

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMOGRAMAN
OPERATOR PADA JAVA
PEKAN 3

Disusun Oleh :

KINAYA NOVRYA MANDA
(2511531016)

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

MUHAMMAD ZAKI AL HAFIZ



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis sampaikan kehadiran Allah SWT. Salawat dan salam disampaikan kepada Nabi Muhammad SAW. Karena thaufik dan hidayah-Nya, laporan praktikum Java ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban dari kegiatan praktikum yang telah dilaksanakan, sekaligus sebagai sarana untuk memperdalam pemahaman mengenai konsep dasar serta penerapan bahasa pemrograman Java.

Melalui praktikum ini, penulis memperoleh pengalaman langsung dalam memahami operator yang terdapat pada java dan implementasi logika pemrograman menggunakan Java. Diharapkan laporan ini dapat memberikan gambaran yang jelas mengenai materi yang telah dipelajari serta hasil dari percobaan yang dilakukan selama praktikum berlangsung.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa yang akan datang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, serta semua pihak yang telah memberikan bimbingan, arahan, dan dukungan sehingga laporan praktikum ini dapat tersusun. Semoga laporan ini dapat bermanfaat bagi pembaca, khususnya dalam memahami dasar-dasar pemrograman Java.

Padang, September 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR LAMPIRAN	iii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	2
BAB 2 PEMBAHASAN	
2.1 Kode Program.....	3
2.2 Langkah Kerja	11
2.3 Analisis Hasil.....	18
BAB 3 KESIMPULAN	
3.1 Hasil Praktikum	20
3.2 Saran Pengembangan	24
DAFTAR KEPUSTAKAAN	26

DAFTAR LAMPIRAN

Kode Program 2.1 (Kode Program OperatorAritmatika)	3
Kode Program 2.2 (Kode Program OperatorAssignment).....	5
Kode Program 2.3 (Kode Program OperatorRelasional)	7
Kode Program 2.4 (Kode Program OperatorLogika)	9
Gambar 2.1 (Langkah 1 OperatorAritmatika).....	11
Gambar 2.2 (Langkah 2 OperatorAritmatika).....	11
Gambar 2.3 (Langkah 3 OperatorAritmatika).....	12
Gambar 2.4 (Langkah 4 dan 5 OperatorAritmatika)	12
Gambar 2.5 (Langkah 1 OperatorAssignment)	13
Gambar 2.6 (Langkah 2 OperatorAssignment)	13
Gambar 2.7 (Langkah 3 OperatorAssignment)	13
Gambar 2.8 (Langkah 4 dan 5 OperatorAssignment).....	14
Gambar 2.9 (Langkah 1 OperatorRelasional)	15
Gambar 2.10 (Langkah 2 OperatorRelasional)	15
Gambar 2.11 (Langkah 3 OperatorRelasional)	15
Gambar 2.12 (Langkah 4 dan 5 OperatorRelasional)	16
Gambar 2.13 (Langkah 1 OperatorLogika).....	17
Gambar 2.14 (Langkah 2 OperatorLogika).....	17
Gambar 2.15 (Langkah 3 OperatorLogika).....	17
Gambar 2.16 (Langkah 4 dan 5 OperatorLogika)	18
Gambar 3.1 (Hasil Praktikum OperatorAritmatika)	20
Gambar 3.2 (Hasil Praktikum OperatorAssignment)	21
Gambar 3.3 (Hasil Praktikum OperatorRelasional)	22
Gambar 3.4 (Hasil Praktikum OperatorLogika).....	23

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, *operator* memiliki peran yang sangat penting karena hampir semua instruksi program melibatkan proses perhitungan, perbandingan, maupun pengolahan logika. Bahasa pemrograman Java, sebagai salah satu bahasa yang banyak digunakan di dunia industri maupun akademik, menyediakan berbagai macam *operator* yang berfungsi untuk mempermudah pengolahan data dalam program.

Pada praktikum ini, fokus pembelajaran difokuskan pada empat jenis *operator* utama, yaitu *operator aritmatika*, *relasional*, *logika*, dan *assignment*. *Operator aritmatika* digunakan untuk melakukan perhitungan matematis yang menjadi dasar dalam pengolahan angka. *Operator relasional* membantu dalam melakukan perbandingan nilai sehingga dapat digunakan dalam pengambilan keputusan. *Operator logika* berfungsi untuk mengombinasikan kondisi dan menghasilkan nilai kebenaran (*true/false*) yang sangat penting dalam struktur kendali program. Sementara itu, *operator assignment* digunakan untuk memberikan nilai pada variabel sekaligus menyederhanakan proses penulisan kode agar lebih ringkas dan efisien.

Pemahaman terhadap *operator* bukan hanya sebatas teori, melainkan juga harus dilatih melalui praktik langsung dalam penulisan program. Dengan melakukan praktikum ini, mahasiswa diharapkan dapat menguasai penggunaan *operator* secara tepat, memahami cara kerja logika pemrograman, serta terbiasa menyusun kode yang sistematis. Penguasaan terhadap *operator* juga akan menjadi pondasi dasar sebelum mempelajari konsep yang lebih kompleks seperti perulangan, percabangan, *array*, hingga pemrograman berorientasi objek. Dengan demikian, praktikum operator pada Java ini tidak hanya melatih keterampilan teknis, tetapi juga menumbuhkan logika berpikir komputasional yang terstruktur.

1.2 Tujuan Praktikum

Tujuan dilakukannya praktikum ini adalah sebagai berikut :

1. Memahami konsep dan penggunaan operator aritmatika, relasional, logika, dan assignment pada Java.
2. Mampu menuliskan program sederhana yang memanfaatkan operator-operator tersebut.
3. Melatih keterampilan dalam mengombinasikan operator untuk menyelesaikan permasalahan pemrograman.
4. Menjadi bekal dalam memahami materi lanjutan yang membutuhkan penerapan operator.

1.3 Manfaat Praktikum

Manfaat dilakukannya praktikum ini adalah sebagai berikut :

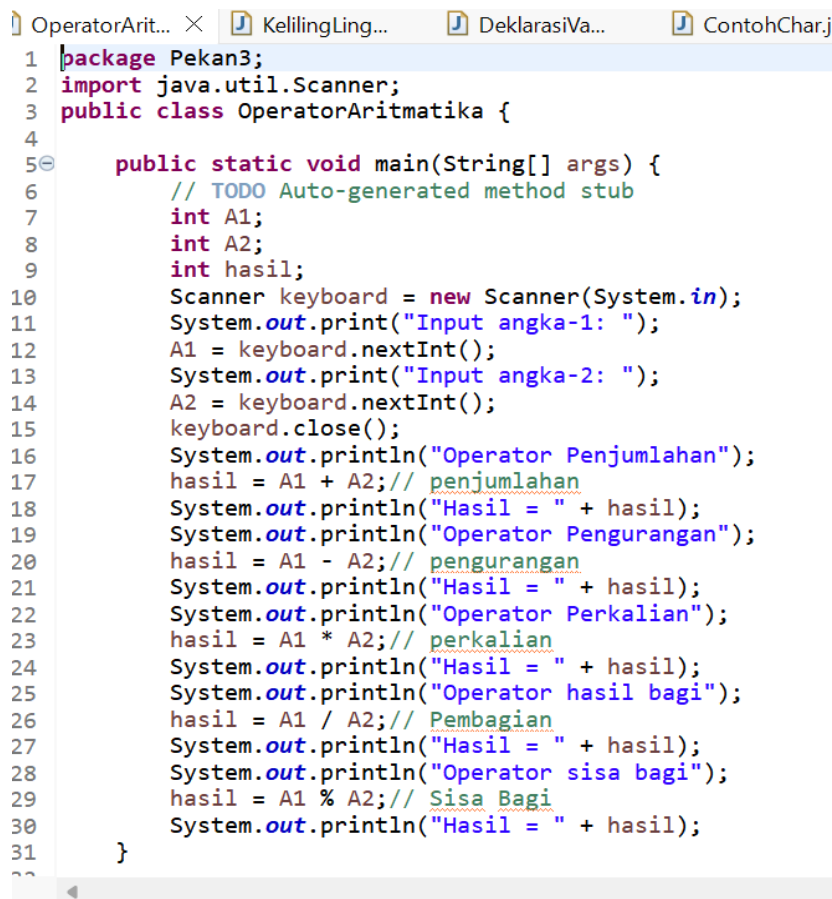
1. Mahasiswa memahami fungsi dan peran operator dalam bahasa Java.
2. Mahasiswa mampu mengimplementasikan operator aritmatika, relasional, logika, dan assignment dalam program sederhana.
3. Mahasiswa terbiasa menggunakan operator sebagai dasar untuk membangun logika pemrograman yang lebih kompleks.
4. Meningkatkan kemampuan analisis dan penyusunan kode yang lebih efektif serta efisien.

BAB II

PEMBAHASAN

2.1 Kode Program

2.1.1 OperatorAritmatika



```

1 package Pekan3;
2 import java.util.Scanner;
3 public class OperatorAritmatika {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int A1;
8         int A2;
9         int hasil;
10        Scanner keyboard = new Scanner(System.in);
11        System.out.print("Input angka-1: ");
12        A1 = keyboard.nextInt();
13        System.out.print("Input angka-2: ");
14        A2 = keyboard.nextInt();
15        keyboard.close();
16        System.out.println("Operator Penjumlahan");
17        hasil = A1 + A2; // penjumlahan
18        System.out.println("Hasil = " + hasil);
19        System.out.println("Operator Pengurangan");
20        hasil = A1 - A2; // pengurangan
21        System.out.println("Hasil = " + hasil);
22        System.out.println("Operator Perkalian");
23        hasil = A1 * A2; // perkalian
24        System.out.println("Hasil = " + hasil);
25        System.out.println("Operator hasil bagi");
26        hasil = A1 / A2; // Pembagian
27        System.out.println("Hasil = " + hasil);
28        System.out.println("Operator sisa bagi");
29        hasil = A1 % A2; // Sisa Bagi
30        System.out.println("Hasil = " + hasil);
31    }
32

```

Kode Program 2.1 (Kode Program OperatorAritmatika)

Uraian Kode Program :

1. Deklarasi Package & Import

- *package Pekan3;* → menempatkan kelas dalam *package*.
- *import java.util.Scanner;* → untuk mengambil input dari keyboard.

2. Deklarasi Kelas

- *public class OperatorAritmatika {* → mendefinisikan kelas utama bernama *OperatorAritmatika*.

3. Metode Main

- *public static void main(String[] args)* → titik awal eksekusi program.

4. Deklarasi Variabel

- *int A1, A2, hasil;* → variabel untuk menyimpan input angka dan hasil operasi.

5. Input Data

Menggunakan *Scanner* untuk membaca angka pertama (A1) dan angka kedua (A2) dari *user*.

6. Operasi Aritmatika

Penjumlahan: $hasil = A1 + A2;$

Pengurangan: $hasil = A1 - A2;$

Perkalian: $hasil = A1 * A2;$

Pembagian: $hasil = A1 / A2;$

Sisa Bagi (*modulus*): $hasil = A1 \% A2;$

7. Output Hasil

Mencetak hasil dari setiap operasi aritmatika ke layar menggunakan *System.out.println()*.

2.1.2 Operator Assignment

```

OperatorArit... DeklarasiVa... ContohChar.java DataMaka
1 package Pekan3;
2 import java.util.Scanner;
3 public class OperatorAssignment {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int A1;
8         int A2;
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("Assignment penambahan ");
16        A1 += A2; // penambahan, sekarang A1 = 15
17        System.out.println("Penambahan : " + A1);
18        System.out.println("Assignment pengurangan ");
19        A1 -= A2; // pengurangan, sekarang A1 = 10
20        System.out.println("Pengurangan : " + A1);
21        System.out.println("Assignment perkalian ");
22        A1 *= A2; // perkalian, sekarang A1 = 50
23        System.out.println("Perkalian : " + A1);
24        System.out.println("Assignment hasil bagi ");
25        A1 /= A2; // hasil bagi, sekarang A1 = 10
26        System.out.println("Pembagian : " + A1);
27        System.out.println("Assignment sisa bagi ");
28        A1 %= A2; // sisa bagi, sekarang A1 = 0
29        System.out.println("Sisa Bagi : " + A1);
30    }
31
--

```

Kode Program 2.2 (Kode Program OperatorAssignment)

Uraian Kode Program :

1. Deklarasi Package & Import

- *package Pekan3;* → menempatkan kelas dalam *package*.
- *import java.util.Scanner;* → untuk mengambil input dari keyboard.

2. Deklarasi Kelas

- *public class OperatorAssignment {* → mendefinisikan kelas utama bernama *OperatorAssignment*

3. Metode Main

- `public static void main(String[] args)` → titik awal eksekusi program.

4. Deklarasi Variabel

- `int A1, A2;` → variabel untuk menyimpan input dari *user*.

5. Input Data

Menggunakan *Scanner* untuk membaca angka pertama (A1) dan angka kedua (A2) dari *user*.

6. Operasi Assignment

Program mendemonstrasikan penggunaan *operator assignment* (`+=`, `-=`, `*=`, `/=`, `%=`) yang menggabungkan operasi aritmatika dengan assignment.

`A1 += A2;` → penjumlahan dengan assignment.

`A1 -= A2;` → pengurangan dengan assignment.

`A1 *= A2;` → perkalian dengan assignment.

`A1 /= A2;` → pembagian dengan assignment.

`A1 %= A2;` → sisa bagi (*modulus*) dengan assignment.

Sisa Bagi (*modulus*): `hasil = A1 % A2;`

7. Output Hasil

Setiap operasi ditampilkan dengan `System.out.println()` agar user bisa melihat perubahan nilai A1 setelah setiap proses assignment.

2.1.3 Operator Relasional

```

1 package Pekan3;
2 import java.util.Scanner;
3 public class OperatorRelasional {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int A1;
8         int A2;
9         boolean hasil;
10        Scanner keyboard = new Scanner(System.in);
11        System.out.print("Input angka-1: ");
12        A1 = keyboard.nextInt();
13        System.out.print("Input angka-2: ");
14        A2 = keyboard.nextInt();
15        keyboard.close();
16        hasil = A1 > A2; // apakah A1 lebih besar A2?
17        System.out.println("A1 > A2 = " + hasil);
18        hasil = A1 < A2; // apakah A1 lebih kecil A2?
19        System.out.println("A1 < A2 = " + hasil);
20        hasil = A1 >= A2; // apakah A1 lebih besar samadengan A2?
21        System.out.println("A1 >= A2 = " + hasil);
22        hasil = A1 <= A2; // apakah A1 lebih kecil samadengan A2?
23        System.out.println("A1 <= A2 = " + hasil);
24        hasil = A1 == A2; // apakah A1 samadengan A2?
25        System.out.println("A1 == A2 = " + hasil);
26        hasil = A1 != A2; // apakah A1 tidak samadengan A2?
27        System.out.println("A1 != A2 = " + hasil);
28    }
29
30 }
31

```

Kode Program 2.3 (Kode Program Operator Relasional)

Uraian Kode Program :

1. Deklarasi Package & Import

- *package Pekan3;* → menempatkan kelas dalam *package*.
- *import java.util.Scanner;* → untuk mengambil input dari keyboard.

2. Deklarasi Kelas

- *public class OperatorRelasional {* → mendefinisikan kelas utama bernama *OperatorRelasional*.

3. Metode Main

- *public static void main(String[] args)* → titik awal eksekusi program.

4. Deklarasi Variabel

- *int A1, A2;* → variabel untuk menyimpan angka input dari user.
- *boolean hasil;* → variabel untuk menyimpan hasil perbandingan (*true/false*).

5. Input Data

Menggunakan *Scanner* untuk membaca angka pertama (A1) dan angka kedua (A2) dari *user*.

6. Operasi Relasional

Program ini mengecek hubungan antara dua angka menggunakan operator relasional:

$A1 > A2$ → apakah A1 lebih besar dari A2?

$A1 < A2$ → apakah A1 lebih kecil dari A2?

$A1 \geq A2$ → apakah A1 lebih besar atau sama dengan A2?

$A1 \leq A2$ → apakah A1 lebih kecil atau sama dengan A2?

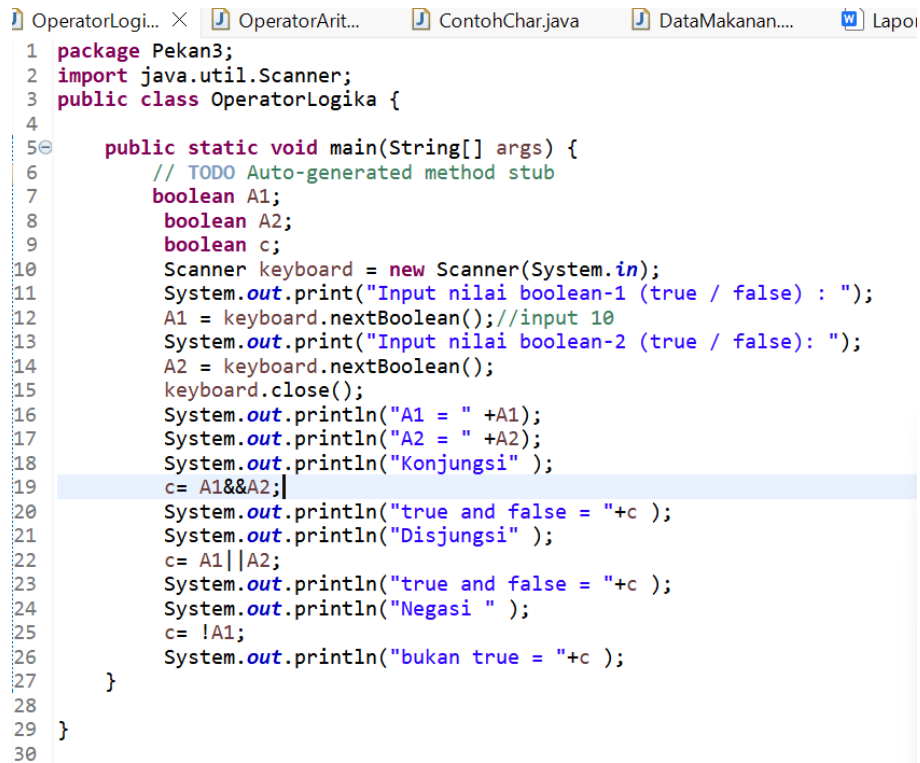
$A1 == A2$ → apakah A1 sama dengan A2?

$A1 != A2$ → apakah A1 tidak sama dengan A2?

7. Output Hasil

Setiap hasil perbandingan ditampilkan dengan *System.out.println()* dalam bentuk *true* atau *false*.

2.1.4 OperatorLogika



```

1 package Pekan3;
2 import java.util.Scanner;
3 public class OperatorLogika {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         boolean A1;
8         boolean A2;
9         boolean c;
10        Scanner keyboard = new Scanner(System.in);
11        System.out.print("Input nilai boolean-1 (true / false) : ");
12        A1 = keyboard.nextBoolean();//input 10
13        System.out.print("Input nilai boolean-2 (true / false): ");
14        A2 = keyboard.nextBoolean();
15        keyboard.close();
16        System.out.println("A1 = " +A1);
17        System.out.println("A2 = " +A2);
18        System.out.println("Konjungsi");
19        c= A1&A2;
20        System.out.println("true and false = "+c );
21        System.out.println("Disjungsi");
22        c= A1||A2;
23        System.out.println("true and false = "+c );
24        System.out.println("Negasi ");
25        c= !A1;
26        System.out.println("bukan true = "+c );
27    }
28
29 }
30

```

Kode Program 2.4 (Kode Program OperatorLogika)

Uraian Kode Program :

1. Deklarasi Package & Import

- *package Pekan3;* → menempatkan kelas dalam *package*.
- *import java.util.Scanner;* → untuk mengambil input dari keyboard.

2. Deklarasi Kelas

- *public class OperatorLogika {* → mendefinisikan kelas utama bernama *OperatorLogika*.

3. Metode Main

- *public static void main(String[] args)* → titik awal eksekusi program.

4. Deklarasi Variabel

- *boolean A1, A2, c;* \rightarrow variabel bertipe *boolean* (hanya bisa bernilai *true* atau *false*).

5. Input Data

- Menggunakan *Scanner* dengan *nextBoolean()* untuk membaca input dari *user* berupa *true* atau *false*.
- User diminta memasukkan dua nilai *boolean* (A1 dan A2).

6. Operator Logika yang Digunakan

Program ini mendemonstrasikan penggunaan operator logika pada tipe data *boolean*:

Konjungsi (AND) $\rightarrow c = A1 \ \&\& \ A2;$

Hasilnya *true* hanya jika kedua nilai *true*.

Disjungsi (OR) $\rightarrow c = A1 \ \|\ A2;$

Hasilnya *true* jika salah satu atau kedua nilai *true*.

Negasi (NOT) $\rightarrow c = !A1;$

Membalik nilai *boolean*, jika *A1 = true* maka hasil *false*, dan sebaliknya.

7. Output Hasil

- Setiap hasil operasi logika ditampilkan ke layar dengan *System.out.println()*.
- Program menunjukkan bagaimana nilai *boolean* berubah setelah diterapkan operator logika.

2.2 Langkah Kerja

2.2.1 OperatorAritmatika

1. Membuat kelas baru bernama *OperatorAritmatika* dalam *package* *Pekan3*.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:

Interfaces:

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Gambar 2.1 (Langkah 1 OperatorAritmatika)

2. Mendeklarasikan variabel *integer* A1, A2, dan hasil.

```

5- public static void main(String[] args) {
6     // TODO Auto-generated method stub
7     int A1;
8     int A2;
9     int hasil;

```

Gambar 2.2 (Langkah 2 OperatorAritmatika)

3. Menginput dua bilangan dari keyboard menggunakan *Scanner*.

```

10     Scanner keyboard = new Scanner(System.in);
11     System.out.print("Input angka-1: ");
12     A1 = keyboard.nextInt();
13     System.out.print("Input angka-2: ");
14     A2 = keyboard.nextInt();
15     keyboard.close();

```

Gambar 2.3 (Langkah 3 OperatorAritmatika)

4. Lakukan operasi aritmatika dasar: Penjumlahan ($A1 + A2$), Pengurangan ($A1 - A2$), Perkalian ($A1 * A2$), Pembagian ($A1 / A2$), *Modulus* / sisa bagi ($A1 \% A2$)

5. Tampilkan hasil setiap operasi dengan *System.out.println()*.

```

16     System.out.println("Operator Penjumlahan");
17     hasil = A1 + A2; // penjumlahan
18     System.out.println("Hasil = " + hasil);
19     System.out.println("Operator Pengurangan");
20     hasil = A1 - A2; // pengurangan
21     System.out.println("Hasil = " + hasil);
22     System.out.println("Operator Perkalian");
23     hasil = A1 * A2; // perkalian
24     System.out.println("Hasil = " + hasil);
25     System.out.println("Operator hasil bagi");
26     hasil = A1 / A2; // Pembagian
27     System.out.println("Hasil = " + hasil);
28     System.out.println("Operator sisa bagi");
29     hasil = A1 % A2; // Sisa Bagi
30     System.out.println("Hasil = " + hasil);
31 }

```

Gambar 2.4 (Langkah 4 dan 5 OperatorAritmatika)

2.2.2 OperatorAssignment

1. Membuat kelas baru bernama *OperatorAssignment* dalam *package Pekan3*.

The screenshot shows the 'New Class' dialog in Eclipse. The 'Source folder' is 'Prakalpro_2025_B_2511531016/src', the 'Package' is 'Pekan3', and the 'Name' is 'OperatorAssignment'. The 'Modifiers' section has 'public' selected. The 'Superclass' is 'java.lang.Object'. The 'Interfaces' section is empty. Under 'Which method stubs would you like to create?', 'public static void main(String[] args)' is checked. There are options for 'Constructors from superclass' and 'Inherited abstract methods', both unchecked. At the bottom, there is an option for 'Generate comments', which is also unchecked.

Gambar 2.5 (Langkah 1 OperatorAssignment)

2. Mendeklarasikan variabel *integer* A1 dan A2.

```

5 public static void main(String[] args) {
6     // TODO Auto-generated method stub
7     int A1;
8     int A2;

```

Gambar 2.6 (Langkah 2 OperatorAssignment)

3. Menginput dua bilangan dari keyboard menggunakan *Scanner*.

```

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();

```

Gambar 2.7 (Langkah 3 OperatorAssignment)

4. Melakukan operasi assignment:

$A1 += A2$ (penjumlahan dan assignment)

$A1 -= A2$ (pengurangan dan assignment)

$A1 *= A2$ (perkalian dan assignment)

$A1 /= A2$ (pembagian dan assignment)

$A1 \% = A2$ (sisa bagi dan assignment)

5. Menampilkan nilai A1 setelah setiap operasi ke layar.

```

15      System.out.println("Assignment penambahan ");
16      A1 += A2; // penambahan, sekarang A1 = 15
17      System.out.println("Penambahan : " + A1);
18      System.out.println("Assignment pengurangan ");
19      A1 -= A2; // pengurangan, sekarang A1 = 10
20      System.out.println("Pengurangan : " + A1);
21      System.out.println("Assignment perkalian ");
22      A1 *= A2; // perkalian, sekarang A1 = 50
23      System.out.println("Perkalian : " + A1);
24      System.out.println("Assignment hasil bagi ");
25      A1 /= A2; // hasil bagi, sekarang A1 = 10
26      System.out.println("Pembagian : " + A1);
27      System.out.println("Assignment sisa bagi ");
28      A1 %= A2; // sisa bagi, sekarang A1 = 0
29      System.out.println("Sisa Bagi : " + A1);
30  }
```

Gambar 2.8 (Langkah 4 dan 5 Operator Assignment)

2.2.3 Operator Relasional

1. Membuat kelas baru bernama *OperatorRelasional* dalam package *Pekan3*.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:

Interfaces:

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Gambar 2.9 (Langkah 1 OperatorRelasional)

2. Mendeklarasikan variabel *integer* A1, A2, dan variabel *boolean* hasil.

```

5 public static void main(String[] args) {
6     // TODO Auto-generated method stub
7     int A1;
8     int A2;
9     boolean hasil;

```

Gambar 2.10 (Langkah 2 OperatorRelasional)

3. Menginput dua bilangan dari keyboard menggunakan *Scanner*.

```

10 Scanner keyboard = new Scanner(System.in);
11 System.out.print("Input angka-1: ");
12 A1 = keyboard.nextInt();
13 System.out.print("Input angka-2: ");
14 A2 = keyboard.nextInt();
15 keyboard.close();

```

Gambar 2.11 (Langkah 3 OperatorRelasional)

4. Lakukan operasi relasional:

$A1 > A2$ (lebih besar)

$A1 < A2$ (lebih kecil)

$A1 \geq A2$ (lebih besar atau sama dengan)

$A1 \leq A2$ (lebih kecil atau sama dengan)

$A1 == A2$ (sama dengan)

$A1 != A2$ (tidak sama dengan)

5. Menampilkan hasil perbandingan (*true/false*) ke layar.

```

16      hasil = A1 > A2; // apakah A1 lebih besar A2?
17      System.out.println("A1 > A2 = " + hasil);
18      hasil = A1 < A2; // apakah A1 lebih kecil A2?
19      System.out.println("A1 < A2 = " + hasil);
20      hasil = A1 >= A2; // apakah A1 lebih besar samadengan A2?
21      System.out.println("A1 >= A2 = " + hasil);
22      hasil = A1 <= A2; // apakah A1 lebih kecil samadengan A2?
23      System.out.println("A1 <= A2 = " + hasil);
24      hasil = A1 == A2; // apakah A1 samadengan A2?
25      System.out.println("A1 == A2 = " + hasil);
26      hasil = A1 != A2; // apakah A1 tidak samadengan A2?
27      System.out.println("A1 != A2 = " + hasil);
28  }
29
30 }
```

Gambar 2.12 (Langkah 4 dan 5 Operator Relasional)

2.2.4 Operator Logika

1. Membuat kelas baru bernama *OperatorLogika* dalam package *Pekan3*.

The screenshot shows the 'New Class' dialog in Eclipse. The 'Source folder' is 'Prakalpro_2025_B_2511531016/src'. The 'Package' is 'Pekan3'. The 'Name' is 'OperatorLogika'. The 'Modifiers' are set to 'public' and 'none'. The 'Superclass' is 'java.lang.Object'. The 'Interfaces' section is empty. Under 'Which method stubs would you like to create?', the checkbox for 'public static void main(String[] args)' is checked. The 'Do you want to add comments?' section has the 'Generate comments' checkbox unchecked.

Gambar 2.13 (Langkah 1 OperatorLogika)

2. Mendeklarasikan variabel *boolean* A1, A2, dan c.

```

5 public static void main(String[] args) {
6     // TODO Auto-generated method stub
7     boolean A1;
8     boolean A2;
9     boolean c;

```

Gambar 2.14 (Langkah 2 OperatorLogika)

3. Menginput dua nilai boolean (*true* atau *false*) dari keyboard menggunakan *Scanner*.

```

10 Scanner keyboard = new Scanner(System.in);
11 System.out.print("Input nilai boolean-1 (true / false) : ");
12 A1 = keyboard.nextBoolean();//input 10
13 System.out.print("Input nilai boolean-2 (true / false): ");
14 A2 = keyboard.nextBoolean();
15 keyboard.close();

```

Gambar 2.15 (Langkah 3 OperatorLogika)

4. Lakukan operasi logika:

$c = A1 \ \&\& \ A2$ (AND / konjungsi \rightarrow true jika keduanya true).

$c = A1 \ || \ A2$ (OR / disjungsi \rightarrow true jika salah satu true).

$c = !A1$ (NOT / negasi \rightarrow membalik nilai A1).

5. Cetak hasil dari setiap operasi logika dengan *System.out.println()*.

```

16      System.out.println("A1 = " +A1);
17      System.out.println("A2 = " +A2);
18      System.out.println("Konjungsi" );
19      c= A1&&A2;
20      System.out.println("true and false = "+c );
21      System.out.println("Disjungsi" );
22      c= A1||A2;
23      System.out.println("true and false = "+c );
24      System.out.println("Negasi " );
25      c= !A1;
26      System.out.println("bukan true = "+c );
27  }
28
29 }
```

Gambar 2.16 (Langkah 4 dan 5 OperatorLogika)

2.3 Analisis Hasil

Program **OperatorAritmatika** menunjukkan bagaimana operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan *modulus* bekerja pada tipe data numerik di Java. Misalnya, jika $A1 = 10$ dan $A2 = 3$, maka hasil yang diperoleh adalah penjumlahan 13, pengurangan 7, perkalian 30, pembagian 3 (karena *integer division* hanya menghasilkan bilangan bulat), dan *modulus* 1. Hal ini sesuai dengan teori bahwa operator aritmatika (+, -, *, /, %) digunakan untuk operasi perhitungan dasar dalam Java, sebagaimana dijelaskan dalam *Oracle Java Documentation*.

Pada **OperatorAssignment**, program memperlihatkan bagaimana operasi aritmatika dapat langsung digabungkan dengan assignment. Contohnya jika $A1 = 10$ dan $A2 = 5$, maka setelah operasi $A1 += A2$, nilai A1 menjadi 15, kemudian

A1 -= A2 kembali menjadi 10, A1 *= A2 menjadi 50, A1 /= A2 menjadi 10, dan A1 %= A2 menghasilkan 0. Hal ini mendukung teori bahwa operator assignment (+=, -=, *=, /=, %=) merupakan bentuk singkat dari operasi aritmatika sekaligus penyimpanan hasil ke variabel yang sama. Teori ini dijelaskan dalam *Java Tutorials* bahwa operator assignment membantu efisiensi penulisan kode dan mempersingkat ekspresi program.

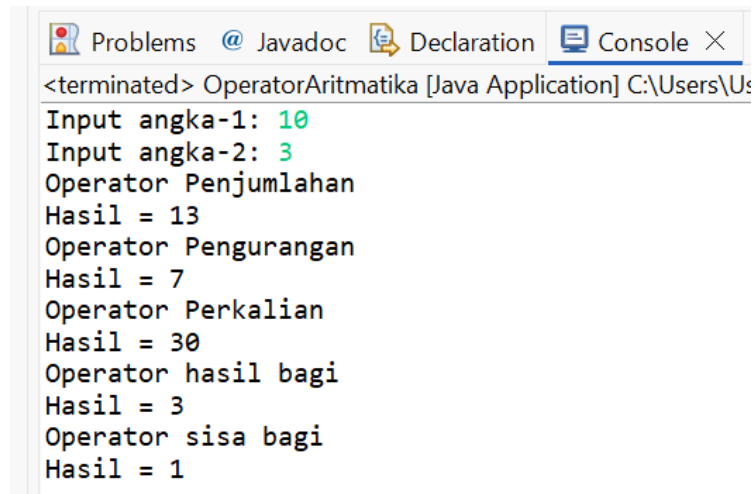
Sementara itu, **Operator Relasional** memberikan hasil berupa nilai *boolean* (*true* atau *false*) berdasarkan perbandingan dua bilangan. Jika A1 = 7 dan A2 = 5, maka ekspresi A1 > A2 bernilai *true*, A1 < A2 bernilai *false*, A1 >= A2 bernilai *true*, A1 <= A2 bernilai *false*, A1 == A2 bernilai *false*, dan A1 != A2 bernilai *true*. Hasil ini sesuai dengan teori bahwa operator relasional (>, <, >=, <=, ==, !=) digunakan untuk membandingkan dua nilai dan selalu menghasilkan tipe data *boolean*. Menurut *Java Language Specification*, operator relasional berperan penting dalam pengambilan keputusan (*if*, *switch*) maupun pengulangan (*while*, *for*).

Terakhir, **Operator Logika** memperlihatkan bagaimana nilai *boolean* dapat diproses menggunakan operator logika. Jika input A1 = *true* dan A2 = *false*, maka A1 && A2 menghasilkan *false* karena kedua nilai tidak sama-sama *true*, A1 || A2 menghasilkan *true* karena salah satu bernilai *true*, dan !A1 menghasilkan *false* karena negasi membalik nilai aslinya. Hal ini mendukung teori bahwa operator logika (&&, ||, !) digunakan untuk menggabungkan atau memanipulasi ekspresi *boolean*. Menurut *Oracle Java Documentation*, operator logika sering dipakai dalam kontrol alur program untuk menentukan kondisi yang lebih kompleks.

BAB III KESIMPULAN

3.1 Hasil Praktikum

3.1.1 OperatorAritmatika



```
<terminated> OperatorAritmatika [Java Application] C:\Users\Us
Input angka-1: 10
Input angka-2: 3
Operator Penjumlahan
Hasil = 13
Operator Pengurangan
Hasil = 7
Operator Perkalian
Hasil = 30
Operator hasil bagi
Hasil = 3
Operator sisa bagi
Hasil = 1
```

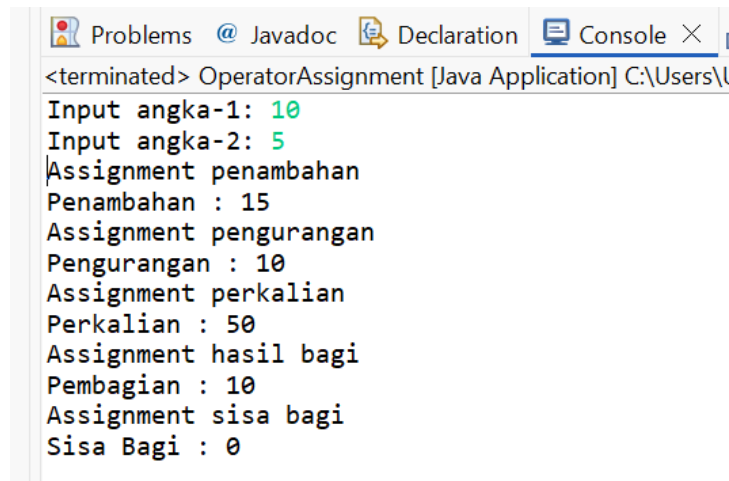
Gambar 3.1 (Hasil Praktikum OperatorAritmatika)

Program pada gambar di atas adalah program Java yang menampilkan operasi aritmatika dasar menggunakan dua input angka, yaitu 10 dan 3. Program ini melakukan beberapa operasi sebagai berikut:

- Penjumlahan ($10 + 3$) menghasilkan 13.
- Pengurangan ($10 - 3$) menghasilkan 7.
- Perkalian (10×3) menghasilkan 30.
- Pembagian ($10 \div 3$) menghasilkan 3 (karena tipe data yang digunakan kemungkinan *integer*, sehingga hanya menampilkan hasil bagi bulat tanpa desimal).
- Sisa bagi ($10 \% 3$) menghasilkan 1, yaitu sisa dari pembagian 10 dibagi 3.

Hasil program menunjukkan penerapan operator aritmatika di Java, yaitu +, -, *, /, dan %, yang digunakan untuk melakukan operasi matematika dasar terhadap dua bilangan.

3.1.2 OperatorAssignment



```
<terminated> OperatorAssignment [Java Application] C:\Users\l
Input angka-1: 10
Input angka-2: 5
Assignment penambahan
Penambahan : 15
Assignment pengurangan
Pengurangan : 10
Assignment perkalian
Perkalian : 50
Assignment hasil bagi
Pembagian : 10
Assignment sisa bagi
Sisa Bagi : 0
```

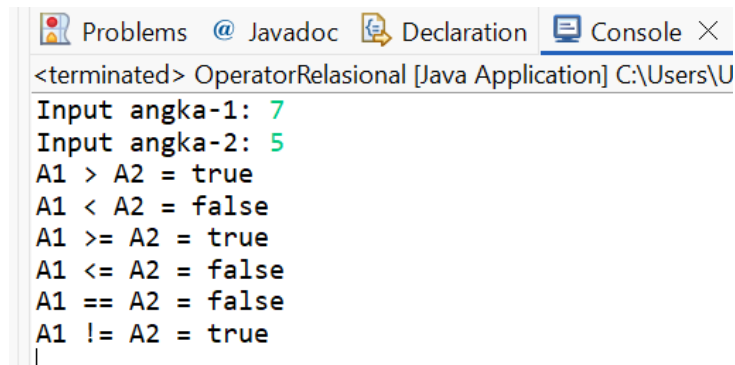
Gambar 3.2 (Hasil Praktikum OperatorAssignment)

Program pada gambar di atas adalah program Java yang menampilkan penggunaan operator assignment dengan dua input angka, yaitu 10 dan 5. Operator assignment digunakan untuk melakukan operasi aritmatika sekaligus menyimpan hasilnya pada variabel. Berikut penjelasan hasilnya:

- Assignment penambahan (+=) $\rightarrow 10 + 5 = 15$
- Assignment pengurangan (-=) $\rightarrow 15 - 5 = 10$
- Assignment perkalian (*=) $\rightarrow 10 * 5 = 50$
- Assignment pembagian (/=) $\rightarrow 50 / 5 = 10$ (pembagian bilangan bulat)
- Assignment sisa bagi (%=) $\rightarrow 10 \% 5 = 0$ (sisa pembagian habis)

Program ini menunjukkan bagaimana operator assignment bekerja, yaitu menggabungkan operasi aritmatika dengan proses penyimpanan nilai baru pada variabel, sehingga kode menjadi lebih ringkas dan efisien.

3.1.3 Operator Relasional



```

Problems @ Javadoc Declaration Console X
<terminated> OperatorRelasional [Java Application] C:\Users\U
Input angka-1: 7
Input angka-2: 5
A1 > A2 = true
A1 < A2 = false
A1 >= A2 = true
A1 <= A2 = false
A1 == A2 = false
A1 != A2 = true

```

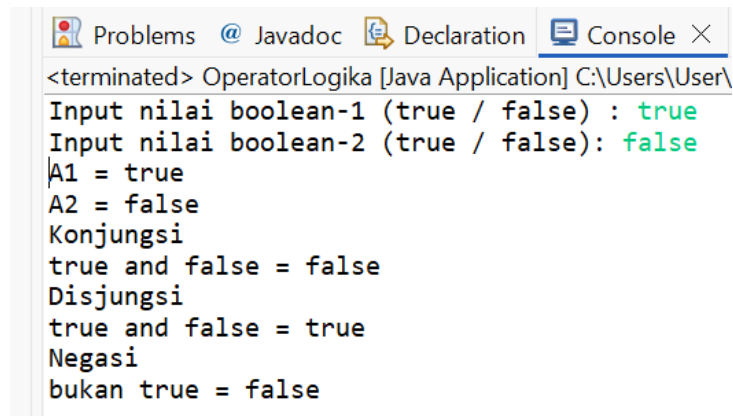
Gambar 3.3 (Hasil Praktikum Operator Relasional)

Program pada gambar di atas adalah program Java yang menampilkan penggunaan operator relasional dengan dua input angka, yaitu 7 dan 5. Operator relasional digunakan untuk membandingkan dua nilai dan menghasilkan output berupa true atau false. Berikut penjelasan hasilnya:

- $A1 > A2 \rightarrow 7 > 5 \rightarrow \text{true}$
- $A1 < A2 \rightarrow 7 < 5 \rightarrow \text{false}$
- $A1 \geq A2 \rightarrow 7 \geq 5 \rightarrow \text{true}$
- $A1 \leq A2 \rightarrow 7 \leq 5 \rightarrow \text{false}$
- $A1 == A2 \rightarrow 7 == 5 \rightarrow \text{false}$
- $A1 != A2 \rightarrow 7 != 5 \rightarrow \text{true}$

Program ini menunjukkan bagaimana operator relasional bekerja dalam Java untuk menentukan hubungan antara dua bilangan, apakah lebih besar, lebih kecil, sama dengan, atau tidak sama dengan.

3.1.4 OperatorLogika



```

Problems @ Javadoc Declaration Console X
<terminated> OperatorLogika [Java Application] C:\Users\User\
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 and false = true
Negasi
bukan true = false

```

Gambar 3.4 (Hasil Praktikum OperatorLogika)

Program pada gambar di atas adalah program Java yang menampilkan penggunaan operator logika dengan dua input *boolean*, yaitu true dan false. Operator logika digunakan untuk menggabungkan atau memanipulasi ekspresi *boolean* dan menghasilkan output berupa true atau false. Berikut penjelasan hasilnya:

- Konjungsi (AND)

$A1 \ \&\& \ A2 \rightarrow \text{true} \ \&\& \ \text{false} \rightarrow \text{false}$

- Disjungsi (OR)

$A1 \ || \ A2 \rightarrow \text{true} \ || \ \text{false} \rightarrow \text{true}$

- Negasi (NOT)

$!A1 \rightarrow !\text{true} \rightarrow \text{false}$

Program ini menunjukkan bagaimana operator logika bekerja dalam Java. Operator AND menghasilkan true hanya jika kedua operand bernilai true, operator OR menghasilkan true jika salah satu operand bernilai true, dan operator NOT digunakan untuk membalikkan nilai boolean.

3.2 Saran Pengembangan

Saran pengembangan dari keempat kode program yang telah dibuat dapat diarahkan pada perluasan fungsi serta penerapan dalam konteks yang lebih nyata sehingga tidak hanya berfokus pada teori dasar, tetapi juga memberikan pengalaman pemrograman yang lebih mendalam.

Pada program **operator aritmatika**, pengembangan dapat dilakukan dengan menggunakan berbagai tipe data seperti *float* dan *double* untuk menunjukkan perbedaan hasil antara bilangan bulat dan bilangan desimal. Selain itu, dapat ditambahkan juga contoh kasus yang lebih aplikatif, misalnya menghitung rata-rata nilai mahasiswa atau perhitungan diskon, sehingga pemanfaatan operator aritmatika lebih terasa kegunaannya dalam kehidupan sehari-hari.

Pada program **operator assignment**, pengembangan bisa dilakukan dengan mengombinasikan penggunaan assignment dengan struktur perulangan seperti *for* atau *while*, sehingga mahasiswa dapat memahami efisiensi penggunaan operator ini pada pengolahan data dalam jumlah besar.

Selanjutnya, pada program **operator relasional**, pengembangannya dapat diarahkan dengan mengintegrasikannya ke dalam logika percabangan *if-else* atau *switch-case*. Hal ini akan membantu menunjukkan bagaimana operator relasional bekerja dalam proses pengambilan keputusan. Contoh aplikasinya dapat berupa sistem penilaian mahasiswa (apakah lulus atau tidak berdasarkan nilai).

Terakhir, pada program **operator logika**, pengembangan dapat dilakukan dengan memperluas contoh kondisi yang lebih kompleks, seperti pembuatan simulasi login sederhana dengan kombinasi *username* dan *password*. Dengan demikian, mahasiswa dapat melihat bagaimana operator logika berperan dalam menggabungkan banyak kondisi *boolean* untuk menghasilkan keputusan yang tepat.

Secara keseluruhan, pengembangan keempat program ini bertujuan agar mahasiswa tidak hanya memahami fungsi dasar operator dalam Java, tetapi juga mampu mengaplikasikannya pada kasus nyata. Selain itu, pengembangan ini juga melatih keterampilan berpikir logis dan sistematis dalam menyusun program, serta memberikan wawasan bahwa konsep dasar seperti operator adalah fondasi penting bagi pemrograman yang lebih kompleks di kemudian hari.

DAFTAR PUSTAKA

- [1] Oracle, “Java Operators,” *The Java™ Tutorials: Learning the Java Language*, Oracle, 2023. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/opsummary.html>. [Accessed: Sep. 23, 2025].
- [2] Oracle, “Expressions, Statements, and Blocks,” *The Java™ Tutorials: Learning the Java Language*, Oracle, 2023. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/expressions.html>. [Accessed: Sep. 23, 2025].
- [3] Oracle, “The Java® Language Specification – Operators,” *Oracle Java SE Documentation*, Oracle, 2023. [Online]. Available: <https://docs.oracle.com/javase/specs/>. [Accessed: Sep. 23, 2025].
- [4] Oracle, “Conditional and Logical Operators,” *The Java™ Tutorials: Learning the Java Language*, Oracle, 2023. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op2.html>. [Accessed: Sep. 23, 2025].