

Nama & NPM	Topik:	Tanggal:
G. Krisna Prabowo Barus G1F024062	Operator Java	12 September 2024
[Nomor Soal] Identifikasi Masalah:		
1) Uraikan permasalahan dan variabel 2) Rincikan sumber informasi yang relevan (buku / webpage) 3) Uraikan rancangan solusi yang diusulkan (jika ada). 4) Analisis susunan solusi, parameter solusi (jika ada).		
[Nomor Soal] Analisis dan Argumentasi		
1) Uraikan rancangan solusi yang diusulkan. 2) Analisis solusi, kaitkan dengan permasalahan.		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
1) Rancang desain solusi atau algoritma 2) Tuliskan kode program dan luaran <ul style="list-style-type: none"> a) Beri komentar pada kode b) Uraikan luaran yang dihasilkan c) Screenshot/ Capture potongan kode dan hasil luaran 		
[Nomor Soal] Kesimpulan		
1) Analisa <ul style="list-style-type: none"> a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? 2) Evaluasi <ul style="list-style-type: none"> a) Apa konsekuensi dari skenario pemrograman ini? b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada) 3) Kreasi <ul style="list-style-type: none"> a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi? b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada) 		

LATIHAN 1

[No. 1] Identifikasi Masalah:

Pada soal ini kita di diperlukan untuk menambahkan baris code `System.out.println ("a + b = " + (a + b));` dan mengubah operator (+) dengan tanda (-, *, /, %), kemudian menganalisa perhitungan matematika yang terjadi.

[No.1] Analisis dan Argumentasi

Pada kode ini, saya menggunakan berbagai macam operator aritmatika melakukan berbagai perhitungan dasar antara dua bilangan a dan b, yaitu 20 dan 3. Penjumlahan (+) menghasilkan 23, karena ini adalah operasi dasar yang menjumlahkan kedua nilai.

Pengurangan (-) menghasilkan 17, yang merupakan hasil dari pengurangan 20 dengan 3.

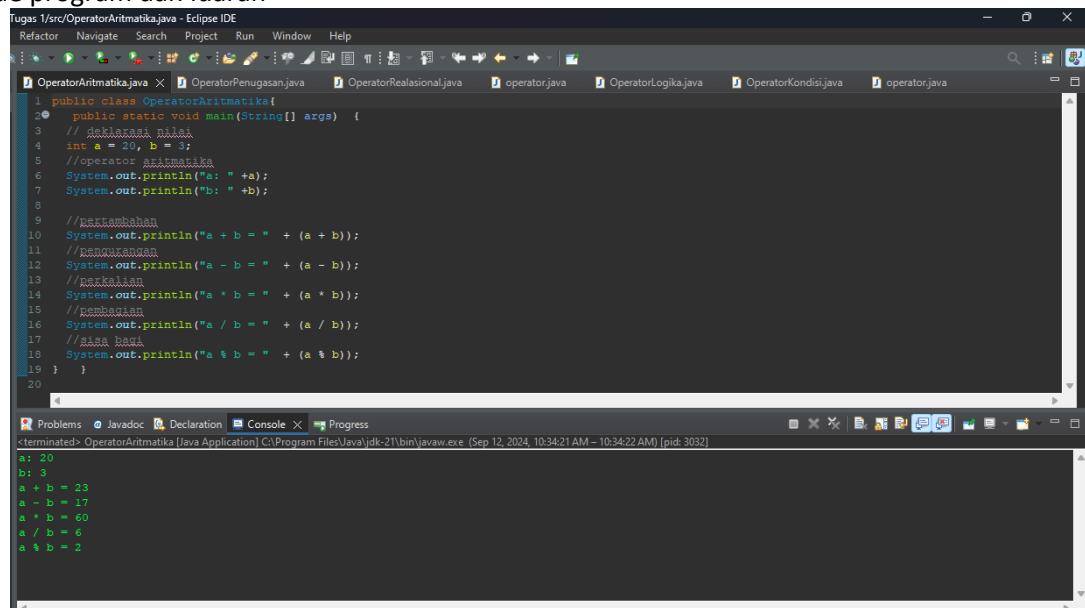
Untuk operasi perkalian (*) mendapatkan hasil 60 karena 20 dikalikan dengan 3. Kemudian untuk pembagian (/) menghasilkan 6. Dan terakhir, operasi modulus (%) menghasilkan 2, yang merupakan sisa pembagian dari 20 dibagi dengan 3.

[No.1] Penyusunan Algoritma dan Kode Program

Algoritma:

- (a) Mulai
- (b) Membuka Eclipse
- (c) Membuat Java Project baru
- (d) Membuat class baru di dalamnya
- (e) Memasukan code yang diberikan
- (f) Mengubah code sesuai soal
- (g) Tekan run dan code akan berjalan
- (h) Selesai

Kode program dan luaran



```
1 public class OperatorAritmatika {
2     public static void main(String[] args) {
3         // deklarasi nilai
4         int a = 20, b = 3;
5         //operator aritmatika
6         System.out.println("a: " + a);
7         System.out.println("b: " + b);
8
9         //penambahan
10        System.out.println("a + b = " + (a + b));
11        //pengurangan
12        System.out.println("a - b = " + (a - b));
13        //perkalian
14        System.out.println("a * b = " + (a * b));
15        //pembagian
16        System.out.println("a / b = " + (a / b));
17        //sisa hasil
18        System.out.println("a % b = " + (a % b));
19    }
20 }
```

Output Console:

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

Analisa luaran yang dihasilkan:

Program yang dibuat merupakan operator aritmatika java yang digunakan untuk menghitung bilangan-bilangan. Code dan Luaran sudah benar.

[No.1] Kesimpulan

Analisa

Pada dasarnya, operator aritmatika seperti +, -, *, /, dan % dapat digunakan untuk melakukan berbagai perhitungan matematis dasar. Pada code yang telah dibuat, bisa dilihat bagaimana operator-operator ini bekerja dengan angka a (20) dan b (3), menghasilkan nilai sesuai dengan operasinya. Penjumlahan dan pengurangan langsung memberikan hasil penambahan dan pengurangan biasa, sementara perkalian mengalikan kedua nilai. Pembagian menggunakan tipe data integer membulatkan hasil ke bawah, hanya menampilkan bilangan bulat. Sisa dari pembagian dapat diketahui menggunakan operator modulus (%).

LATIHAN 2

[No. 2] Identifikasi Masalah:

Pada soal ini kita diperlukan untuk membandingkan hasil dari contoh 1 dan contoh 2

[No.2] Analisis dan Argumentasi

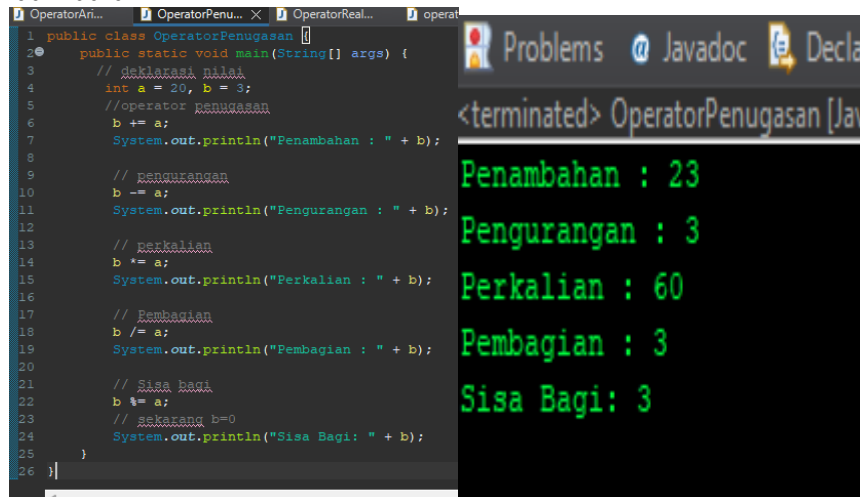
Pada Contoh 1 dan Contoh 2, dapat dilihat penggunaan operator aritmatika dan operator penugasan yang memiliki perbedaan signifikan dalam cara kerjanya. Pada Contoh 1, operator aritmatika seperti +, -, *, /, dan % hanya digunakan untuk melakukan perhitungan antara dua bilangan, namun tidak mengubah nilai asli dari variabel yang terlibat. Sebagai contoh, saat dilakukan penjumlahan $a + b$, hasilnya adalah 23, tetapi nilai variabel b tetap 3. Sedangkan, pada Contoh 2, operator penugasan seperti +=, -=, *=, /=, dan %= digunakan untuk melakukan perhitungan sekaligus mengubah nilai variabel secara langsung. Misalnya, dalam operasi $b += a$, nilai b yang awalnya 3 berubah menjadi 23 setelah penambahan. Setiap operasi pada Contoh 2 memperbarui nilai variabel b , sedangkan pada Contoh 1, variabel tetap tidak berubah.

[No.2] Penyusunan Algoritma dan Kode Program

Algoritma:

- (a) Mulai
- (b) Membuka Eclipse
- (c) Membuat Java Project baru
- (d) Membuat class baru di dalamnya
- (e) Memasukkan code yang diberikan
- (f) Tekan Run pada eclipse
- (g) Selesai

Kode program dan luaran



```
1 public class OperatorPenugasan {
2     public static void main(String[] args) {
3         // deklarasi nilai
4         int a = 20, b = 3;
5         //operator penugasan
6         b += a;
7         System.out.println("Penambahan : " + b);
8
9         // pengurangan
10        b -= a;
11        System.out.println("Pengurangan : " + b);
12
13        // perkalian
14        b *= a;
15        System.out.println("Perkalian : " + b);
16
17        // Pembagian
18        b /= a;
19        System.out.println("Pembagian : " + b);
20
21        // Sisa bagi
22        b %= a;
23        // sekarang b=0
24        System.out.println("Sisa Bagi: " + b);
25    }
26 }
```

<terminated> OperatorPenugasan [Java]

Penambahan : 23
Pengurangan : 3
Perkalian : 60
Pembagian : 3
Sisa Bagi: 3

Analisa luaran yang dihasilkan:

Program ini adalah sebuah program yang digunakan untuk tidak hanya melakukan operasi aritmatika, tetapi juga mengubah nilai variabel b setelah setiap operasi. Code dan Luaran sudah benar.

[No.2] Kesimpulan

Analisa

Operator aritmatika dan operator penugasan memiliki fungsi yang berbeda. Pada Contoh 1, operator aritmatika seperti +, -, *, /, dan % hanya menghitung hasil tanpa mengubah nilai variabel a dan b. Sebaliknya, pada Contoh 2, operator penugasan (+=, -=, *=, /=, %=) mengubah nilai variabel b setelah setiap operasi. Perbedaan utama adalah bahwa operator penugasan memperbarui nilai variabel secara langsung, sementara operator aritmatika hanya menghitung hasil tanpa memodifikasi variabel.

LATIHAN 3

[No. 3] Identifikasi Masalah:

Pada soal ini kita diperlukan untuk mengubah nilai A = 4 dan B = 4, dan menganalisa perubahan yang terjadi, kemudian membandingkan bagaimana perbedaan nilai A dan B mempengaruhi nilai luaran.

[No.3] Analisis dan Argumentasi

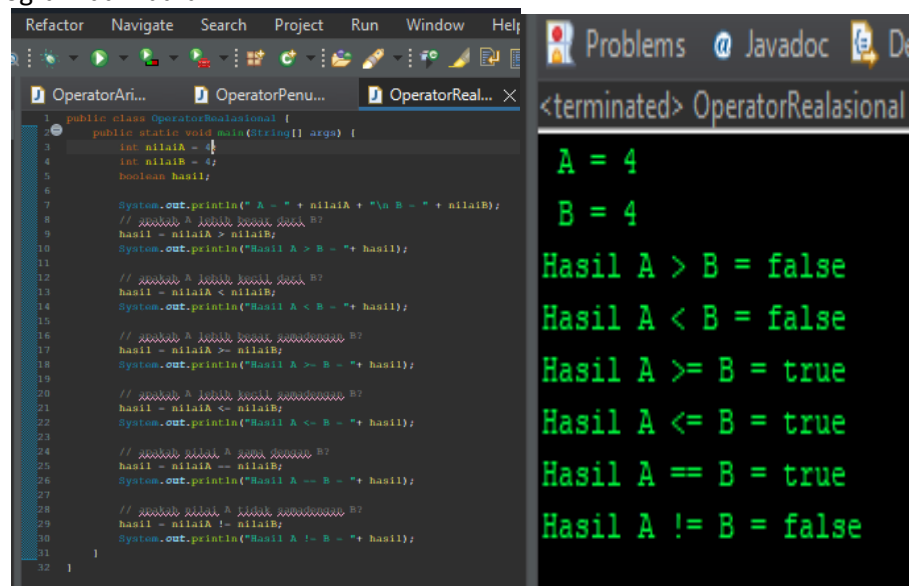
Ketika nilai variabel A dan B diubah menjadi 4 dan 4, hasil dari operasi relasional mengalami perubahan signifikan. Pada contoh awal dengan nilai A = 12 dan B = 4, hasil dari operasi seperti A > B adalah true karena 12 memang lebih besar dari 4, sedangkan A < B adalah false karena 12 tidak lebih kecil dari 4. Dalam hal ini, beberapa hasil true dan false tercermin dalam perbandingan, tergantung pada hubungan antara nilai-nilai tersebut. Namun, ketika kedua nilai diubah menjadi sama, yaitu A = 4 dan B = 4, hasilnya berubah secara dramatis. Semua perbandingan yang melibatkan kesetaraan, seperti A == B, menjadi true karena kedua nilai tersebut identik, sedangkan perbandingan ketidaksamaan, seperti A != B, menjadi false. Selain itu, hasil untuk operator >= dan <= juga berubah menjadi true, karena nilai A sama dengan nilai B, sehingga memenuhi kedua kondisi "lebih besar atau sama dengan" dan "lebih kecil atau sama dengan".

[No.3] Penyusunan Algoritma dan Kode Program

Algoritma:

- Mulai
- Membuka Eclipse
- Membuat Java Project baru
- Membuat class baru di dalamnya
- Memasukan code dari E-Learning
- Mengubah code sesuai dengan yang diperlukan
- Tekan Run pada Eclipse
- Selesai

Kode program dan luaran



The screenshot displays the Eclipse IDE interface. On the left, the 'OperatorReal...' class is open, showing a Java program that sets A = 4 and B = 4, then prints the results of various relational operations. On the right, the 'Problems' and 'Javadoc' tabs are visible, and the output of the program is shown in a console window.

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

Output:

```
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
```

Analisa luaran yang dihasilkan

Program ini digunakan untuk menguji hubungan antara nilai dan atau variabel dan selalu menghasilkan nilai true atau false, Luaran dan code sudah benar.

[No.3] Kesimpulan

Analisa

Operator ini digunakan untuk membandingkan dua nilai dan menghasilkan hasil boolean (true atau false) berdasarkan hubungan antara nilai-nilai tersebut. Dalam hasil dari Contoh 3, operator seperti >, <, >=, <=, ==, dan != memberikan informasi tentang bagaimana nilai nilaiA dan nilaiB dibandingkan. Misalnya operator > dan < mengevaluasi apakah nilai pertama lebih besar atau lebih kecil dari nilai kedua, sementara >= dan <= memeriksa apakah nilai pertama lebih besar atau sama dengan, atau lebih kecil atau sama dengan nilai kedua. Operator == dan != membandingkan kesetaraan atau ketidaksamaan nilai.

LATIHAN 4

[No. 4] Identifikasi Masalah:

Pada soal ini kita diperlukan untuk membandingkan hasil Post dan Pre untuk Increment dan Decrement dari kode yang diberikan.

[No.4] Analisis dan Argumentasi

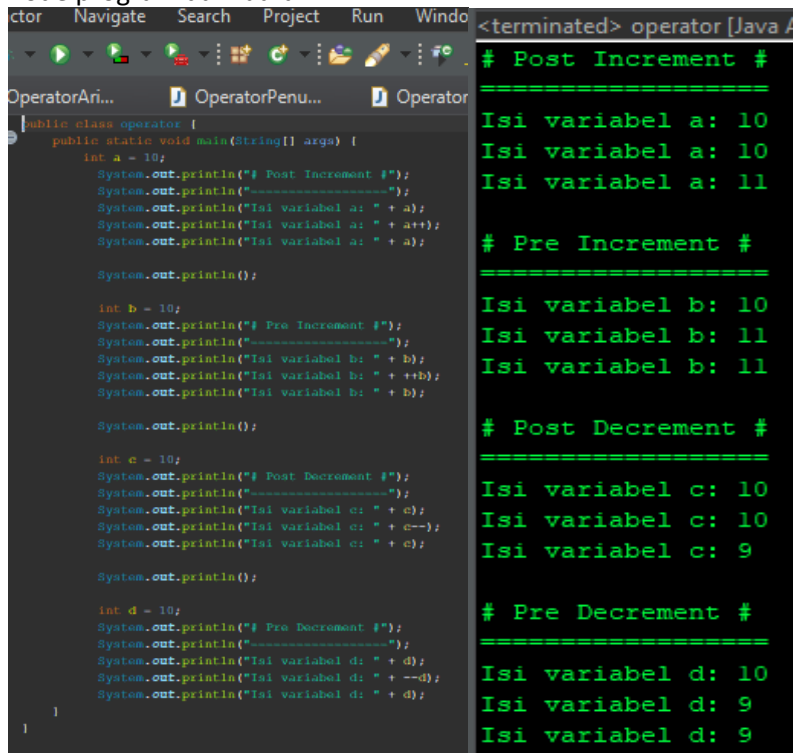
Pada post increment (a++) dan post decrement (c--), perubahan nilai variabel terjadi setelah nilai variabel digunakan dalam ekspresi, artinya bahwa nilai yang ditampilkan sebelum perubahan tetap sesuai dengan nilai awal. Contohnya, dengan a++, nilai a yang dicetak pertama kali adalah 10, dan baru setelah itu nilai a diubah menjadi 11. Seterusnya, pada pre increment (++b) dan pre decrement (--d), perubahan nilai terjadi sebelum nilai variabel digunakan. Oleh karena itu, hasil yang dicetak langsung mencerminkan nilai setelah perubahan. Dengan ++b, nilai b ditingkatkan menjadi 11 sebelum dicetak, sehingga hasilnya langsung 11.

[No.4] Penyusunan Algoritma dan Kode Program

Algoritma:

- Mulai
- Membuka Eclipse
- Membuat Java Project baru
- Membuat class baru di dalamnya
- Memasukkan kode
- Menguji post/pre decrement dan increment
- Tekan Run
- Selesai

Kode program dan luaran



The screenshot shows the Eclipse IDE with a Java class named 'Operator' and its output. The code defines four variables (a, b, c, d) and demonstrates post-increment, pre-increment, post-decrement, and pre-decrement operations. The output shows the state of each variable before and after the operation.

```
public class Operator {  
    public static void main(String[] args) {  
        int a = 10;  
        System.out.println("# Post Increment #");  
        System.out.println("-----");  
        System.out.println("Isi variabel a: " + a);  
        System.out.println("Isi variabel a: " + a++);  
        System.out.println("Isi variabel a: " + a);  
  
        System.out.println();  
  
        int b = 10;  
        System.out.println("# Pre Increment #");  
        System.out.println("-----");  
        System.out.println("Isi variabel b: " + b);  
        System.out.println("Isi variabel b: " + ++b);  
        System.out.println("Isi variabel b: " + b);  
  
        System.out.println();  
  
        int c = 10;  
        System.out.println("# Post Decrement #");  
        System.out.println("-----");  
        System.out.println("Isi variabel c: " + c);  
        System.out.println("Isi variabel c: " + c--);  
        System.out.println("Isi variabel c: " + c);  
  
        System.out.println();  
  
        int d = 10;  
        System.out.println("# Pre Decrement #");  
        System.out.println("-----");  
        System.out.println("Isi variabel d: " + d);  
        System.out.println("Isi variabel d: " + --d);  
        System.out.println("Isi variabel d: " + d);  
    }  
}
```

Output:

```
<terminated> operator [Java A]  
# Post Increment #  
-----  
Isi variabel a: 10  
Isi variabel a: 10  
Isi variabel a: 11  
  
# Pre Increment #  
-----  
Isi variabel b: 10  
Isi variabel b: 11  
Isi variabel b: 11  
  
# Post Decrement #  
-----  
Isi variabel c: 10  
Isi variabel c: 10  
Isi variabel c: 9  
  
# Pre Decrement #  
-----  
Isi variabel d: 10  
Isi variabel d: 9  
Isi variabel d: 9
```

Analisa luaran yang dihasilkan

Kode ini digunakan untuk menaikkan nilai variabel sebesar satu dan menurunkan nilai variabel sebesar satu. Code dan Luaran sudah benar dan sesuai aturan.

[No.4] Kesimpulan

Analisa

Perbedaan antara operator increment (++) dan decrement (--) terletak pada urutan perubahan nilai variabel relatif terhadap penggunaan nilai tersebut dalam ekspresi, operator ++ dan -- merupakan post increment dan post decrement, yang artinya perubahan nilai terjadi setelah nilai variabel digunakan. Dengan demikian, nilai yang dicetak sebelum increment atau decrement tetap mencerminkan nilai awal variabel, sementara perubahan hanya terlihat setelah operasi tersebut. Sebaliknya, operator ++b dan --d adalah pre-increment dan pre-decrement, yang menyebabkan perubahan nilai terjadi sebelum nilai digunakan dalam ekspresi. Oleh karena itu, nilai yang dicetak langsung mencerminkan perubahan yang dilakukan.

LATIHAN 5

[No. 5] Identifikasi Masalah:

Pada soal ini kita diperlukan untuk menambahkan baris kode untuk memeriksa `a || b`, mengubah nilai `a = false` dan `b = false`, dan menganalisa perubahan dan perbedaan boolean yang terjadi, kemudian apabila diketahui pernyataan `a || b && a || !b`. Uraikan urutan logika yang akan dikerjakan, dan analisa luaran true atau false dari pernyataan tersebut!

[No. 5] Analisis dan Argumentasi

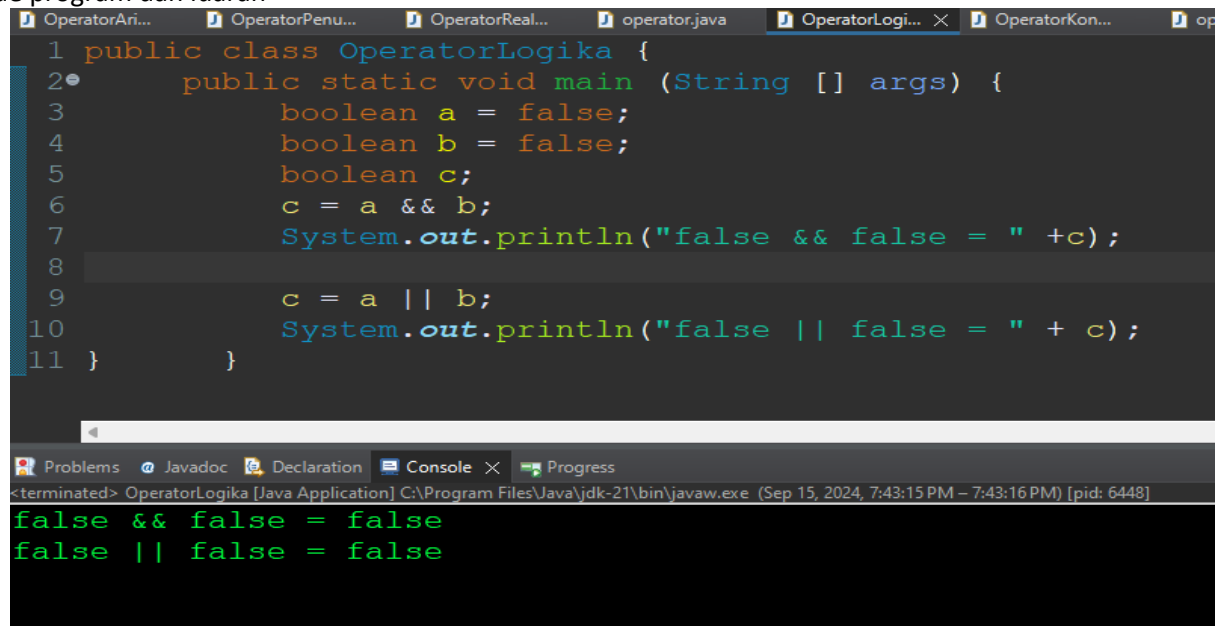
Dalam pernyataan `a || b && a || !b`, evaluasi logika mengikuti prioritas operator di mana negasi (!) dan AND (&&) memiliki prioritas lebih tinggi dibanding OR (||). Dengan nilai `a = false` dan `b = false`, evaluasi dimulai dari `b && a` yang menghasilkan false, dilanjutkan dengan negasi `!b` yang menjadi true. Urutan berikutnya, `a || false` menghasilkan false, namun ekspresi false || true pada bagian akhir memberikan hasil akhir true. Ini karena operator OR hanya memerlukan salah satu operand bernilai true untuk mengembalikan hasil true, sehingga meskipun bagian awal ekspresi bernilai false, hasil akhir tetap true berkat evaluasi `!b`.

[No. 5] Penyusunan Algoritma dan Kode Program

Algoritma:

- (a) Mulai
- (b) Membuka Eclipse
- (c) Membuat Java Project baru
- (d) Membuat class baru di dalamnya
- (e) Memasukan code yang diberikan
- (f) Merubah code sesuai yang diperlukan
- (g) Tekan Run pada Eclipse
- (h) Selesai

Kode program dan luaran



```
1 public class OperatorLogika {
2     public static void main (String [] args) {
3         boolean a = false;
4         boolean b = false;
5         boolean c;
6         c = a && b;
7         System.out.println("false && false = " + c);
8
9         c = a || b;
10        System.out.println("false || false = " + c);
11    }
12 }
```

Problems Javadoc Declaration Console × Progress

<terminated> OperatorLogika [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Sep 15, 2024, 7:43:15 PM – 7:43:16 PM) [pid: 6448]

```
false && false = false
false || false = false
```

Analisa luaran yang dihasilkan:

Kode ini digunakan untuk mengevaluasi ekspresi logika yang melibatkan operator logika seperti AND (&&), OR (||), dan negasi (!) dalam bahasa pemrograman Java. Kode dan Luaran sudah benar dan sesuai aturan.

[No.5] Kesimpulan

Analisa

Operator logika AND (&&), OR (||), dan negasi (!) digunakan untuk mengevaluasi kondisi boolean dalam program, dengan hasil true atau false. Operator AND hanya menghasilkan true jika kedua ekspresi benar, sementara OR membutuhkan salah satu ekspresi bernilai true. Negasi membalikkan nilai boolean. Perbedaan utama antara &&|| dan &| adalah efisiensi, di mana &&|| menghentikan evaluasi ketika hasil sudah pasti.

LATIHAN 6

[No. 6] Identifikasi Masalah:

Pada soal ini kita diperlukan untuk mengubah nilai int dari 80 menjadi 60, kemudian menganalisis hasil dan proses yang terjadi

[No.6] Analisis dan Argumentasi

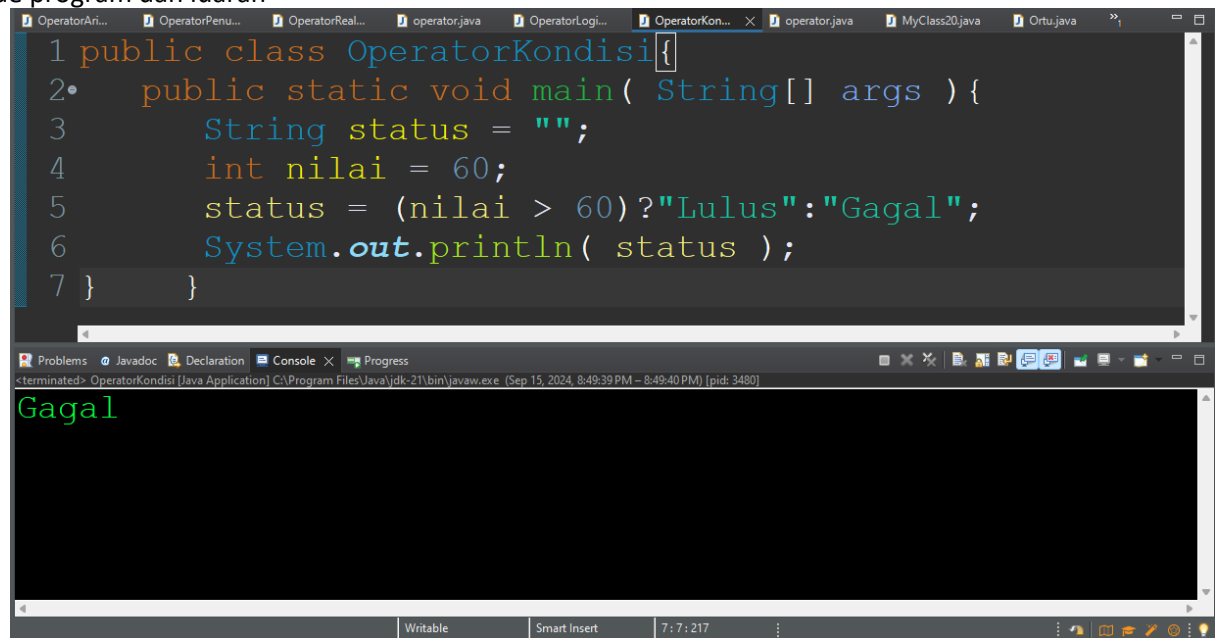
Ketika nilai variabel nilai diubah menjadi 60, program menggunakan operator ternary untuk mengevaluasi kondisi ($\text{nilai} > 60$). Dalam hal ini, kondisi tersebut menghasilkan false karena 60 tidak lebih besar dari 60. Oleh karena itu, bagian yang dieksekusi adalah perintah setelah tanda titik dua (:), yaitu "Gagal". Ternary operator ini bekerja sebagai bentuk singkat dari pernyataan if-else, di mana jika kondisi true, hasilnya adalah "Lulus", dan jika false, hasilnya adalah "Gagal". Dengan demikian, karena kondisi false, program menampilkan "Gagal" sebagai output.

[No.6] Penyusunan Algoritma dan Kode Program

Algoritma:

- (a) Mulai
- (b) Membuka Eclipse
- (c) Membuat Java Project baru
- (d) Membuat class baru di dalamnya
- (e) Memasukkan kode
- (f) Mengubah kode
- (g) Selesai

Kode program dan luaran



The screenshot shows the Eclipse IDE with a Java project. The main editor displays the following code:

```
1 public class OperatorKondisi{
2     public static void main( String[] args ){
3         String status = "";
4         int nilai = 60;
5         status = (nilai > 60)?"Lulus":"Gagal";
6         System.out.println( status );
7     } }
```

The bottom console window shows the output of the program:

```
<terminated> OperatorKondisi [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe. (Sep 15, 2024, 8:49:39 PM - 8:49:40 PM) [pid: 3480]
Gagal
```

Analisa luaran yang dihasilkan:

Program ini adalah sebuah program yang digunakan untuk mengetahui lulus atau tidak dengan menginput bilangan. Code dan Luaran sudah benar.

[No.6] Kesimpulan

Analisa

Operator kondisional atau ternary (?:) adalah cara singkat untuk menyederhanakan pernyataan if else yang hanya memiliki satu perintah di setiap bloknnya. Dengan format (kondisi) ? (jika benar) : (jika salah), operator ini mengevaluasi kondisi dan memilih salah satu dari dua hasil berdasarkan nilai kondisi tersebut. Dalam contoh yang diberikan, ketika nilai nilai diubah menjadi 60, kondisi (nilai > 60) menghasilkan false, sehingga hasilnya adalah "Gagal".

LATIHAN 7

[No. 7] Identifikasi Masalah:

Pada soal ini kita diperlukan untuk memilih 3 perhitungan dari Contoh 7, kemudian uraikan perhitungan biner! Simpulkan hasilnya!

[No.7] Analisis dan Argumentasi

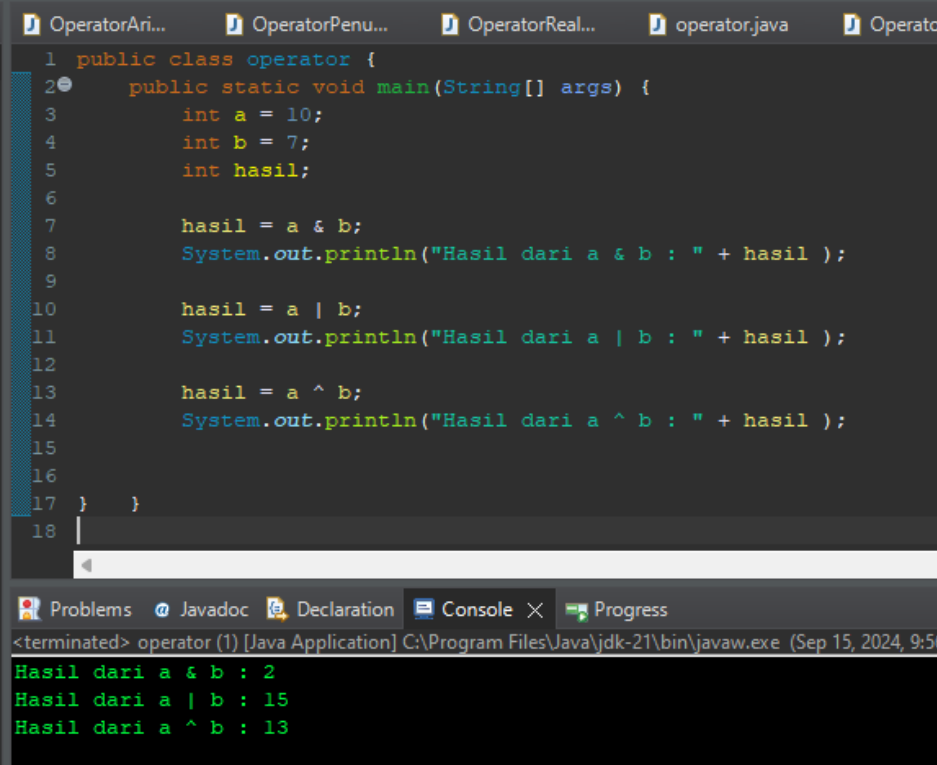
Operasi bitwise seperti AND (&), OR (|), dan XOR (^) mengelola bit-bit dari bilangan biner untuk menghasilkan hasil tertentu. Misalnya, pada $10 \& 7$, bitwise AND menghasilkan 0010 (desimal 2), karena hanya bit ketiga yang sama-sama 1. Untuk OR (|), $10 | 7$ menghasilkan 1111 (desimal 15), karena setidaknya satu dari setiap pasangan bit adalah 1. XOR (^) pada $10 \wedge 7$ menghasilkan 1101 (desimal 13), karena bit yang berbeda menghasilkan 1, sementara bit yang sama menghasilkan 0. Analisis ini menunjukkan bagaimana setiap operator mempengaruhi bit-bit bilangan dan hasil akhir dari operasi tersebut.

[No.7] Penyusunan Algoritma dan Kode Program

Algoritma:

- Mulai
- Membuka Eclipse
- Membuat Java Project baru
- Membuat class baru di dalamnya
- Memasukan code dari E-Learning
- Memilih 3 kode yang ingin digunakan
- Tekan run pada eclipse
- selesai

Kode program dan luaran



```
1 public class operator {
2     public static void main(String[] args) {
3         int a = 10;
4         int b = 7;
5         int hasil;
6
7         hasil = a & b;
8         System.out.println("Hasil dari a & b : " + hasil );
9
10        hasil = a | b;
11        System.out.println("Hasil dari a | b : " + hasil );
12
13        hasil = a ^ b;
14        System.out.println("Hasil dari a ^ b : " + hasil );
15
16    }
17 }
18
```

Problems Javadoc Declaration Console Progress

<terminated> operator (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Sep 15, 2024, 9:5

```
Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
```

Analisa luaran yang dihasilkan

Kode yang menggunakan operator bitwise sering kali digunakan untuk berbagai tujuan dalam pemrograman dan manipulasi data, Luaran dan code sudah benar.

[No.7] Kesimpulan

Analisa

Operator bitwise, seperti AND (&), OR (|), XOR (^), dan NOT (~), serta penggeseran bit (LEFT SHIFT << dan RIGHT SHIFT >>), memungkinkan manipulasi langsung pada bit dalam bilangan biner, menawarkan efisiensi dan kontrol yang lebih mendalam dalam pemrograman. Misalnya, AND (&) menghasilkan 1 hanya jika kedua bit adalah 1, sementara OR (|) menghasilkan 1 jika salah satu bit adalah 1. XOR (^) memberikan 1 jika bit-bit berbeda, dan NOT (~) membalikkan semua bit. Penggeseran bit, seperti LEFT SHIFT (<<), menggeser bit ke kiri, meningkatkan nilai bilangan, sedangkan RIGHT SHIFT (>>) menggeser bit ke kanan, mengurangi nilainya.

Refleksi

Pengalaman saya saat belajar matakuliah computer dan pemrograman terutama materi operator java yaitu saya lumayan mengerti materi yang diberikan tetapi masih ada yang saya kurang mengerti jadi saya akan terus belajar dengan giat lagi.