**LAPORAN TUGAS BESAR**

**PEMROGRAMAN BERBASIS OBJEK**

**KALKULATOR**

# HALAMAN JUDUL



**Disusun Oleh :**

**NAMA : ALFIDA ZUMAROH**

**NIM : 32602200039**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS ISLAM SULTAN AGUNG**

**SEMARANG**

**2022**

# DAFTAR ISI

[HALAMAN JUDUL 1](#_Toc154995288)

[DAFTAR ISI 2](#_Toc154995289)

[DAFTAR GAMBAR 3](#_Toc154995290)

[BAB I Pendahuluan 4](#_Toc154995291)

[**1.1** **Latar belakang** 4](#_Toc154995292)

[**1.2** **Tujuan** 4](#_Toc154995293)

[**1.3** **Manfaat** 4](#_Toc154995294)

[BAB II Konsep Dasar PBO 5](#_Toc154995295)

[**2.1** **Inheritance** 5](#_Toc154995296)

[**2.2** **Polimorfisme** 5](#_Toc154995297)

[**2.3** **Encapsulation** 5](#_Toc154995298)

[**2.4** **Getter dan Setter** 5](#_Toc154995299)

[**2.5** **Interface** 5](#_Toc154995300)

[BAB III Struktur Program 6](#_Toc154995301)

[**3.1** **File Utama ( CalculatorFrame )** 6](#_Toc154995302)

[**3.2** **Kelas Calculator** 8](#_Toc154995303)

[**3.3** **Kelas Operation (AdditionOperation, SubtractionOperation, MultiplicationOperation, DivisionOperation)** 9](#_Toc154995304)

[**3.4** **Interface Calculator Operation** 13](#_Toc154995305)

[BAB IV Implementasi Program 15](#_Toc154995306)

[**4.1** **Menjalankan Program** 15](#_Toc154995307)

[**4.2** **Mengoperasikan Kalkulator** 16](#_Toc154995308)

[BAB V Kesimpulan 19](#_Toc154995309)

[DAFTAR PUSTAKA 21](#_Toc154995310)

# DAFTAR GAMBAR

[Gambar 1. 1 File CalculatorFrame 1 6](#_Toc154995255)

[Gambar 1. 2 File CalculatorFrame 2 6](#_Toc154995256)

[Gambar 1. 3 File CalculatorFrame 3 7](#_Toc154995257)

[Gambar 1. 4 File CalculatorFrame 4 7](#_Toc154995258)

[Gambar 1. 5 File CalculatorFrame 5 7](#_Toc154995259)

[Gambar 1. 6 File CalculatorFrame 6 8](#_Toc154995260)

[Gambar 1. 7 Kelas Calculator 1 9](#_Toc154995261)

[Gambar 1. 8 Kelas Calculator 2 9](#_Toc154995262)

[Gambar 1. 9 Kelas DivisionOperation 10](#_Toc154995263)

[Gambar 1. 10 Kelas MultiplicationOperation 11](#_Toc154995264)

[Gambar 1. 11 Kelas SubtractionOperation 12](#_Toc154995265)

[Gambar 1. 12 Kelas AdditionOperation 13](#_Toc154995266)

[Gambar 1. 13 Interface Calculator Operation 14](#_Toc154995267)

[Gambar 2. 1 Open Neatbeans 15](#_Toc154995268)

[Gambar 2. 2 Open Projects 15](#_Toc154995269)

[Gambar 2. 3 Arahkan ke project 15](#_Toc154995270)

[Gambar 2. 4 Open struktur project 16](#_Toc154995271)

[Gambar 2. 5 Run program 16](#_Toc154995272)

[Gambar 2. 6 Tunggu program muncul 16](#_Toc154995273)

[Gambar 2. 7 Menu Calculator 16](#_Toc154995274)

[Gambar 2. 8 Penambahan 17](#_Toc154995275)

[Gambar 2. 9 Klick calculate to get result 17](#_Toc154995276)

[Gambar 2. 10 Klick clear 17](#_Toc154995277)

[Gambar 2. 11 menu operation 18](#_Toc154995278)

# BAB I Pendahuluan

## **Latar belakang**

Pengembangan kalkulator GUI menjadi relevan sebagai solusi modern dalam memberikan pengguna pengalaman perhitungan yang lebih interaktif dan user-friendly..

## **Tujuan**

Tujuan utama dari pembuatan kalkulator GUI ini adalah memberikan alat bantu perhitungan yang efisien dan mudah digunakan kepada pengguna..

## **Manfaat**

1. Mempermudah pengguna dalam melakukan perhitungan matematika sehari-hari.
2. Menyediakan antarmuka yang intuitif untuk pengalaman pengguna yang lebih baik.

# BAB II Konsep Dasar PBO

## **Inheritance**

Penerapan inheritance digunakan dalam kalkulator untuk mewarisi sifat dan perilaku umum dari kelas Calculator ke kelas operasi matematika..

## **Polimorfisme**

Polimorfisme digunakan untuk memberikan fleksibilitas dalam pemilihan operasi matematika yang sesuai dengan pilihan pengguna..

## **Encapsulation**

Encapsulation digunakan untuk menyembunyikan detail implementasi dan mengakses atribut melalui metode setter dan getter..

## **Getter dan Setter**

Penggunaan metode getter dan setter memungkinkan pengaksesan dan pengubahan atribut dengan kontrol yang lebih baik..

## **Interface**

Penerapan interface pada kalkulator memungkinkan implementasi operasi matematika tanpa peduli pada detail implementasi kelas tersebut..

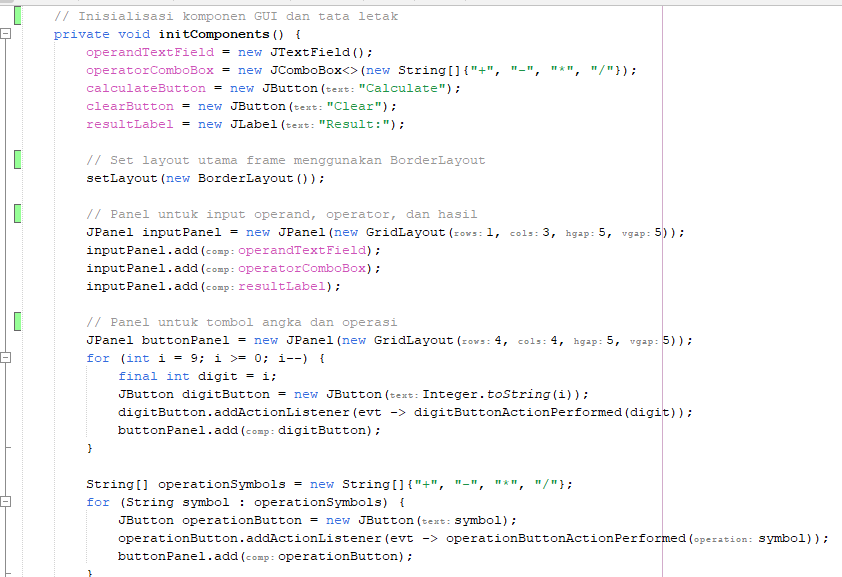
# BAB III Struktur Program

## **File Utama ( CalculatorFrame )**

Kelas utama yang berperan sebagai frame kalkulator, mengintegrasikan antarmuka pengguna dengan operasi matematika.



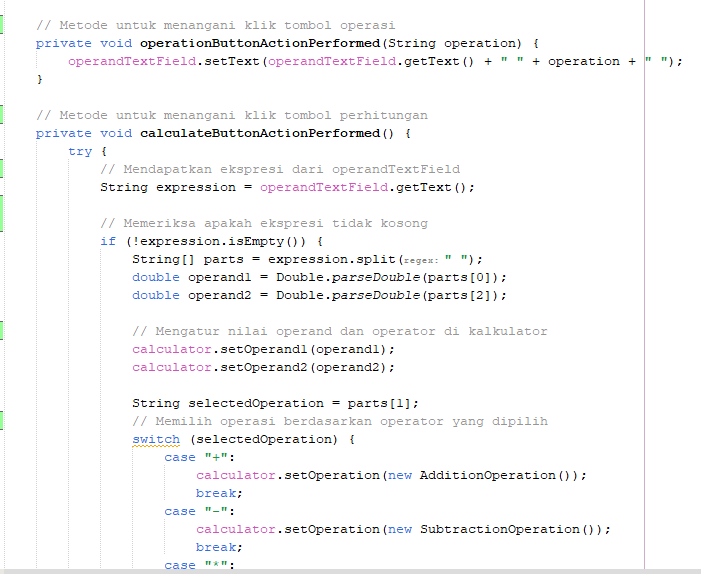
Gambar 1. 1 File CalculatorFrame 1



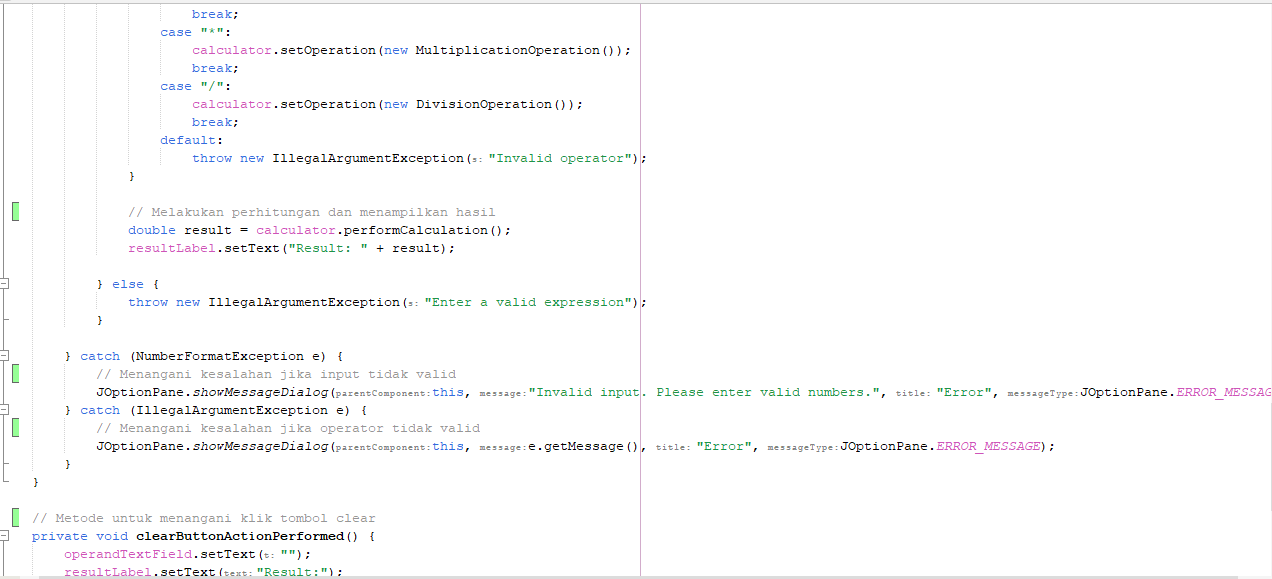
Gambar 1. 2 File CalculatorFrame 2



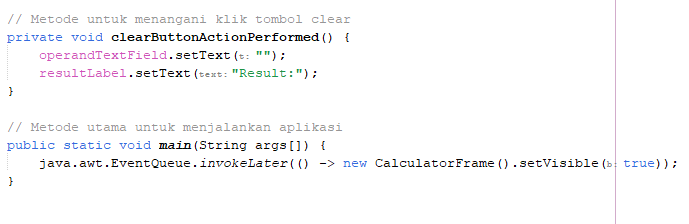
Gambar 1. 3 File CalculatorFrame 3



Gambar 1. 4 File CalculatorFrame 4



Gambar 1. 5 File CalculatorFrame 5



Gambar 1. 6 File CalculatorFrame 6

Penjelasan:

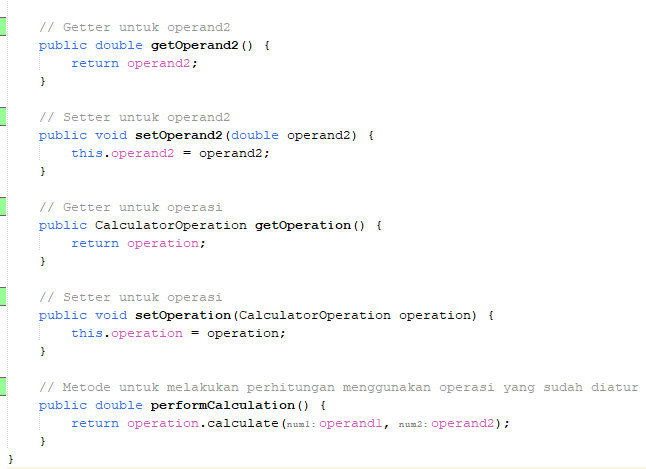
1. Kode ini merupakan implementasi dari kalkulator GUI menggunakan Java Swing.
2. Frame kalkulator diatur dengan layout BorderLayout untuk tata letak yang jelas.
3. Komponen-komponen seperti JTextField, JComboBox, JButton, dan JLabel digunakan untuk input, operasi, dan output.
4. Metode initComponents digunakan untuk inisialisasi komponen-komponen GUI.
5. Aksi setiap tombol (digit, operasi, Calculate, Clear) diimplementasikan dalam metode terpisah.
6. Pada saat Calculate ditekan, ekspresi diambil dari operandTextField, dihitung menggunakan Calculator, dan hasilnya ditampilkan di resultLabel.
7. Aplikasi dijalankan melalui metode main pada thread event dispatch.

## **Kelas Calculator**

Kelas yang bertanggung jawab untuk perhitungan matematika berdasarkan operasi yang dipilih.



Gambar 1. 7 Kelas Calculator 1



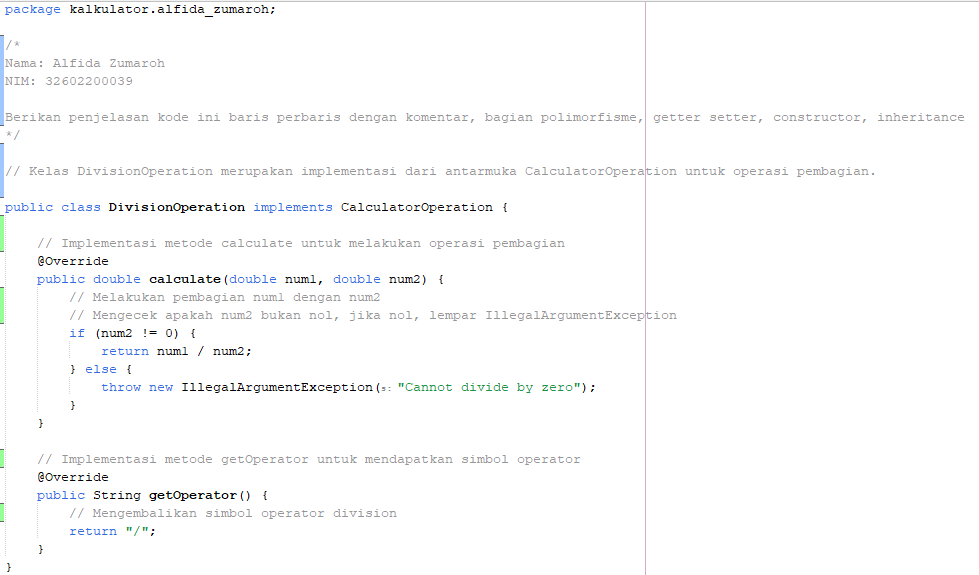
Gambar 1. 8 Kelas Calculator 2

Penjelasan:

1. Constructor (Konstruktor): Konstruktor Calculator digunakan untuk inisialisasi objek. Saat objek dibuat, nilai operand1 dan operand2 diatur menjadi 0, dan operasi default diatur sebagai penjumlahan (new AdditionOperation()).
2. Getter dan Setter: Terdapat getter (getOperand1, getOperand2, getOperation) dan setter (setOperand1, setOperand2, setOperation) untuk mendapatkan dan mengatur nilai operand1, operand2, dan operasi.

## **Kelas Operation (AdditionOperation, SubtractionOperation, MultiplicationOperation, DivisionOperation)**

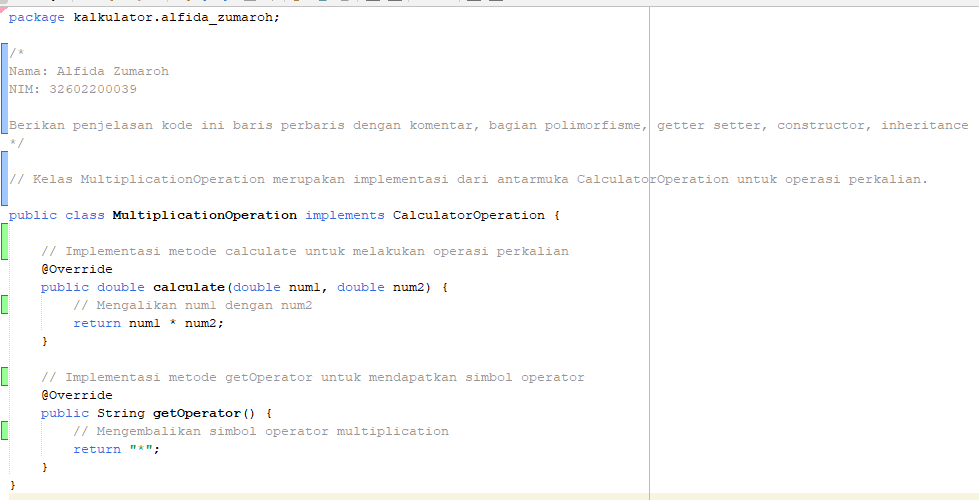
Kelas-kelas operasi matematika yang diintegrasikan ke dalam kalkulator untuk menjalankan perhitungan.



Gambar 1. 9 Kelas DivisionOperation

Penjelasan:

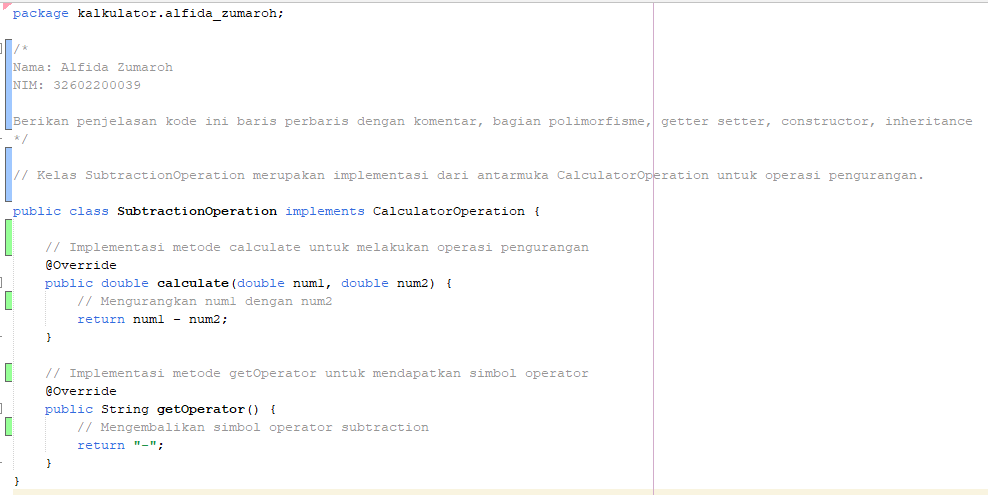
1. Polimorfisme: Kelas DivisionOperation mengimplementasikan antarmuka (CalculatorOperation). Dengan demikian, kelas ini menunjukkan polimorfisme, di mana objek dari kelas ini dapat digunakan di mana pun antarmuka CalculatorOperation diperlukan. Polimorfisme juga terlihat dalam penggunaan metode calculate yang diimplementasikan sesuai dengan kebutuhan operasi pembagian.
2. Getter dan Setter: Tidak ada penggunaan getter dan setter dalam kelas ini karena tidak ada properti yang perlu diakses atau diubah dari luar kelas.
3. Constructor (Konstruktor): Tidak ada konstruktor yang didefinisikan dalam kelas ini. Java akan menyediakan konstruktor default jika tidak ada konstruktor yang didefinisikan.
4. Inheritance (Pewarisan): Kelas DivisionOperation tidak melakukan pewarisan secara langsung. Namun, secara tidak langsung, kelas ini "mewarisi" perilaku dari antarmuka (CalculatorOperation) yang diimplementasikannya.



Gambar 1. 10 Kelas MultiplicationOperation

Penjelasan:

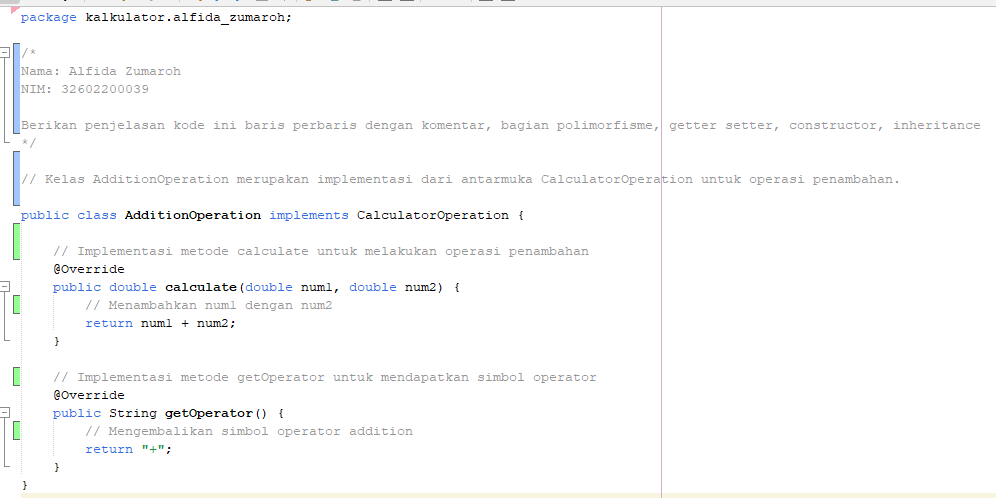
1. Polimorfisme: Kelas MultiplicationOperation mengimplementasikan antarmuka (CalculatorOperation). Dengan demikian, kelas ini menunjukkan polimorfisme, di mana objek dari kelas ini dapat digunakan di mana pun antarmuka CalculatorOperation diperlukan. Polimorfisme juga terlihat dalam penggunaan metode calculate yang diimplementasikan sesuai dengan kebutuhan operasi perkalian.
2. Getter dan Setter: Tidak ada penggunaan getter dan setter dalam kelas ini karena tidak ada properti yang perlu diakses atau diubah dari luar kelas.
3. Constructor (Konstruktor): Tidak ada konstruktor yang didefinisikan dalam kelas ini. Java akan menyediakan konstruktor default jika tidak ada konstruktor yang didefinisikan.
4. Inheritance (Pewarisan): Kelas MultiplicationOperation tidak melakukan pewarisan secara langsung. Namun, secara tidak langsung, kelas ini "mewarisi" perilaku dari antarmuka (CalculatorOperation) yang diimplementasikannya.



Gambar 1. 11 Kelas SubtractionOperation

Penjelasan:

1. Polimorfisme: Kelas SubtractionOperation mengimplementasikan antarmuka (CalculatorOperation). Dengan demikian, kelas ini menunjukkan polimorfisme, di mana objek dari kelas ini dapat digunakan di mana pun antarmuka CalculatorOperation diperlukan. Polimorfisme juga terlihat dalam penggunaan metode calculate yang diimplementasikan sesuai dengan kebutuhan operasi pengurangan.
2. Getter dan Setter: Tidak ada penggunaan getter dan setter dalam kelas ini karena tidak ada properti yang perlu diakses atau diubah dari luar kelas.
3. Constructor (Konstruktor): Tidak ada konstruktor yang didefinisikan dalam kelas ini. Java akan menyediakan konstruktor default jika tidak ada konstruktor yang didefinisikan.
4. Inheritance (Pewarisan): Kelas SubtractionOperation tidak melakukan pewarisan secara langsung. Namun, secara tidak langsung, kelas ini "mewarisi" perilaku dari antarmuka (CalculatorOperation) yang diimplementasikannya.



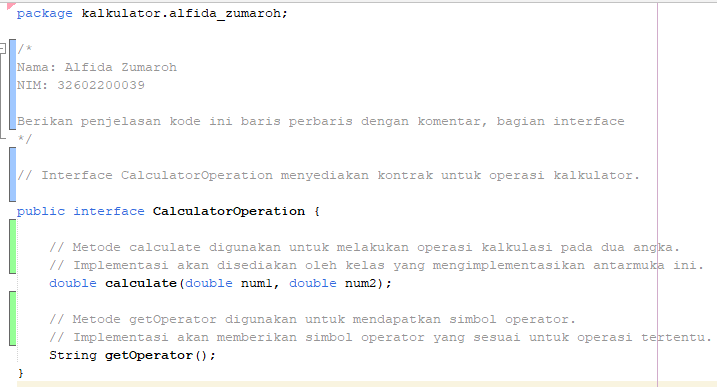
Gambar 1. 12 Kelas AdditionOperation

Penjelasan:

1. Polimorfisme: Kelas AdditionOperation mengimplementasikan antarmuka (CalculatorOperation). Dengan demikian, kelas ini menunjukkan polimorfisme, di mana objek dari kelas ini dapat digunakan di mana pun antarmuka CalculatorOperation diperlukan. Polimorfisme juga terlihat dalam penggunaan metode calculate yang diimplementasikan sesuai dengan kebutuhan operasi penambahan.
2. Getter dan Setter: Tidak ada penggunaan getter dan setter dalam kelas ini karena tidak ada properti yang perlu diakses atau diubah dari luar kelas.
3. Constructor (Konstruktor): Tidak ada konstruktor yang didefinisikan dalam kelas ini. Java akan menyediakan konstruktor default jika tidak ada konstruktor yang didefinisikan.
4. Inheritance (Pewarisan): Kelas AdditionOperation tidak melakukan pewarisan secara langsung. Namun, secara tidak langsung, kelas ini "mewarisi" perilaku dari antarmuka (CalculatorOperation) yang diimplementasikannya.

## **Interface Calculator Operation**

Antarmuka yang diimplementasikan oleh kelas operasi matematika, memberikan kerangka kerja untuk berbagai operasi.



Gambar 1. 13 Interface Calculator Operation

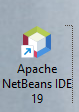
Penjelasan:

1. Interface (Antarmuka): Ini adalah antarmuka CalculatorOperation yang menyediakan kontrak untuk operasi kalkulator. Antarmuka ini memiliki dua metode, calculate untuk melakukan operasi kalkulasi dan getOperator untuk mendapatkan simbol operator yang terkait dengan operasi tersebut.
2. Metode dalam Antarmuka:
3. calculate: Metode ini memerlukan dua parameter (num1 dan num2) dan akan diimplementasikan oleh kelas-kelas yang menggunakan antarmuka ini untuk melakukan operasi kalkulasi sesuai dengan kebutuhan mereka.
4. getOperator: Metode ini mengembalikan string yang berisi simbol operator yang terkait dengan operasi kalkulator tertentu. Metode ini juga akan diimplementasikan oleh kelas-kelas yang menggunakan antarmuka ini.

# BAB IV Implementasi Program

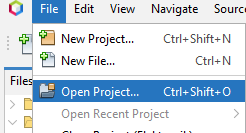
## **Menjalankan Program**

1. Untuk menjalankan program klick neatbeans

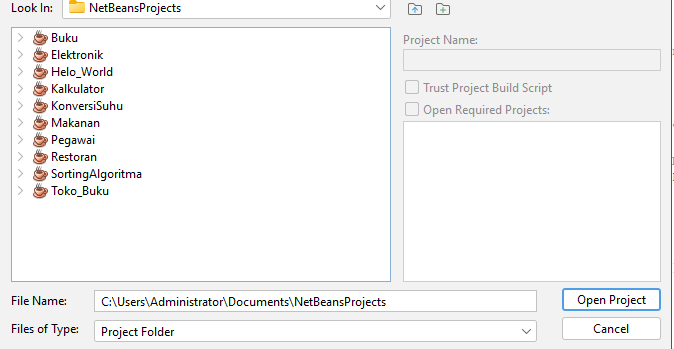


Gambar 2. 1 Open Neatbeans

1. Open new project, arahkan ke elektronik

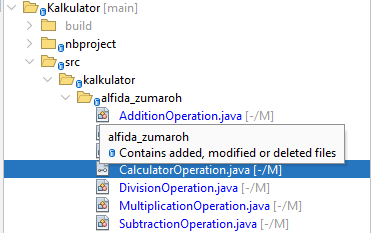


Gambar 2. 2 Open Projects



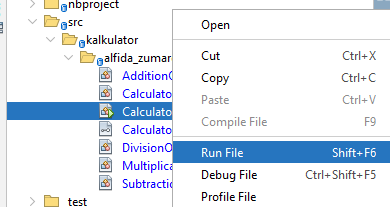
Gambar 2. 3 Arahkan ke project

1. Buka src/kalkulator/alfida\_zumaroh



Gambar 2. 4 Open struktur project

1. Klick file CalculatorFrame.java, lalu klick kanan run file atau shift + f6



Gambar 2. 5 Run program

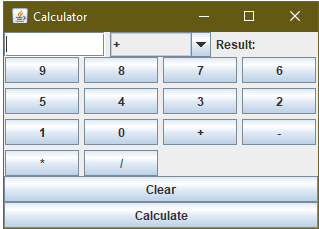
1. Tunggu program muncul



Gambar 2. 6 Tunggu program muncul

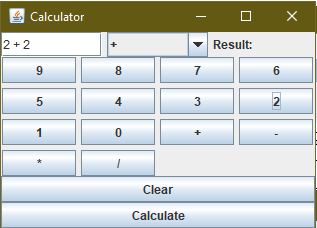
## **Mengoperasikan Kalkulator**

1. Klick Angka yang mau di hitung



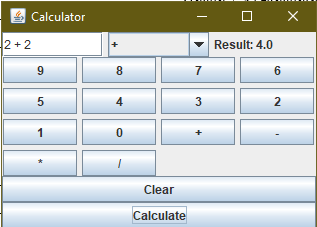
Gambar 2. 7 Menu Calculator

1. Misal isi 2 + 2



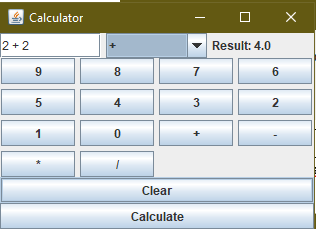
Gambar 2. 8 Penambahan

1. Jika sudah klick calculate, maka akan muncul resul.



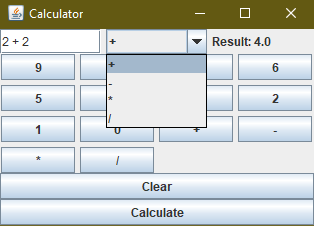
Gambar 2. 9 Klick calculate to get result

1. Jika ingin menghapus klick clear



Gambar 2. 10 Klick clear

1. Operasi bisa dilakukan dengan memilih menu pilihan, selain klick tombol operasi.



Gambar 2. 11 menu operation

# BAB V Kesimpulan

Program kalkulator yang telah dibuat menggunakan bahasa pemrograman Java dan memanfaatkan konsep dasar Pemrograman Berorientasi Objek (PBO). Berikut adalah kesimpulan yang dapat diambil:

1. Inheritance (Pewarisan): Program kalkulator menerapkan konsep pewarisan dengan membuat kelas Calculator yang memiliki sifat-sifat umum yang dapat digunakan oleh kalkulator. Pewarisan membantu dalam menciptakan struktur hierarki yang memudahkan pengelolaan kode dan pemeliharaan.
2. Polimorfisme: Konsep polimorfisme diimplementasikan melalui antarmuka CalculatorOperation. Berbagai operasi kalkulator seperti penjumlahan, pengurangan, perkalian, dan pembagian diimplementasikan sebagai kelas-kelas terpisah yang mengimplementasikan antarmuka ini. Hal ini memungkinkan penggunaan polimorfisme untuk memperlakukan berbagai operasi sebagai objek dengan antarmuka yang sama.
3. Encapsulation (Enkapsulasi): Enkapsulasi diterapkan dengan menyembunyikan detail implementasi operasi kalkulator di dalam kelas Calculator. Variabel-variabel dan metode-metode yang tidak perlu diakses dari luar dikapsulkan untuk menjaga integritas data dan mengurangi kompleksitas.
4. Getter dan Setter: Program menggunakan metode getter dan setter pada kelas Calculator untuk mengakses dan memodifikasi nilai-nilai operand dan operasi kalkulator. Ini membantu dalam mengimplementasikan prinsip enkapsulasi dengan cara yang aman.
5. Interface: Antarmuka CalculatorOperation digunakan untuk menyediakan kontrak untuk operasi-operasi kalkulator. Ini memungkinkan berbagai operasi untuk diimplementasikan dengan cara yang berbeda, tetapi dengan menggunakan antarmuka yang sama.

Keseluruhan, program ini memanfaatkan konsep-konsep dasar Pemrograman Berorientasi Objek untuk menciptakan kalkulator yang modular, mudah dipahami, dan dapat diubah-ubah sesuai kebutuhan.

# DAFTAR PUSTAKA

(Czajkowski and Von Eicken, 1998; Lumpe, Mahmud and Vasa, 2010; Ari Yuana, S.Si, M.Kom, 2015)

Ari Yuana, S.Si, M.Kom, R. (2015) ‘Pemrograman Java’, *Informatika*, pp. 1–77. Available at: https://d1wqtxts1xzle7.cloudfront.net/33519053/JavaIndo.pdf?1398090002=&response-content-disposition=inline%3B+filename%3DDitulis\_oleh.pdf&Expires=1644744989&Signature=hOHnqB2lSAg3O37OjxRlo8mzhmE2pUyaB6R8UbZ75XMmwTHll9mbtKfvvdS4S7MgZTY9R46hVzgrAIej4uG5YJK.

Czajkowski, G. and Von Eicken, T. (1998) ‘JRes: A Resource Accounting Interface for Java’, *SIGPLAN Notices (ACM Special Interest Group on Programming Languages)*, 33(10), pp. 21–35. Available at: https://doi.org/10.1145/286942.286944.

Lumpe, M., Mahmud, S. and Vasa, R. (2010) ‘On the use of properties in Java applications’, *Proceedings of the Australian Software Engineering Conference, ASWEC*, pp. 235–244. Available at: https://doi.org/10.1109/ASWEC.2010.35.