBAHASAN	3
2.1 Instalasi PluginWindowBuilder di Eclipse	3
2.2 Pembuatan Aplikasi Operasi Aritmatika Berbaasis GUI.....	4
2.2.1 Desain GUI Aplikasi Operasi Aritmatika	5
2.2.2 Kode Program untuk Aplikasi Operasi Aritmatika Berbasis GUI	10
BAB III KESIMPULAN	15
3.1 Kesimpulan	15
DAFTAR PUSTAKA	16

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pengembangan perangkat lunak modern, kemampuan sebuah program untuk berinteraksi dengan pengguna melalui tampilan visual menjadi sangat penting. Aplikasi berbasis teks, meskipun mampu menjalankan logika dengan benar, sering kali kurang intuitif dan kurang menarik bagi pengguna awam. Oleh karena itu, Graphic User Interface (GUI) dikembangkan untuk memberikan pengalaman interaksi yang lebih alami, visual, dan responsif.

GUI (Graphical User Interface) adalah antarmuka pengguna yang menggunakan elemen-elemen grafis, seperti ikon, tombol, kontak input, dan indicator visual sebagai sarana interaksi antara pengguna dan program. Berbeda dengan antarmuka berbasis teks yang mengandalkan perintah baris kode, GUI memungkinkan pengguna berinteraksi dengan perangkat lunak melalui klik, ketik, atau gesekan, sehingga mudah dipahami dan digunakan.

Java GUI adalah pemrograman dengan bahasa Java dibuat menggunakan aplikasi yang berbasiskan GUI. Tujuannya adalah menambahkan komponen-komponen yang tidak bisa dibuat dalam basis teks. Komponen-komponen tersebut bisa berupa tombol, gambar, dll [1].

Pada praktikum algoritma dan pemrograman kali ini, mahasiswa diperkenalkan pada pembuatan aplikasi desktop berbasis GUI menggunakan bahasa pemrograman Java dengan bantuan Integrated Development Environment (IDE) Eclipse. Dengan fitur desain visual seperti WindowBuilder, mahasiswa dapat merancang tampilan aplikasi secara intuitif, lalu mengintegrasikannya dengan kode program untuk menjalankan operasi dasar.

1.2 Tujuan Praktikum

1. Melatih kemampuan membuat tampilan aplikasi desktop secara visual menggunakan fitur desain GUI di Eclipse.

2. Menerapkan konsep algoritma dasar ke dalam program yang memiliki antarmuka grafis.
3. Memahami cara menghubungkan desain tampilan dengan kode program agar aplikasi merespons input pengguna.
4. Menghasilkan aplikasi sederhana yang interaktif dan mudah digunakan, bukan hanya berbasis teks di konsol.

1.3 **Manfaat Praktikum**

1. Memberikan pengalaman membuat program yang lebih menarik dan nyata, mirip aplikasi yang sering dipakai sehari-hari.
2. Membantu memahami bahwa program itu tidak sekedar logika, tetapi juga membutuhkan tampilan yang bisa dipakai oleh orang lain.
3. Melatih kemampuan menggabungkan desain dan pemrograman.
4. Menjadi langkah awal untuk belajar membuat aplikasi desktop yang lebih kompleks di mata kuliah berikutnya.

BAB II

PEMBAHASAN

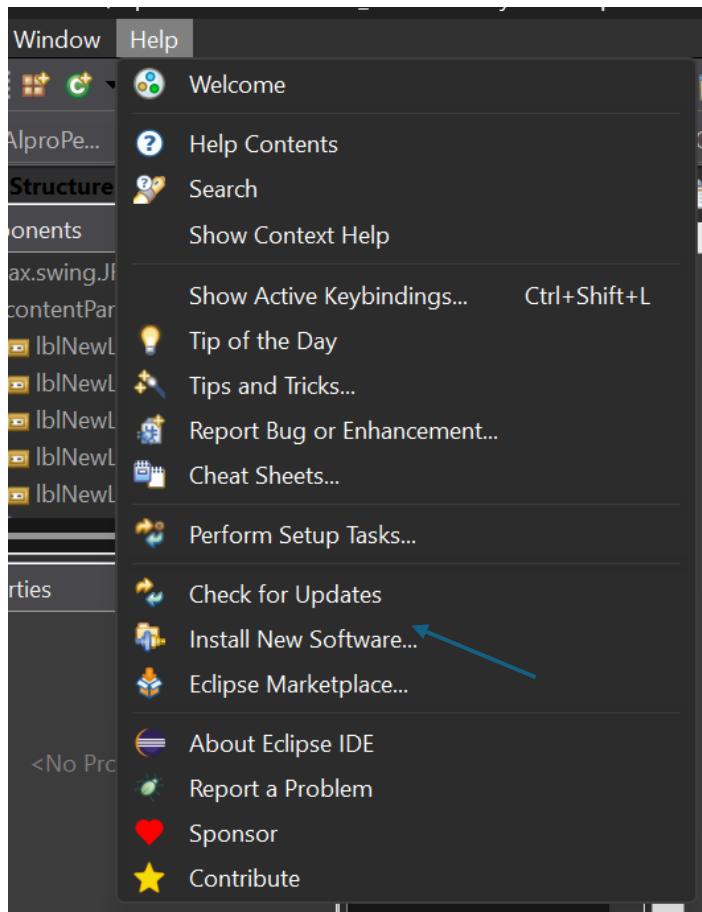
2.1 Instalasi Plugin WindowBuilder di Eclipse

Sebelum memulai praktikum, terlebih dahulu menginstal plugin WindowBuilder di Eclipse. Plugin ini tidak tersedia secara bawaan, jadi harus dipasang manual, dengan tujuan bisa mendesain tampilan aplikasi secara visual tanpa harus menulis semua kode tampilan dari nol.

WindowBuilder adalah perancang GUI Java dua arah yang kuat dan mudah digunakan untuk Eclipse lainnya. Memfasilitasi pembuatan antarmuka pengguna grafis (GUI) untuk aplikasi Java menggunakan toolkit Swing dan SWT.

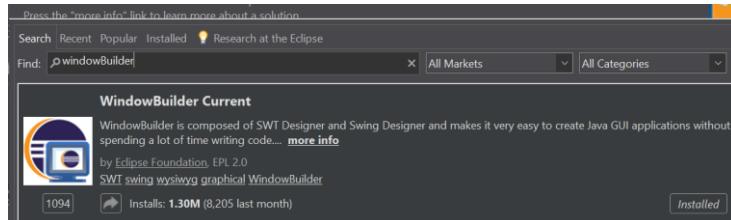
Berikut langkah-langkah meng-instal WindowBuilder di Eclipse:

1. Buka Eclipse, lalu masuk ke menu Help dan cari Eclipse Marketplace.



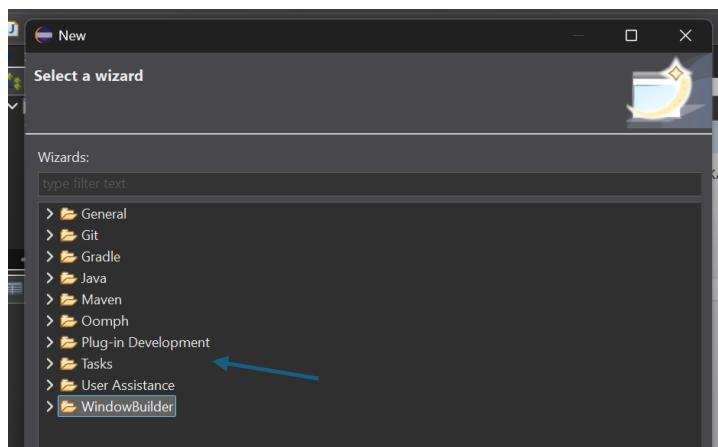
Gambar 2.1: Pilihan dalam menu Help

2. Kemudian cari WindowBuilder pada kolom pencarian, lalu instal



Gambar 2.2: WindowBuilder pada Eclipse Marketplace

3. Setelah proses pengunduhan dan instal selesai, Eclipse diminta untuk restart agar plugin bisa aktif. Setelah restart, cek apakah windowBuilder sudah terinstal atau belum dengan cara klik kanan lalu pilih opsi other, jika windowBuilder sudah ada maka penginstalan sukses dilakukan.



Gambar 2.3: WindowBuilder yang sudah terinstal

4. WindowBuilder siap untuk digunakan.

2.2 Pembuatan Aplikasi Operasi Aritmatika Berbasis GUI

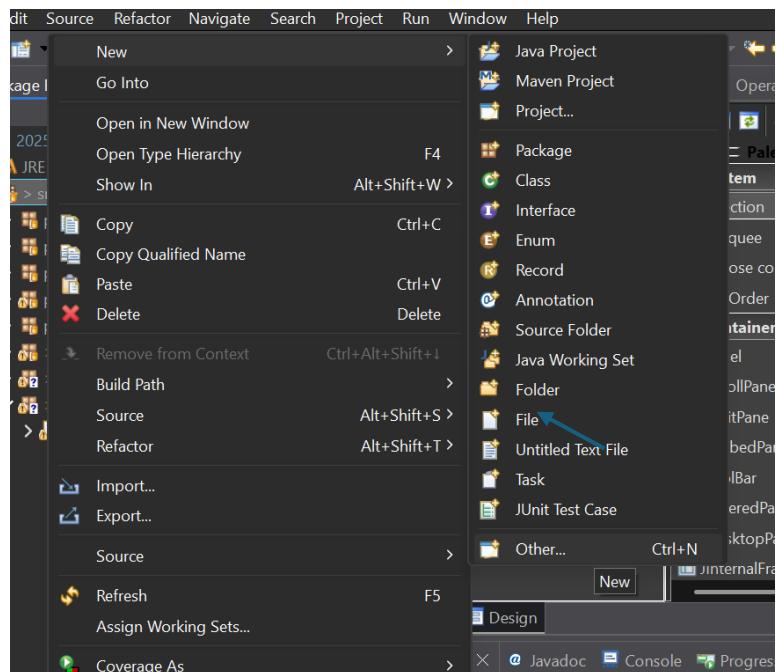
Aplikasi operasi aritmatika berbasis GUI adalah program sederhana yang memungkinkan pengguna melakukan perhitungan matematika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulo melalui tampilan visual. Berbeda dengan program berbasis teks yang hanya menampilkan input dan output di konsol, aplikasi ini menggunakan elemen-elemen grafis seperti kotak input, tombol, dan label hasil agar lebih interaktif dan mudah digunakan. Dalam praktikum ini, konsep tersebut

diwujudkan dengan memadukan desain antarmuka dan logika pemrograman dalam satu kesatuan aplikasi Java yang utuh.

2.2.1 Desain GUI Aplikasi Operasi Aritmatika

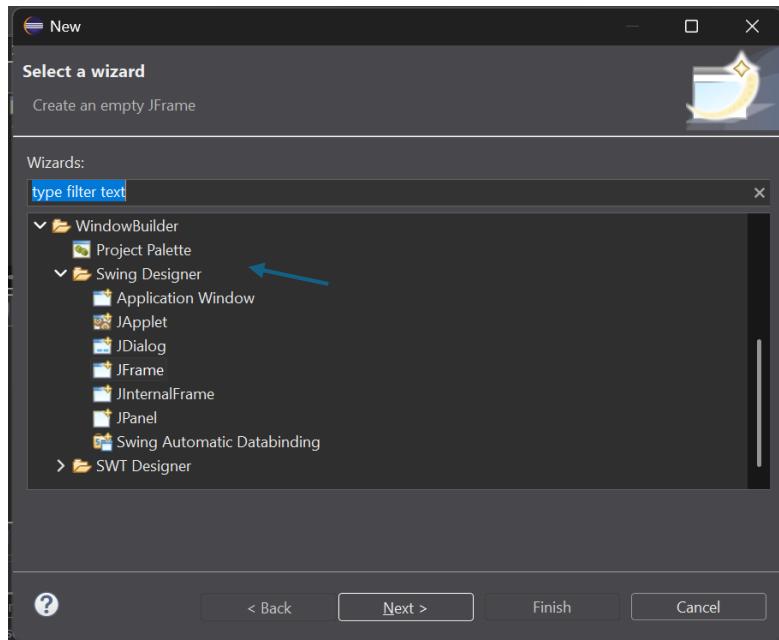
Desain GUI aplikasi aritmatika bertujuan untuk menyusun tampilan yang memudahkan pengguna berinteraksi dengan program. Pada tahap ini, kami belum menulis kode perhitungannya, melainkan focus pada penataan komponen visual seperti dua kotak input untuk memasukkan angka, tombol pilihan operasi perhitungan, dan bagian untuk menampilkan hasil. Tampilan dirancang agar sederhana, rapi, dan intuitif sehingga pengguna bisa langsung paham cara menggunakan tanpa perlu instruksi tambahan. Berikut langkah-langkah pembuatan desain GUI aplikasi Operasi Aritmatika:

1. Untuk mulai membuat desain aplikasi buat terlebih dahulu package baru dengan nama ‘package8_2511532015’, kemudian buat class baru dengan cara klik kanan pada package tersebut lalu pilih New dan pilih Other.



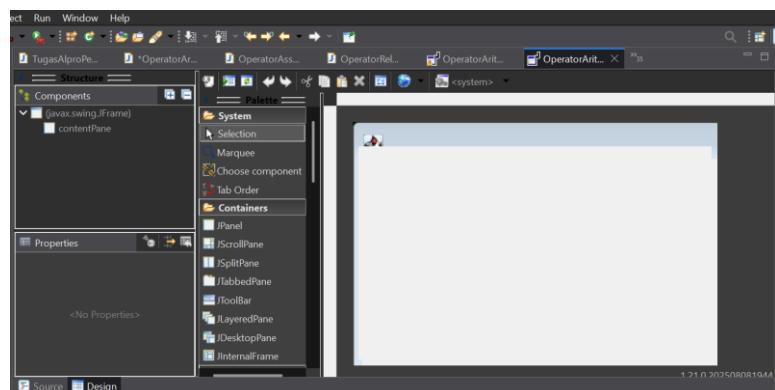
Gambar 2.4: Pilihan pada klik kanan package8

2. Pada jendela yang muncul cari WindowBuilder lalu pilih Swing Designer kemudian pilih JFrame, lalu pilih next. File tersebut diberi nama ‘OperatorAritmatikaGUI_2511532015’.



Gambar 2.5: Menu pilihan pada WindowBuilder

3. Setalah file diberi nama, Eclipse secara otomatis membuka tampilan desain visual berbasis WindowBuilder, dan siap digunakan untuk mulai Menyusun antarmuka aplikasi operasi aritmatika.



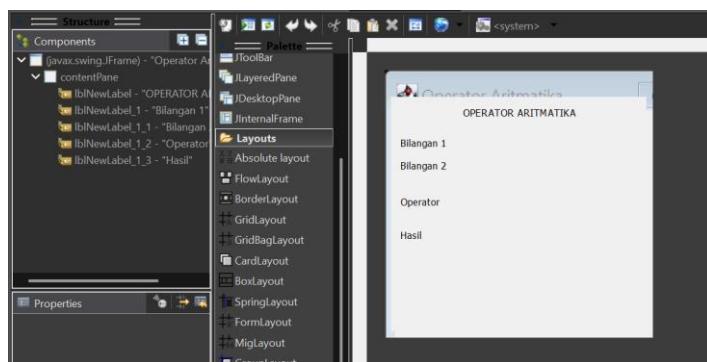
Gambar 2.6: Tampilan desain visual berbasis
WindowBuilder

4. Pertama, sesuaikan ukuruan jendela desain agar sesuai dengan kebutuhan tampilan aplikasi, dengan cara mengklik dan menarik sudut kanan bawah area desain ke arah dalam hingga

ukurannya terlihat cukup untuk menampung semua komponen tanpa terlalu besar atau terlalu kecil.

5. Setelah ukuran jendela pas, tambahkan lima buah label dengan cara drag and drop dari palet komponen ke dalam jendela desain.
 - Label pertama diberi teks “Operator Aritmatika” dan digunakan sebagai judul aplikasi, ubah Horizontal Alignment menjadi center agar tampilannya berada di tengah.
 - Label kedua diberi teks “Bilangan 1” sebagai petunjuk untuk input angka pertama.
 - Label ketiga diberi teks “Bilangan 2” sebagai petunjuk untuk input angka kedua.
 - Label keempat diberi teks “Operator” untuk menjelaskan pemilihan jenis operasi.
 - Label kelima diberi teks “Hasil” yang nantinya akan diperbarui secara otomatis saat program menjalankan hitungan.

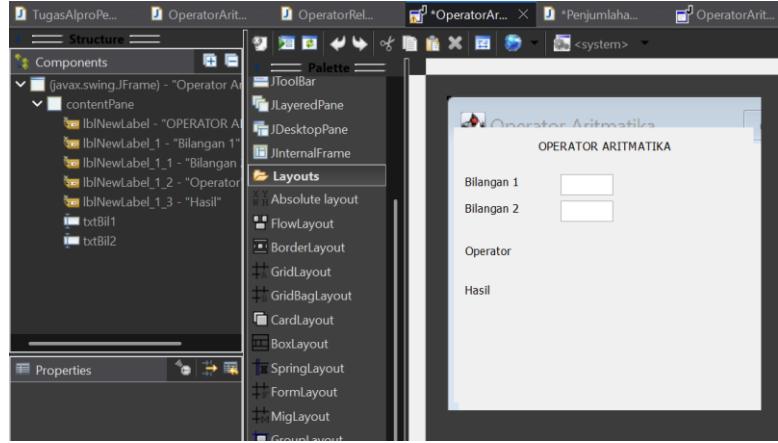
Semua label ditempatkan secara berurutan dan rapi.



Gambar 2.7: Tampilan desain dengan label: Judul, Bil 1, Bil 2, Operator, dan Hasil.

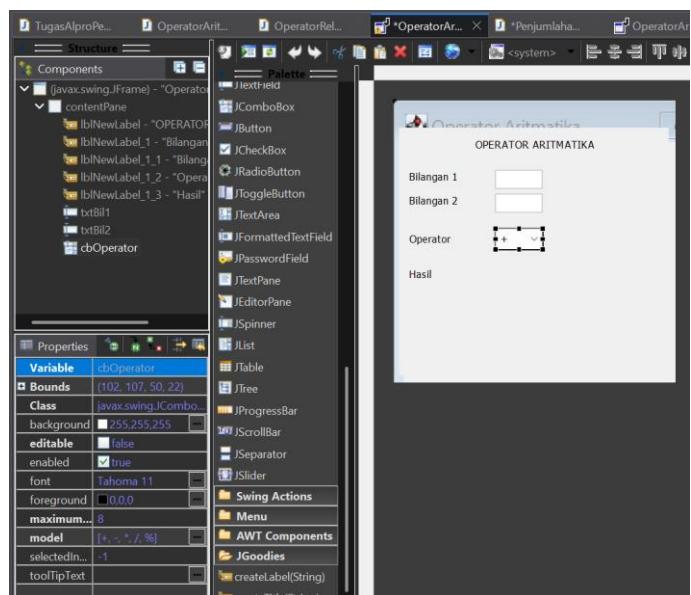
6. Di samping label Bilangan 1 dan Bilangan 2 tambahkan masing-masing satu tempat untuk meng-inputkan angkanya, yaitu dengan menambahkan JTextField ke dalam desain, ubah

nama variabelnya menjadi txtBil1 dan txtBil2 serta horizontal alignmentnya ke center agar teks input tampil rapi di tengah.



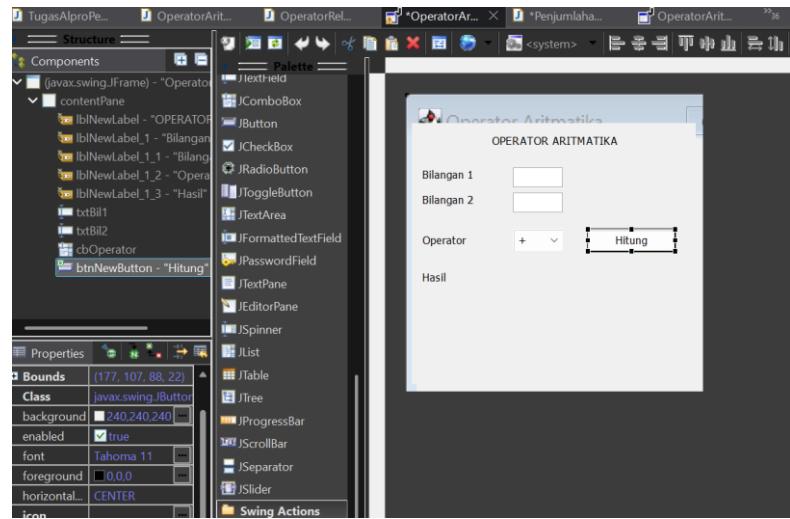
Gambar 2.8: Penambahan komponen JTextField untuk input bilangan

7. Tambahkan komponen JSpinner pada desain di samping label ‘Operator’ untuk memilih operator aritmatika (+, -, *, /, %), ubah variabelnya menjadi cbOperator. Agar input operator selalu valid, properti editable pada cbOperator diatur menjadi false, sehingga pengguna tidak dapat mengetik bebas dan hanya bisa memilih dari daftar pilihan saja. Daftar tersebut bersisi symbol-simbol yang ditambahkan melalui properti model (+, -, *, /, %) yang nantinya akan digunakan dalam proses perhitungan.



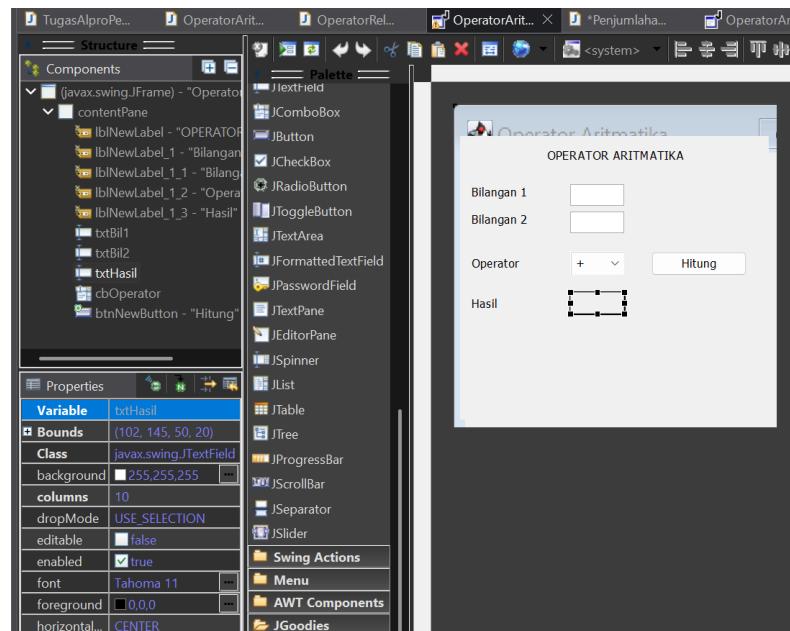
Gambar 2.9: Penambahan JSpinner pada desain

8. Di samping komponen cbOperatror, tambahkan sebuah JButton dengan variabelnya diubah menjadi ‘Hitung’ yang berfungsi untuk menjalankan proses perhitungan saat diklik oleh user.



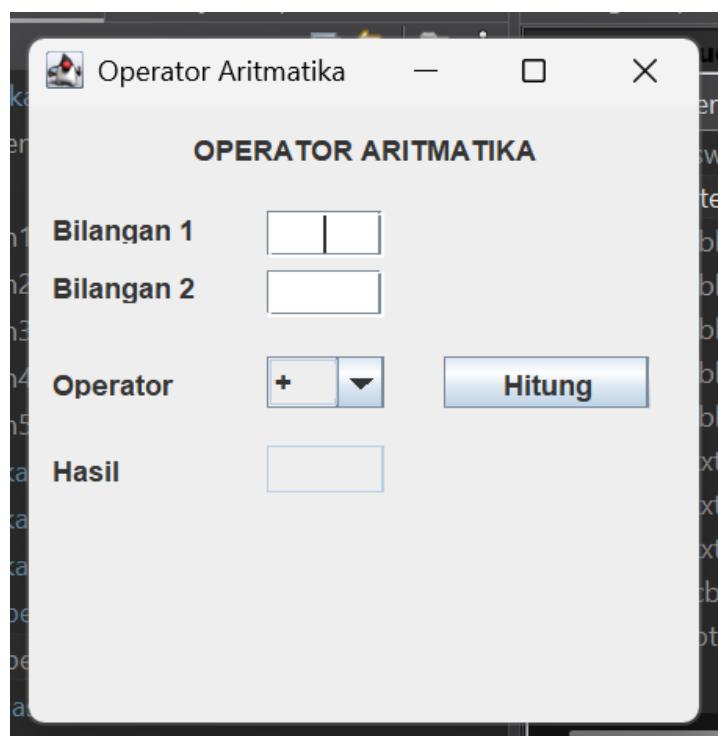
Gambar 2.10: Penambahan JButton

9. Di samping label ‘Hasil’ tambahkan juga komponen JTextField sebagai tempat menampilkan output perhitungan. JTextField ini diatur editable nya menjadi false supaya pengguna hanya bisa melihat hasil bukan mengubah secara manual.



Gambar 2.11: Penambahan komponen terakhir pada desain

10. Ketika dijalankan program tampilan antarmuka GUI akan muncul dengan dua kolom input (txtBil1 dan txtBil2) untuk memasukkan bilangan, pilihan operator melalui cbOperator, tombol ‘Hitung’, serta kolom txtHasil untuk menampilkan Output. Namun, fungsi perhitungan belum berjalan karena logika pemrograman belum ditambahkan dalam kode. Artinya, meskipun pengguna sudah mengisi bilangan dan memilih operator, klik pada tombol ‘Hitung’ belum menghasilkan apa-apa. Selanjutnya kami menulis kode program untuk menghubungkan antarmuka dengan logika operasi aritmatika.



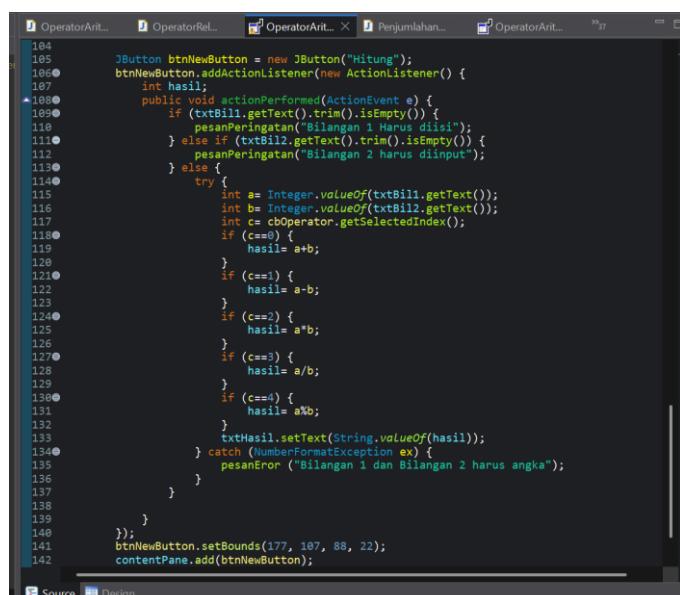
Gambar 2.12: Tampilan Output setelah dijalankan.

2.2.2 Kode Program untuk Operasi Aritmatika Berbasis GUI

Setelah antarmuka GUI dirancang pada tahap sebelumnya, langkah selanjutnya adalah mengimplementasikan logika program agar aplikasi dapat melakukan operasi aritmatika berdasarkan input pengguna. Pada bagian ini, ditambahkan kode program yang menghubungkan komponen GUI, seperti txtBil1, txtBil2,

cbOperator, hitung dengan fungsi perhitungan, sehingga hasil operasi dapat ditampilkan secara dinamis saat tombol ‘Hitung’ diklik. Berikut langkah-langkah dalam pembuatan kode programnya:

1. Sebelum kode logika perhitungan ditambahkan, program sudah memiliki kerangka dasar antarmuka GUI yang dibuat otomatis oleh IDE (seperti Eclipse WindowBuilder). Kerangka ini berisi deklarasi komponen seperti JFrame, JLabel, JTextField, JSpinner, dan JButton yang disusun secara visual.
2. Kode fungsional utama yaitu logika perhitungan saat tombol “Hitung” diklik belum ada maka tambahkan kode logika tersebut secara manual di bagian bawah deklarasi komponen, tepat sebelah bagian yang berkaitan dengan label ‘Hasi’ dan txtHasil. Penambahan ini berupa event listener pada tombol ‘Hitung’ yang berisi seluruh proses yaitu validasi input, pemilihan operator, perhitungan, dan penampilan hasil. Dengan demikian, program awal hanya menampilkan antarmuka sedangkan ‘otak’ nya beru aktif setelah kode tambahan dimasukkan.
3. Ketikkan kode tambahannya:



```

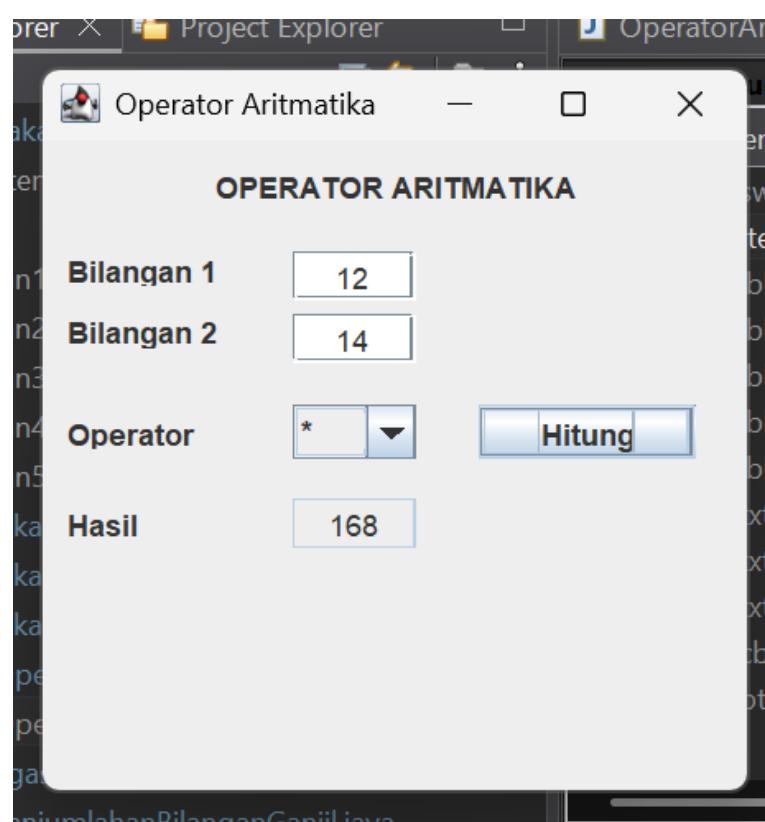
184
185
186 JButton btnNewButton = new JButton("Hitung");
187 btnNewButton.addActionListener(new ActionListener() {
188     int hasil;
189     public void actionPerformed(ActionEvent e) {
190         if (txtBill1.getText().trim().isEmpty()) {
191             pesanPeringatan("Bilangan 1 Harus diisi");
192         } else if (txtBill2.getText().trim().isEmpty()) {
193             pesanPeringatan("Bilangan 2 harus diinput");
194         } else {
195             try {
196                 int a = Integer.valueOf(txtBill1.getText());
197                 int b = Integer.valueOf(txtBill2.getText());
198                 int c = cbOperator.getSelectedIndex();
199                 if (c==0) {
200                     hasil= a+b;
201                 }
202                 if (c==1) {
203                     hasil= a-b;
204                 }
205                 if (c==2) {
206                     hasil= a*b;
207                 }
208                 if (c==3) {
209                     hasil= a/b;
210                 }
211                 if (c==4) {
212                     hasil= a%b;
213                 }
214             }
215             txtHasil.setText(String.valueOf(hasil));
216         } catch (NumberFormatException ex) {
217             pesanError ("Bilangan 1 dan Bilangan 2 harus angka");
218         }
219     }
220 });
221 btnNewButton.setBounds(177, 107, 88, 22);
222 contentPane.add(btnNewButton);
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242

```

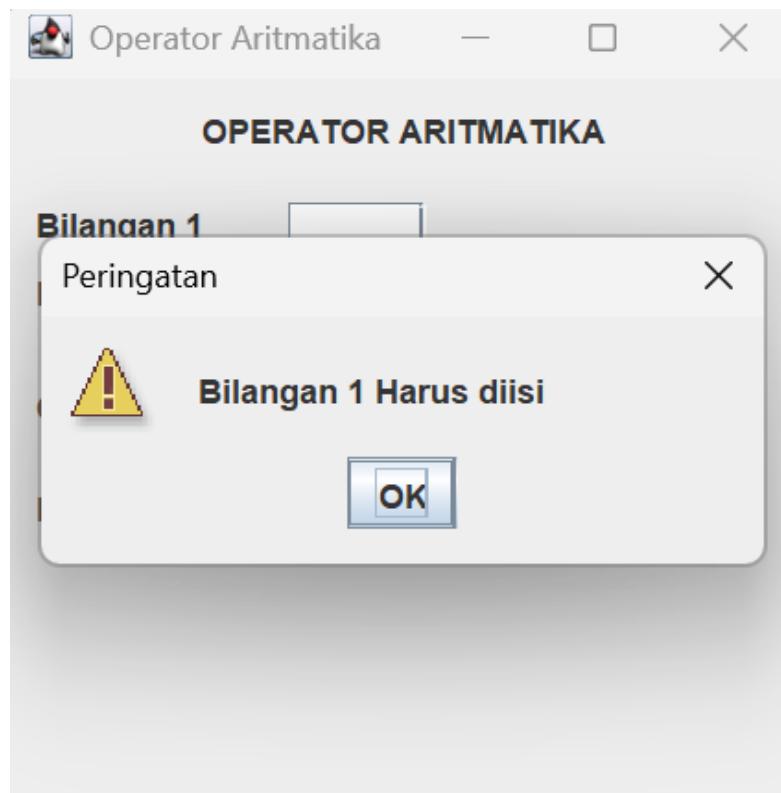
Gambar 2.13: Kode Program logika perhitungan

4. Pada bagian awal penanganan klik tombol ‘Hitung’, program terlebih dahulu memeriksa apakah kolom input Bilangan 1 dan Bilangan 2 telah diisi. Jika Bilangan 1 kosong, program akan menampilkan pesan peringatan “Bilangan 1 harus diisi. Begitupun sebaliknya, jika Bilangan 1 terisi dan Bilangan 2 kosong program tetap akan menampilkan pesan peringatan “Bilangan 2 harus diisi”. Pengecekan ini memastikan bahwa proses perhitungan hanya bisa dilanjutkan jika kedua kolom input telah berisi data.
5. Di dalam blok try, program pertama-tama mengambil teks yang dimasukkan pengguna dari kolom txtBil1 dan txtBil2, lalu mengubah kedua teks tersebut menjadi bilangan bulat (tipe data int) menggunakan perintah IntegerValueOf(). Proses ini diperlukan karena semua input dari pengguna selalu berupa teks, sedangkan operasi aritmatika hanya dapat dilakukan pada angka yang sebenarnya. Selanjutnya program membaca pilihan operator dari cbOperator melalui metode getSelectedIndex() yang menghasilkan nilai indeks berupa angka (0 untuk penjumlahan, 1 untuk pengurangan dan seterusnya). Nilai indeks ini disimpan dalam variabel c, lalu digunakan dalam rangkaian kondisi if untuk menentukan jenis operasi yang akan dijalankan.
6. Hasil perhitungan dari operasi yang sesuai kemudian disimpan dalam variabel hasil, yang selanjutnya diubah Kembali menjadi teks agar dapat ditampilkan di dalam kolom txtHasil.
7. Blok catch digunakan untuk menangani kemungkinan kesalahan saat mengonversi input teks menjadi angka, jika user masukkan data non numerik (seperti huruf atau simbol), program akan menampilkan pesan error “Bilangan 1 dan Bilangan 2 harus angka” sehingga aplikasi tetap stabil dan tidak crash.

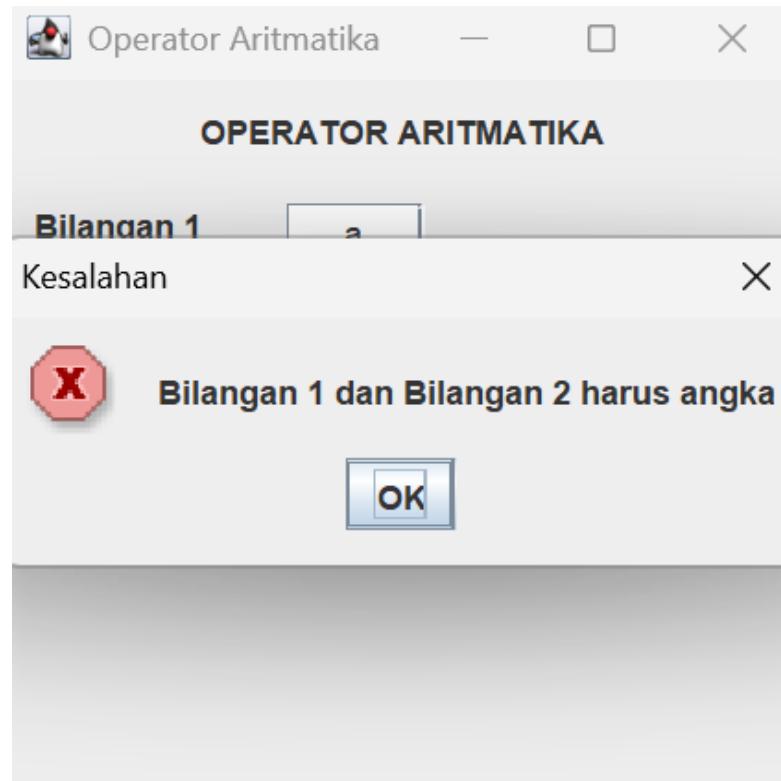
- Setelah kode program tambahan selesai, dan program dijalankan, tampilan GUI muncul dan menunggu input pengguna. Saat tombol “Hitung” diklik, program memvalidasi input, mengonversi teks ke angka, melakukan perhitungan sesuai operator yang dipilih, lalu menampilkan hasilnya atau menampilkan pesan error jika input tidak valid.



Gambar 2.14: Hasil output yang berhasil



Gambar 2.15: Hasil output pesan peringatan



Gambar 2.16: Hasil output pesan eror

BAB III

KESIMPULAN

3.1 Kesimpulan

Praktikum ini berhasil memperkenalkan konsep dasar pembuatan aplikasi desktop berbasis Graphic User Interface (GUI) menggunakan bahasa pemrograman Java dan IDE Eclipse dengan bantuan plugin WindowBuilder. Melalui praktikum ini, mahasiswa mampu merancang tampilan aplikasi secara visual dengan komponen-komponen GUI seperti JLabel, JTextField, JSpinner, dan JButton, lalu mengintegrasikannya dengan logika pemrograman untuk menjalankan operasi aritmatika dasar (penjumlahan, pengurangan, perkalian, pembagian, dan modulo).

Proses pembuatan aplikasi dimulai dari instalasi plugin WindowBuilder, desain antarmuka yang intuitif dan rapi, hingga penulisan kode program yang menghubungkan input pengguna dengan proses perhitungan. Aplikasi yang dihasilkan tidak hanya mampu melakukan perhitungan, tetapi juga dilengkapi fitur validasi input seperti pengcekan kelengkapan data dan konversi teks ke angka serta penanganan error untuk mencegah crash saat pengguna memasukkan data tidak valid.

Dengan demikian, praktikum ini tidak hanya melatih kemampuan teknis dalam pemrograman GUI, tetapi juga memperdalam pemahaman bahwa sebuah program yang baik harus menggabungkan logika yang akurat dengan antarmuka yang user-friendly, sehingga dapat digunakan secara efektif oleh pengguna awam. Hasil akhir berupa aplikasi operasi aritmatika berbasis GUI membuktikan bahwa Java mampu menghasilkan perangkat lunak interaktif yang menyerupai aplikasi nyata yang sering digunakan sehari-hari.

DAFTAR PUSATAKA

- [1] S. Hidayatullah, "Java GUI," [Online]. Available: <https://id.scribd.com/doc/98242962/java-GUI>. [Accessed 18 November 2025].