

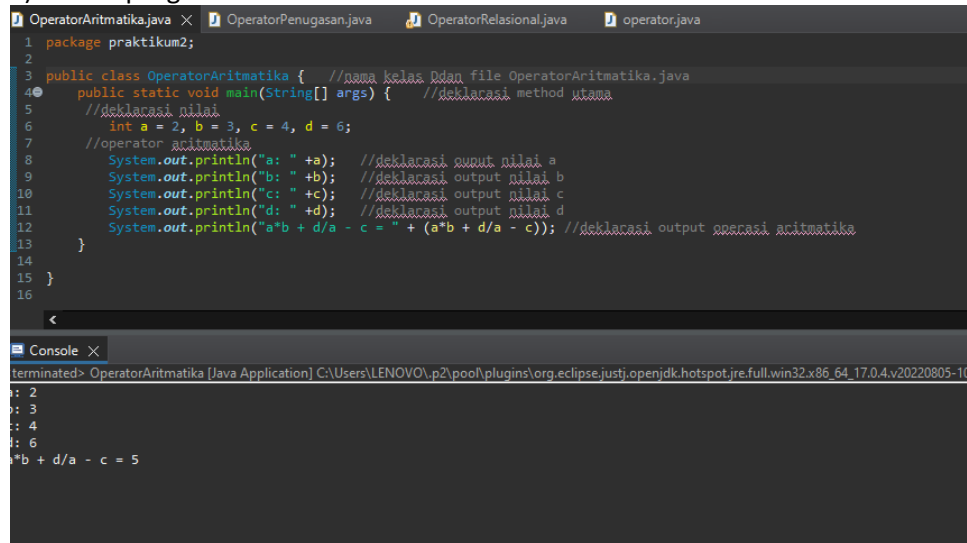
## Lembar Kerja Individu

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Sinta Ezra Wati Gulo G1A022040</b>	<b>Operator</b>	<b>9 September 2022</b>
<b><u>LATIHAN 1</u></b>		
<b>[Nomor 1] Identifikasi Masalah:</b>		
<p>1) Uraikan permasalahan dan variable</p> <pre>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));     } }</pre> <p>Luaran:</p> <pre>a: 20 b: 3 a + b = 23</pre> <p>Diketahui variabel pada program :</p> <pre>a = 20 b = 3</pre> <p>Pada program tersebut tidak terdapat pesan kesalahan dan merupakan program yang menggunakan operator aritmatika yaitu penambahan.</p> <p>Soal :</p> <p>Susun kode Java untuk perhitungan dengan ekspresi <math>(2*3 + 6 / 2 - 4)</math> Serta menyimpulkan urutan prioritas operator yang dijalankan ekspresi tersebut!</p> <p>2) Sumber informasi</p> <p>Materi : <a href="https://rumahilmu.org">Pemrograman Java (rumahilmu.org)</a></p> <p>Video : <a href="https://www.youtube.com/watch?v=PzCMZOboxZM">https://www.youtube.com/watch?v=PzCMZOboxZM</a></p>		
<b>[Nomor 1] Analisis dan Argumentasi</b>		
<p>1) Saya mengusulkan soal tersebut dapat diselesaikan dengan memasukkan nilai variabelnya dan membuat deklarasi operasi aritmatikanya dengan memasukkan operator kali(*), bagi(/), tambah(+), kurang(-).</p> <p>2) Alasan solusi ini agar luaran yang dihasilkan sesuai dengan permintaan pada soal, yaitu agar menghasilkan perhitungan dengan ekspresi <math>(2*3 + 6/2 - 4)</math>.</p>		
<b>[Nomor 1] Penyusunan Algoritma dan Kode Program</b>		
<p>1) Algoritma</p> <ul style="list-style-type: none"> <li>➤ Buat project java</li> <li>➤ Buat package</li> <li>➤ Nama kelas dan file (OperatorAritmatika.java)</li> <li>➤ Deklarasi method utama</li> <li>➤ Deklarasi nilai</li> <li>➤ Deklarasi output nilai a, b, c, dan d</li> <li>➤ Deklarasi output operasi aritmatika <math>(2*3 + 6 / 2 - 4)</math></li> </ul>		

## ➤ Luaran

### 2) Kode program dan luaran

#### a) Kode program



```
OperatorAritmatika.java x OperatorPenugasan.java OperatorRelasional.java operator.java
1 package praktikum2;
2
3 public class OperatorAritmatika { //nama kelas dan file OperatorAritmatika.java
4     public static void main(String[] args) { //deklarasikan method utama
5         //deklarasikan nilai
6         int a = 2, b = 3, c = 4, d = 6;
7         //operator aritmatika
8         System.out.println("a: " + a); //deklarasikan output nilai a
9         System.out.println("b: " + b); //deklarasikan output nilai b
10        System.out.println("c: " + c); //deklarasikan output nilai c
11        System.out.println("d: " + d); //deklarasikan output nilai d
12        System.out.println("a*b + d/a - c = " + (a*b + d/a - c)); //deklarasikan output operasi aritmatika
13    }
14 }
15 }
16 }

Console x
terminated> OperatorAritmatika [Java Application] C:\Users\LENOVO\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220805-10
a: 2
b: 3
c: 4
d: 6
a*b + d/a - c = 5
```

#### b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sesuai dengan kode yang disusun. Pada luaran juga dapat kita lihat bahwa hasil dari  $(2*3 + 6/2 - 4)$  adalah 5.

Sehingga dapat disimpulkan bahwa urutan prioritas pada soal tersebut adalah :

- Perkalian dan pembagian (dimulai dari yang paling kiri)
- Kemudian penjumlahan dan pengurangan (dimulai dari yang paling kiri)

### [Nomor 1] Kesimpulan (Analisa)

Pada program tersebut saya menggunakan bentuk kelas public agar program tersebut juga dapat diakses oleh publik, tidak saya sendiri. Kemudian berdasarkan program yang saya jalankan dapat dilihat bahwa pada operasi aritmatika terdapat prioritas operator, dengan urutan prioritasnya adalah tanda kurung (...)/[...] > tanda kali (\*) / tanda bagi (/) / modulus (%) > tanda tambah (+) / tanda kurang (-).

## LATIHAN 2

### [Nomor 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

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

Luaran:

Penambahan : 23

Diketahui variabel pada program :

a = 20

b = 3

Pada program tersebut tidak terdapat pesan kesalahan dan merupakan program yang menggunakan operator penugasan yaitu penambahan.

Soal :

Susun kode untuk menampilkan perhitungan dengan operator ( -=, \*=, /=, %=)! Simpulkan hasilnya!

- 2) Sumber informasi

Materi : [Pemrograman Java \(rumahilmu.org\)](http://rumahilmu.org)

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

### [Nomor 2] Analisis dan Argumentasi

- 1) Pada program tersebut saya tambahkan deklarasi operator penugasan ( -=) untuk penugasan pengurangan, ( \*=) untuk penugasan perkalian, ( /=) untuk penugasan pembagian, dan ( %=) untuk penugasan modulus/sisa bagi.
- 2) Alasan solusi ini agar luaran yang dihasilkan sesuai dengan permintaan pada soal, yaitu agar menampilkan perhitungan dengan operator ( -=, \*=, /=, %=).

### [Nomor 2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
  - Buat project java
  - Buat package
  - Nama kelas dan file (OperatorPenugasan.java)
  - Deklarasi method utama
  - Deklarasi nilai
  - Deklarasi output nilai a, b, c, dan d
  - Penugasan ( -=, \*=, /=, %=)
  - Deklarasi output operasi penugasan ( -=, \*=, /=, %=)
  - Luaran

## 2) Kode program dan luaran

### a) Kode program

```
OperatorPenugasan.java X
1 package praktikum2;
2
3 public class OperatorPenugasan { //nama kelas dan file OperatorPenugasan.java
4     public static void main(String[] args) { //deklarasikan method utama
5         //deklarasikan nilai
6         int a = 20, b = 3;
7         System.out.println("a: " + a); //deklarasikan output nilai a
8         System.out.println("b: " + b); //deklarasikan output nilai b
9
10        //operator penugasan
11        b += a; //penugasan penambahan
12        System.out.println("Penambahan : " + b); //deklarasikan output penambahan
13
14        b -= a; //penugasan pengurangan
15        System.out.println("Pengurangan : " + b); //deklarasikan output pengurangan
16
17        b *= a; //penugasan perkalian
18        System.out.println("Perkalian : " + b); //deklarasikan output perkalian
19
20        b /= a; //penugasan pembagian
21        System.out.println("Pembagian : " + b); //deklarasikan output pembagian
22
23        b %= a; //penugasan modulus atau sisa hasil
24        System.out.println("Modulus : " + b); //deklarasikan output modulus
25    }
26
27 }
```

```
Console X
<terminated> OperatorPenugasan [Java Application] C:\Users\LENOVO\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win
a: 20
b: 3
Penambahan : 23
Pengurangan : 3
Perkalian : 60
Pembagian : 3
Modulus : 3
```

### b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sesuai dengan kode yang disusun. Pada luaran juga dapat kita lihat hasil dari tiap penugasan ( -=, \*=, /=, %=)

Sehingga dapat disimpulkan bahwa hasil pertama mempengaruhi hasil selanjutnya. Misalkan pada penambahan nilai a=20 dan b=3, sehingga ketika di run hasilnya adalah 23. Namun pada penugasan selanjutnya nilai b=23 (karena pada penambahan  $b += a$  yang artinya  $b = b + a$ , sehingga nilai b selanjutnya adalah 23 bukan 3 lagi), jadi dengan nilai b=23 dikurang dengan nilai a=20 menghasilkan 3 (dengan ekspresi  $b -= a$  yang artinya  $b = b - a$ , dengan begitu nilai b selanjutnya adalah 3 bukan 23 lagi). Jadi hingga perhitungan selanjutnya akan seperti itu.

### [Nomor 2] Kesimpulan (Analisa)

Pada program tersebut saya menggunakan bentuk kelas public agar program tersebut juga dapat diakses oleh publik, tidak saya sendiri. Kemudian berdasarkan program yang saya jalankan dapat dilihat bahwa pada operasi penugasan hasil pada operasi pertama mempengaruhi nilai pada operasi selanjutnya karena jika dimisalkan pada operasi penambahan  $a += b$  yang artinya adalah  $a = a + b$ , sehingga nilai a yang digunakan pada operasi berikutnya adalah hasil dari  $a + b$ . Hal tersebut berlaku sampai operasi selanjutnya lagi.

### LATIHAN 3

#### [Nomor 3] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

```
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
```

Diketahui variabel pada program :

a = 12

b = 4

Pada program tersebut tidak terdapat pesan kesalahan, namun tidak semua luarannya menghasilkan TRUE. Sehingga untuk memenuhi permintaan soal harus membuat/mengubah hubungan antara nilai atau variabel nya yang agar semua luaran nya menghasilkan TRUE.

Soal :

Susunlah perintah kode dengan operator relasional (<, >, <=, >=, =, ==, !=) untuk nilai a dan b yang menghasilkan luaran TRUE!

2) Sumber informasi

Materi : [Pemrograman Java \(rumahilmu.org\)](http://rumahilmu.org)

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

### [Nomor 3] Analisis dan Argumentasi

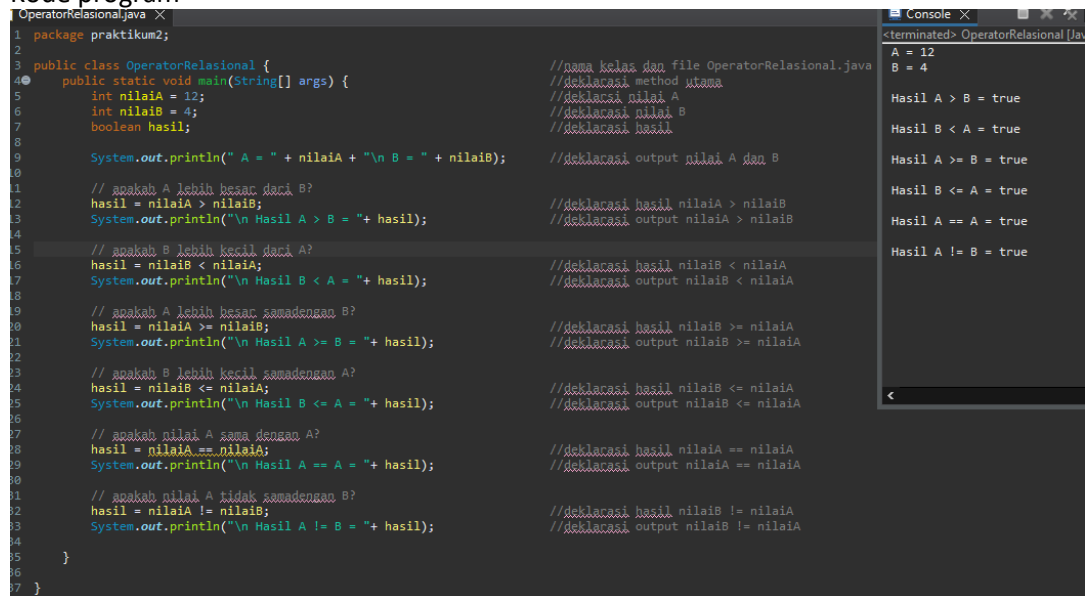
- 1) Agar program tersebut menghasilkan luaran TRUE semua saya mengubah pada bagian deklarasi hasilnya, misalnya pada deklarasi hasil nya nilai nilaiA < nilaiB maka saya ubah menjadi nilaiB < nilaiA sehingga yang tadi luaran nya adalah FALSE menjadi TRUE. Begitu pun pada program yang lain yang menghasilkan FALSE saya perbaiki pada deklarasi hasil nya.
- 2) Alasan solusi ini agar luaran yang dihasilkan sesuai dengan permintaan pada soal, yaitu agar semua luaran yang dihasilkan menampilkan TRUE.

### [Nomor 3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
  - Buat project java
  - Buat package
  - Nama kelas dan file (OperatorPenugasan.java)
  - Deklarasi method utama
  - Deklarasi nilai
  - Deklarasi hasil (menggunakan tipe data boolean)
  - Deklarasi output nilai a, b, c, dan d
  - Deklarasi hasil operator relasional
  - Deklarasi output operator relasional (<, >, <=, >=, ==, !=)
  - Luaran

### 3) Kode program dan luaran

#### a) Kode program



```
1 package praktikum2;
2
3 public class OperatorRelasional {
4     public static void main(String[] args) {
5         int nilaiA = 12;
6         int nilaiB = 4;
7         boolean hasil;
8
9         System.out.println("A = " + nilaiA + " B = " + nilaiB);
10
11         // apakah A lebih besar dari B?
12         hasil = nilaiA > nilaiB;
13         System.out.println("\n Hasil A > B = " + hasil);
14
15         // apakah B lebih kecil dari A?
16         hasil = nilaiB < nilaiA;
17         System.out.println("\n Hasil B < A = " + hasil);
18
19         // apakah A lebih besar samadengan B?
20         hasil = nilaiA >= nilaiB;
21         System.out.println("\n Hasil A >= B = " + hasil);
22
23         // apakah B lebih kecil samadengan A?
24         hasil = nilaiB <= nilaiA;
25         System.out.println("\n Hasil B <= A = " + hasil);
26
27         // apakah nilai A sama dengan A?
28         hasil = nilaiA == nilaiA;
29         System.out.println("\n Hasil A == A = " + hasil);
30
31         // apakah nilai A tidak samadengan B?
32         hasil = nilaiA != nilaiB;
33         System.out.println("\n Hasil A != B = " + hasil);
34     }
35 }
```

Console Output:

```
<terminated> OperatorRelasional (Ja
A = 12
B = 4
Hasil A > B = true
Hasil B < A = true
Hasil A >= B = true
Hasil B <= A = true
Hasil A == A = true
Hasil A != B = true
```

#### b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sesuai dengan program yang disusun. Dengan memainkan logika, kita dapat menyusun kode yang menghasilkan TRUE pada semua luaran programnya tanpa mengubah value pada variabelnya.

### [Nomor 3] Kesimpulan (Analisa)

Pada program tersebut saya menggunakan bentuk kelas public agar program tersebut juga dapat diakses oleh publik, tidak saya sendiri. Kemudian berdasarkan program yang saya jalankan dapat

dilihat bahwa operasi relasional digunakan untuk menguji hubungan antara nilai dan atau variabel dan selalu menghasilkan nilai true atau false. Sehingga dengan memanfaatkan operator relasional dan logika, kita dapat menggunakan operator relasional untuk menghasilkan luaran TRUE atau FALSE.

## LATIHAN 4

### **[Nomor 4] Identifikasi Masalah:**

- 1) Uraikan permasalahan dan variabel

```
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

Diketahui variabel pada program :  
a = 5

Pada program tersebut tidak terdapat pesan kesalahan, dan program tersebut adalah contoh dari operator post increment. Namun hasilnya belum keluar sesuai dengan penggunaan increment itu sendiri karena hasilnya masih tetap 5, hal tersebut karena tidak menampilkan kembali hasil yang sudah di increment.

Soal :

- 4.1. Susunlah kode program untuk menghasilkan luaran nilai a = 5 dan b = 6 dengan pre/post increment dan pre/post decrement.
- 4.2. Simpulkan hasil perbandingan Anda (pre/post increment, pre/post decrement)!

- 2) Sumber informasi

Materi : [Pemrograman Java \(rumahilmu.org\)](http://rumahilmu.org)

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

### **[Nomor 4] Analisis dan Argumentasi**

- 1) Agar program tersebut menghasilkan luaran a=5 dan b=6, saya melakukan increment dahulu lalu decrement agar hasil keduanya saling berkesinambungan.
- 2) Alasan solusi ini agar luaran yang dihasilkan sesuai dengan permintaan pada soal, yaitu agar semua luaran yang dihasilkan menampilkan nilai a=5 dan b=6.

### **[Nomor 4] Penyusunan Algoritma dan Kode Program**

- 1) Algoritma
  - Buat project java
  - Buat package
  - Nama kelas dan file (Operator.java)
  - Deklarasi method utama
  - Deklarasi nilai
  - Deklarasi output untuk menghasilkan luaran "(Pre-Increment)"
  - Deklarasi output nilai a sebelum di increment
  - Deklarasi output pre-increment nilai a
  - Deklarasi output nilai b dari nilai a yang sudah di increment
  - Deklarasi output untuk menghasilkan luaran "(Pre-Decrement)"
  - Deklarasi output nilai b sebelum di decrement



- Deklarasi output pre-decrement nilai a
- Deklarasi output nilai a yang sudah di decrement
- Deklarasi output untuk menghasilkan luaran "(Post-Increment)"
- Deklarasi output nilai a sebelum di increment
- Deklarasi output post-increment nilai a
- Deklarasi output nilai b dari nilai a yang sudah di increment
- Deklarasi output untuk menghasilkan luaran "(Post-Decrement)"
- Deklarasi output nilai b sebelum di decrement
- Deklarasi output post-decrement nilai a
- Deklarasi output nilai a yang sudah di decrement
- Luaran

## 2) Kode program dan luaran

### 4.1 Susunan kode program untuk menghasilkan a=5 dan b=6 dengan menggunakan pre/post increment dan pre/post decrement

#### a) Kode program

```

1 package praktikum2;
2 public class operator {
3     public static void main(String[] args) {
4         int a = 5;
5
6         //deklarasikan pre-increment
7         System.out.println("(Pre-Increment)");
8         System.out.println("a: " + ++a);
9         System.out.println("b: " + (++a));
10        System.out.println("a: " + ++a);
11
12        //deklarasikan pre-decrement
13        System.out.println("(Pre-Decrement)");
14        System.out.println("b: " + --b);
15        System.out.println("a: " + (--a));
16        System.out.println("a: " + --a);
17
18        //deklarasikan post-increment
19        System.out.println("(Post-Increment)");
20        System.out.println("a: " + a++);
21        System.out.println("b: " + (a++));
22        System.out.println("b: " + a++);
23
24        //deklarasikan post-decrement
25        System.out.println("(Post-Decrement)");
26        System.out.println("b: " + b--);
27        System.out.println("b: " + (b--));
28        System.out.println("a: " + a--);
29    }
30 }
31

```

Console Output:

```

<terminated> Operator [a
(Pre-Increment)
a: 5
b: 6
b: 6
(Pre-Decrement)
b: 6
a: 5
a: 5
(Post-Increment)
a: 5
b: 6
b: 6
a: 5
(Post-Decrement)
b: 6
b: 6
a: 5

```

#### b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sesuai dengan program yang disusun. Dengan memanfaatkan pre/post increment dan pre/post decrement kita dapat menghasilkan luaran a=5 dan b=6.

### 4.2 Kesimpulan hasil perbandingan pre/post increment dan pre/post decrement

- Perbedaan antara increment dan decrement sendiri adalah jika increment merupakan operator yang digunakan untuk menambahkan +1 pada nilai yang dimasukkan sebelumnya, sedangkan decrement merupakan operator yang digunakan untuk mengurangi -1 pada nilai yang dimasukkan sebelumnya.
- Perbedaan antara pre-increment dan post-increment, sebenarnya fungsinya tidak ada perbedaan yaitu sama-sama menambahkan +1. Perbedaannya hanya terletak pada prosesnya, dimana pre-increment menambahkan nilai yang dimasukkan sebanyak 1 angka lalu hasilnya langsung tampil (misal : a > ++a), sedangkan pada post-increment setelah menambahkan nilai yang dimasukkan sebanyak 1 angka maka perlu membuat deklarasi outputnya lagi diakhir yang menampilkan nilai yang dimasukkan (misal : a > a++ > a), baru luaran dengan penambahan 1 angkanya dapat muncul.
- Perbedaan antara pre-decrement dan post-decrement, sebenarnya fungsinya tidak ada perbedaan yaitu sama-sama mengurangi -1. Perbedaannya hanya terletak pada

prosesnya, dimana pre-decrement mengurangi nilai yang dimasukkan sebanyak 1 angka lalu hasilnya langsung tampil (misal :  $a > --a$ ), sedangkan pada post-decrement setelah mengurangi nilai yang dimasukkan sebanyak 1 angka maka perlu membuat deklarasi outputnya lagi diakhir yang menampilkan nilai yang dimasukkan (misal :  $a > a-- > a$ ), baru luaran dengan pengurangan 1 angkanya dapat muncul.

#### **[Nomor 4] Kesimpulan (Analisa)**

Pada program tersebut saya menggunakan bentuk kelas public agar program tersebut juga dapat diakses oleh publik, tidak saya sendiri. Kemudian berdasarkan program yang saya jalankan dapat dilihat bahwa operasi increment dan decrement adalah operator yang digunakan pada operand bilangan bulat dan memiliki fungsi yang berbeda. Increment berfungsi untuk menaikkan nilai variabel sebesar satu dan Decrement berfungsi untuk menurunkan nilai variabel sebesar satu. Masing-masing operator terbagi menjadi pre/post yang mana perbedaannya hanya terletak pada proses menghasilkan luarannya hingga dapat menghasilkan penambahan atau pengurangan sebanyak satu.

## LATIHAN 5

### [Nomor 5] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

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

Luaran:

Hasil logika (a && b) : false

Diketahui variabel pada program :

a = true

b = false

Pada program tersebut tidak terdapat pesan kesalahan, namun luarannya masih menghasilkan FALSE, sementara pada soal diminta untuk menghasilkan TRUE. Sehingga untuk menghasilkan TRUE maka nilai variabelnya harus diubah.

Soal :

Susun kode program dengan mengubah nilai a dan b untuk menghasilkan luaran true dengan operator && dan operator | |. Beri kesimpulan!

- 2) Sumber informasi

Materi : [Pemrograman Java \(rumahilmu.org\)](http://rumahilmu.org)

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

### [Nomor 5] Analisis dan Argumentasi

- 1) Agar program tersebut menghasilkan luaran TRUE yang perlu dilakukan adalah dengan mengubah nilai a dan b. Luaran TRUE akan keluar apabila kedua nilai variabel bernilai TRUE, jadi karena nilai a sudah true maka yang perlu di ubah hanyalah nilai b yang tadinya adalah FALSE, diubah menjadi TRUE.
- 2) Alasan solusi ini agar luaran yang dihasilkan sesuai dengan permintaan pada soal, yaitu agar luaran nya menghasilkan TRUE.

### [Nomor 5] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
  - Buat project java
  - Buat package
  - Nama kelas dan file (operator2.java)
  - Deklarasi method utama
  - Deklarasi nilai a
  - Deklarasi nilai b
  - Deklarasi output hasil menggunakan &&
  - Deklarasi output hasil menggunakan | |
  - Luaran
- 2) Kode program dan luaran

a) Kode program

```
OperatorRelasional.java  operator.java  operator2.java X
1 package praktikum2;
2
3 public class operator2 {
4     public static void main(String[] args) {
5         boolean a = true;
6         boolean b = true;
7
8         System.out.println("Hasil logika (a && b) : " + (a && b));
9         System.out.println("Hasil logika (a || b) : " + (a || b));
10    }
11 }
12
13
<
Console X
<terminated> operator2 [Java Application] C:\Users\LENOVO\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220805-10
Hasil logika (a && b) : true
Hasil logika (a || b) : true
```

b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sesuai dengan program yang disusun. Dengan mengubah kedua nilai menjadi TRUE, maka hasilnya adalah TRUE. Dengan menyatakan TRUE (1) dan FALSE (0), maka operator && akan menghasilkan nilai TRUE apabila kedua nilai bernilai TRUE, sedangkan untuk operator || akan menghasilkan nilai TRUE apabila kedua nilainya bernilai TRUE atau jika salah satu nilainya bernilai TRUE.

**[Nomor 5] Kesimpulan (Analisa)**

Pada program tersebut saya menggunakan bentuk kelas public agar program tersebut juga dapat diakses oleh publik. Berdasarkan program juga dapat dilihat bahwa untuk menghasilkan true pada && dan || maka kedua nilai yang di input harus bernilai true ataupun pada || juga bisa menghasilkan true apabila salah satu nilainya bernilai true. Operator logika sendiri digunakan untuk membandingkan dua nilai variabel atau lebih, dimana hasilnya adalah boolean true atau false.

## LATIHAN 6

### [Nomor 6] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
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

Diketahui variabel pada program :  
nilai = 80

Pada program tersebut tidak terdapat pesan kesalahan, program tersebut merupakan contoh untuk operator kondisional/ternary.

Soal :

Susun kode program! Dengan informasi berikut:

Diketahui nama variabel Jam = 12

Apabila jam < 12 maka tampil "Selamat Pagi", apabila jam > 12 maka tampil "Selamat Malam".

- 2) Sumber informasi

Materi : [Pemrograman Java \(rumahilmu.org\)](http://rumahilmu.org)

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

### [Nomor 6] Analisis dan Argumentasi

- 1) Agar program tersebut menghasilkan luaran jam < 12 maka tampil "Selamat Pagi" dan jam > 12 maka tampil "Selamat Malam". Namun selain itu ada yang perlu diperbaiki pada soal, yaitu value pada nilai variabel jam (jam = 12), karena berdasarkan contoh yang sudah diberikan sebelumnya, nilai variabel tidak sama dengan nilai yang ada pada ekspresi1. Misalnya jika nilai variabel jam adalah int jam = 12, maka ekspresi1 seharusnya (jam < 13) ataupun (jam > 13). Namun karena fokus saya adalah untuk menghasilkan jam < 12 "Selamat pagi" dan jam > 12 "Selamat malam" maka yang saya ubah adalah nilai variabelnya, bukan jam = 12 tetapi diganti dengan jam = 7.
- 2) Alasan solusi ini agar luaran yang dihasilkan sesuai dengan permintaan pada soal, yaitu agar luaran nya menghasilkan "Selamat pagi" apabila jam < 12 dan "Selamat malam" apabila jam > 12.

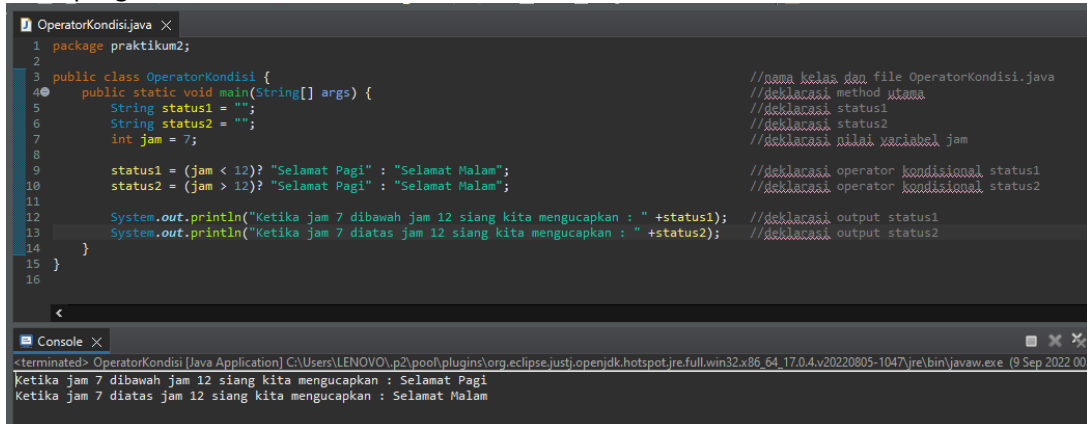
### [Nomor 6] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
  - Buat project java
  - Buat package
  - Nama kelas dan file (OperatorKondisi.java)
  - Deklarasi method utama
  - Deklarasi status1
  - Deklarasi status2
  - Deklarasi output operator kondisional status1
  - Deklarasi output operator kondisional status2

➤ Luaran

2) Kode program dan luaran

a) Kode program



```
1 package praktikum2;
2
3 public class OperatorKondisi {
4     public static void main(String[] args) {
5         String status1 = "";
6         String status2 = "";
7         int jam = 7;
8
9         status1 = (jam < 12)? "Selamat Pagi" : "Selamat Malam";
10        status2 = (jam > 12)? "Selamat Pagi" : "Selamat Malam";
11
12        System.out.println("Ketika jam 7 dibawah jam 12 siang kita mengucapkan : " +status1);
13        System.out.println("Ketika jam 7 diatas jam 12 siang kita mengucapkan : " +status2);
14    }
15 }
16
```

Console

```
<terminated> OperatorKondisi [Java Application] C:\Users\LENOVO\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220805-1047\jre\bin\javaw.exe (9 Sep 2022 00:
Ketika jam 7 dibawah jam 12 siang kita mengucapkan : Selamat Pagi
Ketika jam 7 diatas jam 12 siang kita mengucapkan : Selamat Malam
```

b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sesuai dengan program yang disusun. Dengan memanfaatkan operator kondisional kita dapat membuat luaran “Selamat pagi” dan “Selamat malam”. Dimana luaran “Selamat pagi” dihasilkan jika jam < 12 dan “Selamat malam” dihasilkan jika jam > 12. Dimana sebenarnya “Selamat pagi” merupakan ekspresi2 atau kondisi jika nilai boolean ekspresi1 (jam = 7) bernilai true, sedangkan “Selamat malam” merupakan ekspresi3 atau kondisi jika nilai boolean ekspresi1 (jam = 7) bernilai false.

**[Nomor 6] Kesimpulan (Analisa)**

Pada program tersebut saya menggunakan bentuk kelas public agar program tersebut juga dapat diakses oleh publik. Operator kondisional atau ternary adalah penyederhanaan dari bentuk if.else yang setiap blok dari if dan else hanya terdiri dari satu statement/perintah. Dimana operator ternary memiliki bentuk umum ekspresi1? ekspresi2 : ekspresi3, dengan keterangan ekspresi1 adalah boolean dengan hasil true atau false, ekspresi2 adalah nilai boolean ekspresi1 jika bernilai true, dan ekspresi3 adalah nilai boolean ekspresi1 jika bernilai false. Dengan begitu jika dikaitkan dengan soal, kita dapat menghasilkan selamat pagi dan selamat malam dengan memposisikannya sebagai true atau false.

## LATIHAN 7

### [Nomor 7] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
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 );  
    }  
}
```

Luaran:

```
Hasil dari a & b : 6  
Hasil dari a | b : 7  
Hasil dari a ^ b : 1
```

Diketahui variabel pada program :

```
a = 10  
b = 7
```

Pada program tersebut tidak terdapat pesan kesalahan, program tersebut merupakan contoh untuk operator bitwise yaitu operator untuk operasi bit (biner) dan berlaku untuk tipe data int, long, short, char, dan byte, karena akan menghitung dari bit-ke-bit.

Soal :

Susun kode tambahan dari Contoh 7 untuk melakukan perhitungan dengan operator (>>, <<). Hubungkan hasil luaran dengan perhitungan manual bilangan biner!

- 3) Sumber informasi

Materi : [Pemrograman Java \(rumahilmu.org\)](http://rumahilmu.org)  
[Bitwise Operator in Java \(javatpoint.com\)](http://javatpoint.com)  
[Tutorial Belajar Java: Jenis-jenis Operator Bitwise | Duniaikom](http://Duniaikom)  
Video : <https://www.youtube.com/watch?v=PzCMZObexZM>  
<https://youtu.be/Hgn6Tu1HTtl>

### [Nomor 7] Analisis dan Argumentasi

- 1) Pada soal diminta untuk menyusun kode tambahan dari contoh 7 dengan menggunakan operator >> dan <<. Sehingga saya hanya perlu menambahkan kode a << n dan atau a >> n serta b << n dan b >> n, dengan n adalah 1 (saya pilih 1 agar lebih simpel).
- 3) Alasan solusi ini agar luaran yang dihasilkan sesuai dengan permintaan pada soal, yaitu agar luaran nya menghasilkan perhitungan untuk operator >> dan <<.

### [Nomor 7] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
  - Buat project java

- Buat package
- Nama kelas dan file (OperatorBitwise.java)
- Deklarasi method utama
- Deklarasi nilai a
- Deklarasi nilai b
- Deklarasi hasil
- Deklarasi hasil a & b
- Deklarasi output a & b
- Deklarasi hasil a | b
- Deklarasi output a | b
- Deklarasi hasil a ^ b
- Deklarasi output a ^ b
- Deklarasi hasil a << 1
- Deklarasi output a << 1
- Deklarasi hasil a >> 1
- Deklarasi output a >> 1
- Deklarasi hasil b << 1
- Deklarasi output b << 1
- Deklarasi hasil b >> 1
- Deklarasi output b >> 1
- Luaran

## 2) Kode program dan luaran

### a) Kode program

```

1 package praktikum2;
2
3 public class OperatorBitwise {
4     public static void main(String[] args) {
5         int a = 10;
6         int b = 7;
7         int hasil;
8
9         hasil = a & b;
10        System.out.println("Hasil dari a & b : " + hasil);
11
12        hasil = a | b;
13        System.out.println("Hasil dari a | b : " + hasil);
14
15        hasil = a ^ b;
16        System.out.println("Hasil dari a ^ b : " + hasil);
17
18        hasil = a << 1;
19        System.out.println("Hasil dari a << 1 : " + hasil);
20
21        hasil = a >> 1;
22        System.out.println("Hasil dari a >> 1 : " + hasil);
23
24        hasil = b << 1;
25        System.out.println("Hasil dari b << 1 : " + hasil);
26
27        hasil = b >> 1;
28        System.out.println("Hasil dari b >> 1 : " + hasil);
29    }
30 }
31 }

```

```

<terminated> OperatorBitwise [Java]
Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
Hasil dari a << 1 : 20
Hasil dari a >> 1 : 5
Hasil dari b << 1 : 14
Hasil dari b >> 1 : 3

```

### b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sesuai dengan program yang disusun. Dengan memanfaatkan operator bitwise saya dapat menyelesaikan soal untuk membuat program dengan menggunakan operator << dan >>. Kemudian jika 10 dikonversikan ke bilangan biner akan menghasilkan 1010 atau 00001010 dan 7 jika dikonversikan ke bilangan biner akan menghasilkan 0111 atau 00000111. Berdasarkan luaran yang dihasilkan dapat dilihat hubungan antara hasil luaran dengan perhitungan manual bilangan biner adalah sebagai berikut :

- a & b (a AND b), dimana pada operator & hanya jika kedua nilai bernilai 1 maka akan menghasilkan 1, selain itu maka hasilnya 0. Pada soal 1010 dioperasikan dengan 0111 menghasilkan 0010 atau jika dikonversi ke desimal adalah 2.
- a | b (a OR b), dimana pada operator | hanya jika kedua nilainya bernilai 1 atau salah satunya bernilai 1, jika kedua nilainya bernilai 0 maka hasilnya bernilai 0.



Pada soal 1010 dioperasikan dengan 0111 menghasilkan 1111 atau jika dikonversi ke desimal adalah 15.

- $a \wedge b$  (a XOR b), dimana pada operator  $\wedge$  hanya jika kedua nilai tidak sama (0 dengan 1 atau 1 dengan 0) maka akan menghasilkan 1, jika kedua nilainya bernilai sama (1 dengan 1 atau 0 dengan 0) maka hasilnya 0. Pada soal 1010 dioperasikan dengan 0111 menghasilkan 1101 atau jika dikonversi ke desimal adalah 13.
- $a \ll 1$  (a left shift sebanyak 1 bit), dimana pada operator  $\ll$  akan menggeser posisi bit ke bit sebelah kiri sebanyak n kali (pada program yang saya buat, saya geser sebanyak 1 bit). Pada soal yaitu 1010 atau 00001010 digeser 1 menjadi 00010100 diambil menjadi 10100 atau didesimalkan menjadi 20.
- $a \gg 1$ , (a right shift sebanyak 1 bit), dimana pada operator  $\gg$  akan menggeser posisi bit ke bit sebelah kanan sebanyak n kali (pada program yang saya buat, saya geser sebanyak 1 bit). Pada soal yaitu 1010 atau 00001010 digeser 1 menjadi 00000101 diambil menjadi 0101 atau didesimalkan menjadi 5.
- $b \ll 1$  (b left shift sebanyak 1 bit), dimana pada operator  $\ll$  akan menggeser posisi bit ke bit sebelah kiri sebanyak n kali (pada program yang saya buat, saya geser sebanyak 1 bit). Pada soal yaitu 0111 atau 00000111 digeser 1 menjadi 00001110 diambil menjadi 1110 atau didesimalkan menjadi 14.
- $b \gg 1$ , (b right shift sebanyak 1 bit), dimana pada operator  $\gg$  akan menggeser posisi bit ke bit sebelah kanan sebanyak n kali (pada program yang saya buat, saya geser sebanyak 1 bit). Pada soal yaitu 0111 atau 00000111 digeser 1 menjadi 10000011 diambil menjadi 0011 atau didesimalkan menjadi 3.

#### **[Nomor 7] Kesimpulan (Analisa)**

Pada program tersebut saya menggunakan bentuk kelas public agar program tersebut juga dapat diakses oleh publik. Operasi bitwise adalah untuk operasi bit (biner) dan berlaku untuk tipe data int, long, short, char, dan byte, karena akan menghitung dari bit-ke-bit. Dimana operatornya meliputi AND (&), OR (|), XOR (^), NOT (~), Left shift (<<), Right shift (>>), dan Right shift tidak bertanda (>>>). Dengan dikaitkan pada program yang dibuat, setiap operator memiliki fungsi dan cara kerjanya masing-masing. Dan dengan soal, dimana operator << digunakan untuk menggeser bit ke kiri sebanyak n kali dan operator >> untuk menggeser bit ke kanan sebanyak n kali. Serta perhitungan operator bitwise ini berhubungan erat dengan perhitungan bilangan biner, jadi harus dapat memahami perhitungan bilangan biner juga.

#### **REFLEKSI**

Dari praktikum ini saya mendapatkan banyak pengalaman tentang bagaimana cara menggunakan operator aritmatika, penugasan, relasional, increment dan decrement, boolean logika, kondisional, dan bitwise. Pada prosesnya sendiri saya mengalami beberapa kendala karena ada beberapa soal yang sulit dimengerti maksudnya, namun dengan bantuan teman-teman yang lain saya akhirnya dapat menyelesaikan kesulitan saya dengan cukup lancar. Soal-soal yang diberikan disusun agar dapat memahami dan tau cara membuat program berdasarkan materi yang telah diajarkan yaitu tentang operator, serta soalnya cukup menantang untuk dikerjakan dan membuat otak saya harus bekerja lebih keras lagi.