

Nama & NPM :	Topik :	Tanggal :
Donna Zelvya Gizella Alvera Deadora G1F024018	Operator Java	12 September 2024

1. Operator Aritmatika

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Contoh 1: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
public class OperatorAritmatika{
    public static void main(String[] args) {
        // deklarasi nilai
        int a = 20, b = 3;
        //operator aritmatika
        System.out.println("a: " +a);
        System.out.println("b: " +b);
        System.out.println("a + b = " (a + b)); //menampilkan hasil penjumlahan
    } }
```

Luaran:

Exception in thread "main" java.lang.Error: Unresolved compilation problems: Syntax error on token ""a + b = "" , AssignmentOperator expected after this token The left-hand side of an assignment must be a variable.

Penjelasan :

Diketahui dari soal diatas, tipe data yang digunakan pada variabel a dan b yaitu tipe data int, karena kedua variabel tersebut menyatakan nilai berupa bilangan bulat positif. Permasalahan yang menyebabkan program tidak dapat menghasilkan output yang sesuai adalah karena ada operator yang kurang. Operator yang kurang yaitu operator + pada kode program `System.out.println("a + b = " (a + b));` yang digunakan untuk menghubungkan atau memanggil nilai.

- 2) Rincikan sumber informasi yang relevan (buku / webpage)

<https://www.youtube.com/watch?v=PzCMZOexZM>

Latihan 1

1.1. Rekomendasikan perbaikan kode agar program Contoh 1 dapat berjalan!

Penjelasan :

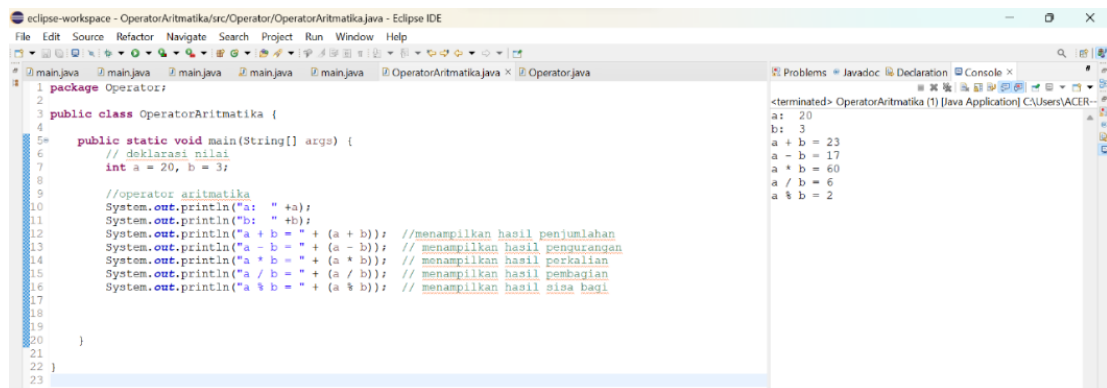
Perbaikan yang dapat dilakukan terhadap kode program yang ada agar program dapat berjalan yaitu dengan menambahkan operator + pada kode program `System.out.println("a + b = " (a + b));` , sehingga ketika sudah diperbaiki, kode program menjadi `System.out.println("a + b = " + (a + b));` , maka program akan menampilkan atau menghasilkan output yang tepat dan sesuai. Operator tersebut digunakan untuk menggabungkan teks dan hasil perhitungan. Tambahkan juga baris untuk menampilkan perhitungan dengan operator sesuai dengan yang diperintahkan, yaitu meliputi pengurangan (-), perkalian (*), pembagian (/), dan sisa bagi (%).

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan operator + pada perintah `System.out.println`. Operator + digunakan untuk menyatukan string dan hasil perhitungan. Selain itu, menambahkan baris untuk menampilkan perhitungan dengan operator lain yaitu meliputi operator -, *, /, dan %.
- 2) Alasan solusi ini karena dengan menambahkan operator + yang merupakan penghubung dari elemen-elemen yang ada di input, maka program akan berjalan lancar.
- 3) Perbaiki kode program dengan cara menambahkan operator + pada perintah `System.out.println` untuk menggabungkan string dengan hasil perhitungan.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Membuat kelas dan method
 - (c) Deklarasikan variabel
 - (d) Tambahkan operasi aritmatika berupa penjumlahan, pengurangan, perkalian, pembagian, dan sisa bagi.
 - (e) Menampilkan hasilnya.
 - (f) Selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran



The screenshot shows the Eclipse IDE with a Java project named 'OperatorAritmatika'. The source code in 'OperatorAritmatika.java' defines a class with a static method 'main' that performs arithmetic operations on variables 'a' (20) and 'b' (3). The code uses `System.out.println` to display the results of addition, subtraction, multiplication, division, and modulus. The console on the right shows the output of the program, which matches the expected results of these operations.

```
package Operator;

public class OperatorAritmatika {

    public static void main(String[] args) {
        // deklarasi nilai
        int a = 20, b = 3;

        //operator aritmatika
        System.out.println("a: " + a);
        System.out.println("b: " + b);
        System.out.println("a + b = " + (a + b)); //menampilkan hasil penjumlahan
        System.out.println("a - b = " + (a - b)); // menampilkan hasil pengurangan
        System.out.println("a * b = " + (a * b)); // menampilkan hasil perkalian
        System.out.println("a / b = " + (a / b)); // menampilkan hasil pembagian
        System.out.println("a % b = " + (a % b)); // menampilkan hasil sisa bagi
    }
}
```

Output in Console:

```
<terminated> OperatorAritmatika (1) [Java Application] C:\Users\ACER-
a: 20
b: 3
a + b = 23
a - b = 17
a * b = 60
a / b = 6
a % b = 2
```

- b) Analisa luaran yang dihasilkan

Pada program diatas, operasi yang dilakukan yaitu beberapa operasi dasar aritmatika yang meliputi penjumlahan, pengurangan, perkalian, pembagian, dan sisa bagi. Luaran yang dihasilkan program tersebut sudah tepat dan sesuai.

[No.1] Kesimpulan

- 1) Analisa

Hal yang menyebabkan program ini error dan tidak dapat berjalan yaitu karena kurang tanda operator berupa + pada perintah `System.out.println("a + b = " (a + b));`; yang menjadi sebuah penghubung. Setelah melakukan perbaikan kode program `System.out.println("a + b = " + (a + b));`, program dapat berjalan lancar.

2. Operator Penugasan

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Contoh 2: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
public class OperatorPenugasan {  
    public static void main(String[] args) {  
        // deklarasi nilai  
        int a = 20, b = 3;  
        //operator penugasan  
        b += a; //melakukan perhitungan penjumlahan  
        System.out.println("Penambahan : " + b); // menampilkan hasil  
        perhitungan penjumlahan  
    }  
}
```

Luaran:

Penambahan : 23

Penjelasan :

Pada contoh diatas, Langkah yang harus dilakukan yaitu menambah baris untuk menampilkan perhitungan dengan operator (-=, *=, /=, %=). Variabel yang digunakan pada program tersebut yaitu variabel int yang digunakan untuk menyimpan nilai berupa bilangan bulat positif.

- 2) Rincikan sumber informasi yang relevan (buku / webpage)

<https://www.youtube.com/watch?v=PzCMZObexZM>

Latihan 2

2.1. Tambahkan baris Contoh 2 untuk menampilkan perhitungan dengan operator (-=, *=, /=, %=)!

2.2. Berikan argumentasi tentang perbedaan luaran dan waktu eksekusi Contoh 1 dan Contoh 2!

Penjelasan :

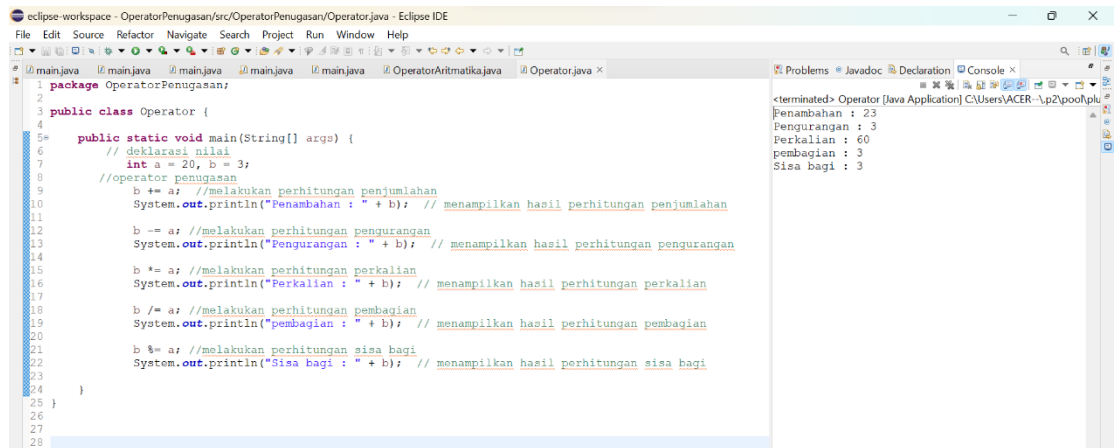
Pada contoh 1, operator yang digunakan yaitu operator aritmatika yang digunakan untuk melakukan perhitungan. Luaran dari operator aritmatika berupa hasil dari operasi yang dilakukan, mulai dari penjumlahan, pengurangan, perkalian, pembagian, dan sisa bagi. Sebagai contoh pada kode program **int a = 20, b = 3;** lalu dicetak **System.out.println("a + b = " + (a + b));** , maka luaran yang dihasilkan yaitu hasil dari penjumlahan a+b. Dan untuk operasi berikutnya, tetap menggunakan nilai dari deklarasi nilai **int a = 20, b = 3;**. Pada contoh 2, operator yang digunakan yaitu operator penugasan yang memberikan nilai variabel dan bergabung dengan operator aritmatika. Operator penugasan ini berfungsi untuk memperbarui nilai variabel. Luaran dari operator aritmatika adalah nilai baru dari variabel yang akan dimasukkan ke operasi selanjutnya.

[No. 2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan baris baru untuk menampilkan perhitungan dengan operator (-=, *=, /=, %=) sesuai dengan perintah pada soal.
- 2) Alasan solusi ini karena dengan menambahkan operator yang diminta dan diperlukan, maka output yang dihasilkan akan sesuai dengan yang diinginkan.

[No. 2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Membuat kelas dan method
 - (c) Deklarasikan variabel
 - (d) Menambah baris untuk menampilkan perhitungan dengan operator ($+=$, $-=$, $*=$, $/=$, $%=$)
 - (e) Menampilkan hasilnya.
 - (f) Selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran



The screenshot shows the Eclipse IDE with a Java project named 'OperatorPenugasan'. The main class 'Operator' contains a static void main method. The code declares two integer variables, 'a' and 'b', with initial values of 20 and 3 respectively. It then performs five arithmetic operations: addition (+=), subtraction (-=), multiplication (*=), division (/=), and modulus (%=). Each operation is followed by a System.out.println statement that displays the operation and the result. The console output on the right shows the results of these operations: Penambahan : 23, Pengurangan : 3, Perkalian : 60, Pembagian : 3, and Sisa bagi : 3.

```
1 package OperatorPenugasan;
2
3 public class Operator {
4
5     public static void main(String[] args) {
6         // deklarasi nilai
7         int a = 20, b = 3;
8         //operator penugasan
9         b += a; //melakukan perhitungan penjumlahan
10        System.out.println("Penambahan : " + b); // menampilkan hasil perhitungan penjumlahan
11
12        b -= a; //melakukan perhitungan pengurangan
13        System.out.println("Pengurangan : " + b); // menampilkan hasil perhitungan pengurangan
14
15        b *= a; //melakukan perhitungan perkalian
16        System.out.println("Perkalian : " + b); // menampilkan hasil perhitungan perkalian
17
18        b /= a; //melakukan perhitungan pembagian
19        System.out.println("Pembagian : " + b); // menampilkan hasil perhitungan pembagian
20
21        b %= a; //melakukan perhitungan sisa bagi
22        System.out.println("Sisa bagi : " + b); // menampilkan hasil perhitungan sisa bagi
23    }
24 }
25
26
27
28
```

Console Output:

```
<terminated> Operator [Java Application] C:\Users\ACER~1\p2\poo\p...
Penambahan : 23
Pengurangan : 3
Perkalian : 60
Pembagian : 3
Sisa bagi : 3
```

- b) Luaran yang dihasilkan

Pada operator pemugasan diatas, menampilkan perhitungan dengan beberapa operator yaitu ($+=$, $-=$, $*=$, $/=$, $%=$) dan luaran yang dihasilkan sudah sesuai.

[No. 2] Kesimpulan

1) Analisa

Pada operator penugasan, nilai yang digunakan untuk melakukan operasi baik operasi penjumlahan, pengurangan, perkalian, pembagian, dan sisa bagi adalah nilai baru dari variabel atau hasil dari operasi pertama. Sebagai contoh pada operator penjumlahan ($+=$), operator ini menambahkan nilai variabel a dan variabel b lalu hasilnya disimpan di variabel b. Untuk operator selanjutnya, nilai baru dari b yang dihasilkan dari operasi sebelumnya akan digunakan dalam operasi selanjutnya. Begitu pula untuk operator-operator lainnya, nilai yang dimasukkan yaitu nilai baru dari operasi sebelumnya.

3. Operator Relasional

[No. 3] Identifikasi Masalah

- 1) Uraikan permasalahan dan variabel

Contoh 3: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
public class OperatorRelasional {  
    public static void main(String[] args) {  
        int nilaiA = 12;  
        int nilaiB = 4;  
        boolean hasil;  
        System.out.println(" A = " + nilaiA + "\n B = " + nilaiB);  
        // apakah A lebih besar dari B?  
        hasil = nilaiA > nilaiB;  
        System.out.println("\n Hasil A > B = " + hasil);  
        // apakah A lebih kecil dari B?  
        hasil = nilaiA < nilaiB;  
        System.out.println("\n Hasil A < B = " + hasil);  
        // apakah A lebih besar samadengan B?  
        hasil = nilaiA >= nilaiB;  
        System.out.println("\n Hasil A >= B = " + hasil);  
        // apakah A lebih kecil samadengan B?  
        hasil = nilaiA <= nilaiB;  
        System.out.println("\n Hasil A <= B = " + hasil);  
        // apakah nilai A sama dengan B?  
        hasil = nilaiA == nilaiB;  
        System.out.println("\n Hasil A == B = " + hasil);  
        // apakah nilai A tidak samadengan B?  
        hasil = nilaiA != nilaiB;  
        System.out.println("\n Hasil A != B = " + hasil);  
    }  
}
```

Luaran :

A = 12

B = 4

Hasil A > B = true

Hasil A < B = false

Hasil A >= B = true

Hasil A <= B = false

Hasil A == B = false

Hasil A != B = true

- 2) Rincikan sumber informasi yang relevan

<https://www.youtube.com/watch?v=PzCMZOexZM>

Latihan 3

- 3.1. Ubahlah nilai A = 4 dan B = 4 pada Contoh 3. Simpulkan perubahan yang terjadi!

Penjelasan :

Setelah nilai A dan B diubah menjadi 4, perubahan yang terjadi yaitu :

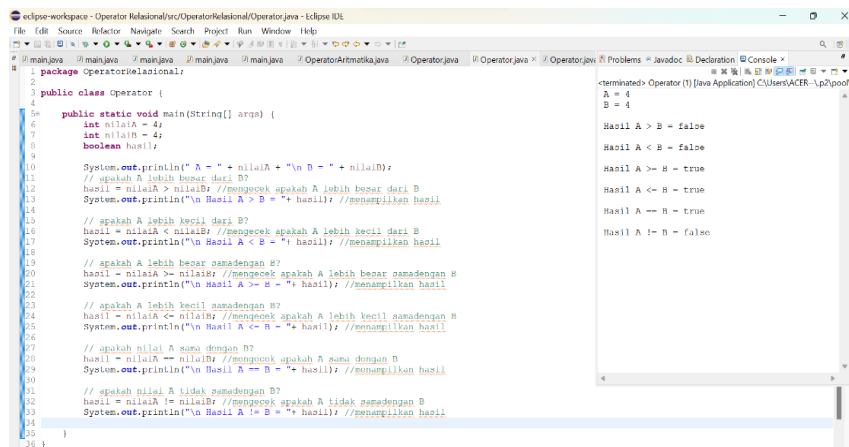
1. Hasil $A > B = \text{false}$, karena nilai A tidak lebih besar dari B tetapi sama dengan.
2. Hasil $A < B = \text{false}$ karena nilai A tidak kurang dari B tetapi sama dengan.
3. Hasil $A \geq B = \text{true}$, karena A sama dengan atau lebih besar dari B.
4. Hasil $A \leq B = \text{true}$, karena karena nilai A sama dengan atau lebih kecil dari B.
5. Hasil $A == B = \text{true}$, karena nilai A sama dengan B.
6. $A != B = \text{false}$, karena nilai A sama dengan B.

[No. 3] Analisis dan Argumentasi

Operator relasional digunakan untuk membandingkan dua variabel. Pada kode program tersebut, perubahan terhadap nilai A dan nilai B akan mempengaruhi hasil luaran dan hubungan antara dua variabel tersebut.

[No. 3] Penyusunan algoritma dan kode program

- 1) Algoritma
 - (a) Mulai
 - (b) Membuat kelas dan method
 - (c) Deklarasikan variabel
 - (d) Mengecek apakah nilai A lebih besar, lebih kecil, lebih besar samadengan, lebih kecil samadengan, sama dengan, dan tidak samadengan dari nilai B.
 - (e) Menampilkan hasilnya.
 - (f) Selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran



The screenshot shows the Eclipse IDE with a Java file named 'Operator.java'. The code defines a class 'Operator' with a static method 'main' that takes an array of strings as input. Inside the 'main' method, two integer variables 'nilaiA' and 'nilaiB' are both assigned the value 4. A series of conditional statements using relational operators (>, <, >=, <=, ==, !=) are used to compare 'nilaiA' and 'nilaiB'. The results of these comparisons are stored in a 'boolean hasil' variable and printed to the console using 'System.out.println'. The output window on the right shows the results: 'A = 4', 'B = 4', 'Hasil A > B = false', 'Hasil A < B = false', 'Hasil A >= B = true', 'Hasil A <= B = true', 'Hasil A == B = true', and 'Hasil A != B = false'.

```
1 package OperatorRelasional;
2
3 public class Operator {
4
5     public static void main(String[] args) {
6         int nilaiA = 4;
7         int nilaiB = 4;
8         boolean hasil;
9
10        System.out.println("A = " + nilaiA + " \n B = " + nilaiB);
11        // apakah A lebih besar dari B?
12        hasil = nilaiA > nilaiB; //mengecek apakah A lebih besar dari B
13        System.out.println("Hasil A > B = " + hasil); //menampilkan hasil
14
15        // apakah A lebih kecil dari B?
16        hasil = nilaiA < nilaiB; //mengecek apakah A lebih kecil dari B
17        System.out.println("Hasil A < B = " + hasil); //menampilkan hasil
18
19        // apakah A lebih besar samadengan B?
20        hasil = nilaiA >= nilaiB; //mengecek apakah A lebih besar samadengan B
21        System.out.println("Hasil A >= B = " + hasil); //menampilkan hasil
22
23        // apakah A lebih kecil samadengan B?
24        hasil = nilaiA <= nilaiB; //mengecek apakah A lebih kecil samadengan B
25        System.out.println("Hasil A <= B = " + hasil); //menampilkan hasil
26
27        // apakah nilai A sama dengan B?
28        hasil = nilaiA == nilaiB; //mengecek apakah A sama dengan B
29        System.out.println("Hasil A == B = " + hasil); //menampilkan hasil
30
31        // apakah nilai A tidak samadengan B?
32        hasil = nilaiA != nilaiB; //mengecek apakah A tidak samadengan B
33        System.out.println("Hasil A != B = " + hasil); //menampilkan hasil
34
35    }
36 }
```

- b) Luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai yaitu hasil antara hubungan nilai A dan B.

[No. 3] Kesimpulan

- 1) Analisa

Perubahan nilai variabel A dan B secara langsung mempengaruhi hasil dari operator relasional yang digunakan dalam program.

4. Operator Increment dan Decrement

[No. 4] Identifikasi Masalah

- 1) Uraikan permasalahan dan kode program

Contoh 4: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
public class operator {  
    public static void main(String[] args) {  
        // deklarasi nilai  
        int a = 5;  
  
        System.out.println("a: " + a);  
        System.out.println("b: " + (a++));  
    }  
}
```

Luaran:

a: 5

b: 5

Penjelasan :

Pada kode program diatas, langkah pertama yang harus dilakukan yaitu menentukan operator yang sesuai agar pada luaran menghasilkan a = 5 dan b = 6. Tipe data yang digunakan pada program tersebut yaitu tipe data **int** yang digunakan untuk menyatakan bilangan bulat.

- 2) Rincikan sumber informasi yang relevan

<https://www.youtube.com/watch?v=PzCMZObexZM>

Latihan 4.

4.1. Berikan saran operasi apa yang diperlukan (pre/post increment, pre/post decrement) agar Contoh 4 menghasilkan nilai a = 5 dan b = 6?

4.2. Simpulkan hasil eksperimen Anda!

Penjelasan :

Ketika **System.out.println("a: " + a);** dijalankan, nilai yang ditampilkan adalah 5, yaitu nilai awal dari variabel a dan setelah pre increment **System.out.println("b: " + (++a));** nilai yang ditampilkan yaitu 6 sebagai b. Hal ini bisa terjadi karena pre increment digunakan untuk menambah nilai a sebanyak satu.

[No. 4] Analisis dan argumentasi

Operator increment adalah operator yang berfungsi untuk menaikkan nilai variabel sebanyak satu, sedangkan operator decrement digunakan untuk menurunkan nilai variabel sebanyak satu. Pada program yang terdapat di soal, operator yang digunakan yaitu post increment. Untuk mendapatkan nilai a = 5 dan b = 6, maka saya mengusulkan untuk menggunakan operator pre increment.

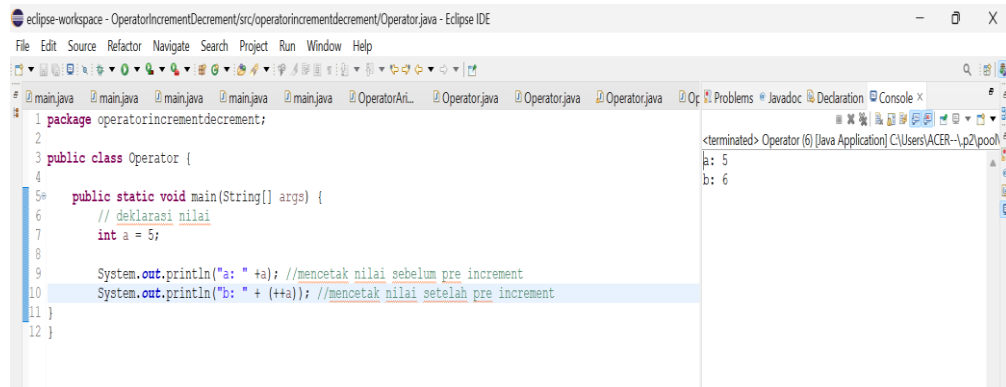
[No. 4] Penyusunan algoritma dan kode program

- 1) Algoritma
 - (a) Mulai
 - (b) Membuat kelas
 - (c) Membuat method
 - (d) Mendeklarasikan variabel

- (e) Mencetak nilai variabel sebelum pre increment
- (f) Mencetak nilai variabel setelah pre increment
- (g) Selesai

2) Kode program dan luaran

- a) Screenshot/ Capture potongan kode dan hasil luaran



The screenshot shows the Eclipse IDE interface. The main editor displays a Java file named `Operator.java` with the following code:

```
1 package operatorincrementdecrement;
2
3 public class Operator {
4
5     public static void main(String[] args) {
6         // deklarasi nilai
7         int a = 5;
8
9         System.out.println("a: " + a); //mencetak nilai sebelum pre increment
10        System.out.println("b: " + (++a)); //mencetak nilai setelah pre increment
11    }
12 }
```

The console window on the right shows the output of the program:

```
<terminated> Operator (6) [Java Application] C:\Users\ACER\AppData\Local\Temp\poolh
a: 5
b: 6
```

- b) Luaran yang dihasilkan

Luaran yang dihasilkan yaitu $a=5$ dan $b=6$ yang sudah sesuai dengan perintah yang diberikan.

[No. 4] Kesimpulan

1) Analisa

Program yang telah direkomendasi menggunakan operator pre increment telah menampilkan luaran yang diinginkan yaitu $a = 5$ dan $b = 6$, yang sesuai dengan perintah yang diberikan. Pre increment digunakan untuk menambah atau meningkatkan nilai satu angka

5. Operator Logika

[No. 5] Identifikasi Masalah

- 1) Uraikan permasalahan dan kode program

Contoh 5: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
public class OperatorLogika {  
    public static void main(String[] args) {  
        // deklarasi nilai  
        boolean a = true;  
        boolean b = false;  
  
        System.out.println("Hasil logika (a && b) : " + (a && b)); //menampilkan hasil  
        logika AND  
    } }  

```

Luaran:

Hasil logika (a && b) : false

- 2) Rincikan sumber informasi yang relevan

<https://www.youtube.com/watch?v=LcFgl0yrKEw>

Latihan 5

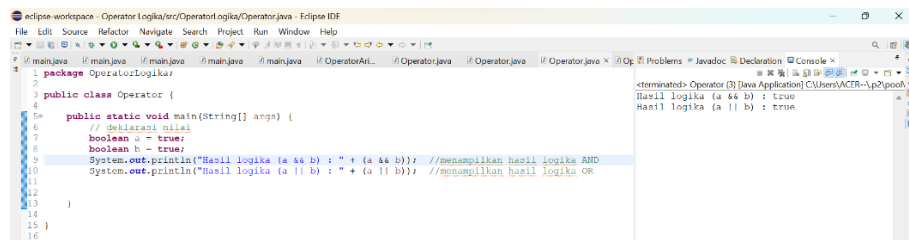
- 5.1. Rekomendasikan berapa nilai a dan b apabila ingin menghasilkan luaran *true* dengan operator && dan operator || ?
- 5.2. Berikan kesimpulan dari latihan 5.1.

[No. 5] Analisis dan argumentasi

Agar dapat menghasilkan luaran *true* dengan operator && dan operator ||, maka, nilai a dan b pada operator && harus *true* semua serta pada operator || nilai a dan b salah satunya *true* atau keduanya *true*.

[No. 5] Algoritma dan penyusunan kode program

- 1) Algoritma
 - (a) Mulai
 - (b) Membuat kelas dan method
 - (c) Deklarasi variabel
 - (d) Melakukan operasi logika
 - (e) Menampilkan hasil
 - (f) Selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran



```
1 package OperatorLogika;
2
3 public class Operator {
4
5     public static void main(String[] args) {
6         // Deklarasi nilai
7         boolean a = true;
8         boolean b = true;
9         System.out.println("Hasil logika (a && b) : " + (a && b)); //menampilkan hasil logika AND
10        System.out.println("Hasil logika (a || b) : " + (a || b)); //menampilkan hasil logika OR
11    }
12
13 }
14
15
16
```

Console Output:

```
<terminated> Operator (3) [Java Application] C:\Users\ACER\AppData\Local\Temp\
Hasil logika (a && b) : true
Hasil logika (a || b) : true
```

b) Luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai dan tepat dan menghasilkan luaran true dengan operator && dan operator || .

[No. 5] Kesimpulan

1. Analisa

Untuk mendapatkan nilai true pada operator AND (&&), maka nilai variabel keduanya harus true. Pada operator OR (||), untuk mendapatkan atau menghasilkan nilai true, maka, nilai variabel keduanya harus true atau salah satu variabel true.

6. Operator Kondisional (Ternary)

[No. 6] Identifikasi Masalah

- 1) Uraikan permasalahan dan kode program

Contoh 6:

```
public class OperatorKondisi{  
    public static void main( String[] args ){  
        String status = "";  
        int nilai = 80;  
        status = (nilai > 60)?"Lulus":"Gagal";  
        System.out.println( status );  
    } }  

```

Luaran:

Lulus

- 2) Rincikan sumber informasi yang relevan

<https://www.youtube.com/watch?v=LcFgl0yrKEw>

Latihan 6

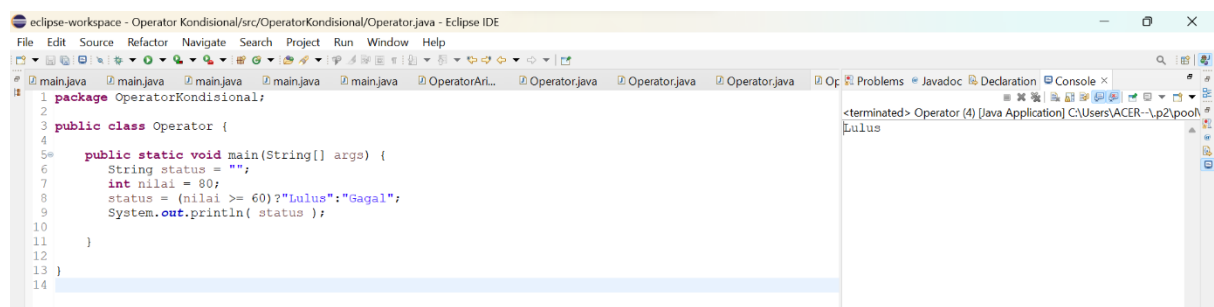
Rekomendasikan apa bentuk tanda operator agar nilai = 60 memenuhi untuk Lulus !

[No. 6] Analisis dan argumentasi

Untuk memecahkan masalah ini, saya mengusulkan untuk mengganti operator = menjadi operator >=. Hal ini dikarenakan, Ketika menggunakan operator >=, maka nilai = 60 akan lulus dan nilai > 60 akan lulus juga.

[No. 6] Penyusunan algoritma dan kode program

- 1) Algoritma
 - (a) Mulai
 - (b) Membuat kelas dan method
 - (c) Deklarasi variabel
 - (d) Menentukan nilai menggunakan operator ternary
 - (e) Menampilkan hasil
 - (f) Selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran



b) Luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai yaitu nilai = 60 lulus dan nilai > 60 juga lulus.

[No. 6] Kesimpulan

1. Analisa

Pada operator kondisional atau ternary, membantu dalam membuat sebuah Keputusan. Untuk memastikan nilai 60 memenuhi syarat untuk Lulus, operator yang perlu digunakan yaitu \geq (lebih besar atau sama dengan). Ini memastikan bahwa nilai 60 dianggap memenuhi syarat untuk Lulus.

7. Operator Bitwise

[No. 7] Identifikasi Masalah

1) Uraikan permasalahan dan kode program

Contoh 7: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
public class OperatorBitwise {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 7;  
        int hasil;  
  
        hasil = a & b;  
        System.out.println("Hasil dari a & b : " + hasil );  
  
        hasil = a | b;  
        System.out.println("Hasil dari a | b : " + hasil );  
  
        hasil = a ^ b;  
        System.out.println("Hasil dari a ^ b : " + hasil );  
  
        hasil = ~a;  
        System.out.println("Hasil dari ~a : " + hasil );  
  
        hasil = a >> 1;  
        System.out.println("Hasil dari a >> 1 : " + hasil );  
  
        hasil = b << 2;  
        System.out.println("Hasil dari b << 2 : " + hasil );  
    } } }
```

Luaran:

Hasil dari a & b : 2

Hasil dari a | b : 15

Hasil dari a ^ b : 13

Hasil dari ~a : -11

Hasil dari a >> 1 : 5

Hasil dari b << 2 : 28

2) Rincikan sumber informasi yang relevan

<https://www.youtube.com/watch?v=LcFgl0yrKEw>

Latihan 7

Evaluasi penyebab hasil ~a = -11 ? Buktikan jawaban Anda dalam perhitungan biner!

[No. 7] Analisis dan argumentasi

Langkah pertama yang saya lakukan untuk mendapatkan jawaban dari pertanyaan diatas adalah mencari informasi mengenai berapa nilai biner 10 dan biner 10 = 1010. Operator bitwise NOT (~) membalikkan setiap bit menjadi nilai yang berlawanan. Ini berarti semua bit 0 diubah menjadi 1, dan semua bit 1 diubah menjadi 0. Berikut langkah-langkah penyebab hasil ~a=-11.

1. Nilai int a = 10 dan biner dari bilangan 10 direpresentasikan sebagai 0000 1010 (dalam 8 bit).
2. Kemudian, setelah menerapkan operator NOT ~ yang berfungsi untuk membalikkan setiap bit dari bilangan biner, maka menjadi 1111 0101 dan merupakan bentuk dua's complement untuk angka negatif.
3. Untuk menghitung nilai desimal dari 1111 0101, hitung kembali inversi 1111 0101, yaitu 0000 1010.
4. Tambahkan 1 : $1 + 0000\ 1010 = 0000\ 1011$ dan dalam desimal hasilnya 11.
5. Karena bit pertama atau paling kiri adalah 1, ini menunjukkan angka negatif, jadi hasil akhirnya adalah -11.

[No. 7] Penyusunan algoritma dan kode program

- 1) Algoritma
 - (a) Mulai
 - (b) Inisialisasi variabel
 - (c) Mencari dan menentukan biner 10
 - (d) Menerapkan Operator Bitwise NOT (~)
 - (e) Hitung nilai desimal dari hasil
 - (f) Selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran

The screenshot shows the Eclipse IDE with a Java file named `Operator.java` in the package `operatorbitwise`. The code defines a class `Operator` with a `main` method that performs various bitwise operations on `a = 10` and `b = 7`.

```

1 package operatorbitwise;
2
3 public class Operator {
4
5     public static void main(String[] args) {
6         int a = 10;
7         int b = 7;
8         int hasil;
9
10        // Operasi Bitwise AND
11        hasil = a & b; //menghitung bitwise AND
12        System.out.println("Hasil dari a & b : " + hasil); //menampilkan hasil
13        // Operasi Bitwise OR
14        hasil = a | b; //menghitung bitwise OR
15        System.out.println("Hasil dari a | b : " + hasil); //menampilkan hasil
16        // Operasi Bitwise XOR
17        hasil = a ^ b; //menghitung bitwise XOR
18        System.out.println("Hasil dari a ^ b : " + hasil); //menampilkan hasil
19        // Operasi Bitwise NOT
20        hasil = ~a; //menghitung bitwise NOT
21        System.out.println("Hasil dari ~a : " + hasil); //menampilkan hasil
22        // Operasi Bitwise Shift kanan
23        hasil = a >> 1; //menggeser ke kanan 1 bit
24        System.out.println("Hasil dari a >> 1 : " + hasil); //menampilkan hasil
25        // Operasi Bitwise Shift kiri
26        hasil = b << 2; //menggeser ke kiri 2 bit
27        System.out.println("Hasil dari b << 2 : " + hasil); //menampilkan hasil
28
29    }
30
31 }
32

```

The console output on the right shows the results of these operations:

```

<terminated> Operator (7) [Java Application] C:\Users\ACER~1\p2\poo...
Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
Hasil dari ~a : -11
Hasil dari a >> 1 : 5
Hasil dari b << 2 : 28

```

[No.7] Kesimpulan

1. Analisa

Operator NOT (~) merupakan operator yang digunakan untuk membalikkan setiap bit menjadi nilai yang berlawanan, Dimana 0 akan diubah menjadi 1 dan begitu pula sebaliknya. Sebagai contoh, penerapan operator NOT(~) pada bilangan 10 menghasilkan 1111 0101. Konversi 1111 0101 ke desimal dengan cara dua's complement menunjukkan bahwa nilai yang sebenarnya adalah -11.

Refleksi :

Dalam menyelesaikan tugas ini, Saya mendapatkan banyak pengetahuan baru dan pengalaman bery. Masih banyak materi yang kurang Saya pahami dan mengalami kesulitan dalam pengerjaannya. Namun, saya tidak menyerah dan terus mempelajari setiap materi yang sekiranya membuat Saya kesulita. Selain itu, berkat bantuan dari teman-teman, yang selalu dengan senang hati membantu Saya, tugas ini berhasil saya selesaikan dengan banyak keluhan, tangisan, senyuman, dan kopi setiap malam.

