

**LAPORAN PRATIKUM PEKAN 8
ALGORITMA DAN PEMROGRAMAN**

“Pemograman GUI 1”



Dosen Pengampu:

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

Asisten Lab:

Muhammad Zaki Al Hafiz

Disusun Oleh:

Hamdi Sidqi Alifi

2511531009

Fakultas Teknologi Informasi

Departemen Informatika

Universitas Andalas

2025

KATA PENGANTAR

Puji dan syukur senantiasa penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan ini guna memenuhi laporan praktikum mata kuliah Algoritma Pemrograman, dengan judul: “Pemograman GUI 1”. Laporan ini berisikan hasil analisis dari praktikum pemograman java penulis di Praktikum Kelas D yang dilaksanakan pada hari Rabu, tanggal 19 November 2025.

Pada kesempatan ini, Penulis ingin mengucapkan terima kasih kepada Bapak DR. Wahyudi. .S.T.M.T, dosen pengampu mata kuliah Algoritma Pemrograman, yang telah memberikan tugas ini. Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah mendukung serta membantu penulis dalam penyelesaian laporan praktikum Algoritma Pemrograman.

Penulis sadar bahwa laporan ini masih memiliki beberapa kekurangan dikarenakan terbatasnya pengalaman dan pengetahuan yang penulis miliki. Oleh karena itu, penulis harap adanya bentuk saran dan kritik yang membangun dari berbagai pihak.

Akhir kata, penulis berharap laporan ini dapat bermanfaat bagi perkembangan dunia pendidikan.

Padang, 20 November 2025

Hamdi Sidqi Alifi
2511531009

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN	1
1.1Latar	Belakang
.....	1
1.2.....	Tujuan
.....	1
1.3Manfaat	Pratikum
.....	1
BAB II PEMBAHASAN.....	2
2.1	Teori
.....	2
2.2Kode	Pemrograman
.....	4
BAB III.....	KESIMPULAN
14	
3.1	Kesimpulan
.....	14
3.2.....	Saran
.....	14
DAFTAR PUSTAKA	3

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kehidupan sehari-hari, kita seringkali menggunakan aplikasi-aplikasi yang memiliki pengaruh besar terhadap hidup kita. Tidak jarang pula kita memperhatikan perubahan-perubahan tata letak tombol-tombol yang kita gunakan pada setiap pembaharuan (update) aplikasi.

Tata letak yang kita perhatikan setiap hari tersebut dapat disebut dengan *Graphical User Interface* (GUI). GUI mendeskripsikan bagaimana proses tatap muka pengguna dengan suatu perangkat lunak, tentu saja dengan disajikan dengan tampilan yang semenarik mungkin.

Pada Java, kita bisa mengembangkan GUI untuk membuat aplikasi kita menjadi lebih menarik, karena program kita memiliki output yang tidak hanya berbentuk tulisan melainkan terdapat visual bentuk yang interaktif. Untuk melakukan ini maka kita harus mengunduh sebuah GUI Designer untuk Java. Pada Eclipse terdapat sebuah plugin bernama WindowBuilder yang berguna untuk membuat GUI dengan menggunakan bahasa pemrograman Java.

1.2 Tujuan

1. Menjelaskan mengenai penggunaan plugin WindowBuilder pada perangkat IDE seperti Eclipse.
2. Menjelaskan cara mengembangkan sebuah aplikasi yang dapat memproses algoritma dasar dengan plugin WindowBuilder.

1.3 Manfaat Pratikum

1. Memahami konsep penggunaan WindowBuilder
2. Mampu memahami dan menerapkan aplikasi algoritma sederhana dengan WindowBuilder.

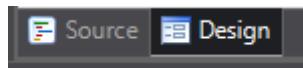
BAB II

PEMBAHASAN

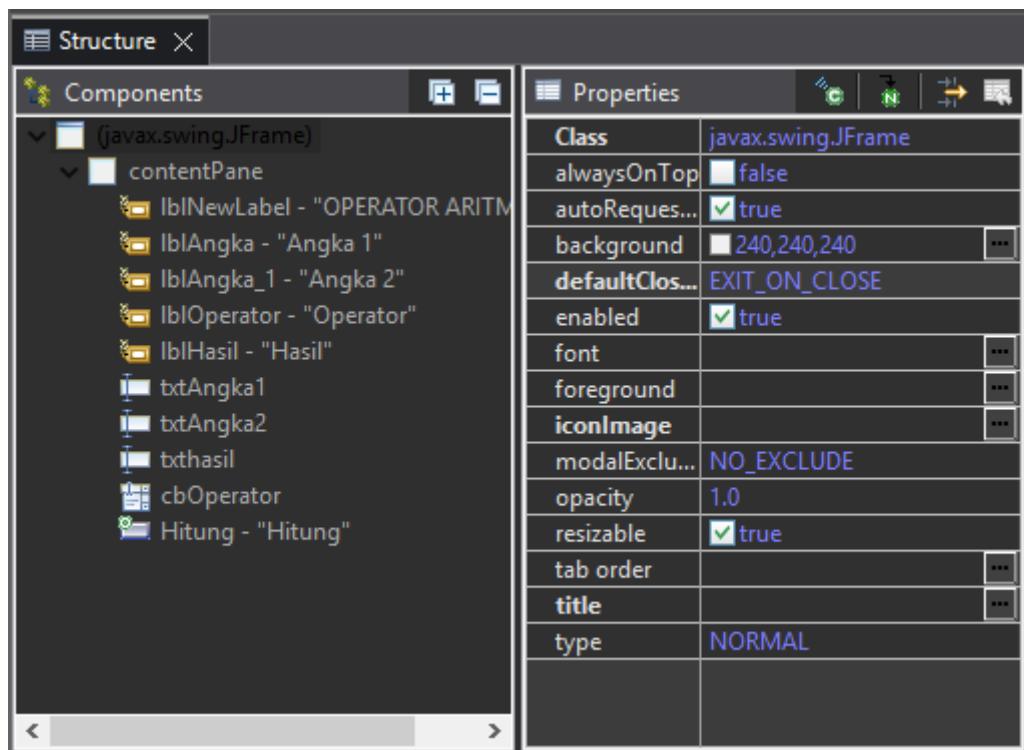
2.1 Teori

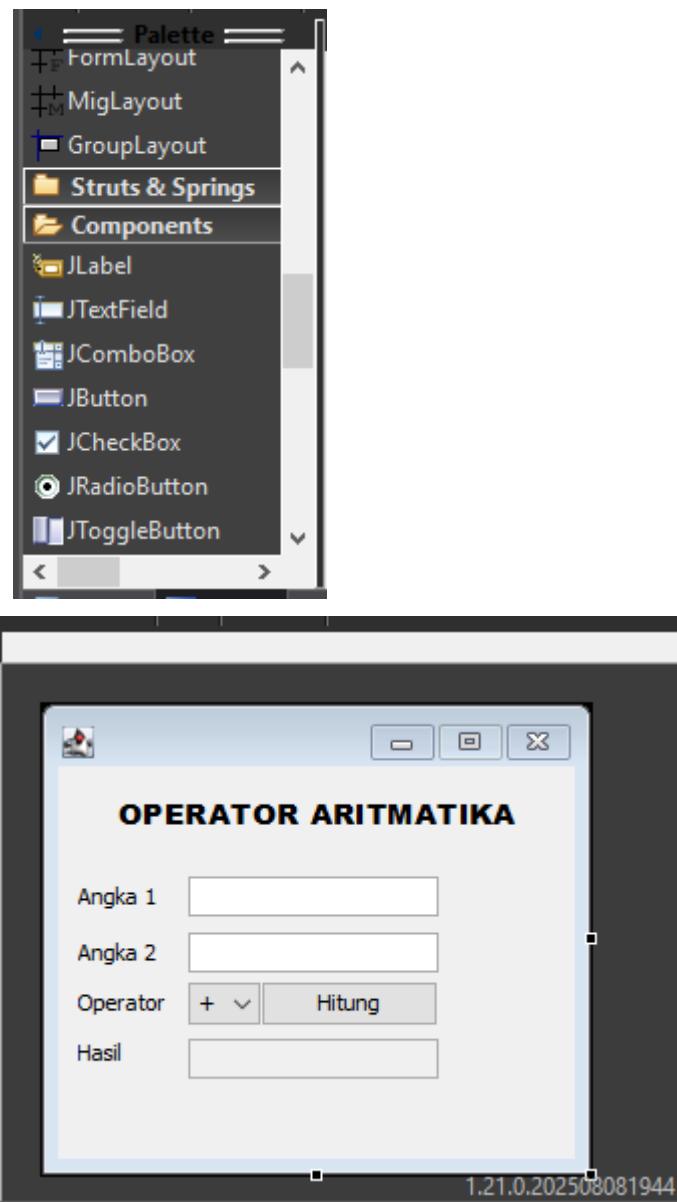
Selama ini kita telah mempelajari berbagai hal, program yang kita jalankan biasanya memiliki tampilan hasil pada jendela output. Kita dapat pula memasukkan sebuah nilai ke dalam suatu variabel apabila kita menggunakan utilitas Scanner. Namun sebenarnya ada cara yang lebih interaktif dari pada cara yang telah kita praktikkan pada pekan-pekan sebelumnya, yaitu dengan menggunakan WindowBuilder.

Dalam penggunaan WindowBuilder akan ada beberapa panel baru. Jika pada program-program sebelumnya tidak menunjukkan sub-tab, pada project berbasis WindowBuilder terdapat dua sub-tab, seperti gambar di bawah ini.

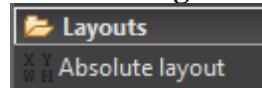


Sub-tab Source berisikan beberapa line kode, kode-kode tersebut menghasilkan sebuah template jendela yang dapat di modifikasi pada sub-tab Design. Pada sub-tab Design terdapat panel Structure, Palette dan Window Preview.

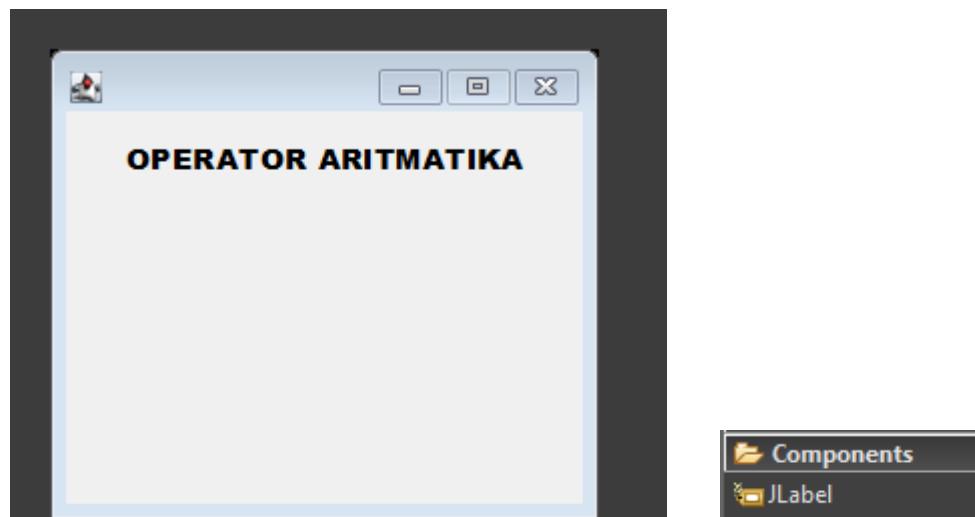




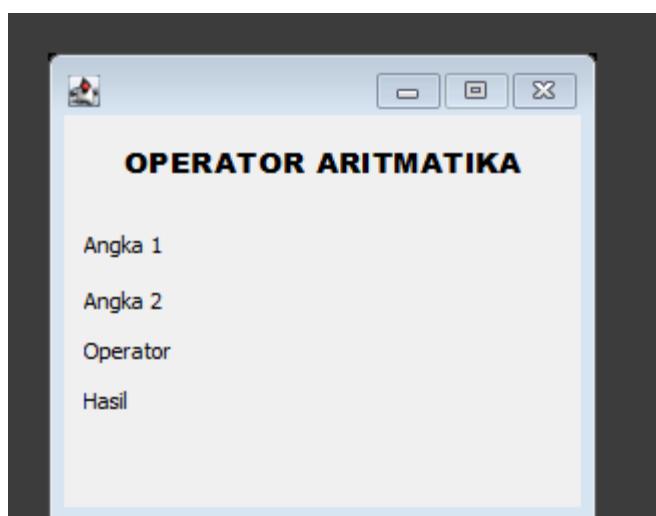
2.2 Kode Pemrograman



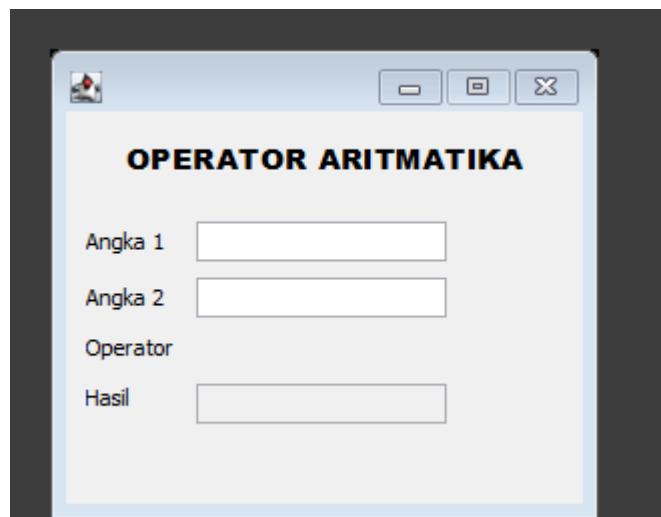
Pada langkah pertama, penulis menerapkan absolute layout pada window, hal ini bertujuan agar komponen-komponen yang kita pasang berada pada posisi dan ukuran yang tetap walaupun jendela aplikasinya kita tarik dan regangkan saat program di compile.



Properties	
Variable	lblNewLabel
Constructor	(Constructor properties)
Bounds	(10, 0, 238, 47)
Class	javax.swing.JLabel
background	■ 240,240,240
displayedM...	
enabled	<input checked="" type="checkbox"/> true
font	Arial Black 14 Bold
foreground	□ 0,0,0
horizontalA...	CENTER
icon	
labelFor	
text	OPERATOR ARITMATIKA
toolTipText	
verticalAlig...	CENTER



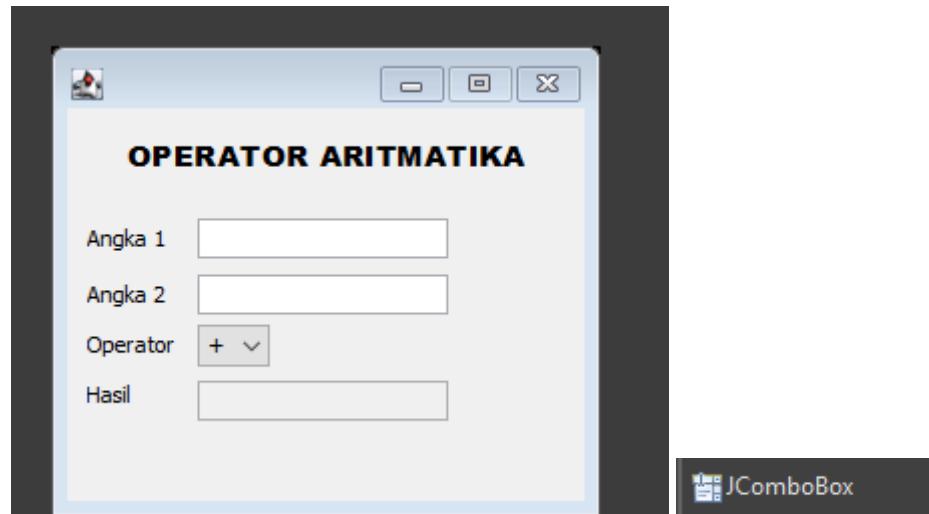
Untuk membuat label “OPERATOR ARITMATIKA”, “Angka 1”, “Angka 2”, “Operator”, ”Hasil” seperti gambar diatas, kita dapat menggunakan komponen JLabel yang berada pada panel palette. Semua komponen dapat kita atur sesuai keinginan kita.

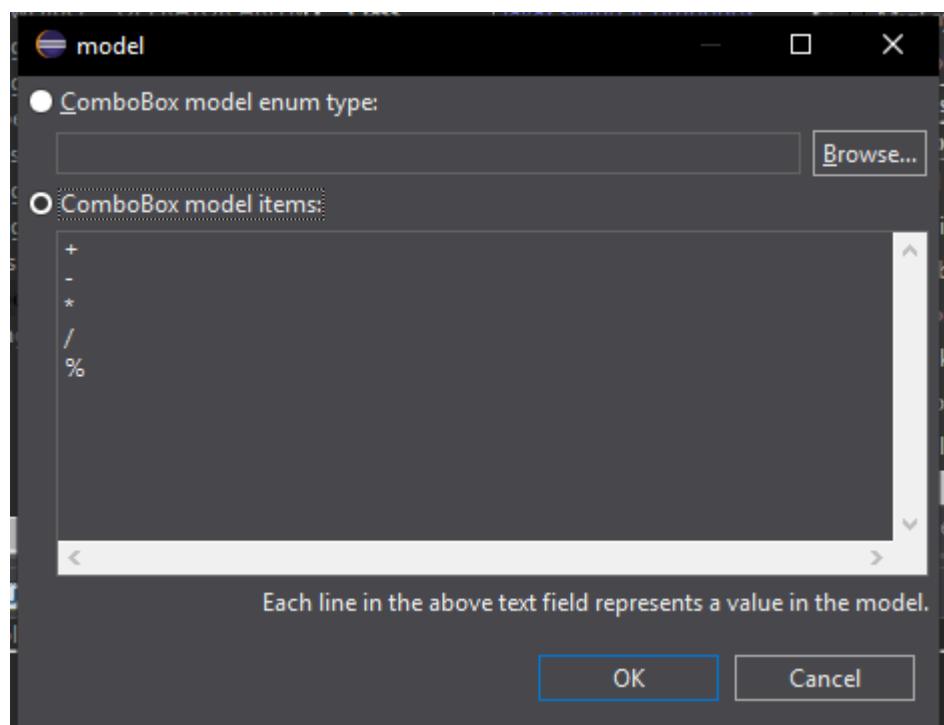
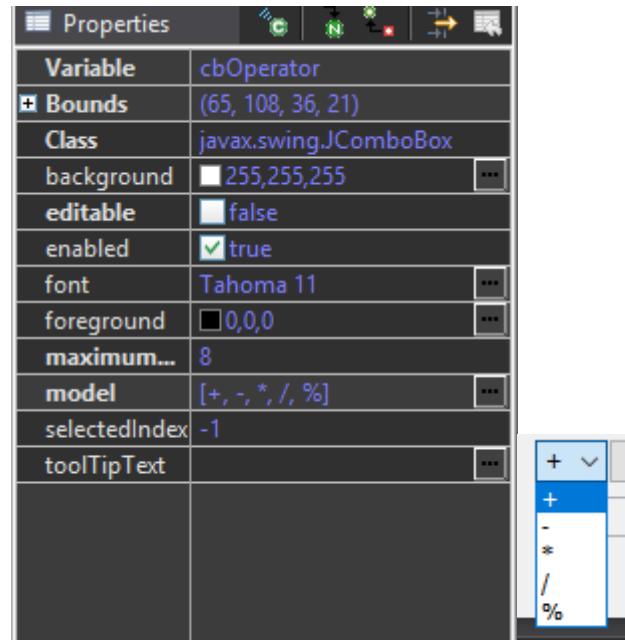


| I JTextField

Dengan menggunakan Component Class JTextField, kita dapat membuat kotak yang dapat kita input melalui Scanner. Kita juga bisa mengatur agar kotak tersebut tidak menerima hasil input dengan cara menonaktifkan centang pada opsi editable

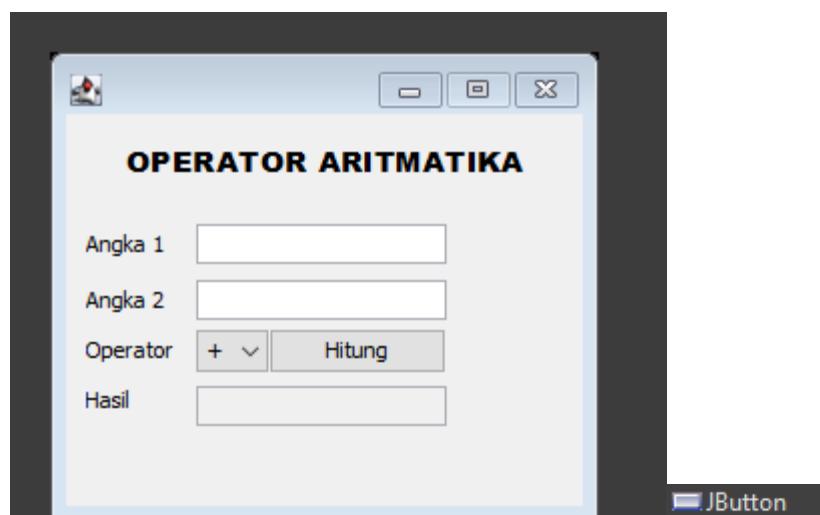
Properties	
Variable	txthasil
+ Bounds	(65, 136, 125, 20)
Class	javax.swing.JTextField
background	■ 255,255,255
columns	10
dropMode	USE_SELECTION
editable	<input type="checkbox"/> false
enabled	<input checked="" type="checkbox"/> true
font	Tahoma 11
foreground	□ 0,0,0
horizontalA...	CENTER
text	
toolTipText	





Component Class JcomboBox berfungsi sebagai penyedia opsi pilihan, pada pemograman algoritma ini penulis menyediakan 5 operator yaitu penjumlahan, pengurangan, perkalian, pembagian, dan sisa bagi. Perlu diketahui bahwa model items yang tertera hanya berperan sebagai penanda saja untuk item no 0 hingga 4.

Properties	
Variable	Hitung
Constructor	(Constructor properties)
+ Bounds	(101, 107, 89, 23)
Class	javax.swing.JButton
background	■ 240,240,240
enabled	<input checked="" type="checkbox"/> true
font	Tahoma 11
foreground	□ 0,0,0
horizontalA...	CENTER
icon	[...]
mnemonic(...)	
selectedIcon	[...]
text	Hitung
toolTipText	
verticalAlig...	CENTER



Kemudian komponen terakhir yang kita gunakan pada program ini adalah Component Class JButton, yang berfungsi sebagai penanda mulainya program jika diberikan aksi yang mana merupakan ketikan dari input mouse/touchpad.

Sekarang untuk mengembangkan program, kita perlu menekan tombol JButton yang telah kita berikan text label “Hitung” sebelumnya. Yang akan memuat kita ke sub-tab source pada line tertentu yang memuat Component Class JButton.

```

 JButton Hitung = new JButton("Hitung");
 Hitung.addActionListener(new ActionListener() {
    int hasil;
    public void actionPerformed(ActionEvent e) {
        if (txtAngka1.getText().trim().isEmpty()) {
            pesanPeringatan("Inputkan angka 1");
        } else if (txtAngka2.getText().trim().isEmpty()) {
            pesanPeringatan("Inputkan angka 2");
        } else {
            try {
                int a = Integer.valueOf(txtAngka1.getText());
                int b = Integer.valueOf(txtAngka2.getText());
                int c = cbOperator.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("Angka 1 dan Angka 2 tidak valid");
            }
        }
    }
}

```

Pastikan program source code memiliki variabel yang sesuai dengan nama nama komponen yang ada pada sub-tab design.

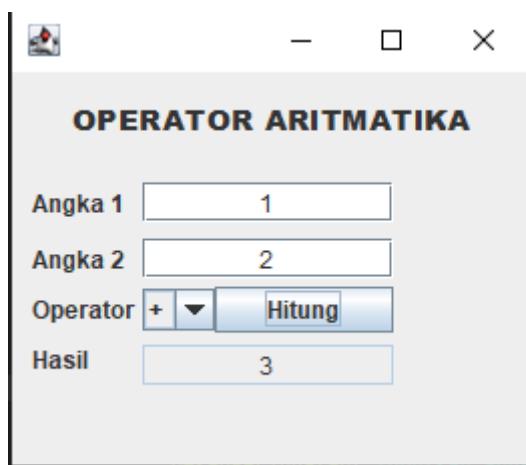
```

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);
}

```

Kita juga memerlukan barisan barisan program berikut untuk menjalankan fitur pesan error dan pesan peringatan mengenai input yang dibaca.

Contoh Output :



OPERATOR ARITMATIKA

Angka 1	<input type="text" value="1"/>
Angka 2	<input type="text" value="2"/>
Operator	<input type="button" value="-"/> <input type="button" value="*"/> <input style="background-color: #ADD8E6; border: none; padding: 0 5px 0 5px;" type="button" value="/"/> Hitung
Hasil	<input type="text" value="-1"/>

OPERATOR ARITMATIKA

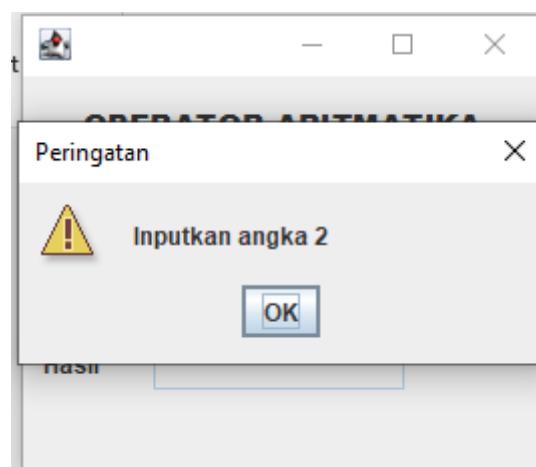
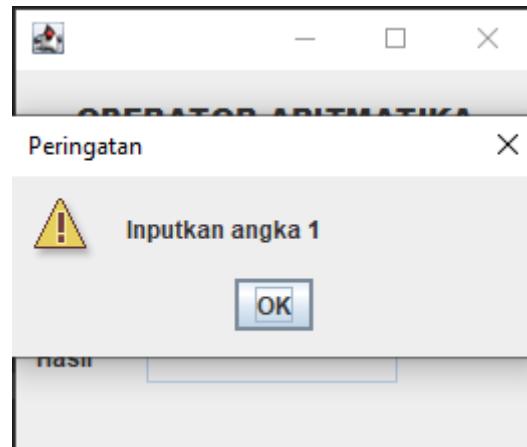
Angka 1	<input type="text" value="1"/>
Angka 2	<input type="text" value="2"/>
Operator	<input type="button" value="-"/> <input type="button" value="*"/> <input style="background-color: #ADD8E6; border: none; padding: 0 5px 0 5px;" type="button" value="/"/> Hitung
Hasil	<input type="text" value="2"/>

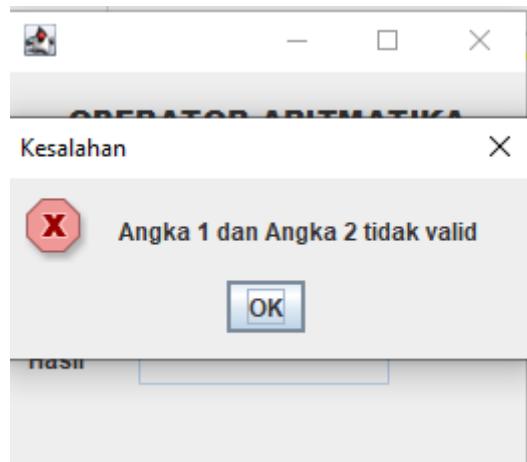
OPERATOR ARITMATIKA

Angka 1	<input type="text" value="1"/>
Angka 2	<input type="text" value="2"/>
Operator	<input type="button" value="-"/> <input type="button" value="*"/> <input style="background-color: #ADD8E6; border: none; padding: 0 5px 0 5px;" type="button" value="/"/> Hitung
Hasil	<input type="text" value="0"/>

OPERATOR ARITMATIKA

Angka 1	<input type="text" value="1"/>
Angka 2	<input type="text" value="2"/>
Operator	%
<input type="button" value="Hitung"/>	
Hasil	<input type="text" value="1"/>





BAB III **KESIMPULAN**

3.1 Kesimpulan

Praktikum pengembangan GUI ini sukses memberikan wawasan mendalam mengenai fondasi perancangan antarmukaDari pengalaman langsung ini, dapat disimpulkan bahwa GUI memiliki peran esensial dalam aplikasi karena menjadi penentu utama bagaimana pengguna dapat berinteraksi dengan perangkat lunak.

Dengan mengaplikasikan berbagai komponen seperti JButton, JLabel, JTextField, dan JComboBox, saya mampu menciptakan desain aplikasi yang tidak hanya fungsional tetapi juga interaktif.

Penggunaan WindowBuilder sangatlah efisien dalam mendesain GUI. Alat ini memfasilitasi perancangan visual melalui teknik *drag and drop*. Keunggulannya terletak pada kemampuannya untuk memproduksi kode Java yang rapi dan minim dependensi eksternal, ditambah fitur *editing* dua arah. Hal ini menjadikannya instrumen yang ideal untuk kegiatan belajar-mengajar, memungkinkan kami melihat secara simultan korelasi antara tampilan grafis dan struktur kode, sehingga logika di balik setiap komponen menjadi jauh lebih mudah dipahami.

3.2 Saran

Penulis harap kita sebagai mahasiswa informatika harus dapat memahami dan mampu mengembangkan aplikasi dengan menggunakan windowbuilder.

DAFTAR PUSTAKA

- [1] Coderanch. (2025). Absolute Layout GUI. coderanch.com .Diakses pada tanggal 20 November 2025 dari
[https://coderanch.com/t/717654/java/Absolute-Layout-GUI#:~:text=The%20Sample%20Output%20uses%206,even%20on%20the%20same%20display.](https://coderanch.com/t/717654/java/Absolute-Layout-GUI#:~:text=The%20Sample%20Output%20uses%206,even%20on%20the%20same%20display)
- [2] Coursera. (2025). What Is a GUI. coursera.org. Diakses pada tanggal 20 November 2025 dari
[https://www.coursera.org/articles/gui#:~:text=A%20graphical%20user%20interface%20\(GUI\)%20is%20an,cursors%2C%20and%20buttons%2C%20to%20make%20computing%20commands](https://www.coursera.org/articles/gui#:~:text=A%20graphical%20user%20interface%20(GUI)%20is%20an,cursors%2C%20and%20buttons%2C%20to%20make%20computing%20commands)
- [3] Eclipse Foundation. *WindowBuilder / A powerful, easy-to-use, bidirectional Java GUI designer.* eclipse.dev. Diakses pada tanggal 20 November 2025 dari <https://eclipse.dev/windowbuilder/>