

Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Rana Qonitah Helida G1A022017	Operator	09 September 2022

[No. 1] Identifikasi Masalah:

1) Uraian permasalahan dan variabel

Adapun permasalahan yang terjadi, yaitu adanya pesan luaran yang menyatakan error pada pemrograman disebabkan terdapat kesalahan pada program yang dijalankan. Variabel yang digunakan untuk menyimpan data adalah a dan b. Dimana masing-masing a dan b merupakan tipe data int dan a bernilai 20 sedangkan b bernilai 2.

Soal:

```
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
    } }
```

Pesan kesalahan pada soal :

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

Latihan

- 1.1. Rekomendasikan perbaikan kode agar program Contoh 1 dapat berjalan!
- 1.2. Tambahkan baris untuk menampilkan perhitungan dengan operator (-, *, / , %) pada Contoh 1!

Sumber informasi :

Informasi mengenai penyelesaian masalah ini saya dapatkan melalui channel YouTube "Rumah Belajar Raflesia" dalam materi operator.

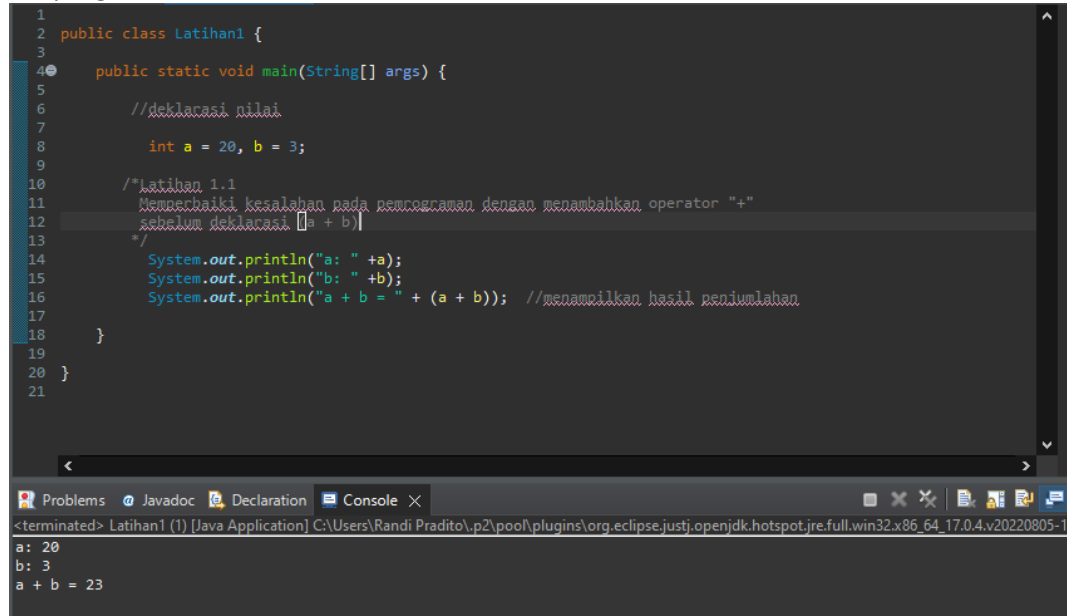
[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan operator tambah yang disimbolkan dengan tanda (+)
- 2) Alasan solusi ini karena jika tidak ditambahkan operator "+" yang digunakan untuk pemanggilan (penambahan) suatu variabel, maka program tidak dapat membaca atau menyelesaikan pemrograman
- 3) Perbaikan kode program dengan cara menambahkan operator tambah "+" setelah tanda kutip dua sebelum (a + b) berfungsi untuk melakukan pemanggilan (penambahan) terhadap variabel (a + b).

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mendeklarasikan data nilai, dalam hal ini

- ```
int a = 20, b = 3;
```
- (b) Melakukan operasi dengan operator yang diinginkan pada soal
  - (c) Melakukan pemanggilan variabel dengan menggunakan tanda "+" sebelum variabel yang akan di print out.
  - (d) Melakukan print.out data untuk menghasilkan luaran.
  - (e) Pada latihan 1.2 nantinya, menambahkan baris di bawahnya untuk melakukan operasi dengan operator aritmatika lainnya, yaitu (-), (\*), (/), dan (%).
- 2) Kode program dan luaran :



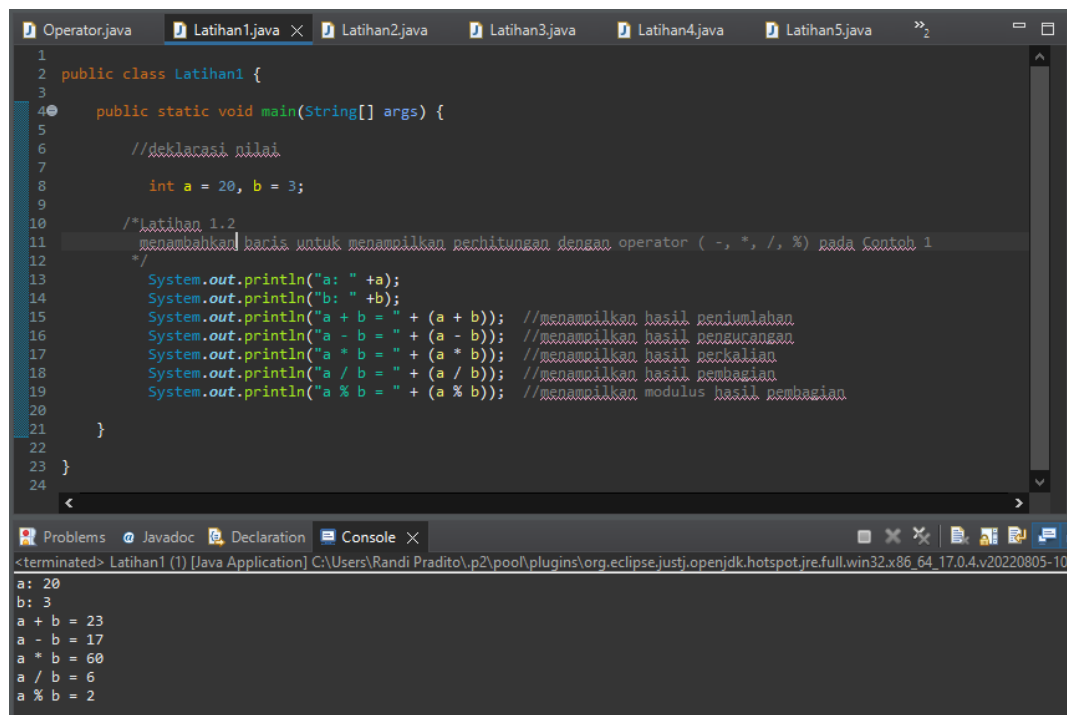
```
1
2 public class Latihan1 {
3
4 public static void main(String[] args) {
5
6 //deklarasasi nilai
7
8 int a = 20, b = 3;
9
10 /*Latihan 1.1
11 Memeriksa kesalahan pada pemrograman dengan menambahkan operator "+"
12 sebelum deklarasasi [a + b]
13 */
14 System.out.println("a: " +a);
15 System.out.println("b: " +b);
16 System.out.println("a + b = " + (a + b)); //menampilkan hasil penjumlahan
17
18 }
19
20 }
21
```

Problems Javadoc Declaration Console

<terminated> Latihan1 (1) [Java Application] C:\Users\Randi Pradito\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.4.v20220805-1

a: 20  
b: 3  
a + b = 23

Kode Latihan 1.1 (perbaikan kode)



```
1
2 public class Latihan1 {
3
4 public static void main(String[] args) {
5
6 //deklarasasi nilai
7
8 int a = 20, b = 3;
9
10 /*Latihan 1.2
11 menambahkan baris untuk menampilkan perhitungan dengan operator (-, *, /, %) pada Contoh 1
12 */
13 System.out.println("a: " +a);
14 System.out.println("b: " +b);
15 System.out.println("a + b = " + (a + b)); //menampilkan hasil penjumlahan
16 System.out.println("a - b = " + (a - b)); //menampilkan hasil pengurangan
17 System.out.println("a * b = " + (a * b)); //menampilkan hasil perkalian
18 System.out.println("a / b = " + (a / b)); //menampilkan hasil pembagian
19 System.out.println("a % b = " + (a % b)); //menampilkan modulus hasil pembagian
20
21 }
22
23 }
24
```

Operator.java Latihan1.java Latihan2.java Latihan3.java Latihan4.java Latihan5.java

Problems Javadoc Declaration Console

<terminated> Latihan1 (1) [Java Application] C:\Users\Randi Pradito\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.4.v20220805-10

a: 20  
b: 3  
a + b = 23  
a - b = 17  
a \* b = 60  
a / b = 6  
a % b = 2

Kode latihan 1.2

- a) Pada gambar pertama (latihan 1.1), kode telah diperbaiki, sehingga tidak ada lagi kesalahan di dalamnya dan hasil luaran pun tidak error. Lalu, pada gambar kedua (latihan

- 1.2) diminta untuk menambahkan baris lagi untuk operasi dengan operator aritmatika lainnya, yaitu (-), (\*), (/), dan (%) seperti yang terlihat di atas.
- b) Berdasarkan luaran kedua program di atas, hasilnya sudah sesuai dengan operasi perhitungan yang diinginkan sesuai operatornya dan tidak ada lagi error pada program pertama setelah diperbaiki.

#### [No.1] Kesimpulan

##### Analisa

- Pada latihan 1.1, terdapat kesalahan atau error pada luaran. Setelah dianalisa, ternyata pada program tersebut, saat menuliskan print out, tidak terdapat operator (+) yang menunjukkan pemanggilan terhadap variabel yang diinginkan. Sehingga terjadilah error ketika program di-run. Maka dari itu, penyelesaian dari masalah di atas dapat dilakukan dengan menambahkan operator (+) yang berfungsi untuk melakukan pemanggilan terhadap variabel yang diinginkan. Dalam hal ini variabelnya adalah (a + b) atau menunjukkan hasil operasi dari (a + b).
  - Pada latihan 1.2, diperintahkan untuk menambah baris untuk menampilkan operasi pengurangan (-), perkalian (\*), pembagian (/), dan modulus pembagian (%). Dimana dalam kasus ini telah saya tambahkan mengikuti prosedur yang sama seperti pada latihan 1.1, sehingga menghasilkan luaran yang sesuai.
- 

#### [No. 2] Identifikasi Masalah

```
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

##### Latihan Soal

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!

##### Sumber informasi :

Informasi mengenai penyelesaian masalah ini saya dapatkan melalui channel YouTube "Rumah Belajar Raflesia" dalam materi operator.

#### [No.2] Analisis dan Argumentasi

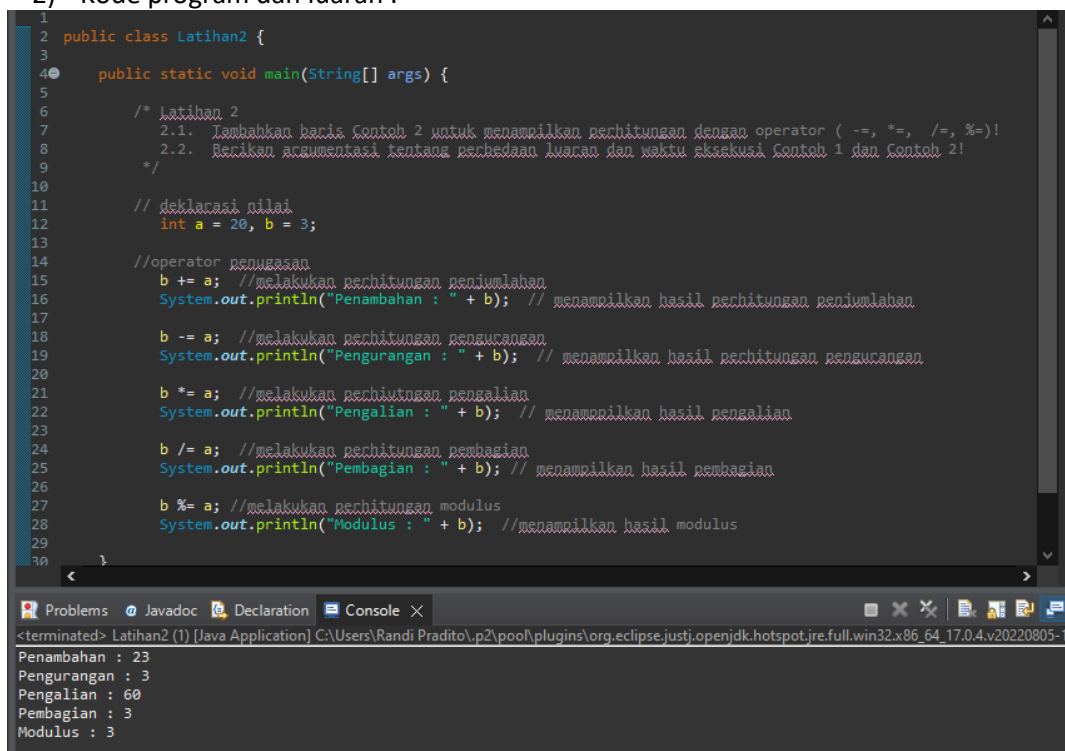
Dalam kasus ini, diminta untuk menampilkan perhitungan dengan operator (-=, \*=, /=, %=) berdasarkan data yang ada. Nantinya, luaran yang dihasilkan pada program ini akan berbeda

dengan luaran pada contoh 1 (latihan 1.2). Hal ini dikarenakan operator yang digunakan berbeda. Pada contoh 1 program dijalankan dengan operator aritmatika saja tanpa penugasan (gabungan), sedangkan pada contoh 2 menggunakan operator penugasan gabungan yang bersifat berantai.

## [No. 2] Penyusunan algoritma dan kode pemrograman

- 1) Algoritma latihan 2.1 :
  - a) Mendeklarasikan data nilai, dalam hal ini  
`int a = 20, b = 3;`
  - b) Melakukan operasi dengan operator penugasan gabungan yang diinginkan pada soal, yaitu (`-=`, `*=`, `/=`, `%=`).
  - c) Melakukan print.out data untuk menghasilkan luaran.

- 2) Kode program dan luaran :



```
1 public class Latihan2 {
2
3
4 public static void main(String[] args) {
5
6 /* latihan 2
7 2.1. Tambahkan baris contoh 2 untuk menampilkan perhitungan dengan operator (-=, *=, /=, %=)!
8 2.2. Berikan argumentasi tentang perbedaan luaran dan waktu eksekusi contoh 1 dan contoh 2!
9 */
10
11 // deklarasi nilai
12 int a = 20, b = 3;
13
14 //operator penugasan
15 b += a; //melakukan perhitungan penjumlahan
16 System.out.println("Penambahan : " + b); // menampilkan hasil perhitungan penjumlahan
17
18 b -= a; //melakukan perhitungan pengurangan
19 System.out.println("Pengurangan : " + b); // menampilkan hasil perhitungan pengurangan
20
21 b *= a; //melakukan perhitungan pengalian
22 System.out.println("Pengalian : " + b); // menampilkan hasil pengalian
23
24 b /= a; //melakukan perhitungan pembagian
25 System.out.println("Pembagian : " + b); // menampilkan hasil pembagian
26
27 b %= a; //melakukan perhitungan modulus
28 System.out.println("Modulus : " + b); //menampilkan hasil modulus
29
30 }
31 }
```

Penambahan : 23  
Pengurangan : 3  
Pengalian : 60  
Pembagian : 3  
Modulus : 3

Kode Latihan 2.1

- a) Pada program di atas, telah ditambahkan baris selanjutnya untuk mengoperasikan data dengan operator penugasan gabungan lainnya, yaitu (`-=`, `*=`, `/=`, dan `%=`).
- b) Setelah melakukan penambahan operasi yang diperintahkan, terbentuklah luaran seperti yang terdapat pada gambar di atas. Dapat kita lihat, data yang ditampilkan dalam pemrograman ini sama dengan data yang ada pada contoh 1 di latihan 1.1 dan 1.2, yaitu :  
`int a = 20, int b = 3;`  
Namun, pada saat melihat luarannya, hasil yang ditampilkan akan berbeda. Dimana hal ini akan menjawab pertanyaan 2.1.  
Pada dasarnya, kode penugasan pada contoh 1 dan 2 memang berbeda. Pada contoh 1, penugasan yang dilakukan merupakan penugasan biasa, dimana hasil dari penugasan yang dilakukan langsung dikeluarkan (tidak terpengaruh dengan data terakhir) seperti yang dapat dilihat pada gambar "Kode Latihan 1.2". Sedangkan pada latihan 2.1, operasi yang digunakan

adalah **operasi penugasan gabungan (bergabung dengan operator aritmatika)**, dimana dalam operasi ini data yang digunakan (data yang akan terpanggil) adalah data terbaru berdasarkan urutan penugasannya atau dengan kata lain memberikan nilai ke dalam variabel tertentu. Hal ini dapat dilihat pada gambar “Kode Latihan 2.1” sebagai contoh, dapat dilihat dari pemrograman berikut :

```
b += a; //melakukan perhitungan penjumlahan
System.out.println("Penambahan : " + b); // menampilkan hasil perhitungan
penjumlahan dari b + a = 23
```

```
b -= a; //melakukan perhitungan pengurangan
System.out.println("Pengurangan : " + b); // menampilkan hasil perhitungan
pengurangan b - a = 3
```

```
/* perhitungan b - a = 3 bisa terjadi karena pada operasi pertama, telah
didapatkan angka baru dari b, yaitu 23 berdasarkan operasi (b + a), sehingga
selanjutnya didapatlah (b - a) = (23 - 3) = 3. Maka untuk operasi setelahnya,
akan selalu berubah mengikuti hasil operasi terakhir.
*/
```

## [No.2] Kesimpulan

### Analisa

Berdasarkan latihan yang telah dikerjakan, ternyata pada operasi contoh 2 ini menggunakan operasi penugasan gabungan yang berbeda dengan operasi penugasan biasa seperti pada contoh 1 di latihan 1.1 dan 1.2. Pada operasi penugasan gabungan ini, akan mengoperasikan hasil operasi terakhir berdasarkan urutan pengerjaannya/penulisannya (dari atas ke bawah). Berbeda dengan penugasan biasa yang angkanya tidak berubah mengikuti hasil operasi terakhir, melainkan tetap berpacu pada data awal yang telah disediakan.

---

## [No. 3] Identifikasi Masalah

```
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

### Latihan Soal

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

#### Sumber informasi :

Informasi mengenai penyelesaian masalah ini saya dapatkan melalui channel YouTube “Rumah Belajar Raflesia” dalam materi operator.

### [No.3] Analisis dan Argumentasi

Dalam kasus ini, diminta untuk mengubah nilai A = 4 dan B = 4 serta menyimpulkan perubahan apa yang terjadi. Seperti yang kita ketahui, pada pemrograman di atas, menggunakan operator relasional untuk membandingkan dua variabel yang akan menghasilkan luaran berupa data boolean **true or false**. Berdasarkan sifat dari operator ini, maka data yang ada sangat mempengaruhi luaran yang dihasilkan. Sebab, operator akan membandingkan data yang ada agar bisa diketahui apakah hasilnya akan bernilai true atau false.

### [No. 3] Penyusunan algoritma dan kode pemrograman

1) Algoritma latihan 3.1 :

- a) Mendeklarasikan data nilai, dalam hal ini  
`int nilai A = 4,    int nilai B = 4;`  
`boolean hasil ;`
- b) Melakukan operasi dengan operator yang diinginkan pada soal, yaitu operator relasional (`>, <, >=, <=, ==`).
- c) Melakukan `print.out` data untuk menghasilkan luaran.

2) Kode Program dan Luaran :

```
2 public class Latihan3 {
3
4 public static void main(String[] args) {
5
6 /* Latihan 3
7 3.1. Ubahlah nilai A = 4 dan B = 4 pada Contoh 3. Simulkan perubahan yang terjadi!
8 */
9
10 int nilaiA = 4;
11 int nilaiB = 4;
12 boolean hasil;
13
14 System.out.println(" A = " + nilaiA + "\n B = " + nilaiB);
15 // apakah A lebih besar dari B?
16 hasil = nilaiA > nilaiB;
17 System.out.println("\n Hasil A > B = " + hasil);
18
19 // apakah A lebih kecil dari B?
20 hasil = nilaiA < nilaiB;
21 System.out.println("\n Hasil A < B = " + hasil);
22
23 // apakah A lebih besar samadengan B?
24 hasil = nilaiA >= nilaiB;
25 System.out.println("\n Hasil A >= B = " + hasil);
26
27 // apakah A lebih kecil samadengan B?
28 hasil = nilaiA <= nilaiB;
29 System.out.println("\n Hasil A <= B = " + hasil);
30
31 // apakah nilai A sama dengan B?
32 hasil = nilaiA == nilaiB;
33 System.out.println("\n Hasil A == B = " + hasil);
34
35 // apakah nilai A tidak samadengan B?
36 hasil = nilaiA != nilaiB;
37 System.out.println("\n Hasil A != B = " + hasil);
38
39 }
40
41 }
42 }
```

Problems Javadoc Declaration Console

<terminated> Latihan3 (1) [Java Application] C:\Users\Randi Pradito\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.4.v20220809-1047\jre\bin\javaw.exe (Sep 8, 2022)

```
A = 4
B = 4

Hasil A > B = false
Hasil A < B = false
Hasil A >= B = true
Hasil A <= B = true
Hasil A == B = true
Hasil A != B = false
```

Kode Latihan 3.1

- a) Pada pemrogramana tersebut, datanya telah diganti dari data awal, yaitu A = 12 dan B = 4, menjadi A = 4 dan B = 4. Sebelum nilai dari A diubah, maka luaran yang dihasilkan adalah
- 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

Namun, setelah dilakukan perubahan nilai A, maka luarannya akan menjadi :

A = 4  
B = 4

Hasil A > B = false  
Hasil A < B = false  
Hasil A >= B = true  
Hasil A <= B = true

```
Hasil A == B = true
Hasil A != B = false
```

- b) Hal ini dapat terjadi karena hasil operasi relasional akan memberikan hasil yang sesuai terhadap nilai yang dibandingkan. Sebagai contoh, pada saat nilai A = 12 dan B = 4, lalu dioperasikan dengan operator ">" (lebih besar), sehingga saat akan membandingkan nilai A dan B (A > B), maka luarannya akan bernilai **true**, karena memang benar bahwasanya 12 lebih besar daripada 4. Begitu pula jika nilai pada variabel A diubah menjadi 4. Maka saat membandingkan nilai tersebut (A > B), akan menghasilkan luaran **false**, karena nilai A dan B sama atau A = B, bukan A > B ataupun A < B. Hal ini berlaku untuk perbandingan yang lainnya.

### [No.3] Kesimpulan

#### Analisa

Pada pemrograman tersebut, kita diminta untuk mengganti nilai variabel A menjadi 4 sehingga sama dengan nilai variabel B. Dalam hal ini, operasi yang digunakan adalah operasi relasional yang digunakan untuk membandingkan dua variabel yang hasilnya akan berbentuk boolean. Jika kita lihat pada soal, luaran yang dihasilkan data awal A = 12 dan B = 4 berbeda dengan hasil luaran dari data A = 4 dan B = 4. Hal ini dikarenakan nilai yang dibandingkan telah berubah yang akhirnya menyebabkan logika pun berubah sehingga hasil antara keduanya tidak sama.

---

### [No. 4] Identifikasi Masalah

```
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
```

#### Latihan Soal

- 4.1. Berikan saran operasi apa yang diperlukan (pre/post increment, pre/post increment) agar Contoh 4 menghasilkan nilai a = 5 dan b = 6?
- 4.2. Simpulkan hasil eksperimen Anda!

#### Sumber informasi :

Informasi mengenai penyelesaian masalah ini saya dapatkan melalui channel YouTube "Rumah Belajar Raflesia" dalam materi operator.

### [No. 4] Analisis dan argumentasi

Dari kode di atas, kita ketahui hasil operasi menggunakan operator post-increment (a++) akan menghasilkan luaran nilai 5. Lalu, pada soal 4.1 meminta agar program tersebut menghasilkan nilai a = 5 dan b = 6. Untuk mewujudkan hal tersebut, dapat kita lakukan dengan dua cara, yang pertama, mengubah operator menjadi **pre-increment** dan yang kedua dengan tetap menggunakan **post-increment**, tetapi melakukan pemanggilan lagi setelahnya. Hal tersebut dikarenakan adanya perbedaan konsep pada operator post-increment dan pre-increment



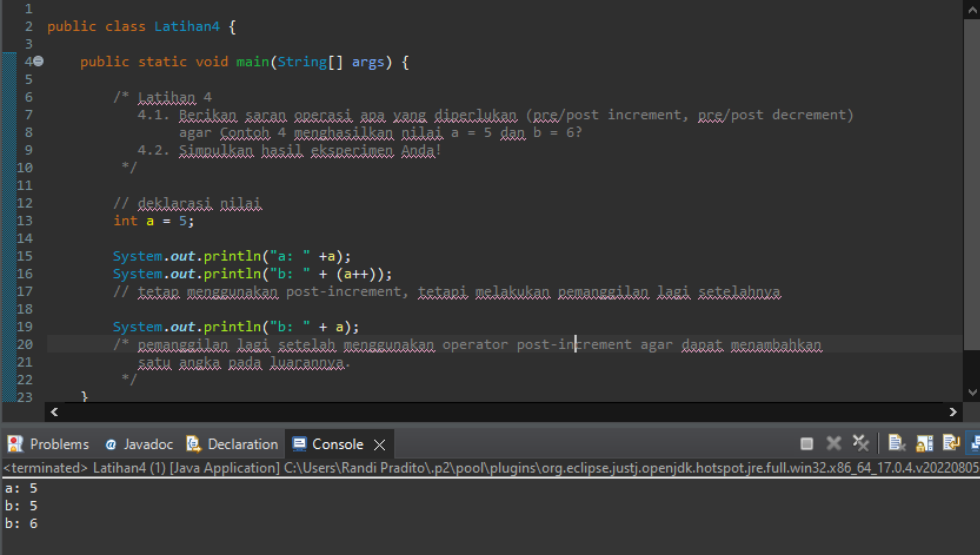
meskipun sama-sama berfungsi menambahkan sebanyak 1 angka. Pada operator post increment, penambahan dilakukan setelah pemanggilan setelah penambahan tersebut. Sedangkan pada operator pre-increment, penambahan angka langsung dapat dihasilkan dari operasi yang telah dituliskan.

#### [No. 4] Penyusunan algoritma dan kode pemrograman

1) Algoritma latihan 4.1 :

- (a) Melakukan deklarasi nilai, dalam hal ini :  
`int a = 5;`
- (b) Menganalisa data agar sesuai dengan perintah yang diberikan
- (c) Menyusun kode yang tepat berdasarkan hasil analisis yang telah dilakukan untuk menghasilkan luaran yang diinginkan (menggunakan pre-increment atau post-increment).

2) Kode Program dan Luaran :



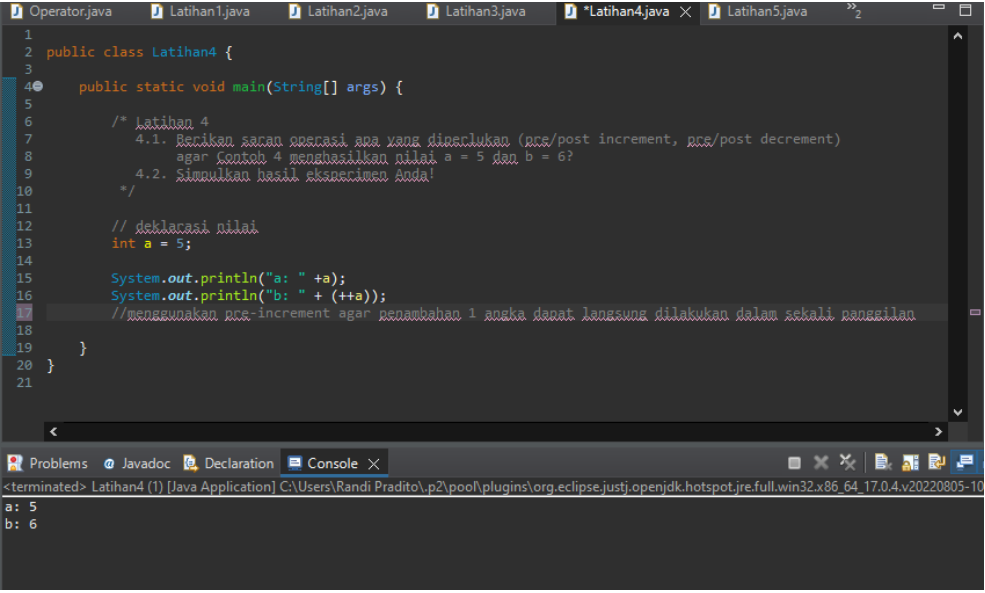
```
1
2 public class Latihan4 {
3
4 public static void main(String[] args) {
5
6 /* latihan 4
7 4.1. Berikan saran operasi apa yang diperlukan (pre/post increment, pre/post decrement)
8 agar contoh 4 menghasilkan nilai a = 5 dan b = 6?
9 4.2. Simpulkan hasil eksekusi Anda!
10 */
11
12 // deklarasi nilai
13 int a = 5;
14
15 System.out.println("a: " + a);
16 System.out.println("b: " + (a++));
17 // tetap menggunakan post-increment, tetapi melakukan pemanggilan lagi setelahnya
18
19 System.out.println("b: " + a);
20 /* pemanggilan lagi setelah menggunakan operator post-increment agar dapat menambahkan
21 satu angka pada luarnya.
22 */
23 }
```

Problems Javadoc Declaration Console X

<terminated> Latihan4 (1) [Java Application] C:\Users\Randi Pradito\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.4.v20220805-10

a: 5  
b: 5  
b: 6

Kode Latihan 4.1.1 (Menggunakan operator post-increment)



```
1
2 public class Latihan4 {
3
4 public static void main(String[] args) {
5
6 /* latihan 4
7 4.1. Berikan saran operasi apa yang diperlukan (pre/post increment, pre/post decrement)
8 agar contoh 4 menghasilkan nilai a = 5 dan b = 6?
9 4.2. Simpulkan hasil eksekusi Anda!
10 */
11
12 // deklarasi nilai
13 int a = 5;
14
15 System.out.println("a: " + a);
16 System.out.println("b: " + (++a));
17 // menggunakan pre-increment agar penambahan 1 angka dapat langsung dilakukan dalam sekali pemanggilan
18
19 }
20 }
21
```

Operator.java Latihan1.java Latihan2.java Latihan3.java \*Latihan4.java X Latihan5.java

Problems Javadoc Declaration Console X

<terminated> Latihan4 (1) [Java Application] C:\Users\Randi Pradito\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.4.v20220805-10

a: 5  
b: 6

Kode Latihan 4.1.2 (Menggunakan operator pre-increment)

- a) Pada operasi latihan 4.1.1, saya menggunakan operator post-increment untuk menambahkan nilai dari variabel a, sehingga mendapatkan angka 6 dengan melakukan pemanggilan lagi setelah ditugaskan menggunakan operator tersebut.  
Pada operasi latihan 4.1.2, saya menggunakan operator pre-increment untuk menambahkan nilai dari variabel a, sehingga langsung mendapatkan angka 6 dalam sekali deklarasi.
- b) Kedua program di atas telah sesuai dengan perintah yang diberikan serta menghasilkan luaran yang diinginkan.

#### [No.4] Kesimpulan

##### **Analisa**

Pada program di atas, kita bisa menggunakan kedua tipe operator increment, yaitu pre dan post increment. Namun, cara kerja kedua operator tersebut sedikit berbeda. Jika kita ingin melakukan penambahan angka dengan sekali pemanggilan, hal tersebut bisa dilakukan dengan menggunakan operator pre-increment. Sebaliknya, jika kita ingin menggunakan operator post-increment, maka kita harus melakukan pemanggilan lagi setelahnya seperti yang dapat dilihat pada kode latihan di atas.

---

#### [No. 5] Identifikasi Masalah:

```
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

##### **Latihan Soal**

- 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.

##### **Sumber informasi :**

Informasi mengenai penyelesaian masalah ini saya dapatkan melalui channel YouTube "Rumah Belajar Raflesia" dalam materi operator.

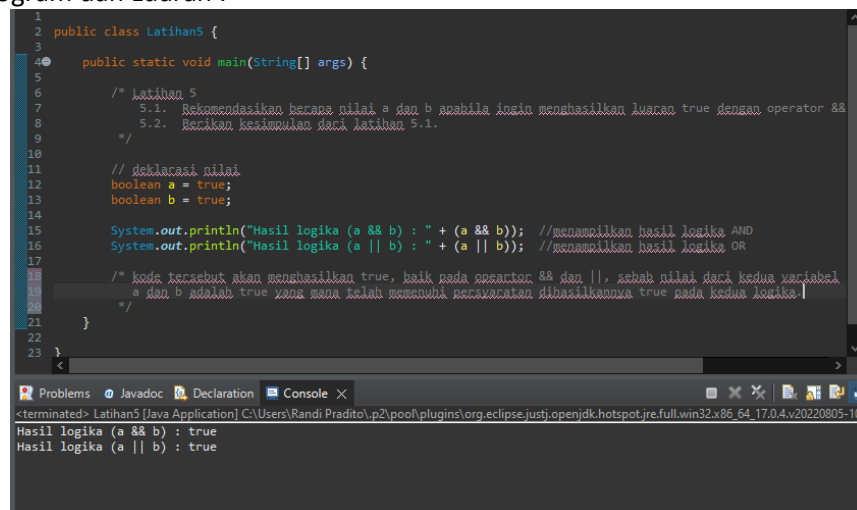
#### [No. 5] Analisis dan argumentasi :

Dari kode di atas, kita dapat mengetahui bahwa dari hasil operasi menggunakan operator logika true or false, akan dihasilkan luaran false berdasarkan nilai boolean a = true dan b = false. Lalu, pada latihan 5.1 kita diminta untuk merekomendasikan berapa nilai a dan b apabila ingin menghasilkan luaran true dengan operator && (and) dan operator || (or). Perlu diketahui bahwa operator logika && dan || memiliki arti yang berbeda. Operator && hanya

akan menghasilkan kondisi true apabila kedua pernyataan (variabel) yang dibandingkan bernilai true, dan operator `||` akan menghasilkan kondisi true apabila salah satu atau kedua pernyataan (variabel) bernilai true. Maka dari itu, untuk menghasilkan nilai true pada kedua operator `&&` dan `||` dalam operasi di atas, kita perlu mengganti nilai b menjadi true. Dengan demikian, kedua operator akan menghasilkan nilai true.

**[No. 5] Penyusunan algoritma dan kode pemrograman :**

- 1) Algoritma latihan 5.1 :
  - a) Mendeklarasikan nilai boolean :  
`boolean a = true, boolean b = true;`
  - b) Menganalisa data agar sesuai dengan perintah yang diberikan, dalam hal ini mengubah nilai boolean b yang sebelumnya false menjadi true.
  - c) Menyusun kode yang tepat berdasarkan hasil analisis yang telah dilakukan untuk menghasilkan luaran yang diinginkan.
  - d) Melakukan print.out untuk menghasilkan data yang telah diprogram.
- 2) Kode Program dan Luaran :



```
1 public class Latihan5 {
2
3 public static void main(String[] args) {
4
5 /* latihan 5
6 5.1. Rekomendasikan bahwa nilai a dan b apakah login menghasilkan luaran true dengan operator &&
7 5.2. Revisi hasil latihan 5.1.
8 */
9
10 // deklarasi nilai
11 boolean a = true;
12 boolean b = true;
13
14 System.out.println("Hasil logika (a && b) : " + (a && b)); //menampilkan hasil logika AND
15 System.out.println("Hasil logika (a || b) : " + (a || b)); //menampilkan hasil logika OR
16
17 /* kode tersebut akan menghasilkan true, baik pada operator && dan ||, sebab nilai dari kedua variabel
18 a dan b adalah true yang mana telah memenuhi persyaratan dihasilkannya true pada kedua logika.
19 */
20 }
21 }
22
23
```

Problems Javadoc Declaration Console X

<terminated> Latihan5 [Java Application] C:\Users\Randi Pradito\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64.17.0.4.v20220805-104

Hasil logika (a && b) : true  
Hasil logika (a || b) : true

**Kode Latihan 5.1**

- a) Pada pemrograman di atas, menampilkan operasi dengan operator logika, yaitu `&&` (and) dan `||` (or). Kedua operator ini memiliki syarat yang berbeda dalam menghasilkan luaran true. Untuk operator `&&` (and) akan menghasilkan true apabila kedua variabel yang dibandingkan bernilai true dan untuk operator `||` (or) akan bernilai true apabila salah satu saja atau kedua variabel mempunyai nilai true. Sehingga, untuk menghasilkan luaran true pada kedua operator, maka nilai variabel a harus diubah menjadi true seperti pada gambar di atas.
- b) Hasil luaran pada program di atas sudah sesuai dengan apa yang diperintahkan, yaitu menghasilkan nilai true pada kedua operator `&&` (and) dan `||` (or).

**[No. 5] Kesimpulan :**

**Analisis**

Dalam program dengan menggunakan operator logika, kita perlu memperhatikan syarat luaran yang akan dihasilkan agar sesuai dengan apa yang kita inginkan. Seperti latihan di atas yang menginginkan hasil operasi dari kedua operator bernilai true, maka kita perlu mengganti data yang bernilai false menjadi true, agar bisa menghasilkan luaran true pula pada operator `&&` (and).

#### [No. 6] Identifikasi Masalah:

```
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

#### Latihan Soal

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

#### Sumber informasi :

Informasi mengenai penyelesaian masalah ini saya dapatkan melalui channel YouTube “Rumah Belajar Raflesia” dalam materi operator.

#### [No. 6] Analisis dan argumentasi :

Pada pemrograman di atas, menggunakan operator kondisional untuk menjalankan operasinya. Operator kondisional ini sendiri sama dengan operator ternary yang memiliki 3 operand dimana dalam kasus di atas adalah **nilai**, serta deklarasi **“lulus”** atau **“gagal”**. Untuk menghasilkan luaran dengan nilai = 60 memenuhi untuk lulus, maka yang perlu dilakukan adalah menambahkan operator (=) pada status. Hal ini dikarenakan operator kondisional > (lebih dari) sama dengan sifat aritmatika yang menyatakan lebih dari nilai yang tersedia. Sehingga apabila ingin menyertakan nilai 60 (sama dengan nilai yang dicantumkan) perlu menambahkan operator (=) yang berarti **lebih besar** atau **sama dengan 60**. Dengan demikian, program akan menghasilkan luaran “lulus” untuk nilai = 60.

#### [No. 6] Penyusunan algoritma dan kode pemrograman :

##### 1) Algoritma latihan 6 :

###### a) Mendeklarasikan variabel beserta nilai berikut :

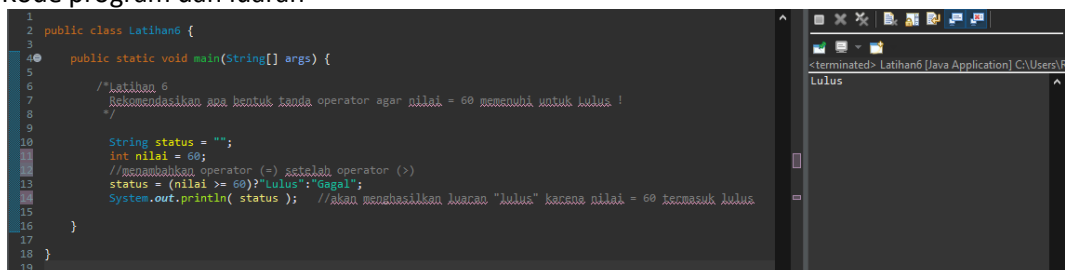
```
String status = "";
int nilai = 80;
status = (nilai > 60)?"Lulus":"Gagal";
```

###### b) Menganalisa data agar sesuai dengan perintah yang diberikan, dalam hal ini bagaimana cara agar nilai = 60 dapat lulus.

###### c) Menyusun kode yang tepat berdasarkan hasil analisis yang telah dilakukan untuk menghasilkan luaran yang diinginkan.

###### d) Melakukan print.out untuk menghasilkan data yang telah diprogram.

##### 2) Kode program dan luaran



```
1 public class Latihan6 {
2
3
4 public static void main(String[] args) {
5
6 /*Latihan 6
7 Rekomendasikan apa bentuk tanda operator agar nilai = 60 memenuhi untuk lulus !
8 */
9
10 String status = "";
11 int nilai = 60;
12 //menambahkan operator (=) setelah operator (>)
13 status = (nilai >= 60)?"Lulus":"Gagal";
14 System.out.println(status); //akan menghasilkan luasan "lulus" karena nilai = 60 termasuk lulus
15
16 }
17 }
18 }
19 }
```

Output: Lulus

Kode Latihan 6

- a) Program di atas menggunakan operator relasional yang dirangkap menjadi operator kondisional untuk mendapatkan luaran suatu kondisi berdasarkan data yang tersedia.
- b) Pada pemrograman di atas, operator (>) sudah ditambahkan dengan operator (=), sehingga dihasilkan luaran yang sesuai dengan perintah, yaitu nilai = 60 "lulus".

#### [No. 6] Kesimpulan :

##### Analisis

Pada program di atas, hasil yang diinginkan adalah status "lulus" untuk nilai = 60. Untuk melakukan hal tersebut, saya menggunakan operator kondisional yang menyatakan 3 operand, yaitu **nilai**, **"lulus"**, dan **"gagal"**, serta operator relasional untuk membandingkan nilai yang tersedia (dalam hal ini memilih operator yang tepat) agar sesuai dengan perintah soal. Adapun operator yang tepat adalah (>=), karena ia akan mengikutsertakan nilai yang tersedia", yaitu 60 untuk mendapatkan pernyataan "lulus". Dimana hal ini sama dengan operasi aritmatika "lebih dari atau sama dengan" terhadap suatu nilai.

#### [No. 7] Identifikasi Masalah:

```
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 : 6
Hasil dari a | b : 7
Hasil dari a ^ b : 1
Hasil dari ~a : -11
Hasil dari a >> 1 : 3
Hasil dari b << 2 : 28
```

## Latihan Soal

Evaluasi penyebab hasil  $\sim a = -11$  ? Buktikan jawaban Anda dalam perhitungan biner!

### Sumber informasi :

Informasi mengenai penyelesaian masalah ini saya dapatkan melalui channel YouTube "Rumah Belajar Raflesia" dalam materi operator.

#### [No. 7] Analisis dan argumentasi :

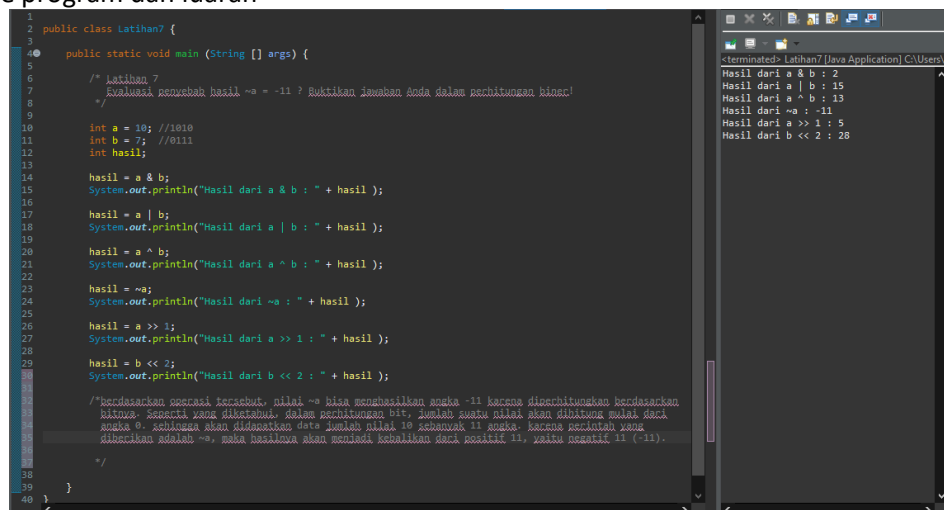
Dalam pemrograman di atas, menggunakan operator bitwise untuk melakukan operasi secara langsung terhadap bit suatu bilangan. Sehingga, luaran yang dihasilkan akan berdasarkan bit bilangan tersebut. Seperti yang dapat dilihat pada contoh di atas, terdapat nilai `int a = 10` dan `int b = 7`. Dimana nilai bit dari bilangan 10 adalah 1010 dan nilai bit dari bilangan 7 adalah 0111. Maka berdasarkan operasi bitwise, jika dioperasikan dengan operator `&`, akan diperhitungkan sebagai berikut :

Misal : berapa hasil (`a & b`), maka operasinya akan melihat di antara bit 10 dan bit 7, yaitu 1010 dan 0111. Bila ada angka 1 (bernilai true) yang sejajar posisinya sehingga keduanya bernilai true, maka itulah hasil dari operasi `&` pada operasi bitwise. Perlu diperhatikan pula posisi angkanya, sebab untuk menentukan hasilnya, kita membutuhkan perhitungan berdasarkan posisi bit tersebut. Hal inilah yang menjadi landasan dari perhitungan dengan operator bitwise. Namun, untuk penyebab hasil dari  $\sim a = -11$  sedikit berbeda. Hal ini dikarenakan perhitungannya tidak melihat bit nilai `a`, melainkan sistem perhitungan dalam biner pada suatu nilai. Yang dimaksud di sini adalah dalam perhitungan biner, selalu dimulai dari angka 0. Sehingga, jumlah angkanya akan menjadi nilai + 1. Dalam kasus ini, karena yang diminta adalah  $\sim a$ , maka akan menghasilkan nilai -11 sebab nilai 10 berjumlah 11 angka jika dihitung berdasarkan perhitungan biner.

#### [No. 7] Penyusunan algoritma dan kode pemrograman :

- 1) Algoritma latihan 6 :
  - a) Mendeklarasikan variabel beserta nilai berikut :

```
int a = 10;
int b = 7;
int hasil;
```
  - b) Menyusun kode yang tepat berdasarkan hasil analisis yang telah dilakukan untuk menghasilkan luaran yang diinginkan.
  - c) Melakukan `print.out` untuk menghasilkan data yang telah diprogram.
  - d) Mengevaluasi hasil dari perhitungan
- 2) Kode program dan luaran



```
1 public class Latihan7 {
2
3 public static void main (String [] args) {
4
5 /* Latihan 7
6 * Evaluasi penyebab hasil ~a = -11 ? Buktikan jawaban Anda dalam perhitungan biner!
7 */
8
9 int a = 10; //1010
10 int b = 7; //0111
11 int hasil;
12
13 hasil = a & b;
14 System.out.println("Hasil dari a & b : " + hasil);
15
16 hasil = a | b;
17 System.out.println("Hasil dari a | b : " + hasil);
18
19 hasil = a ^ b;
20 System.out.println("Hasil dari a ^ b : " + hasil);
21
22 hasil = ~a;
23 System.out.println("Hasil dari ~a : " + hasil);
24
25 hasil = a >> 1;
26 System.out.println("Hasil dari a >> 1 : " + hasil);
27
28 hasil = b << 2;
29 System.out.println("Hasil dari b << 2 : " + hasil);
30
31 /* berdasarkan operasi tersebut, nilai ~a bisa menghasilkan angka -11 karena diperhitungkan berdasarkan
32 * bitnya. Seperti yang diketahui, dalam perhitungan bit, jumlah suatu nilai akan dikurangkan mulai dari
33 * angka 0 sehingga akan didapatkan data jumlah nilai 10 kemudian 11. Untuk kasus tersebut yang
34 * diberikan adalah ~a, maka hasilnya akan menjadi kebalikan dari positif 11, yaitu negatif 11 (-11).
35 */
36
37 }
38
39 }
40 }
```

Output:

```
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
```

- a) Pada program di atas menampilkan operasi dengan operator bitwise, yaitu & (and), | (or), ^ (xor), ~ (not), >> (geser kanan), dan << (geser kiri).
- b) Luaran pada program di atas sudah sesuai dengan operasinya dan sesuai dengan kebutuhan yang diinginkan.

## [No.7] Kesimpulan

### Analisa

Pada program yang telah dibuat di atas dengan menggunakan operator bitwise, nilai yang dihasilkan akan berpaku pada bit dalam suatu bilangan, sehingga hasilnya tampak seperti gambar di atas. Dimana dalam operasi biner ini perlu kesejajaran letak angka untuk menghasilkan luaran yang tepat. Perlu diingat bahwa angka 0 bernilai false dan angka 1 bernilai true. Maka dalam perhitungan logika biner seperti & , | , ^ , dan lainnya sangat berpengaruh pada urutan bit suatu angka. Lalu seperti yang kita ketahui, sistem bilangan biner adalah sistem bilangan basis 2, dimana pada sistem ini setiap nilai (mulai dari kanan ke kiri) akan dihitung menjadi  $2^0$ ,  $2^1$ ,  $2^2$ , dan seterusnya. Kaitannya dengan soal di atas adalah ketika kita mengoperasikan biner 10 dan 7 pada operator bitwise, sebagai contoh menentukan (a & b), maka gambarannya adalah sebagai berikut :

bit 10 = 1010

bit 7 = 0111

Dikarenakan operator & hanya bisa bernilai true jika kedua data bernilai true, maka kita hanya perlu melihat angka 1 yang saling sejajar. pada nilai di atas, angka 1 yang sejajar terletak pada posisi kedua, yang bernilai  $2^1$  yang artinya hasil/luaran dari (a & b) adalah 2.

Untuk penyebab hasil dari ~a = -11 sedikit berbeda. Seperti yang telah disebutkan sebelumnya, hal ini dikarenakan perhitungannya tidak melihat bit nilai a, melainkan sistem perhitungan dalam biner pada suatu nilai. Yang dimaksud di sini adalah dalam perhitungan biner, selalu dimulai dari angka 0. Sehingga, jumlah angkanya akan menjadi (nilai + 1). Dalam kasus ini, karena yang diminta adalah ~a, maka akan menghasilkan nilai -11 sebab nilai 10 berjumlah 11 angka jika dihitung berdasarkan perhitungan biner.

### Refleksi

Selama belajar mengenai sistem operator dalam pemrograman ini, wawasan saya jadi semakin bertambah. Dari yang dulunya tidak terlalu mengerti mengenai apa itu operator, jenis-jenisnya, bagaimana fungsi dan cara mengoperasikannya, akhirnya saya mengerti. Tentu saja terdapat tantangan selama proses belajar ini, karena saya sedikit kesulitan memahami mengenai operator bitwise dan bagaimana cara mengoperasikannya. Mulai dari bilangan binernya, perhitungannya, dan hal-hal detail lainnya mengenai operasi ini. Namun, setelah belajar dengan tekun dan dengan bantuan dari teman-teman, pada akhirnya saya pun mengerti mengenai operasi bitwise tersebut.

Selain itu, setiap soal yang ada mengandung materi yang berbeda. Mulai dari soal operator aritmatika (memberikan pelajaran mengenai perhitungan secara matematis), operator penugasan gabungan (untuk memberikan nilai ke dalam variabel tertentu), operator relasional (untuk menguji hubungan antara nilai dan atau variabel dan selalu menghasilkan nilai true atau false), operator increment dan decrement (yang digunakan untuk menaikkan dan menurunkan nilai variabel sebesar satu), operator logika (untuk membandingkan dua nilai variabel atau lebih, hasilnya boolean true atau false), operator kondisional (penyederhanaan dari bentuk if.else yang setiap blok dari if dan else hanya terdiri dari satu statement/perintah), dan operator bitwise (untuk operasi bit (biner) dan berlaku untuk tipe data int, long, short, char, dan byte, karena akan menghitung dari bit-ke-bit), memnantu saya untuk semakin memahami materi operator ini.