

**LAPORAN PRATIKUM**  
**PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN**  
**PENERAPAN OPERATOR DALAM JAVA**

**Disusun Oleh:**

Muhammad Fharel

2511531010

**Dosen Pengampu:**

Dr. Wahyudi S.T.M.T

**Asisten Pratikum:**

Muhammad Zaki Al Hafiz



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**

**2025**

## **KATA PENGANTAR**

Segala puji penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga laporan praktikum Algoritma dan Pemograman pada tanggal 24 September 2025 yang membahas Operator dalam Java ini dapat terselesaikan. Praktikum ini bertujuan untuk memperkenalkan dan memahami berbagai jenis operator dalam bahasa pemrograman Java, seperti operator aritmatika, logika, relasional, dan assignment, serta penerapannya dalam membangun program yang efisien dan logis. Materi ini penting karena menjadi fondasi dalam memahami pemograman.

Ucapan terima kasih ditujukan kepada dosen pengampu, asisten praktikum, serta rekan-rekan yang telah membantu dalam proses pelaksanaan praktikum. Penulis menyadari bahwa penulisan laporan masih jauh dari kata sempurna, oleh karena itu kritik dan saran sangat diharapkan untuk penyempurnaan di kemudian hari. Semoga laporan ini memberikan manfaat dan menambah wawasan pembaca.

Padang, 27 September 2025

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1    Latar Belakang.....	1
1.2    Tujuan Praktikum .....	1
1.3    Manfaat Praktikum .....	1
<b>BAB II PEMBAHASAN .....</b>	<b>3</b>
2.1    Dasar Teori .....	3
2.1.1    Class Scanner .....	3
2.1.2    Operator Aritmatika .....	3
2.1.3    Operator Logika .....	3
2.1.4    Operator Rasional.....	4
2.1.5    Operator Assignment .....	5
2.2    Pembuatan Package dan Class untuk Pekan 3.....	6
2.3    Program Pertama (Operator Aritmatika).....	8
2.3.1    Output Program Pertama (Operator Aritmatika).....	8
2.3.2    Penjelasan Singkat .....	9
2.4    Program Kedua (Operator Assingment).....	9
2.4.1    Output Program Kedua (Operator Assingment) .....	9
2.4.2    Penjelasan Singkat .....	10
2.5    Program Ketiga (Operator Relasional).....	10
2.5.1    Output Program Ketiga (Operator Relasional) .....	11
2.5.2    Penjelsan Singkat .....	11
2.6    Program Keempat (Operator Logika).....	12

2.6.1	Output Program Keempat (Operator Logika) .....	12
2.6.2	Penjelsan Singkat .....	13
<b>BAB III PENUTUP .....</b>		<b>14</b>
3.1	Kesimpulan.....	14
<b>DAFTAR PUSTAKA .....</b>		<b>15</b>

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Dalam pemrograman, operator merupakan komponen fundamental yang berfungsi untuk memanipulasi data dan mengendalikan logika program. Bahasa Java menyediakan berbagai operator seperti aritmatika, relasional, logika, dan assignment yang menjadi dasar penyusunan instruksi. Tanpa pemahaman yang kuat tentang operator, programmer akan kesulitan dalam membangun program yang efisien.

Penguasaan konsep operator penting karena hampir seluruh sintaks program melibatkan operasi tertentu. Hal ini sejalan dengan pendapat Nugroho (2020) yang menekankan bahwa operator tidak hanya menjadi dasar perhitungan, tetapi juga mendukung implementasi algoritma lebih kompleks seperti pengendalian alur program dan pemrosesan data.

Oleh karena itu, praktikum ini dirancang untuk memberikan pengalaman langsung kepada mahasiswa dalam menggunakan operator pada Java. Praktikum ini sekaligus membangun pondasi untuk mempelajari materi lanjutan seperti struktur kontrol, fungsi, dan pemrograman berorientasi objek.

#### **1.2 Tujuan Praktikum**

Tujuan dari praktikum ini adalah:

1. Memahami konsep dasar operator aritmatika, relasional, logika, dan *assignment* dalam pemrograman Java.
2. Mengimplementasikan operator dalam pemrograman sederhana.
3. Mengembangkan keterampilan menyelesaikan masalah komputasi dengan menggunakan operator Java.

#### **1.3 Manfaat Praktikum**

Manfaat dari praktikum ini adalah:

1. Mahasiswa mampu memahami fungsi dan cara kerja operator dalam Java.
2. Memberikan pengalaman praktis kepada mahasiswa dalam penerapan operator.

3. Menjadi dasar penting untuk materi pemrograman lanjutan.

## BAB II

### PEMBAHASAN

#### 2.1 Dasar Teori

Operator adalah simbol khusus yang digunakan untuk melakukan operasi pada *operand*, baik berupa nilai maupun variabel. Dalam Java, operator dapat dikelompokkan menjadi:

##### 2.1.1 Class Scanner

*Class Scanner* dalam Java digunakan untuk membaca input dari berbagai sumber, seperti *keyboard*, *file*, maupun *string*. Dengan *Scanner*, interaksi antara program dan pengguna menjadi lebih mudah karena input dapat diproses langsung dalam bentuk tipe data yang sesuai, misalnya *int*, *double*, atau *String*. Penggunaan *Scanner* membantu mahasiswa memahami bagaimana sebuah program dapat berkomunikasi dengan pengguna melalui input dan output secara dinamis.

##### 2.1.2 Operator Aritmatika

Operator aritmatika digunakan untuk melakukan operasi dasar matematika seperti penjumlahan (+), pengurangan (-), perkalian (\*), pembagian (/), dan modulus (%). Operator ini hanya berlaku untuk tipe data numerik. Operator aritmatika adalah fondasi utama dalam pemrograman karena hampir semua proses komputasi melibatkan perhitungan matematis.

Operator	Keterangan	Contoh
+	Penjumlahan	$A + B = 15$
-	Pengurangan	$B - A = 5$
*	Perkalian	$A * B = 50$
/	Membagi pembilang dengan penyebut	$B / A = 2$
%	Sisa bagi dari hasil pembagian pembilang dengan penyebut	$B \% A = 0$

Gambar 2.1 Sumber: Dapur Elektronya Galuh

##### 2.1.3 Operator Logika

Operator logika digunakan untuk menggabungkan ekspresi boolean, misalnya && (*AND*), || (*OR*), dan ! (*NOT*). Operator ini sering digunakan dalam percabangan (*if*) dan perulangan (*while*, *for*). Penguasaan

operator logika mempermudah programmer dalam membangun kondisi kompleks yang mendasari alur eksekusi program.

Operator	Keterangan	Contoh
&&	Disebut operator logika AND. Jika kedua operand bernilai bukan nol maka kondisinya benar (true)	(A&&B) is not true
	Disebut operator logika OR. Jika kedua operand bernilai nol maka kondisinya salah (false / not true)	(A  B) is true
!	Disebut operator logika NOT. Fungsinya untuk membalik nilai. Jika hasilnya adalah 0 maka kondisinya salah (false / not true), jika hasilnya adalah 1 maka kondisinya benar (true)	(!A) is true (!B) is not true

Gambar 2.2 Sumber: Dapur Elektronya Galuh

#### 2.1.4 Operator Rasional

Operator relasional digunakan untuk membandingkan dua nilai dan menghasilkan output berupa *true* atau *false*. Contoh operator ini antara lain ==, !=, <, >, <=, dan >=. Operator relasional berperan penting dalam struktur kontrol karena membantu program mengambil keputusan berdasarkan kondisi tertentu.



Operator	Keterangan	Contoh
<code>==</code>	Mengecek apakah dua operand bernilai sama atau tidak, jika nilainya sama maka hasilnya adalah true (benar)	<code>(A==B)</code> is not true
<code>!=</code>	Mengecek apakah dua operand bernilai sama atau tidak, jika nilainya tidak sama maka hasilnya adalah true (benar)	<code>(A!=B)</code> is true
<code>&gt;</code>	Mengecek apakah nilai salah satu operand di depan symbol operator bernilai lebih besar dibandingkan nilai operand di belakang symbol operator, jika iya maka hasilnya adalah true (benar)	<code>(A &gt; B )</code> is not true
<code>&lt;</code>	Mengecek apakah nilai salah satu operand di depan symbol operator bernilai lebih kecil dibandingkan nilai operand di belakang symbol operator, jika iya maka hasilnya adalah true (benar)	<code>(A &lt; B )</code> is true
<code>&gt;=</code>	Mengecek apakah nilai salah satu operand di depan symbol operator bernilai lebih besar atau sama dengan nilai operand di belakang symbol operator, jika iya maka hasilnya adalah true (benar)	<code>(A &gt;= B)</code> is not true
<code>&lt;=</code>	Mengecek apakah nilai salah satu operand di depan symbol operator bernilai lebih kecil atau sama dengan nilai operand di belakang symbol operator, jika iya maka hasilnya adalah true (benar)	<code>(A &lt;= B)</code> is true

Gambar 2.3 Sumber: Dapur Elektronya Galuh

### 2.1.5 Operator Assignment

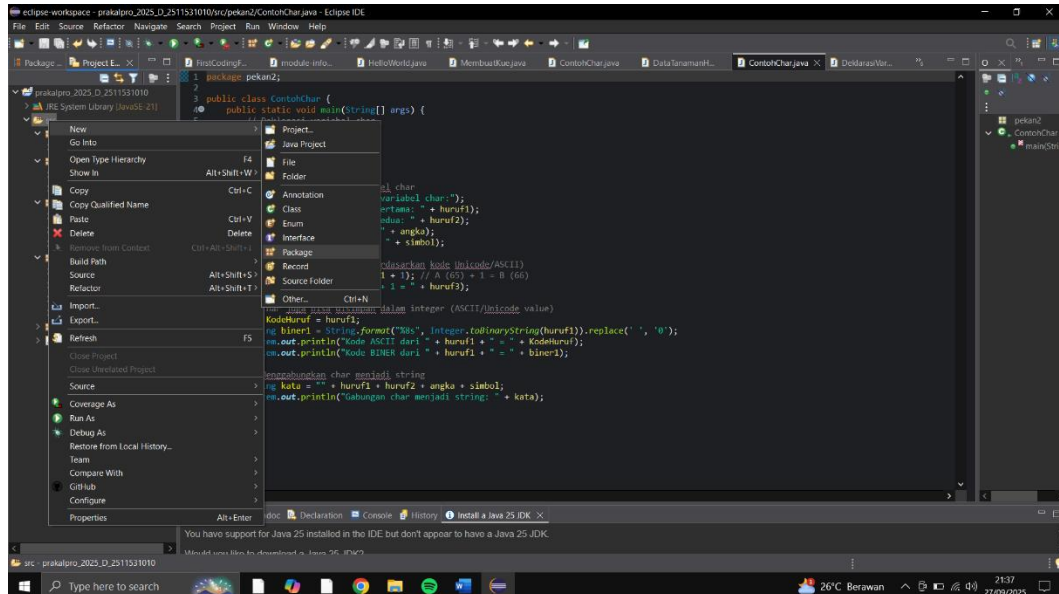
Operator assignment digunakan untuk memberikan nilai pada variabel. Operator dasar adalah tanda sama dengan (`=`), sedangkan variasinya seperti `+=`, `-=`, `*=`, `/=`, dan `%=` memungkinkan penulisan kode menjadi lebih ringkas. Penggunaan operator assignment mendukung efisiensi penulisan program dengan menggabungkan operasi aritmatika dan penugasan sekaligus.

Operator	Fungsi	Contoh	Setara dengan
<code>+=</code>	Menambahkan dan menugaskan	<code>x += 5;</code>	<code>x = x + 5;</code>
<code>-=</code>	Mengurangi dan menugaskan	<code>x -= 3;</code>	<code>x = x - 3;</code>
<code>*=</code>	Mengalikan dan menugaskan	<code>x *= 2;</code>	<code>x = x * 2;</code>
<code>/=</code>	Membagi dan menugaskan	<code>x /= 4;</code>	<code>x = x / 4;</code>
<code>%=</code>	Modulus dan menugaskan	<code>x %= 3;</code>	<code>x = x % 3;</code>

Gambar 2.4

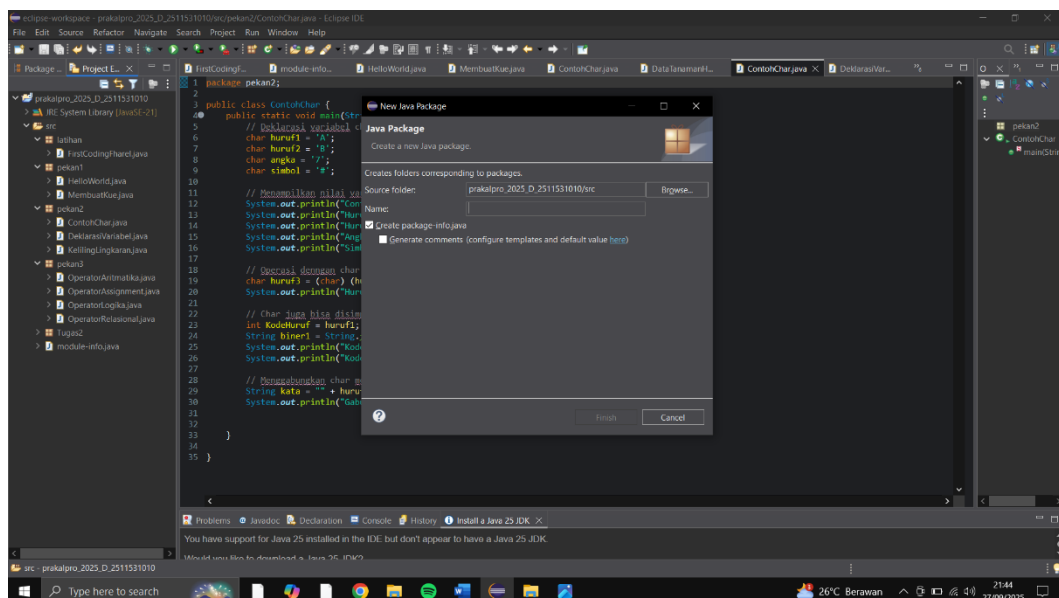
## 2.2 Pembuatan Package dan Class untuk Pekan 3

1. Untuk membuat *package* baru, klik kanan pada src dan tekan “New” setelah itu pilih *package*.



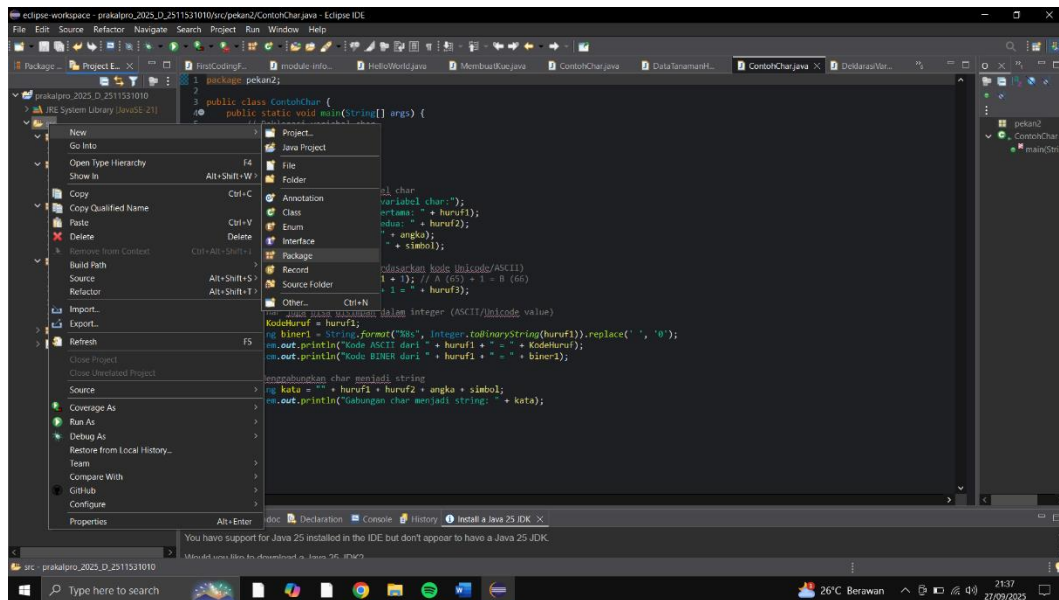
Gambar 2.5

2. Lanjut diberi nama *package* tanpa pakai spasi, huruf kapital ataupun karakter khusus. Seperti “pekan3” dan tekan *Finish*.



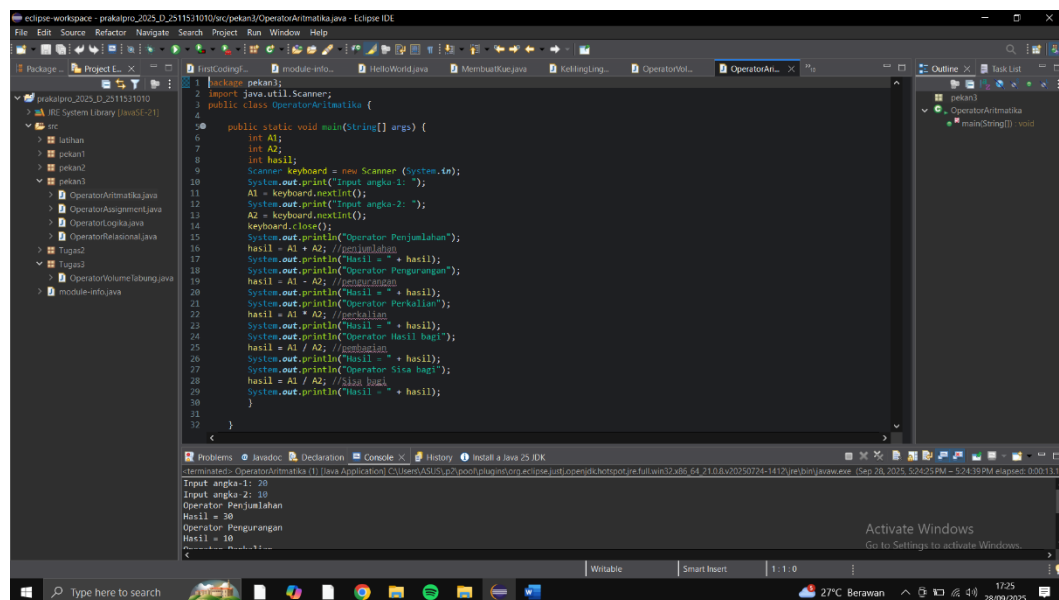
Gambar 2.6

3. Kemudian klik kanan pada package “pekan3” tersebut dan klik “New” dan pilih bagian “Class” untuk memulai membuat program.



Gambar 2.7

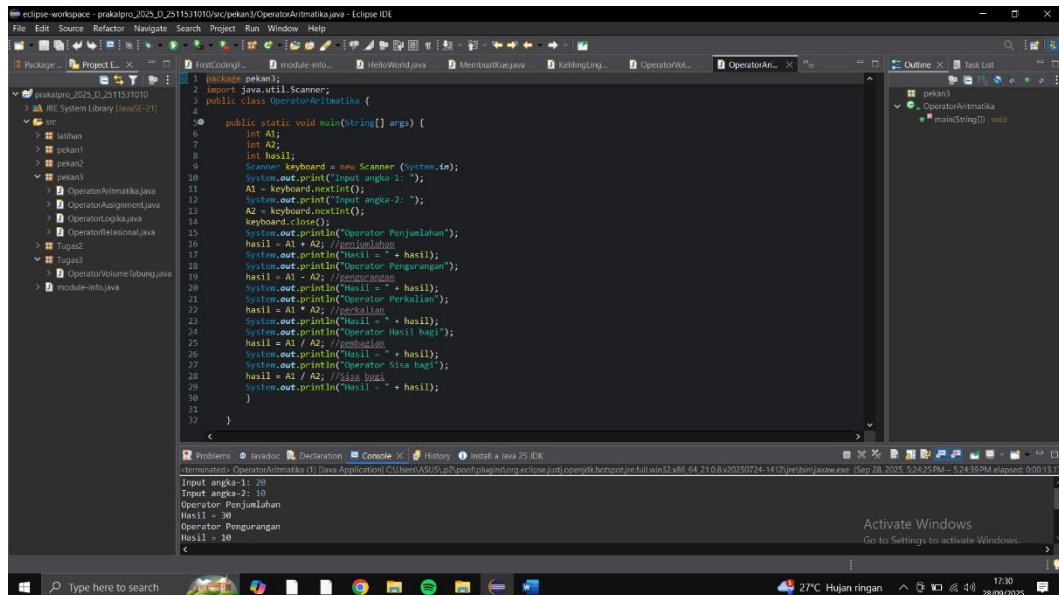
4. Buat nama *Class* yang akan dibuat pada bagian nama tanpa spasi dan menggunakan huruf kapital pada awal kata, lalu klik bagian “*public static void main(string[] args)*”. Kemudian klik *finish*.



Gambar 2.8

## 2.3 Program Pertama (Operator Aritmatika)

Buat *Class* seperti program sebelumnya dan beri nama *Class* tersebut “OperatorAritmatika”. Dan masukkan kode pemrograman berikut.

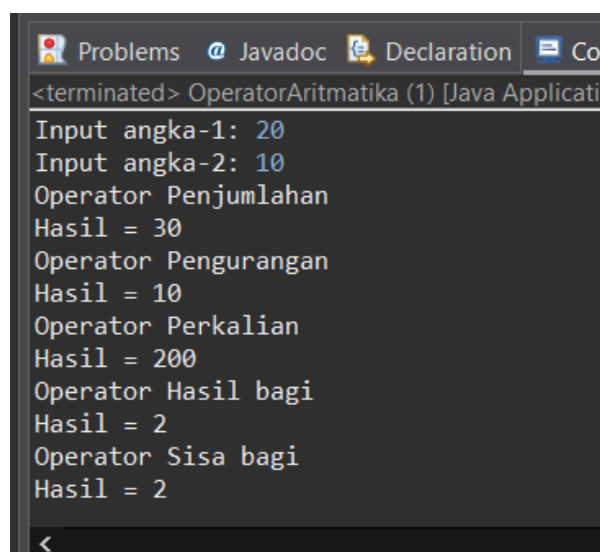


```
1 package pekan3;
2 import java.util.Scanner;
3 public class OperatorAritmatika {
4
5     public static void main(String[] args) {
6         int A1;
7         int A2;
8         int hasil;
9         Scanner keyboard = new Scanner(System.in);
10        System.out.print("Input angka-1: ");
11        A1 = keyboard.nextInt();
12        System.out.print("Input angka-2: ");
13        A2 = keyboard.nextInt();
14        keyboard.close();
15        System.out.println("Operator Penjumlahan");
16        hasil = A1 + A2; //penjumlahan
17        System.out.println("Hasil = " + hasil);
18        System.out.println("Operator Pengurangan");
19        hasil = A1 - A2; //pengurangan
20        System.out.println("Hasil = " + hasil);
21        System.out.println("Operator Perkalian");
22        hasil = A1 * A2; //perkalian
23        System.out.println("Hasil = " + hasil);
24        System.out.println("Operator Hasil bagi");
25        hasil = A1 / A2; //pembagian
26        System.out.println("Hasil = " + hasil);
27        System.out.println("Operator Sisa bagi");
28        hasil = A1 % A2; //sisa bagi
29        System.out.println("Hasil = " + hasil);
30    }
31 }
32 }
```

Gambar 2.9

### 2.3.1 Output Program Pertama (Operator Aritmatika)

Setelah menekan tombol *run*, maka tampil output yang dimana kita diminta untuk menginputkan angka pertama dan kedua, setelah itu hasil operator penjumlahan, pengurangan, perkalian, hasil bagi, sisa hasil bagi akan tampil di layar.



```
<terminated> OperatorAritmatika (1) [Java Application]
Input angka-1: 20
Input angka-2: 10
Operator Penjumlahan
Hasil = 30
Operator Pengurangan
Hasil = 10
Operator Perkalian
Hasil = 200
Operator Hasil bagi
Hasil = 2
Operator Sisa bagi
Hasil = 2
```

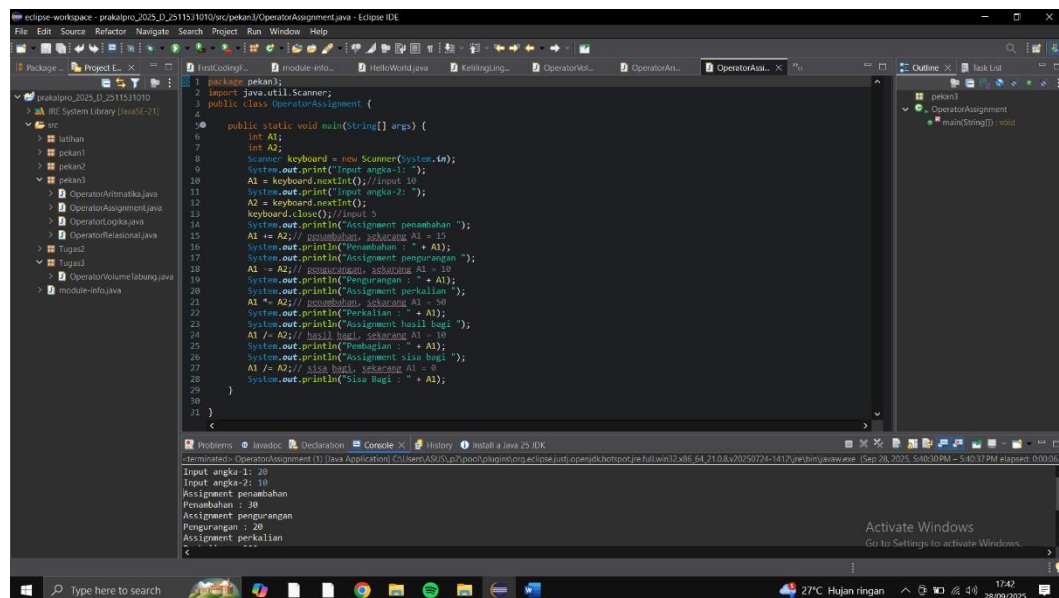
Gambar 2.10

### 2.3.2 Penjelasan Singkat

Program ini dibuat untuk penggunaan operator aritmatika dalam bahasa Java. Program meminta pengguna memasukkan dua buah angka, kemudian memproses angka tersebut dengan berbagai operasi aritmatika, yaitu penjumlahan, pengurangan, perkalian, pembagian, dan sisa bagi. Hasil dari setiap operasi ditampilkan ke layar agar pengguna dapat melihat perbedaan masing-masing operator. Dalam pembacaan *input*, program menggunakan *Class Scanner* dari *library java.util.Scanner*. Dengan adanya program ini, kita dapat memahami cara kerja operator aritmatika di Java secara praktis melalui contoh perhitungan langsung.

### 2.4 Program Kedua (Operator Assingment)

Buat *Class* seperti program sebelumnya dan beri nama *Class* tersebut “OperatorAssingment”. Dan masukkan kode pemograman berikut.



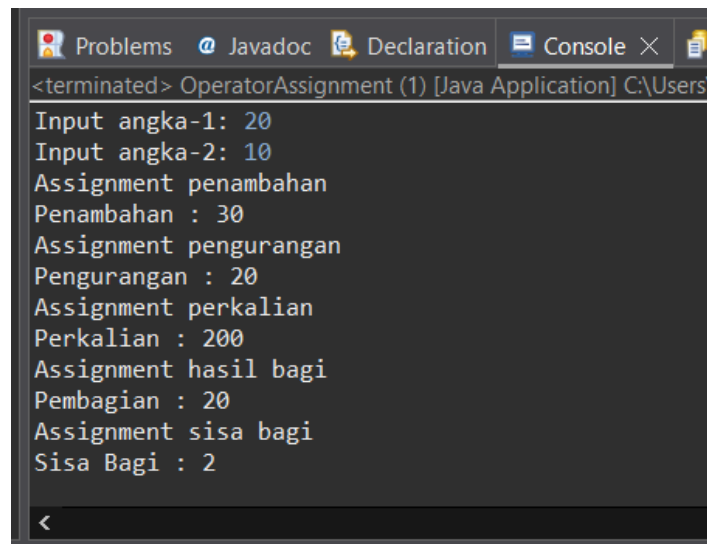
```
package pekan3;
import java.util.Scanner;
public class OperatorAssingment {
    public static void main(String[] args) {
        int A1;
        int A2;
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Input angka-1: ");
        A1 = keyboard.nextInt(); //input 10
        System.out.print("Input angka-2: ");
        A2 = keyboard.nextInt();
        keyboard.close(); //input 5
        System.out.println("Assignment penjumlahan ");
        A1 += A2; // penjumlahan sebanyak A1 = 15
        System.out.println("Penjumlahan : " + A1);
        A1 -= A2; // pengurangan sebanyak A1 = 10
        System.out.println("Assignment pengurangan ");
        System.out.println("Pengurangan : " + A1);
        A1 *= A2; // perkalian sebanyak A1 = 50
        System.out.println("Assignment perkalian ");
        System.out.println("Perkalian : " + A1);
        A1 /= A2; // hasil bagi sebanyak A1 = 10
        System.out.println("Assignment pembagian ");
        System.out.println("Pembagian : " + A1);
        A1 %= A2; // sisa hasil pembagian A1 = 0
        System.out.println("Assignment sisa bagi ");
        System.out.println("Sisa bagi : " + A1);
    }
}
```

Input angka-1: 20  
Input angka-2: 10  
Assignment penjumlahan  
Penjumlahan : 30  
Assignment pengurangan  
Pengurangan : 20  
Assignment perkalian  
Perkalian : 200  
Assignment pembagian  
Pembagian : 2  
Assignment sisa bagi  
Sisa bagi : 0

Gambar 2.11

#### 2.4.1 Output Program Kedua (Operator Assingment)

Setelah menekan tombol *run*, maka tampil output yang dimana kita diminta untuk menginputkan angka pertama dan kedua, setelah itu hasil operator penjumlahan, pengurangan, perkalian, hasil bagi, sisa hasil bagi akan tampil di layar.



```
<terminated> OperatorAssignment (1) [Java Application] C:\Users
Input angka-1: 20
Input angka-2: 10
Assignment penambahan
Penambahan : 30
Assignment pengurangan
Pengurangan : 20
Assignment perkalian
Perkalian : 200
Assignment hasil bagi
Pembagian : 20
Assignment sisa bagi
Sisa Bagi : 2
<
```

Gambar 2.12

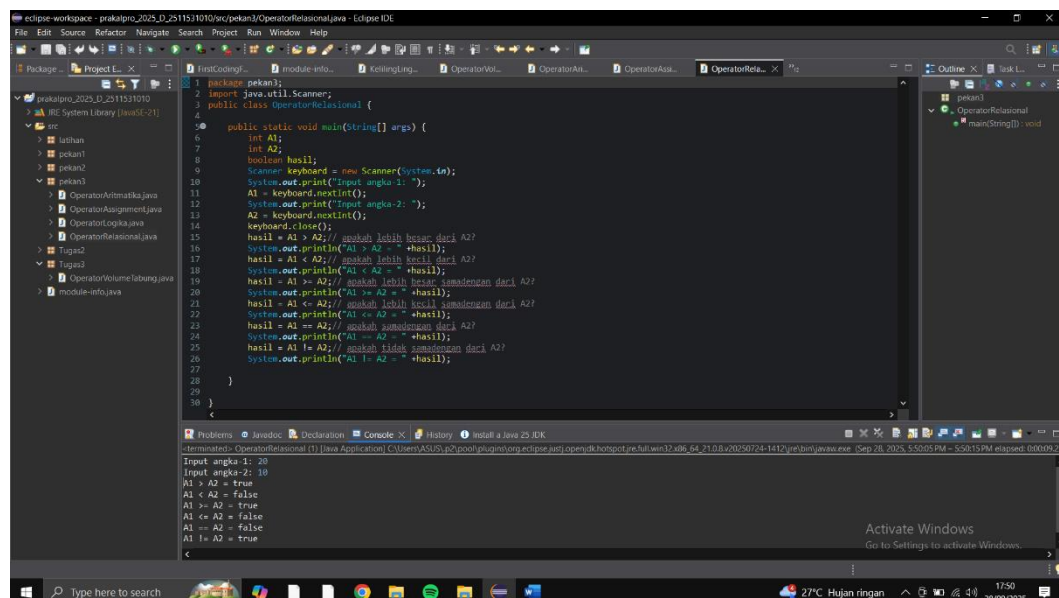
#### 2.4.2 Penjelasan Singkat

Program ini bertujuan untuk memperlihatkan penggunaan berbagai macam operator *assignment* dalam bahasa Java. Program meminta pengguna memasukkan dua buah angka, kemudian angka pertama dimodifikasi dengan operator *assignment* menggunakan angka kedua. Operator yang digunakan meliputi penambahan ( $+=$ ), pengurangan ( $-=$ ), perkalian ( $*=$ ), pembagian ( $/=$ ), dan sisa bagi ( $\%=$ ). Setelah setiap operasi, nilai variabel diperbarui sesuai hasil perhitungan dan langsung ditampilkan ke layar. Dengan cara ini, pengguna dapat memahami bahwa operator *assignment* tidak hanya melakukan perhitungan, tetapi juga secara langsung menyimpan hasilnya kembali ke variabel yang bersangkutan.

#### 2.5 Program Ketiga (Operator Relasional)

Buat *Class* seperti program sebelumnya dan beri nama *Class* tersebut “OperatorRelasional”. Dan masukkan kode pemograman berikut.

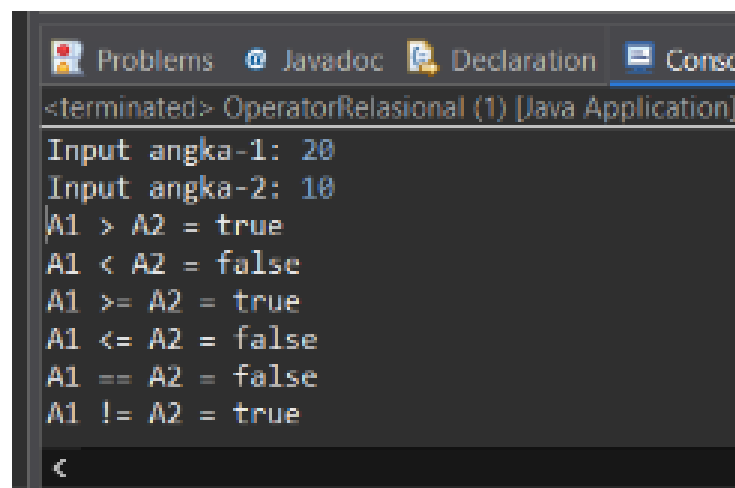




Gambar 2.12

### 2.5.1 Output Program Ketiga (Operator Relasional)

Setelah menekan tombol *run*, maka tampil output yang dimana kita diminta untuk menginputkan angka pertama dan kedua, setelah itu hasil operator akan tampil di layar.



Gambar 2.13

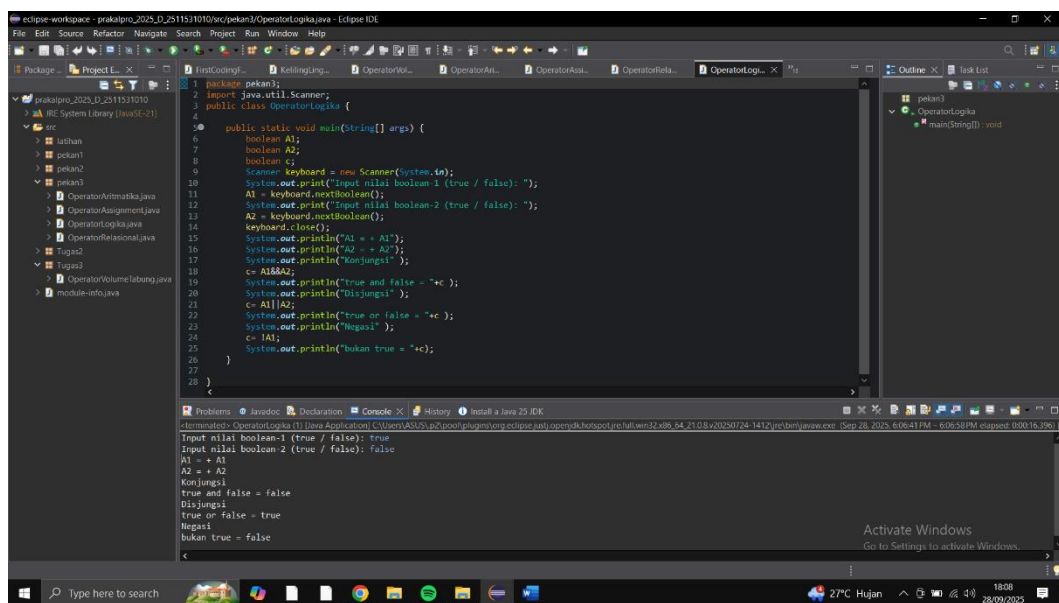
### 2.5.2 Penjelsan Singkat

Program ini digunakan untuk menunjukkan cara kerja operator *assignment* dalam bahasa Java. Setelah pengguna memasukkan dua angka, angka pertama akan dimodifikasi secara langsung menggunakan operator *assignment* dengan angka kedua. Operator yang ditunjukkan meliputi += (penambahan), -= (pengurangan), \*= (perkalian), /= (pembagian), dan %= (sisa bagi). Setiap operasi tidak hanya

melakukan perhitungan, tetapi juga memperbarui nilai variabel secara langsung, sehingga hasil perhitungan otomatis tersimpan pada variabel tersebut. Melalui contoh ini, dapat dipahami bahwa operator *assignment* lebih ringkas dibandingkan penulisan operasi aritmatika biasa karena menggabungkan proses perhitungan dan penyimpanan nilai dalam satu langkah.

## 2.6 Program Keempat (Operator Logika)

Buat *Class* seperti program sebelumnya dan beri nama *Class* tersebut “OperatorLogika”. Dan masukkan kode pemrograman berikut.



```
1 package pekan3;
2 import java.util.Scanner;
3 public class OperatorLogika {
4
5     public static void main(String[] args) {
6         boolean A1;
7         boolean A2;
8         boolean C;
9         Scanner keyboard = new Scanner(System.in);
10        System.out.print("Input nilai boolean-1 (true / false): ");
11        A1 = keyboard.nextBoolean();
12        System.out.print("Input nilai boolean-2 (true / false): ");
13        A2 = keyboard.nextBoolean();
14        keyboard.close();
15        System.out.println("A1 = " + A1);
16        System.out.println("A2 = " + A2);
17        System.out.println("Konjungsi");
18        C = A1 & A2;
19        System.out.println("true and false = " + C);
20        System.out.println("Disjungsi");
21        C = A1 | A2;
22        System.out.println("true or false = " + C);
23        System.out.println("Negasi");
24        C = !A1;
25        System.out.println("bukan true = " + C);
26    }
27
28 }
```

Output Console:

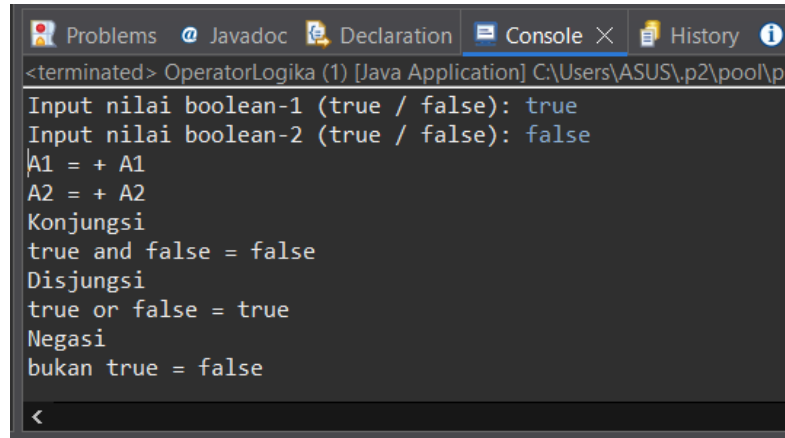
```
Input nilai boolean-1 (true / false): true
Input nilai boolean-2 (true / false): false
A1 = true
A2 = false
Konjungsi
true and false = false
Disjungsi
true or false = true
Negasi
bukan true = false
```

Gambar 2.14

### 2.6.1 Output Program Keempat (Operator Logika)

Setelah menekan tombol *run*, maka tampil output yang dimana kita diminta untuk menginputkan nilai pertama dan kedua apakah *true* atau *false*, setelah itu hasil operator akan tampil di layar.



The image shows a screenshot of a Java IDE's console window. The window has tabs for 'Problems', 'Javadoc', 'Declaration', 'Console', and 'History'. The 'Console' tab is active, displaying the output of a Java application named 'OperatorLogika (1)'. The output shows two prompts for boolean input: 'Input nilai boolean-1 (true / false): true' and 'Input nilai boolean-2 (true / false): false'. Below these, the program performs several operations: 'A1 = + A1', 'A2 = + A2', a conjunction test 'Konjungsi' resulting in 'true and false = false', a disjunction test 'Disjungsi' resulting in 'true or false = true', and a negation test 'Negasi' resulting in 'bukan true = false'. A cursor is visible at the bottom of the console.

```
<terminated> OperatorLogika (1) [Java Application] C:\Users\ASUS\p2\pool\p
Input nilai boolean-1 (true / false): true
Input nilai boolean-2 (true / false): false
A1 = + A1
A2 = + A2
Konjungsi
true and false = false
Disjungsi
true or false = true
Negasi
bukan true = false
<
```

Gambar 2.15

### 2.6.2 Penjelsan Singkat

Program ini dibuat untuk mendemonstrasikan penggunaan operator logika pada bahasa Java dengan tipe data *boolean*. Program meminta pengguna memasukkan dua nilai boolean (*true* atau *false*), lalu menguji hasil operasi logika menggunakan operator konjungsi (&&), disjungsi (||), dan negasi (!). Pada operator &&, hasil bernilai *true* hanya jika kedua nilai bernilai *true*. Operator || menghasilkan *true* jika salah satu *operand* bernilai *true*. Sedangkan operator ! berfungsi membalikkan nilai *boolean*, sehingga jika *input true* maka hasilnya menjadi *false*, dan sebaliknya.

## BAB III

### PENUTUP

#### 3.1 Kesimpulan

Berdasarkan hasil praktikum yang dilakukan, dapat disimpulkan bahwa operator dalam Java merupakan dasar penting dalam membangun logika dan struktur program. Pemahaman atas berbagai jenis operator seperti aritmatika, relasional, logika, dan *assignment* memungkinkan programmer mengolah data sekaligus membuat keputusan secara lebih efisien. Melalui percobaan langsung, mahasiswa mendapatkan pengalaman nyata dalam menggunakan operator untuk menyelesaikan persoalan komputasi sederhana, sekaligus memahami cara kerja serta pengaruhnya terhadap hasil program.

Selain itu, pemanfaatan *Class Scanner* sebagai sarana *input* terbukti mendukung interaksi antara pengguna dan program, sehingga penerapan operator dapat dipahami dalam konteks yang lebih nyata. Praktikum ini juga menegaskan pentingnya ketelitian dalam menuliskan sintaks dan pemilihan operator yang sesuai dengan kebutuhan logika program. Dengan menguasai operator dasar, mahasiswa memiliki bekal yang kuat untuk mempelajari materi pemrograman lanjutan seperti struktur kontrol, fungsi, maupun pemrograman berorientasi objek.

## DAFTAR PUSTAKA

- [1] Suarga. (2019). *Algoritma dan Pemrograman*. Yogyakarta: Deepublish.
- [2] Nugroho, B. (2020). *Dasar Pemrograman Java*. Jakarta: PT Elex Media Komputindo.
- [3] Galuh, R. (2019). *Operator Aritmatika dan Operator Relasi*. Diakses Tersedia pada: <https://galuhratna.alza.web.id/blog/2019/11/operator-aritmatika-dan-operator-relasi/>. [Diakses: 28-Sep-2025].
- [4] Andi, R. (2021). *Pemrograman Java untuk Pemula*. Bandung: Informatika Bandung.