

Nama dan NPM:	Topik:	Tanggal:
Sophina Shafa Salsabila G1A022021	Operator	7 September 2022

[Latihan 1] Identifikasi Masalah:

Susun kode Java untuk perhitungan dengan ekspresi $(2*3 + 6 / 2 - 4)$. Simpulkan urutan prioritas operator yang dijalankan ekspresi tersebut!
Diketahui dari soal adalah ekspresi operator, yaitu $(2*3 + 6 / 2 - 4)$.

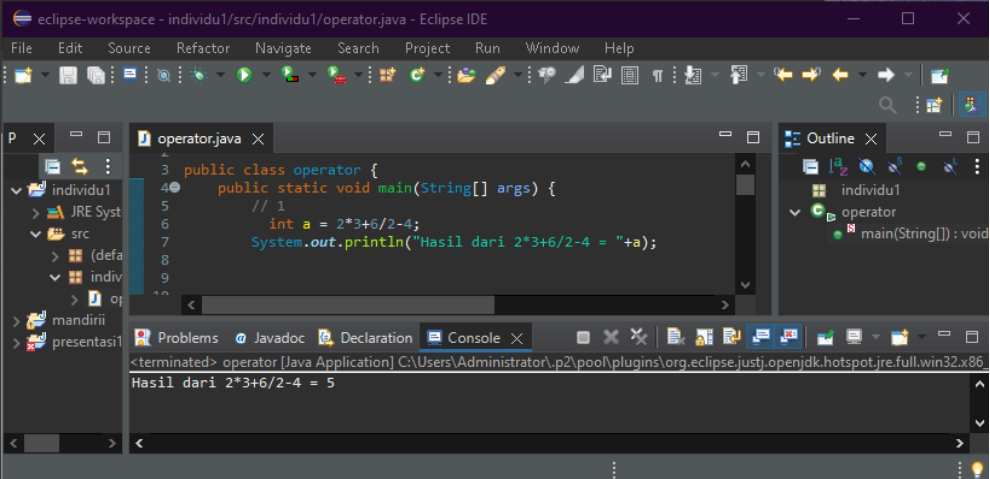
[Latihan 1] Analisis dan Argumentasi

Ekspresi operator tersebut dapat di selesaikan dengan menggunakan operator aritmatika, karena operator yang digunakan adalah (*),(+),(-), dan (/)

[Latihan 1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma penyelesaian ekspresi $(2*3 + 6 / 2 - 4)$ menggunakan pemograman
 - a) Menyusun kembali pemograman dengan membuat kelas baru.
 - b) Gunakan tipe data integer untuk memasukkan variabel hasil dari ekspresi tersebut.
 - c) Panggil variabel hasil dengan menambahkan (+) pada System.out.println
 - d) Jalankan program
 - e) Hasil ekspresi muncul.
- 2) Algoritma penyelesaian ekspresi $(2*3 + 6 / 2 - 4)$ secara manual
 - a) $([2*3] + [6 / 2] - 4)$, menurut operator operasi, tingkatan (*) dan (/) lebih tinggi dari pada (+) dan (-), jadi operator pertama yang dihitung adalah (*) dan (/) seperti yang ditandai oleh [].
 - b) $([6 + 3] - 4)$, karena (+) dan (-) berada pada tingkatan yang sama, maka pada ekspresi tersebut yang dikerjakan duluan adalah operator (+) seperti yang ditandai oleh [].
 - c) $(9 - 4) = 5$, maka didapatlah hasil yang sama dengan hasil pada program diatas.
 - d) Hasil ekspresi adalah 5.
- 3) Kode program dan luaran

a)



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

```

1 public class operator {
2     public static void main(String[] args) {
3         // 1
4         int a = 2*3+6/2-4;
5         System.out.println("Hasil dari 2*3+6/2-4 = "+a);
6     }
7 }

```

The 'Outline' view on the right shows the class structure. The 'Console' view at the bottom shows the output of the program:

```

<terminated> operator [Java Application] C:\Users\Administrator\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Hasil dari 2*3+6/2-4 = 5

```

- b) Analisa luaran yang dihasilkan
Tipe data dan operator yang digunakan sudah sesuai, luaran yang dihasilkan pun sama dengan hasil perhitungan manual.

[Latihan 1] Kesimpulan

Pada program itu saya menggunakan tipe data integer karena operator yang digunakan adalah operator aritmatika, sehingga luaran yang dihasilkan berupa angka desimal. adapun perhitungan manual yang dilakukan adalah dengan memperhatikan urutan prioritas operator, dan didapatkan hasil perhitungan pemrograman dan hasil perhitungan manual itu sama.

[Latihan 2] Identifikasi Masalah:

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

Pada contoh diketahui :

Variabel a = 20

Variabel b = 3

Variabel tersebut telah dijalankan program dengan menggunakan operator (+=) dan menghasilkan luaran 23. Bagaimana hasil perhitungan jika program tersebut ditambahkan operator (-=, *=, /=, %=)?

[Latihan 2] Analisis dan Argumentasi

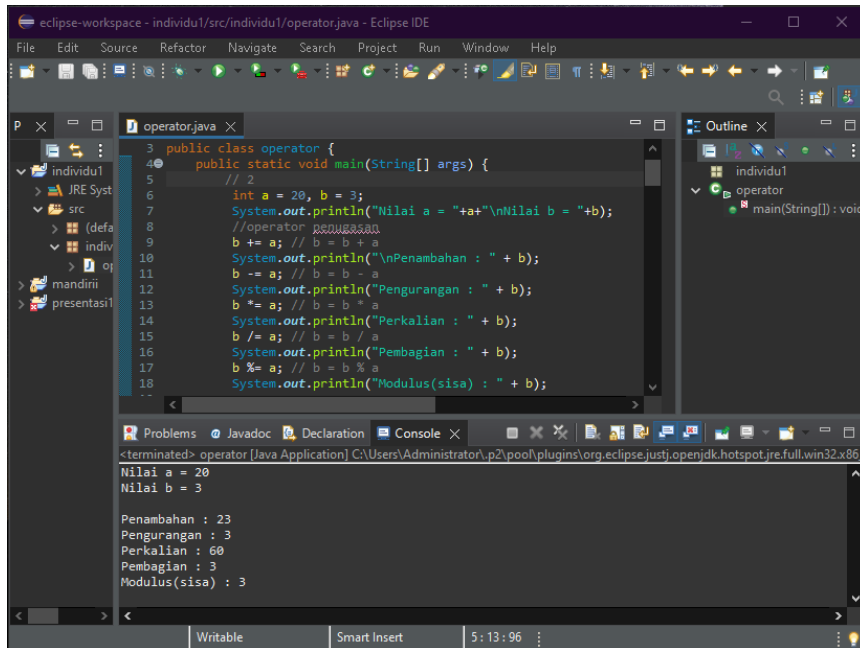
Penambahan operator (-=, *=, /=, %=) pada pemrograman tersebut dapat dilakukan dengan menambahkan operator penugasan yang baru serta System.out.println untuk memanggil hasil dari penugasan baru tersebut.

[Latihan 2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma penyelesaian penggunaan operator penugasan menggunakan program.
 - a) Menyusun kembali pemrograman dengan membuat kelas baru.
 - b) Tulis pemrograman sebelumnya yang telah terdapat deklarasi nilai a dan b serta operator(+=).
 - c) tambahkan variabel untuk menjalankan operator (b -= a).
 - d) Panggil variabel hasil dengan menambahkan (+b) pada System.out.println
 - e) tambahkan variabel untuk menjalankan operator (b += a).
 - f) Panggil variabel hasil dengan menambahkan (+b) pada System.out.println
 - g) tambahkan variabel untuk menjalankan operator (b /= a).
 - h) Panggil variabel hasil dengan menambahkan (+b) pada System.out.println
 - i) tambahkan variabel untuk menjalankan operator (b %= a).
 - j) Panggil variabel hasil dengan menambahkan (+b) pada System.out.println
 - k) Jalankan program
 - l) Hasil ekspresi muncul.
- 2) Algoritma penyelesaian penggunaan operator penugasan secara manual
 - a) Nilai a = 20 dan nilai b = 3
 - b) b += a, berarti b = 3 + 20 = 23
(Nilai b telah berubah menjadi 23)
 - c) b -= a, berarti b = 23 – 20 = 3
(Nilai b telah berubah menjadi 3)
 - d) b *= a, berarti b = 3 * 20 = 60
(Nilai b telah berubah menjadi 60)
 - e) b /= a, berarti b = 60 / 20 = 3
(Nilai b telah berubah menjadi 3)
 - f) b %= a, berarti b = 3 % 20 = 3
 - g) Hasil ekspresi adalah 5.

3) Kode program dan luaran

a)



```
public class operator {
    public static void main(String[] args) {
        // 2
        int a = 20, b = 3;
        System.out.println("Nilai a = "+a+"\nNilai b = "+b);
        //operator
        b += a; // b = b + a
        System.out.println("\nPenambahan : " + b);
        b -= a; // b = b - a
        System.out.println("Pengurangan : " + b);
        b *= a; // b = b * a
        System.out.println("Perkalian : " + b);
        b /= a; // b = b / a
        System.out.println("Pembagian : " + b);
        b %= a; // b = b % a
        System.out.println("Modulus(sisa) : " + b);
    }
}
```

Console Output:

```
<terminated> operator [Java Application] C:\Users\Administrator\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Nilai a = 20
Nilai b = 3

Penambahan : 23
Pengurangan : 3
Perkalian : 60
Pembagian : 3
Modulus(sisa) : 3
```

b) Analisa luaran yang dihasilkan

Tipe data dan operator yang digunakan sudah sesuai, luaran yang dihasilkan pun sama dengan hasil perhitungan manual.

[Latihan 2] Kesimpulan

Operator penugasan mengandung deklarasi ($=$), jadi nilai dari variabel pertama akan berubah ubah karena mengikuti hasil terakhir. Kemudian pada operator terakhir, yaitu modulus ($\%$) atau sisa dari pembagian menghasilkan 3. nilai tersebut diperoleh dari 3 bagi 20 menghasilkan 0.15 maka disimpulkan hasil dari pembagian tersebut adalah 0 jadi modulusnya adalah 3

[Latihan 3] Identifikasi Masalah:

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

Diketahui dari soal:

Variabel nilaiA = 12

Variabel nilaiB = 4

Perhitungan ($a > b$), ($a >= b$), ($a != b$) menghasilkan luaran true.

perhitungan ($a < b$), ($a <= b$), ($a == b$) menghasilkan luaran false. Susun ulang variabel pada ekspresi ini agar menghasilkan luaran true.

[Latihan 3] Analisis dan Argumentasi

Pada pemograman tersebut, agar hasil luaran ekspresi ($a < b$), ($a <= b$), ($a == b$) adalah true tanpa mengganti operator dan nilai variabel, maka kita dapat menukar variabel - variabel pada operator tersebut agar sesuai.

[Latihan 3] Penyusunan Algoritma dan Kode Program

1) Algoritma penyelesaian ekspresi ($a < b$), ($a <= b$), ($a == b$) agar menjadi true.

a) Menyusun kembali pemograman dengan membuat kelas baru.

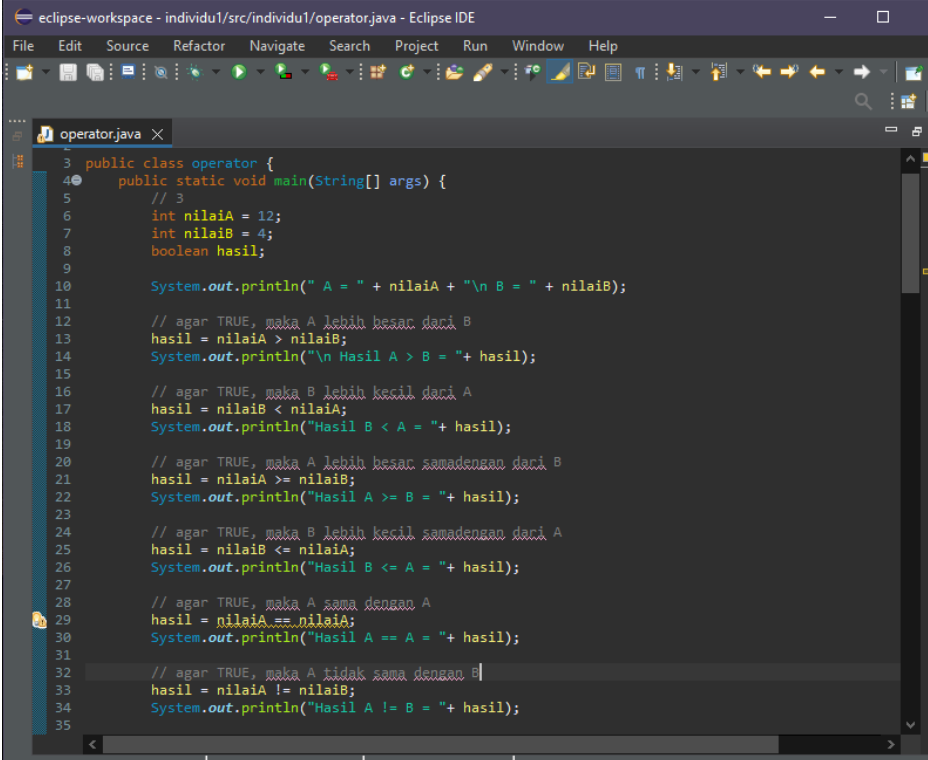
b) Tulis pemograman sebelumnya yang telah terdapat deklarasi nilai a dan b serta operator. Tipe data hasil yang digunakan adalah boolean.

c) Pada deklarasi operator ($a < b$), tukar variabel a dan variabel b (variabel a menjadi b dan sebaliknya).

- d) Pada deklarasi operator ($a \leq b$), tukar variabel a dan variabel b (variabel a menjadi b dan sebaliknya).
- e) Pada deklarasi operator ($a == b$), tukar variabel b menjadi a.
- f) Jalankan program
- g) Semua hasil menjadi true.

2) Kode program dan luaran

a)

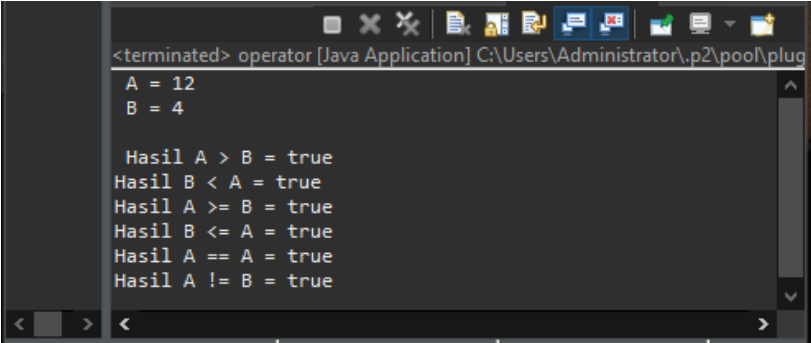


```

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

```

Luaran :



```

<terminated> operator [Java Application] C:\Users\Administrator\p2\pool\plug
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 telah sesuai dengan yang diperintahkan.

[Latihan 3] Kesimpulan

Pada program tersebut, untuk menghasilkan luaran true tanpa mengubah nilai variabel dan operator yang digunakan dapat dilakukan dengan menukar variabel pada ekspresi (variabel a menjadi variabel b, atau sebaliknya)

[Latihan 4] Identifikasi Masalah:

- 4.1 Susunlah kode program untuk menghasilkan luaran nilai a = 5 dan b = 6 dengan pre/post increment dan pre/post decrement.

Diketahui dari soal :

Variabel nilai a= 5

Variabel nilai b = 6

[Latihan 4] Analisis dan Argumentasi

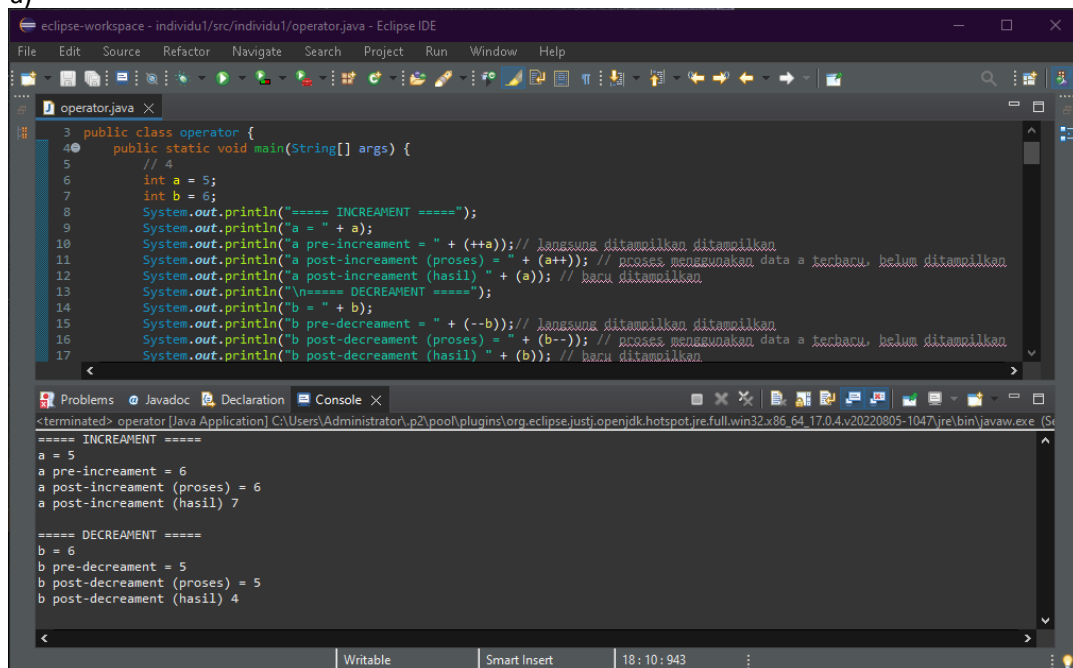
Susunan program untuk menghasilkan nilai yang diinginkan dengan menggunakan pre atau post increment dan decrement dapat dilakukan dengan menuliskan operator increment dan decrement pada System.out.println. Misalnya System.out.println(++a) untuk pre increment.

[Latihan 4] Penyusunan Algoritma dan Kode Program

- 1) Algoritma penyelesaian pre/post - increment/decrement.
 - a) Menyusun kembali pemrograman dengan membuat kelas baru.
 - b) Tambahkan variabel nilai a = 5 dengan menggunakan tipe data integer.
 - c) Tambahkan variabel nilai b = 6 dengan menggunakan tipe data integer.
 - d) Tambahkan System.out.println dengan (a) untuk menampilkan nilai a.
 - e) Tambahkan System.out.println dengan (++a) untuk menampilkan nilai a setelah ditambah dengan menggunakan operator pre-increment.
 - f) Tambahkan System.out.println dengan (a++) untuk memproses nilai a setelah ditambah dengan menggunakan operator post-increment
 - g) Tambahkan System.out.println dengan (a) untuk menampilkan nilai a setelah ditambah dengan menggunakan operator post-increment.
 - h) Tambahkan System.out.println dengan (b) untuk menampilkan nilai b.
 - i) Tambahkan System.out.println dengan (--b) untuk menampilkan nilai b setelah ditambah dengan menggunakan operator pre-decrement.
 - j) Tambahkan System.out.println dengan (b--) untuk memproses nilai b setelah ditambah dengan menggunakan operator post-decrement
 - k) Tambahkan System.out.println dengan (b) untuk menampilkan nilai b setelah ditambah dengan menggunakan operator post-decrement.
 - l) Jalankan program.
 - m) Muncul hasil yang diperintahkan.

3) Kode program dan luaran

a)



```
3 public class operator {
4     public static void main(String[] args) {
5         // 4
6         int a = 5;
7         int b = 6;
8         System.out.println("==== INCREMENT =====");
9         System.out.println("a = " + a);
10        System.out.println("a pre-increment = " + (++a)); // langsung ditampilkan
11        System.out.println("a post-increment (proses) = " + (a++)); // proses memproses data a terakhir, belum ditampilkan
12        System.out.println("a post-increment (hasil) = " + (a)); // baru ditampilkan
13        System.out.println("\n==== DECREMENT =====");
14        System.out.println("b = " + b);
15        System.out.println("b pre-decrement = " + (--b)); // langsung ditampilkan
16        System.out.println("b post-decrement (proses) = " + (b--)); // proses memproses data b terakhir, belum ditampilkan
17        System.out.println("b post-decrement (hasil) = " + (b)); // baru ditampilkan
18    }
19 }
```

```
<terminated> operator [Java Application] C:\Users\Administrator\AppData\Local\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220805-1047\jre\bin\javaw.exe (S
==== INCREMENT =====
a = 5
a pre-increment = 6
a post-increment (proses) = 6
a post-increment (hasil) 7

==== DECREMENT =====
b = 6
b pre-decrement = 5
b post-decrement (proses) = 5
b post-decrement (hasil) 4
```

b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan telah sesuai dengan yang diperintahkan.

[Latihan 4] Kesimpulan

Pre increment adalah penambahan nilai variabel sebanyak 1 angka dengan cara menambahkan tanda (++) pada awal variabel (++a) sedangkan pre decrement adalah pengurangan nilai variabel sebanyak 1 angka dengan cara menambahkan tanda (--) pada awal variabel (--b) . hasil penambahan dan pengurangan tersebut akan langsung muncul pada pemanggilan pertama seperti yang telah dijalankan pada program diatas.

Post increment adalah penambahan nilai variabel sebanyak 1 angka dengan cara menambahkan tanda (++) pada akhir variabel (a++) sedangkan post decrement adalah pengurangan nilai variabel sebanyak 1 angka dengan cara menambahkan tanda (--) pada akhir variabel (b--) . hasil penambahan dan pengurangan tersebut akan diproses dahulu pada pemanggilan pertama dan akan muncul pada pemanggilan selanjutnya seperti yang telah dijalankan pada program diatas.

[Latihan 5] Identifikasi Masalah:

Susun kode program dengan mengubah nilai a dan b untuk menghasilkan luaran true dengan operator && dan operator | |. Beri kesimpulan!
Diketahui dari soal, tipe data yang digunakan adalah boolean karean operator yang digunakan adalah operator logika. Variabel a = true dan variabel b = false. Ubah variabel untuk menghasilkan luaran true dengan menggunakan operator AND (&&) dan OR (| |).

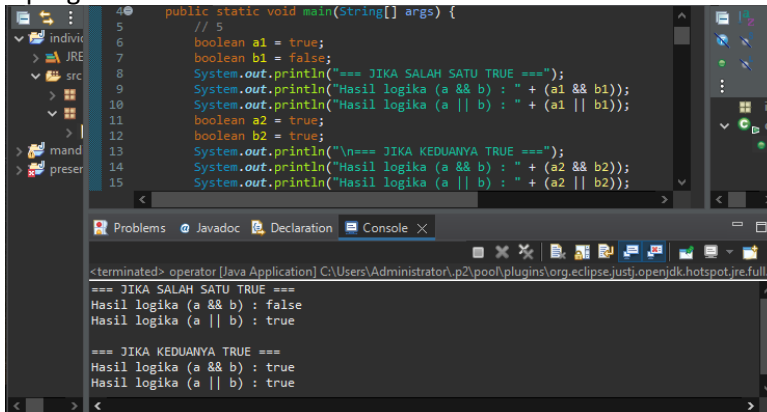
[Latihan 5] Analisis dan Argumentasi

Jika menjalankan operator AND dan OR dengan menggunakan variabel yang telah diketahui pada soal, maka program tersebut akan menghasilkan false pada operator AND dan true pada operator OR. Agar hasil luaran keduanya sama - sama true, maka dapat dilakukan dengan mengganti nilai variabel b menjadi true, karena operator AND hanya akan true jika kedua variabel true.

[Latihan 5] Penyusunan Algoritma dan Kode Program

- 1) Algoritma penyelesaian ekspresi pada operator logika agar menghasilkan luaran true.
 - a) Menyusun kembali pemograman dengan membuat kelas baru.
 - b) Tulis pemograman sebelumnya yang telah terdapat deklarasi nilai a dan b dengan tipe data boolean tanpa mengganti nilai variabel sebagai perbandingan
 - c) Tulis kembali program sebelumnya dengan mengganti nilai variabel b menjadi true.
 - d) Jalankan program
 - e) Didapat luaran yang menghasilkan true semua.
- 2) Kode program dan luaran

a)



```
public static void main(String[] args) {  
    // 5  
    boolean a1 = true;  
    boolean b1 = false;  
    System.out.println("=== JIKA SALAH SATU TRUE ===");  
    System.out.println("Hasil logika (a && b) : " + (a1 && b1));  
    System.out.println("Hasil logika (a || b) : " + (a1 || b1));  
    boolean a2 = true;  
    boolean b2 = true;  
    System.out.println("\n=== JIKA KEDUANYA TRUE ===");  
    System.out.println("Hasil logika (a && b) : " + (a2 && b2));  
    System.out.println("Hasil logika (a || b) : " + (a2 || b2));  
}
```

Problems Javadoc Declaration Console

```
<terminated> operator [Java Application] C:\Users\Administrator\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.w  
=== JIKA SALAH SATU TRUE ===  
Hasil logika (a && b) : false  
Hasil logika (a || b) : true  
  
=== JIKA KEDUANYA TRUE ===  
Hasil logika (a && b) : true  
Hasil logika (a || b) : true
```

- b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan telah sesuai dengan yang diperintahkan.

[Latihan 5] Kesimpulan

Pada program tersebut diketahui bahwa operator && atau AND akan menghasilkan true jika nilai kedua variabel adalah true. Operator || atau OR akan menghasilkan true jika kedua dan/atau salah satu variabel bernilai true. Untuk menghasilkan luaran true dengan operator && dan || maka nilai dari a dan b harus sama sama true.

[Latihan 6] Identifikasi Masalah:

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".
Dari soal diketahui bahwa nilai variabel yang digunakan adalah 12, ekspresi 2 adalah "Selamat Pagi" dan ekspresi 3 adalah "Selamat Malam".

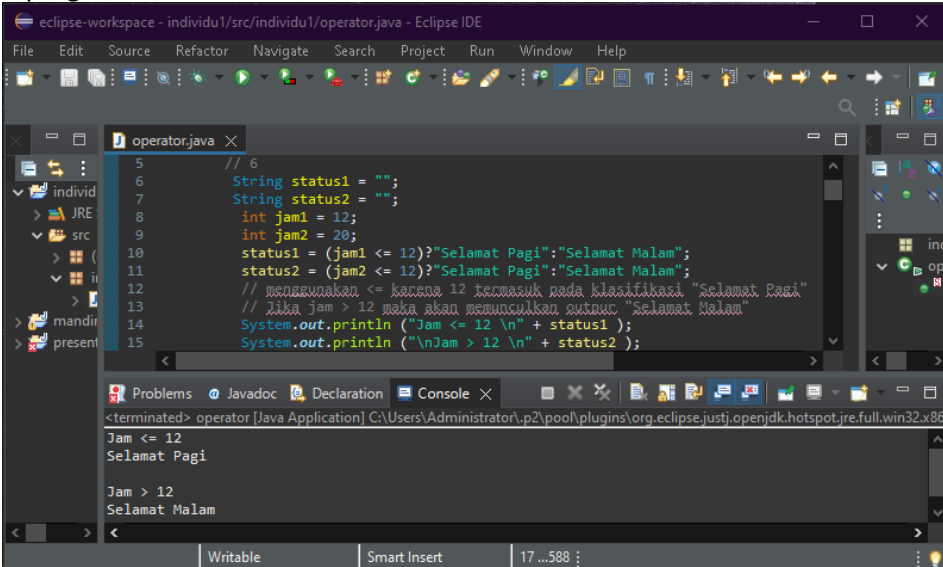
[Latihan 6] Analisis dan Argumentasi

Penyelesaian masalah tersebut menggunakan operator kondisional atau ternary. Variabel jam digunakan pada ekspresi 1, yaitu (jam<=12). Menggunakan operator (<=) karena nilai 12 juga termasuk pada nilai true dan akan menghasilkan luaran "Selamat Pagi" (Ekspresi 2).

[Latihan 6] Penyusunan Algoritma dan Kode Program

- 3) Algoritma penyelesaian ekspresi pada operator logika agar menghasilkan luaran true.
 - a) Menyusun kembali pemograman dengan membuat kelas baru.
 - b) Tulis pemograman seperti pada contoh.
 - c) Ganti nama variabel nilai menjadi jam1, dan ganti nilai variabel menjadi 12.
 - d) Tambahkan variabel jam2 dengan nilai 20 sebagai perbandingan.
 - e) Tulis System.out.println dengan menambahkan operator kondisional (jam1 <=12?"Selamat Pagi":"Selamat Malam"
 - f) Tambahkan System.out.println dengan menambahkan operator kondisional (jam2 <=12?"Selamat Pagi":"Selamat Malam"
 - g) Jalankan program
 - h) Didapat luaran seperti yang diperintahkan
- 4) Kode program dan luaran

a)



```
5 // 6
6 String status1 = "";
7 String status2 = "";
8 int jam1 = 12;
9 int jam2 = 20;
10 status1 = (jam1 <= 12)?"Selamat Pagi":"Selamat Malam";
11 status2 = (jam2 <= 12)?"Selamat Pagi":"Selamat Malam";
12 // menggunakan <= karena 12 termasuk pada klasifikasi "Selamat Pagi"
13 // jika jam > 12 maka akan memunculkan output "Selamat Malam"
14 System.out.println ("Jam <= 12 \n" + status1);
15 System.out.println ("Jam > 12 \n" + status2);
```

Console Output:

```
<terminated> operator [Java Application] C:\Users\Administrator\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jdk-17.0.2\bin\java.exe
Jam <= 12
Selamat Pagi
Jam > 12
Selamat Malam
```

- c) Analisa luaran yang dihasilkan

Luaran yang dihasilkan telah sesuai dengan yang diperintahkan.

[Latihan 6] Kesimpulan

Operator kondisional atau ternary adalah operator yang akan menghasilkan luaran tergantung ekspresi 1, jika ekspresi 1 bernilai true, maka luaran akan otomatis ke ekspresi 2. jika ekspresi 1 bernilai false, maka luaran akan otomatis ke ekspresi 3.

[Latihan 7] Identifikasi Masalah:

Susun kode tambahan dari Contoh 7 untuk melakukan perhitungan dengan operator (>, <).
Hubungkan hasil luaran dengan perhitungan manual bilangan biner!
Diketahui dari soal variabel a = 10 dan b = 7 serta operasi menggunakan operator logika &, |,
dan ^, Bagaimanakah hasil perhitungan dengan menggunakan operator (>) dan (<).

[Latihan 7] Analisis dan Argumentasi

Pada pemograman, simbol (>>) dan (<<) merupakan operator bitwise yang berarti geser kekanan atau ke kiri. Yang bergeser adalah angka desimal yang telah dikonversikan menjadi angka biner.

[Latihan 7] Penyusunan Algoritma dan Kode Program

- 1) Algoritma penyelesaian penggunaan operator penugasan menggunakan program.
 - a) Menyusun kembali pemrograman dengan membuat kelas baru.
 - b) Tulis pemrograman sebelumnya yang telah terdapat deklarasi nilai a dan b serta operator(&, |, ^).
 - c) tambahkan variabel untuk menjalankan operator ($a \gg 1$).
 - d) Panggil variabel hasil dengan menambahkan (+a) pada `System.out.println`
 - e) tambahkan variabel untuk menjalankan operator ($b \ll 2$).
 - f) Panggil variabel hasil dengan menambahkan (+b) pada `System.out.println`
 - g) Jalankan program
 - h) Hasil ekpresi muncul.
- 2) Algoritma penyelesaian penggunaan operator penugasan secara manual
 - a) Nilai a = 10 maka jika dikonversikan menjadi biner a = 1010
 - b) ($a \gg 1$) berarti nilai biner a geser kekanan sebanyak 1, maka nilai biner a menjadi 0101
 - c) Nilai biner 0101 dikonversikan jadi bilangan desimal menghasilkan 5
 - d) Jadi nilai ($a \gg 1$) = 5
 - e) Nilai b = 7 maka jika dikonversikan menjadi biner b = 0111
 - f) ($b \gg 2$) berarti nilai biner b geser ke kiri sebanyak 2, maka nilai biner b menjadi 0001 1100
 - g) Nilai biner 0001 1100 dikonversikan jadi bilangan desimal menghasilkan 28
 - h) Jadi nilai ($b \ll 2$) = 28
- 3) Kode program dan luaran

a)

```

1 // 6
2 int a = 10; // bindec = 1010
3 int b = 7; // bindec = 0111
4 int hasil;
5
6 hasil = a & b; // 1010 & 0111 = 0010 = 2 (AND)
7 System.out.println("Hasil dari a & b : " + hasil );
8
9 hasil = a | b; // 1010 | 0111 = 1111 = 15 (OR)
10 System.out.println("Hasil dari a | b : " + hasil );
11
12 hasil = a ^ b; // 1010 ^ 0111 = 1101 = 13 (XOR)
13 System.out.println("Hasil dari a ^ b : " + hasil );
14
15 hasil = a >> 1; // 1010 >> 1 = 0101 = 5
16 System.out.println("Hasil dari a >> 1 : " + hasil );
17
18 hasil = b << 2; // 0111 << 2 = 0001 1100 = 28
19 System.out.println("Hasil dari b << 2 : " + hasil );
20
21
22
23

```

```

<terminated> operator [Java Application] C:\Users\Administrator\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jdk\bin\java.exe -Xms4m -Xmx4m -XX:PermSize=256m -XX:MaxPermSize=256m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Duser.language=en -Duser.region=US -Duser.timezone=UTC -Djava.class.path=C:\Users\Administrator\p2\poo\workspace\operator\bin\operator.jar C:\Users\Administrator\p2\poo\workspace\operator\bin\operator.jar
Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
Hasil dari a >> 1 : 5
Hasil dari b << 2 : 28

```

b) Analisa luaran yang dihasilkan

Tipe data dan operator yang digunakan sudah sesuai, luaran yang dihasilkan pun sama dengan hasil perhitungan manual

[Latihan 7] Kesimpulan

<p>Operator Bitwise berarti melakukan operasi perhitungan dengan menggunakan bilangan biner. Operator (\gg) berarti pergeseran kekanan nilai yang telah dikonversikan menjadi bilangan biner. Seperti pada contoh diketahui $a = 10$ maka bilangan biner dari $a = 1010$, jika dilakukan operasi ($a \gg 1$) berarti nilai a geser ke kanan sebanyak 1, maka nilai biner a menjadi 0101, jika dikonversikan ke bilangan desimal maka nilai a setelah operasi adalah 5, sama seperti program diatas.</p>
--

<p>Kemudian operator (\ll) berarti pergeseran ke kanan nilai yang telah dikonversikan menjadi bilangan biner. Pada contoh diketahui $b = 7$ maka bilangan biner dari $b = 0111$, jika dilakukan operasi ($b \ll 2$) berarti nilai b geser ke kiri sebanyak 2, maka nilai biner dari b menjadi 00011100, jika dikonversikan ke bilangan desimal maka nilai a setelah operasi adalah 28.</p>
