

LAPORAN PRATIKUM  
ALGORITMA DAN PEMROGRAMAN  
PENERAPAN OPERATOR DALAM JAVA



disusun Oleh:  
Annisa Layli Ramadhani  
2511532024

Dosen Pengampu: Wahyudi. Dr. S.T.M.T  
Asisten Pratikum: Rahmad Dwirizki Olders

DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
TAHUN 2025

## **KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Allah SWT karena dengan rahmat-Nya laporan praktikum Algoritma dan Pemrograman tanggal 22 September 2025 dapat diselesaikan. Laporan ini membahas materi tentang Penerapan Operator dalam Java sebagai dasar penting dalam memahami pemrograman. Penulis mengucapkan terima kasih kepada dosen, asisten, dan rekan-rekan yang telah membantu selama praktikum. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran sangat diharapkan. Semoga laporan ini bermanfaat bagi pembaca.

Padang, 25 September 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan .....	1
1.3 Manfaat .....	2
BAB II .....	3
2.1 Dasar Teori .....	3
2.1.1 Class Scanner.....	3
2.1.2 Operator Aritmatika.....	3
2.1.3 Operator <i>Assignment</i> .....	4
2.1.4 Operator Relasional .....	4
2.1.5 Operator Logika.....	5
2.2 Langkah Kerja.....	5
2.2.1 Pembuatan <i>Package</i> dan <i>Class</i> Pekan 3 .....	5
2.2.2 Program Pertama: Operator Aritmatika .....	7
2.2.3 Program Kedua: Operator <i>Assignment</i> .....	7
2.2.4 Program Ketiga: Operator Relasional .....	8
2.2.5 Program Keempat: Operator Logika .....	9
2.2.6 <i>Commit</i> dan <i>Push</i> ke GitHub.....	10
BAB III.....	12
3.1 Kesimpulan .....	12
3.2 Saran.....	12
DAFTAR PUSTAKA .....	13

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam pemrograman Java, operator memiliki peran penting untuk menyusun logika dan mengolah data. Operator aritmatika digunakan untuk melakukan perhitungan dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian, serta modulus. Operator relasional berfungsi membandingkan dua nilai dan menghasilkan output berupa *true* atau *false*. Operator logika berfungsi menghubungkan ekspresi logis dengan simbol untuk menentukan kondisi lebih dari satu pernyataan. Sementara itu, operator *assignment* digunakan untuk memberikan nilai pada variabel dengan cara yang lebih praktis.

Selain operator, Java juga menyediakan *class Scanner* yang berfungsi untuk menerima input dari pengguna melalui *keyboard*. Dengan *Scanner*, program menjadi lebih interaktif karena dapat menyesuaikan hasil sesuai data yang dimasukkan.

Pemahaman tentang operator dan penggunaan *Scanner* sangat penting sebagai dasar dalam belajar pemrograman Java. Dengan menguasai materi ini, kita dapat membuat program sederhana yang mampu melakukan perhitungan, perbandingan, hingga interaksi dengan pengguna, yang nantinya menjadi fondasi untuk mempelajari konsep pemrograman yang lebih kompleks.

### **1.2 Tujuan**

Tujuan dari pelaksanaan praktikum sebagai berikut:

- 1.2.1 Memahami penggunaan operator aritmatika, relasional, logika, dan assignment dalam Java.
- 1.2.2 Mempelajari cara mengambil input dari pengguna menggunakan *class Scanner*.
- 1.2.3 Mengimplementasikan operator dan input dalam program sederhana yang interaktif.

### **1.3 Manfaat**

Manfaat dari pelaksanaan praktikum sebagai berikut:

- 1.3.1 Membantu mahasiswa memahami dasar-dasar logika pemrograman Java.
- 1.3.2 Menjadi bekal awal untuk membangun program dengan tingkat kompleksitas lebih tinggi.
- 1.3.3 Melatih keterampilan dalam mengolah data, membuat percabangan, dan merancang program yang interaktif.

## BAB II

### PEMBAHASAN

#### 2.1 Dasar Teori

##### 2.1.1 Class Scanner

*Scanner* merupakan *class* yang menyediakan fungsi-fungsi untuk mengambil input dari *keyboard*. Agar kita bisa menggunakan *Scanner*, kita perlu mengimpornya ke dalam kode:

```
Import java.util.Scanner;
```

##### 2.1.2 Operator Aritmatika

Operator aritmatika merupakan operator dasar dalam bahasa pemrograman Java yang digunakan untuk melakukan perhitungan matematis terhadap data numerik, baik bilangan bulat (*integer*) maupun bilangan desimal (*floating point*). Operator ini menjadi fondasi penting karena hampir semua program butuh proses perhitungan, baik yang sederhana maupun kompleks.

Operator	Penggunaan	Deskripsi
+	$A1 + A2$	Menambahkan A1 dengan A2
-	$A1 - A2$	Mengurangkan A1 dengan A2
*	$A1 * A2$	Mengalikan A1 dengan A2
/	$A1 / A2$	Membagi A1 dengan A2
%	$A1 \% A2$	Menghasilkan sisa hasil bagi antara A1 dengan A2

### 2.1.3 Operator Assignment

Operator *assignment* (penugasan) fungsinya untuk memberikan tugas pada variabel tertentu. Pada java operator *assignment* digunakan untuk memberikan nilai pada sebuah variabel.

Operator	Penggunaan	Ekivalen dengan
+=	A1 += A2	A1 = A1+A2
-=	A1 -= A2	A1 = A1-A2
*=	A1 *= A2	A1 = A1*A2
/=	A1 /= A2	A1 = A1/A2
%=	A1 %= A2	A1 = A1%A2

### 2.1.4 Operator Relasional

Operator relasional berfungsi untuk membandingkan 2 variabel dalam Java. Nilai yang dihasilkan dari operator ini berupa *boolean*, yaitu: *true* dan *false*. Operator ini terdiri dari:

Operator	Penggunaan	Deskripsi
>	A1 > A2	Menghasilkan <i>true</i> jika A1 besar dari A2
<	A1 < A2	Menghasilkan <i>true</i> jika A1 kecil dari A2
>=	A1 >= A2	Menghasilkan <i>true</i> jika A1 besar atau sama dengan A2
<=	A1 <= A2	Menghasilkan <i>true</i> jika A1 kecil atau sama dengan A2
==	A1 == A2	Menghasilkan <i>true</i> jika A1 sama dengan A2
!=	A1 != A2	Menghasilkan <i>true</i> jika A1 tidak sama dengan A2

### 2.1.5 Operator Logika

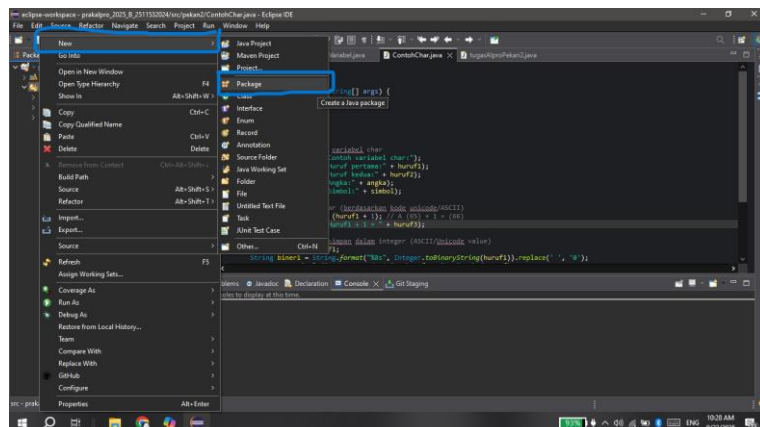
Operator logika dalam Java digunakan untuk menghubungkan dua atau lebih ekspresi logis dan menghasilkan nilai berupa *true* atau *false*.

Operator	Penggunaan	Deskripsi
&&	A1&&A2	Menghasilkan <i>true</i> jika A1 dan A2, bernilai <i>boolean true</i>
	A1   A2	Menghasilkan <i>true</i> jika salah satu A1 atau A2 bernilai <i>true</i>
!	!A1	Menghasilkan <i>true</i> jika A1 bernilai <i>false</i>

## 2.2 Langkah Kerja

### 2.2.1 Pembuatan *Package* dan *Class* Pekan 3

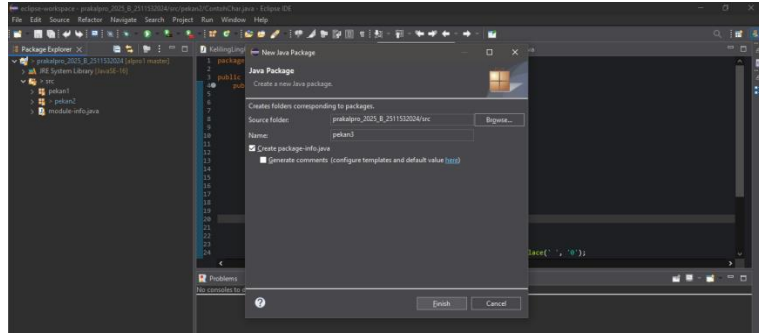
1. Buka aplikasi eclipse, lalu buat package baru pada bagian src klik kanan, lalu pilih “New” dan klik tulisan Package.



### 2.2.1 Pembuatan *Package*

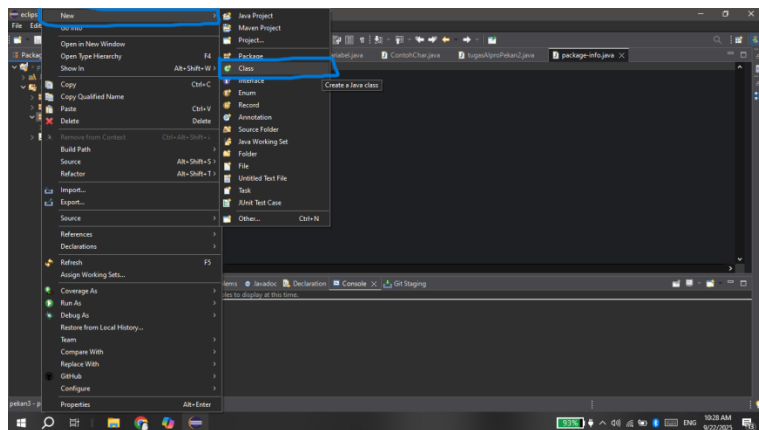
2. Kemudian buat nama packagenya tanpa spasi, huruf kapital, karakter khusus. beri nama “pekan3”. Lalu klik finish.





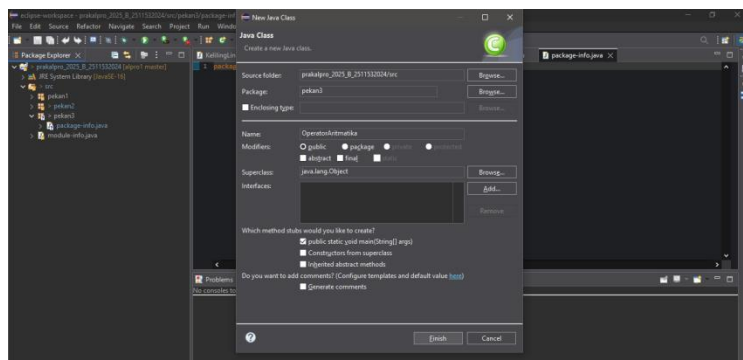
### 2.2.2 Pembuatan *Package*

3. Ketika package sudah jadi. Klik kanan pada bagian “pekan3”. Lalu klik “New” dan pilih bagian “Class” untuk memulai membuat program.



### 2.2.3 Pembuatan *Class*

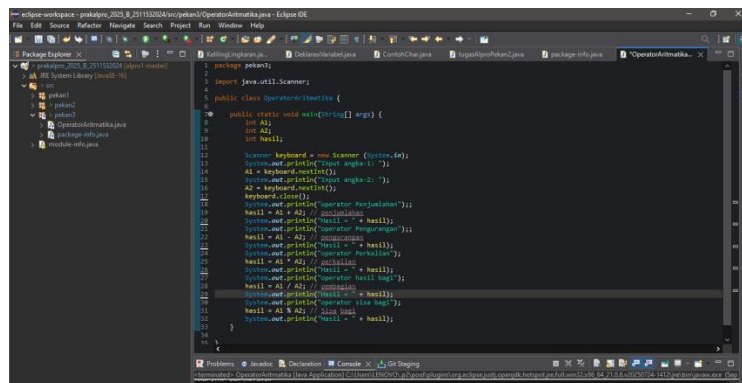
4. Buat nama Class program yang akan dibuat pada bagian nama tanpa spasi dan menggunakan huruf kapital pada awal kata, lalu klik bagian “public static void main(string[]args)”. Kemudian klik “finish”.



### 2.2.4 Pembuatan *Class*

## 2.2.2 Program Pertama: Operator Aritmatika

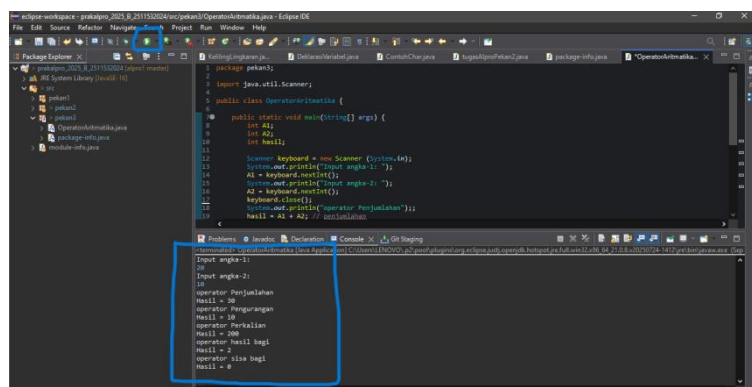
1. Masukkan sintaks berikut.



```
1 package petan2;  
2 import java.util.Scanner;  
3  
4 public class OperatorAritmatika {  
5     public static void main(String[] args) {  
6         Scanner keyboard = new Scanner (System.in);  
7         System.out.println("Input angka-1: ");  
8         A1 = keyboard.nextInt();  
9         System.out.println("Input angka-2: ");  
10        A2 = keyboard.nextInt();  
11        keyboard.close();  
12  
13        System.out.println("operator Penjumlahan");  
14        Hasil = A1 + A2; // penjumlahan  
15        System.out.println("Hasil = " + Hasil);  
16  
17        System.out.println("operator Pengurangan");  
18        Hasil = A1 - A2; // pengurangan  
19        System.out.println("Hasil = " + Hasil);  
20  
21        System.out.println("operator Perkalian");  
22        Hasil = A1 * A2; // perkalian  
23        System.out.println("Hasil = " + Hasil);  
24  
25        System.out.println("operator Pembagian");  
26        Hasil = A1 / A2; // pembagian  
27        System.out.println("operator sisa bagi");  
28        Hasil = A1 % A2; // sisa bagi  
29        System.out.println("Hasil = " + Hasil);  
30    }  
31 }
```

2.2.5 sintaks operator aritmatika

2. Lalu klik bagian “run” untuk melihat hasil sintaks yang sudah di inputkan. Lalu akan menghasilkan output seperti berikut.



```
Input angka-1:  
10  
Input angka-2:  
20  
operator Penjumlahan  
Hasil = 30  
operator Pengurangan  
Hasil = 10  
operator Perkalian  
Hasil = 200  
operator hasil bagi  
Hasil = 2  
operator sisa bagi  
Hasil = 0
```

2.2.6 Running kode dan output sintaks

## 2.2.3 Program Kedua: Operator Assignment

1. Ulangi pembuatan Class seperti program sebelumnya dan beri nama Class tersebut “OperatorAssignment”. Lalu masukkan sintaks seperti berikut.

```

package petakari;

import java.util.Scanner;

public class OperatorAssignment {

    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();
        System.out.print("Input angka-2: ");
        A2 = keyboard.nextInt();
        keyboard.close();
        System.out.println("Assignment penambahan");
        A1 += A2; // rumus hasil: a1 + a2
        System.out.println("Assignment pengurangan");
        A1 -= A2; // rumus hasil: a1 - a2
        System.out.println("Assignment perkalian");
        A1 *= A2; // rumus hasil: a1 * a2
        System.out.println("Assignment hasil bagi");
        A1 /= A2; // rumus hasil: a1 / a2
        System.out.println("Assignment sisa bagi");
        A1 %= A2; // rumus hasil: a1 % a2
        System.out.println("Assignment sisa bagi");
    }
}

```

2.2.7 sintaks operator assignment

2. Klik “run” untuk melihat hasil sintaks yang sudah di buat.

```

Input angka-1: 15
Input angka-2: 10
Assignment penambahan
Assignment pengurangan
Assignment perkalian
Assignment hasil bagi
Assignment sisa bagi
Assignment sisa bagi

```

2.2.8 running code dan output sintaks

## 2.2.4 Program Ketiga: Operator Relasional

1. Ulangi pembuatan Class seperti program sebelumnya dan beri nama Class tersebut “OperatorRelasional”. Lalu masukkan sintaks seperti berikut.

```

package petakari;

import java.util.Scanner;

public class OperatorRelasional {

    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();
        System.out.print("Input angka-2: ");
        A2 = keyboard.nextInt();
        keyboard.close();
        System.out.println("Assignment A1 lebih besar A2?");
        System.out.println("A1 > A2 = " + hasil);
        System.out.println("Assignment A1 lebih kecil A2?");
        System.out.println("A1 < A2 = " + hasil);
        System.out.println("Assignment A1 sama dengan A2?");
        System.out.println("A1 == A2 = " + hasil);
        System.out.println("Assignment A1 tidak sama dengan A2?");
        System.out.println("A1 != A2 = " + hasil);
    }
}

```

2.2.9 sintaks operator relasional

2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.

```

1 package pekan3;
2
3 import java.util.Scanner;
4
5 public class OperatorRelasional {
6     int A1;
7     boolean A2;
8     Scanner keyboard = new Scanner(System.in);
9     System.out.print("Input angka-1: ");
10    A1 = keyboard.nextInt();
11    System.out.print("Input angka-2: ");
12    A2 = keyboard.nextInt();
13    keyboard.close();
14    hasil = A1 > A2; // apakah A1 lebih besar A2?
15    System.out.println("A1 > A2 = " + hasil);
16    hasil = A1 < A2; // apakah A1 lebih kecil A2?
17    System.out.println("A1 < A2 = " + hasil);
18 }

```

```

Input angka-1: 10
Input angka-2: 20
A1 > A2 = false
A1 < A2 = true
A1 > A2 = false
A1 < A2 = true
A1 > A2 = false
A1 < A2 = true

```

2.2.10 running code dan output sintaks

## 2.2.5 Program Keempat: Operator Logika

1. Ulangi pembuatan Class seperti program sebelumnya dan beri nama Class tersebut “OperatorLogika”. Lalu masukkan sintaks seperti berikut.

```

1 package pekan3;
2
3 import java.util.Scanner;
4
5 public class OperatorLogika {
6     boolean A1;
7     boolean A2;
8     boolean A3;
9     Scanner keyboard = new Scanner(System.in);
10    System.out.print("Input nilai boolean-1 (true/false): ");
11    A1 = keyboard.getBoolean(); // Input 10
12    System.out.print("Input nilai boolean-2 (true/false): ");
13    A2 = keyboard.getBoolean();
14    keyboard.close();
15    System.out.println("A1 = " + A1);
16    System.out.println("A2 = " + A2);
17    System.out.println("A3 = " + A3);
18    c1 = A1 & A2;
19    System.out.println("true and false = " + c1);
20    c2 = A1 | A2;
21    System.out.println("true and false = " + c2);
22    c3 = A1 ^ A2;
23    System.out.println("true and false = " + c3);
24    c4 = !A1;
25    System.out.println("negasi true = " + c4);
26    System.out.println("bukan true = " + c4);
27 }

```

2.2.11 sintaks operator logika

2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.

```

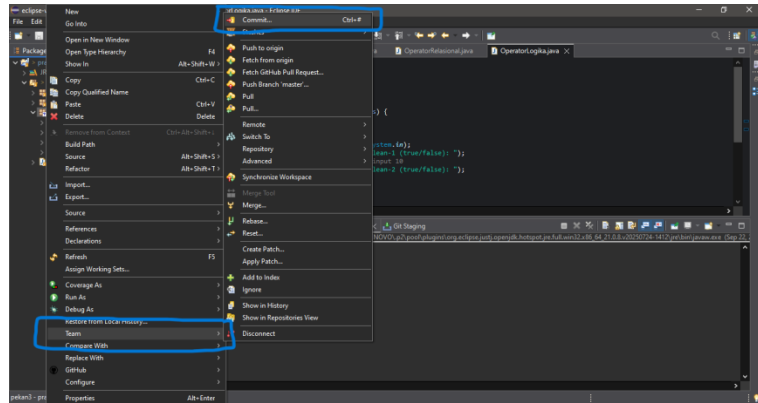
Input nilai boolean-1 (true/false): true
Input nilai boolean-2 (true/false): false
A1 = true
A2 = false
true and false = false
true or false = true
true xor false = true
negasi true = false
bukan true = false

```

2.2.12 running code dan output sintaks

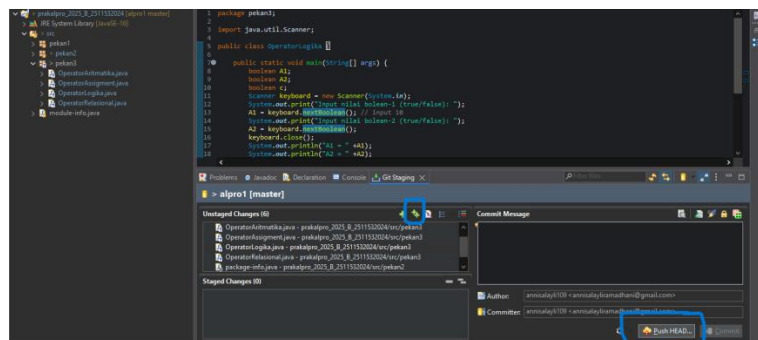
## 2.2.6 Commit dan Push ke GitHub

1. Klik kanan pada bagian “pekan2”. Lalu klik “Team” dan pilih bagian “Commit...”.



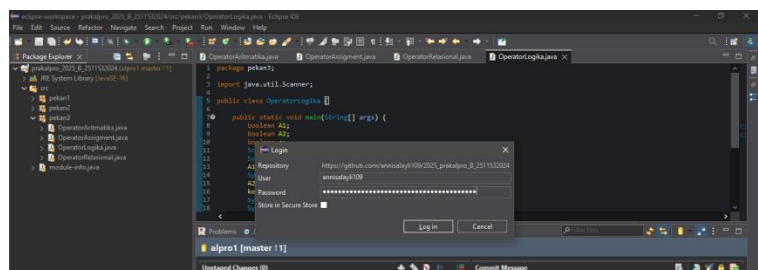
2.2.13 proses commit dan push ke GitHub

2. Klik bagian “Add file” (bagian tanda tambah) tugas yang ingin di push ke github, lalu buat “commit message” untuk tugas berikut. Setelah itu klik “Commit and Push”.



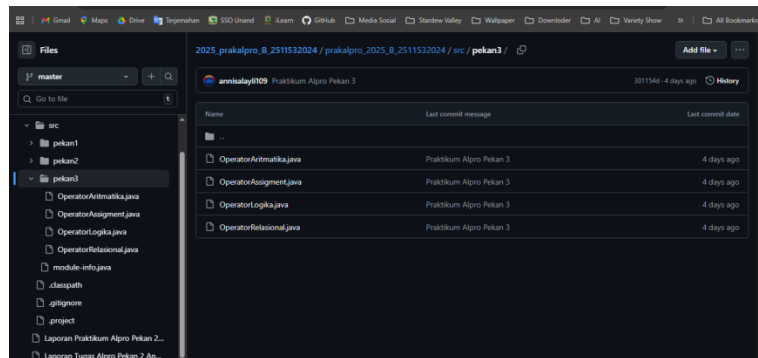
2.2.14 proses commit dan push ke GitHub

3. Lalu masukkan username akun GitHub dan masukkan token GitHub yang sudah dibuat sebelumnya. Lalu klik “Log in”.



2.2.15 proses commit dan push ke GitHub

4. Kemudian cek akun GitHub untuk melihat apakah file yang di push sudah masuk ke akun GitHub. Jika sudah masuk ke akun GitHub maka sudah selesai.



2.2.16 proses commit dan push ke GitHub

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Dari praktikum mengenai operator dan penggunaan class Scanner pada bahasa pemrograman Java dapat disimpulkan bahwa operator memiliki peran penting dalam membangun logika program. Operator aritmatika digunakan untuk melakukan perhitungan, operator assignment untuk memberikan nilai pada variabel, operator relasional untuk membandingkan nilai, dan operator logika untuk menghubungkan ekspresi logis. Selain itu, class Scanner memungkinkan program menerima input dari pengguna sehingga lebih interaktif.

Dengan memahami dan menguasai penggunaan berbagai operator serta Scanner, kita dapat membuat program sederhana yang mampu melakukan perhitungan, perbandingan, pengambilan keputusan, hingga interaksi dengan pengguna. Pemahaman dasar ini menjadi fondasi untuk mempelajari konsep pemrograman Java yang lebih kompleks di tahap selanjutnya.

#### **3.2 Saran**

Dalam mempelajari pemrograman Java, sebaiknya mahasiswa tidak hanya memahami teori mengenai operator dan penggunaan class Scanner, tetapi juga memperbanyak latihan praktik dengan membuat program sederhana. Dengan sering berlatih, pemahaman konsep akan lebih kuat dan keterampilan logika pemrograman dapat berkembang. Selain itu, disarankan untuk terus mengeksplorasi materi lanjutan agar penguasaan bahasa Java semakin mendalam.

## DAFTAR PUSTAKA

- [1] Koding Alza, “Operator pada bahasa pemrograman Java,” *Koding.Alza.Web.Id.* [Online]. Available: <https://koding.alza.web.id/operator-pada-bahasa-pemrograman-java/>. [Accessed: 25-Sep-2025].
- [2] Oracle, “Class Scanner (Java Platform SE 17),” Oracle, 2025. [Online]. Available: <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Scanner.html> [Akses: 25-Sep-2025].
- [3] Oracle, “The Java™ Tutorials,” Oracle, 2025. [Online]. Available: <https://docs.oracle.com/javase/tutorial/> [Akses: 25-Sep-2025].