

Nama & NPM	Topik:	Tanggal:
Marchellino Madani Putra G1F024064	Operator Aritmatika	06 september 2024

[No. 1] Identifikasi Masalah:

```
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));
    } }
```

Luaran:

```
a: 20
b: 3
a - b = 17
```

Latihan 1.

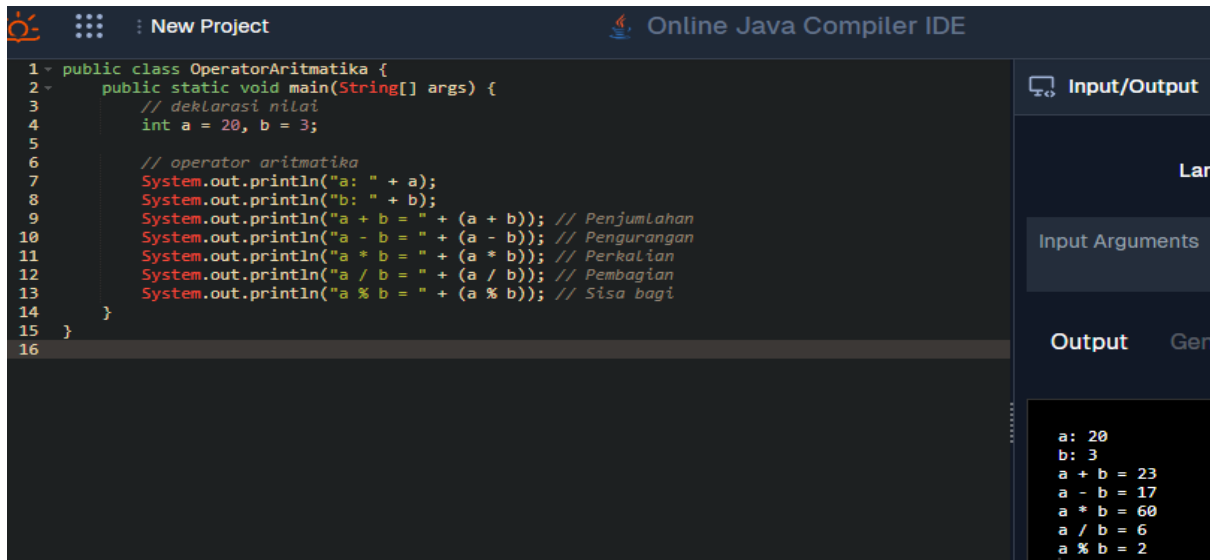
- 1.1. Tambahkan baris `System.out.println("a + b = " + (a + b));` Ubahlah operator (+) dengan tanda (-, *, /, %)
- 1.2. Analisa perhitungan matematika yang terjadi!

[No.1] Analisis dan Argumentasi

- 1) Saya melakukan deklarasi nilai dengan yang sesuai pada soal kemudian menambahkan baris `System.out.println`. dengan mengganti operator atau perintah sesuai dengan yang ada pada soal kemudian melakukan mencetak nilai tersebut agar dapat melihat outputnya sesuai atau tidak dengan yang di perintahkan pada kodenya.
- 2) Anilisis perhitungan matematika output yang di tampilkan sesuai yang pertama adalah melakukan penjumlahan,kemudian pengurangan, perkalian,pembagian dan hasil bagi. Operasi sisa bagi menggunakan oeprasi modulus untuk mendapatkan hasil yang sesuai pembagian menghasilkan 6 karena hasil pembagian dibulatkan ke bawah.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Deklarasi nilai
 - (c) Rubah perintah/operator sesuai soal
 - (d) Cetak nilai
 - selesai
- 2) Kode program dan luaran



The screenshot shows an IDE window titled "New Project" and "Online Java Compiler IDE". The code editor contains the following Java code:

```
1 public class OperatorAritmatika {
2     public static void main(String[] args) {
3         // deklarasi nilai
4         int a = 20, b = 3;
5
6         // operator aritmatika
7         System.out.println("a: " + a);
8         System.out.println("b: " + b);
9         System.out.println("a + b = " + (a + b)); // Penjumlahan
10        System.out.println("a - b = " + (a - b)); // Pengurangan
11        System.out.println("a * b = " + (a * b)); // Perkalian
12        System.out.println("a / b = " + (a / b)); // Pembagian
13        System.out.println("a % b = " + (a % b)); // Sisa bagi
14    }
15 }
16
```

On the right side, there is a panel with tabs for "Input/Output", "Input Arguments", and "Output". The "Output" tab is selected, showing the following output:

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

Saya melakukan deklarasi nilai dan inisial pada variabel dengan yang sesuai pada soal kemudian menambahkan baris `System.out.println`. dengan mengganti operator atau perintah sesuai dengan yang ada pada soal. lakukan operasi aritmatika, setiap baris melakukan operasi aritmatika yang berbeda dan mencetak hasilnya. Setiap baris output memberikan informasi tentang nilai hasil dari operasi aritmatika yang dilakukan dalam program, output yang keluar mencerminkan bagaimana proses yang dilakukan dalam program.

[No.1] Kesimpulan

1) Analisa

Program ini menunjukkan penggunaan operator aritmatika dalam Java. Dengan melakukan atau mencoba proses pemrograman ini, kita dapat melihat bagaimana variabel bekerja dan bagaimana cara mendeklarasikan nilai pada program. Dengan memahami proses aritmatika pada program ini dapat membuat kerja dalam program menjadi lebih efektif, hasil keluaran juga sesuai dengan logika nya.

Nama & NPM	Topik:	Tanggal:
Marchellino Madani Putra G1F024064	Operator Penugasan	06 september 2024

[No. 2] Identifikasi Masalah:

```
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);

        // pengurangan
        b -= a;
        System.out.println("Pengurangan : " + b);

        // perkalian
        b *= a;
        System.out.println("Perkalian : " + b);

        // Pembagian
        b /= a;
        System.out.println("Pembagian : " + b);

        // Sisa bagi
        b %= a;
        // sekarang b=0
        System.out.println("Sisa Bagi: " + b);
    }
}
```

Luaran:

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

Latihan 2.

2.1. Bandingkan hasil Contoh 1 dengan Contoh 2!

[No.2] Analisis dan Argumentasi

- 1) Perbedaan aritmatika dengan penugasan adalah jika pada proses aritmatika nilai baru yang kita tambahkan pada variable akan terkunci, tetapi jika kita menambahkan nilai terbaru lagi pada variable yang sama maka nilai yang barusan kita tambahkan akan menggantikan nilai yang sebelumnya terkunci.
- 2) Pada proses penugasan kita menambahkan nilai pada variable, nilai yang sebelumnya kita tambahkan tidak akan tergantikan tetapi akan menambah atau mengurangi sesuai dengan kriteria penugasannya.

[No.2] Penyusunan Algoritma dan Kode Program

- 3) Algoritma
 - (a) Mulai
 - (b) Deklarasi nilai
 - (b) Kriteria penugasan
 - (c) Cetak nilai
 - (d) selesai
- 4) 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         b -= a;
9         System.out.println("Pengurangan : " + b);
10        b *= a;
11        System.out.println("Perkalian : " + b);
12        b /= a;
13        System.out.println("Pembagian : " + b);
14        b %= a;
15        System.out.println("Sisa Bagi: " + b);
16    }
17 }
18
19
```

Want me to explain code logic?
[Explain Code](#) [Not Now](#)

Input/Output

Language

Input Arguments

Output Generated

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

[No.2] Kesimpulan

2) Analisa

Materi tentang operator penugasan yang kita pelajari ini merupakan dasar yang penting untuk memahami cara kerja variabel dan proses manipulasi nilai. Operator penugasan yang di gunakan seperti +=, -=, *=, /=, dan %= yang kita gunakan untuk mengubah nilai variabel dengan cara yang lebih ringkas dan efisien. Memahami penggunaan operator ini membantu dalam penulisan kode yang lebih rapi dan efektif untuk di pahami.

Nama & NPM	Topik:	Tanggal:
Marchellino Madani Putra G1F024064	Operator Relasional	06 september 2024

[No. 3] Identifikasi Masalah:

Operator Relasional: <, >, <=, >=, =, ==, !=

Operator Relasi untuk menguji hubungan antara nilai dan atau variabel dan selalu menghasilkan nilai true atau false.

Contoh 3: Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle

```
public class OperatorRealasional {
    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("Hasil A > B = "+ hasil);

        // apakah A lebih kecil dari B?
        hasil = nilaiA < nilaiB;
        System.out.println("Hasil A < B = "+ hasil);

        // apakah A lebih besar samadengan B?
        hasil = nilaiA >= nilaiB;
        System.out.println("Hasil A >= B = "+ hasil);

        // apakah A lebih kecil samadengan B?
        hasil = nilaiA <= nilaiB;
        System.out.println("Hasil A <= B = "+ hasil);

        // apakah nilai A sama dengan B?
        hasil = nilaiA == nilaiB;
        System.out.println("Hasil A == B = "+ hasil);

        // apakah nilai A tidak samadengan B?
        hasil = nilaiA != nilaiB;
        System.out.println("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 3

3.1. Ubahlah nilai A = 4 dan B = 4. Analisa perubahan yang terjadi!

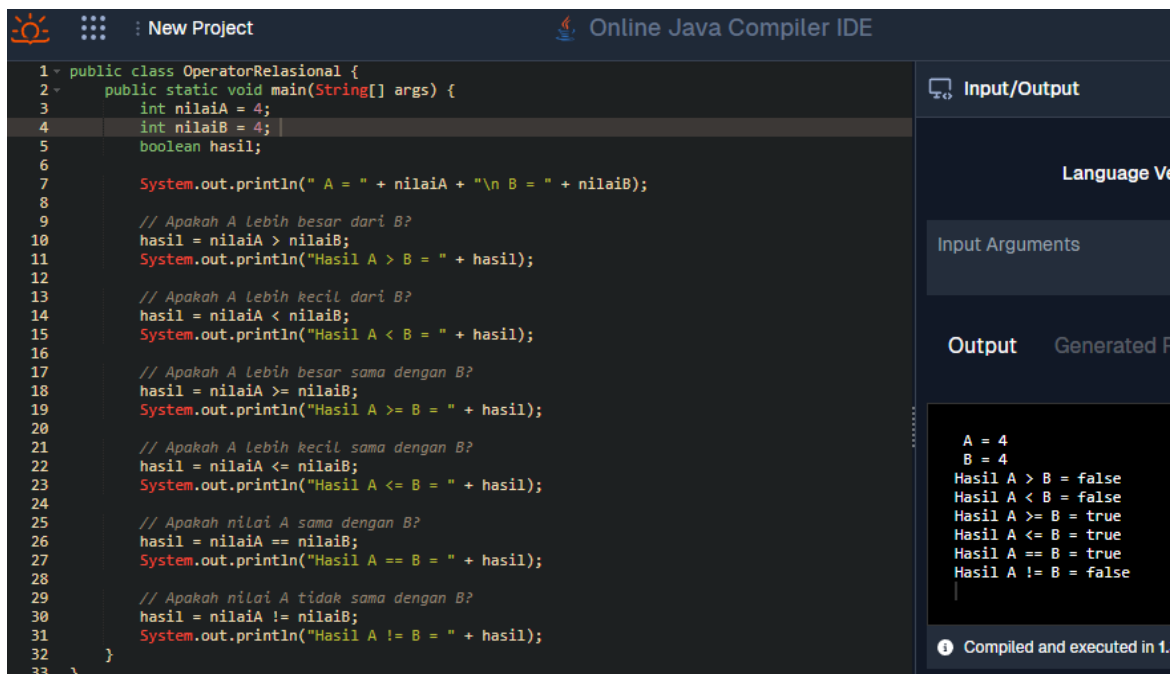
3.2 Bandingkan bagaimana perbedaan nilai A dan B mempengaruhi nilai luaran!

[No.3] Analisis dan Argumentasi

- 1) Hal yang harus dilakukan adalah, melakukan perubahan nilai (A) yang di samakan dengan nilai (B) yaitu 4. Hasil output yang awalnya true akan berubah menjadi false karena adanya perubahan pada nilai (A) begitu pun sebaliknya.
- 2) Ketika kita melakukan perubahan kedua nilai dari (A) dan (B) menjadi sama, hasil dari output sama dengan (==) menjadi true dan tidak sama dengan (!=) menjadi false, yang berarti kondisi ini berubah.

[No.3] Penyusunan Algoritma dan Kode Program

- 5) Algoritma
 - (a) Mulai
 - (b) Deklarasi nilai
 - (c) lakukan perubahan pada nilai A
 - (d) cetak nilai
 - (e) selesai
- 6) Kode program dan luaran



The screenshot shows an online Java compiler interface. On the left, the code editor contains a Java program named `OperatorRelasional`. The program declares two integer variables, `nilaiA` and `nilaiB`, both initialized to 4. It then performs a series of relational operations and prints the results. On the right, the 'Output' panel displays the program's execution results.

```
1 public class OperatorRelasional {
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
9         // Apakah A lebih besar dari B?
10        hasil = nilaiA > nilaiB;
11        System.out.println("Hasil A > B = " + hasil);
12
13        // Apakah A lebih kecil dari B?
14        hasil = nilaiA < nilaiB;
15        System.out.println("Hasil A < B = " + hasil);
16
17        // Apakah A lebih besar sama dengan B?
18        hasil = nilaiA >= nilaiB;
19        System.out.println("Hasil A >= B = " + hasil);
20
21        // Apakah A lebih kecil sama dengan B?
22        hasil = nilaiA <= nilaiB;
23        System.out.println("Hasil A <= B = " + hasil);
24
25        // Apakah nilai A sama dengan B?
26        hasil = nilaiA == nilaiB;
27        System.out.println("Hasil A == B = " + hasil);
28
29        // Apakah nilai A tidak sama dengan B?
30        hasil = nilaiA != nilaiB;
31        System.out.println("Hasil A != B = " + hasil);
32    }
33 }
```

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
```

Compiled and executed in 1.0s

Saya melakukan perubahan pada proses deklarasi nilai yang awalnya nilai (A) ini 12 di rubah menjadi 4 sama seperti nilai (B). Yang menyebabkan terjadinya perubahan pada output antara nilai tersebut true atau false, hal ini lah yang mempengaruhi perubahan pada output.

[No.3] Kesimpulan

3) Analisa

Program ini menunjukkan perubahan nilai yang mempengaruhi pernyataan pada output. Hal inilah yang membuat program mengevaluasi perbedaan yang terjadi, dengan memahami bagaimana nilai mempengaruhi hasil pada output, kita dapat lebih baik dalam menulis kondisi logika dalam pemrograman.

Nama & NPM	Topik:	Tanggal:
Marchellino Madani Putra G1F024064	Operator Logika	06 september 2024

[No. 4] Identifikasi Masalah:

```

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);
    }
}

```

Luaran:

```

# 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
```

Latihan 4.

4.1. Berdasarkan luaran program Contoh 4, bandingkan hasil Post dan Pre untuk Increment dan Decrement!

[No.4] Analisis dan Argumentasi

- 1) Pada hasil output post increment nilai di inialisasikan dengan variable a, nilai dari variable a adalah 10 pada proses post increment nilai variable akan di tambahkan 1. Pada proses post increment kita tidak bisa langsung menampilkan nilai hasilnya yaitu 11, karena menampilkan variable atau nilai sebelumnya terlebih dahulu. Untuk menambahkan nilai variable perlu di buatkan baris atau system.out.println yang baru. Begitu juga pada proses post decrement hanya saja jika decrement nilai pada variable di kurangi 1, nilai decrement di inialisasikan dengan variable c.
- 2) Pada hasil output pre increment nilai di inialisasikan dengan variable b, nilai dari variable b adalah 10 pada proses pre increment nilai variable akan di tambahkan 1. Proses pre increment kita bisa langsung menampilkan nilai hasilnya yaitu 11. jika menambahkan baris atau system.out.println yang baru nilai dari variable tidak akan berubah. Begitu juga pada proses pre decrement hanya saja jika decrement nilai pada variable di kurangi 1, nilai decrement di inialisasikan dengan variable d.
- 3)

[No.4] Penyusunan Algoritma dan Kode Program

- 7) Algoritma
 - (a) Mulai
 - (b) Deklarasi nilai dan variable dengan post increment dan decrement, pre increment dan decrement
 - (c) cetak nilai
 - (d) selesai
- 8) Kode program dan luaran

The screenshot shows a Java IDE with a code editor on the left and an 'Input/Output' panel on the right. The code defines a class 'operator' with a 'main' method. It demonstrates four types of increment and decrement operations: post-increment on variable 'a', pre-increment on variable 'b', post-decrement on variable 'c', and pre-decrement on variable 'd'. Each operation is followed by a print statement to show the value before and after the operation.

```
1 public class operator {
2     public static void main(String[] args) {
3         int a = 10;
4         System.out.println("# Post Increment #");
5         System.out.println("=====");
6         System.out.println("Isi variabel a: " + a);
7         System.out.println("Isi variabel a: " + ++a);
8         System.out.println("Isi variabel a: " + a);
9
10        System.out.println();
11
12        int b = 10;
13        System.out.println("# Pre Increment #");
14        System.out.println("=====");
15        System.out.println("Isi variabel b: " + b);
16        System.out.println("Isi variabel b: " + ++b);
17        System.out.println("Isi variabel b: " + b);
18
19        System.out.println();
20
21        int c = 10;
22        System.out.println("# Post Decrement #");
23        System.out.println("=====");
24        System.out.println("Isi variabel c: " + c);
25        System.out.println("Isi variabel c: " + c--);
26        System.out.println("Isi variabel c: " + c);
27
28        System.out.println();
29
30        int d = 10;
31        System.out.println("# Pre Decrement #");
32        System.out.println("=====");
33        System.out.println("Isi variabel d: " + d);
34        System.out.println("Isi variabel d: " + --d);
35        System.out.println("Isi variabel d: " + d);
36    }
37 }
```

The 'Input/Output' panel on the right shows the following output:

```
# 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
```

At the bottom of the IDE, a status bar indicates 'Compiled and executed'.

Saya melakukan proses analisis perbedaan antara post increment dan decrement, pre increment dan decrement.

[No.4] Kesimpulan

4) Analisa

Program ini menunjukkan cara membedakan luaran atau output perbedaan antara post dan pre. Perbedaan ini menunjukkan bagaimana urutan proses eksekusi memengaruhi nilai yang ditampilkan pada pemrograman. Dalam operasi post, nilai ditampilkan sebelum perubahan, sedangkan dalam operasi pre, nilai ditampilkan setelah perubahan.

Nama & NPM	Topik:	Tanggal:
Marchellino Madani Putra G1F024064	Operator Relasional	06 september 2024

[No. 5] Identifikasi Masalah:

```
public class OperatorLogika {
    public static void main (String [] args) {
        boolean a = true;
        boolean b = false;
        boolean c;
        c = a && b;
        System.out.println("true && false = " +c);
    }
}
```

Luaran:

```
true && false = false
```

Latihan 5

- 5.1. Tambahkan baris kode untuk memeriksa a || b.
- 5.2. Ubahlah nilai a = false dan b = false. Analisa perubahan dan perbedaan boolean yang terjadi!
- 5.2. Apabila diketahui pernyataan a || b && a || !b. Uraikan urutan logika yang akan dikerjakan! Analisa luaran true atau false dari pernyataan tersebut!

[No.5] Analisis dan Argumentasi

- 1) Menamba baris kode untuk memeriksa a OR b, seperti yang kita ketahu bahwa inisialisasi dari variable a adalah true variable b adalah false, yang menampilkan output a || b = true.
- 2) Proses perubahan nilai a = false dan b = false yang terjadi pada Boolean akan menampilkan output yang berbeda karena perubahan dalam inisialisasi variable.dan di lakukan proses pengoperasian ulang. Sesudah keduanya menjadi false, hasil false && false tetap false, karena AND hanya menghasilkan true jika keduanya adalah true. Sesudah keduanya menjadi false, hasil false || false adalah false. OR hanya menghasilkan true jika salah satu adalah true.
- 3) Lalu kita ganti nilai dari a dan b ke dalam ekspresi yang menjadi !b atau !false = true, a || b && a || !b = true

[No.5] Penyusunan Algoritma dan Kode Program

- 9) Algoritma
 - (a) Mulai
 - (b) Deklarasi variabel
 - (c) Menambah baris kode memeriksa OR
 - (d) Mengubah variable menjadi false
 - (e) Cetak nilai
 - (f) Selesai

10) Kode program dan keluaran

```
1 public class OperatorLogika {
2     public static void main(String[] args) {
3         boolean a = true;
4         boolean b = false;
5         boolean c;
6
7         c = a && b;
8         System.out.println("a && b = " + c);
9
10        c = a || b;
11        System.out.println("a || b = " + c);
12
13        a = false;
14        b = false;
15
16        c = a && b;
17        System.out.println("a && b = " + c);
18
19        c = a || b;
20        System.out.println("a || b = " + c);
21
22        c = a || b && a || !b;
23        System.out.println("b || b && b || !b = " + c);
24    }
25 }
```

Input/Output

Language Ver

Input Arguments

Output Generated F

```
a && b = false
a || b = true
a && b = false
a || b = false
b || b && b || !b = true
```

- 4) Saya melakukan penambahan baris kode untuk memeriksa OR, hasil output yang di keluarkan adalah true karena true || false adalah true. Dalam OR jika salah satu true maka akan mengeluarkan true, sedangkan AND akan mengeluarkan hasil true jika keduanya adalah true. Lalu melakukan perubahan nilai a = false dan b = false yang terjadi pada Boolean akan menampilkan output yang berbeda karena perubahan dalam inisialisasi variable. dan di lakukan proses pengoperasian ulang. Lalu kita ganti nilai dari a dan b ke dalam ekspresi yang menjadi !b atau !false = true, a || b && a || !b = true.

[No.5] Kesimpulan

5) Analisa

Dengan demikian, perubahan nilai dari true menjadi false pada kedua variabel mengubah hasil dari operasi logika sesuai dengan aturan. Mencoba berbagai kombinasi nilai untuk true dan false pada operator AND dan OR supaya dapat memahami hasil dan perbedaannya dengan baik.

Nama & NPM	Topik:	Tanggal:
Marchellino Madani Putra G1F024064	OperatorKondisional(Ternary)	06 september 2024

[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 6

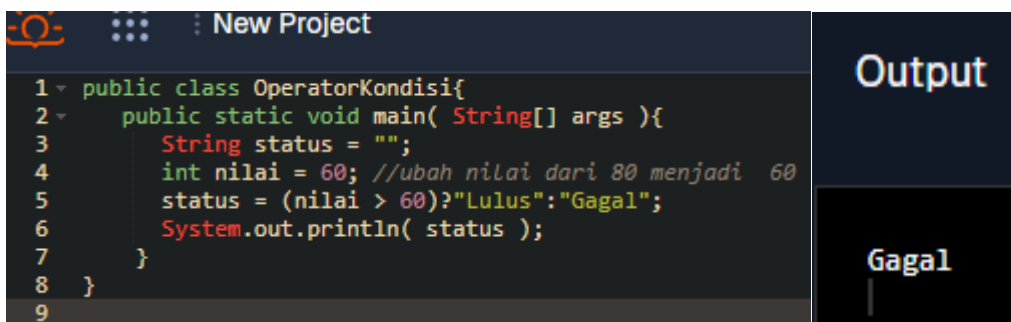
Berdasarkan Contoh 6, ubahlah nilai = 60. Analisis hasil dan proses yang terjadi!

[No.6] Analisis dan Argumentasi

Melakukan perubahan nilai int dari 80 ke 60 hal ini dilakukan untuk melihat perubahan yang terjadi apabila saat proses penilaian apakah nilai tersebut termasuk kriteria lulus atau gagal. Jika nilai yang kita masukkan lebih besar dari kriterianya yaitu 60 maka hasilnya akan lulus tetapi jika kurang dari 60 maka hasilnya akan gagal.

[No.6] Penyusunan Algoritma dan Kode Program

- 11) Algoritma
 - (a) Mulai
 - (b) Deklarasi variabel
 - (c) proses operator ternary
 - (d) Cetak nilai
 - (e) Selesai
- 12) Kode program dan luaran



```
1 public class OperatorKondisi{
2     public static void main( String[] args ){
3         String status = "";
4         int nilai = 60; //ubah nilai dari 80 menjadi 60
5         status = (nilai > 60)?"Lulus":"Gagal";
6         System.out.println( status );
7     }
8 }
9
```

Output

Gagal

Saya melakukan perubahan dari nilai variable yang awalnya bernilai 80 di rubah menjadi 60 untuk melihat proses pada operator kondisional (Ternary). Pada soal awal hasil output adalah lulus, dan ketika dilakukan perubahan nilai pada variable menjadi 60 hasilnya menjadi gagal, hal ini disebabkan karena jika nilai dari variable yang kita masukkan lebih besar dari kriterianya maka hasil outputnya akan menjadi lulus namun sebaliknya jika nilai kurang dari 60 maka hasil output akan menimbulkan gagal.

[No.6] Kesimpulan

6) Analisa

Proses operator kondisional atau (Ternary) ini adalah proses yang melihat kondisi dari variable yang akan menimbulkan lulus atau gagal sesuai dengan kriteria nilai yang diberikan. Oleh karena itu, evaluasi kondisi sangat penting dalam menentukan hasil dan perlu disesuaikan sesuai dengan yang diinginkan.

Nama & NPM	Topik:	Tanggal:
Marchellino Madani Putra G1F024064	Operator Bitwise	06 september 2024

[No. 7] Identifikasi Masalah:

```
public class operator {
    public static void main(String[] args) {
        int a = 10;
        int b = 7;
        int hasil;

        hasil = a & b;
        System.out.println("Hasil dari a & b : " + hasil );

        hasil = a | b;
        System.out.println("Hasil dari a | b : " + hasil );

        hasil = a ^ b;
        System.out.println("Hasil dari a ^ b : " + hasil );

        hasil = ~a;
        System.out.println("Hasil dari ~a : " + hasil );

        hasil = a >> 1;
        System.out.println("Hasil dari a >> 1 : " + hasil );

        hasil = b << 2;
        System.out.println("Hasil dari b << 2 : " + hasil );
    }
}
```

Luaran:

```
Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
Hasil dari ~a : -11
Hasil dari a >> 1 : 5
Hasil dari b << 2 : 28
```

Latihan 7

Pilihlah 3 perhitungan Contoh 7, kemudian uraikan perhitungan biner! Simpulkan hasilnya!

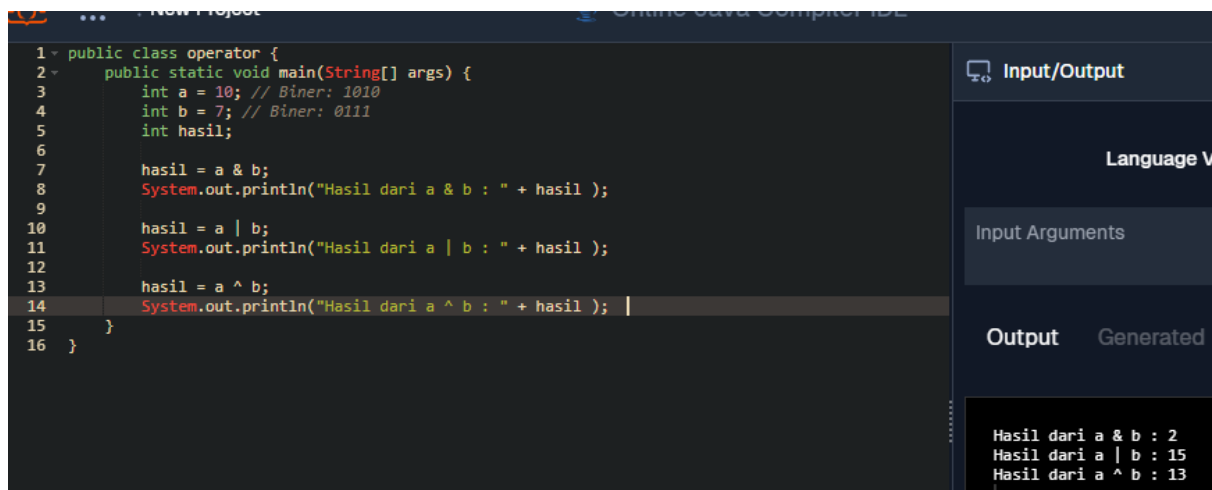
[No.7] Analisis dan Argumentasi

Pada contoh 7 memilih 3 perhitungan, perhitungan yang di pilih adalah AND,OR,XOR. Yang menghasilkan output 2,15,13 nilai awal variable a =10 binernya 1010 variabel b=7 binernya 0111 lalu lakukan perhitungan AND, ketika sudah mendapatkan hasil biner, konversi hasil biner tersebut menjadi decimal. Angka hasil konversi dari biner ke decimal itulah yang akan menjadi output begitu juga pada proses perhitungan OR dan XOR.

[No.7] Penyusunan Algoritma dan Kode Program

- 13) Algoritma
 - (a) Mulai
 - (b) Deklarasi variabel
 - (c) perhitungan biner pada setiap variabel
 - (d) Cetak nilai
 - (e) Selesai

14) Kode program dan luaran



```
1 public class operator {
2     public static void main(String[] args) {
3         int a = 10; // Biner: 1010
4         int b = 7; // Biner: 0111
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 }
```

Input/Output

Language V

Input Arguments

Output Generated

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

Saya memilih 3 perhitungan yaitu AND, OR dan XOR . Yang menghasilkan output 2, 15 dan 13 nilai awal variable a =10 binernya 1010 variabel b=7 binernya 0111 lalu lakukan perhitungan biner pada AND, yang hasil binernya adalah 0010 ketika sudah mengetahui nilai dari biner konversi hasil biner tersebut menjadi decimal. Angka hasil konversi dari biner ke decimal itulah yang akan menjadi output begitu juga pada proses perhitungan OR dan XOR. Cara merubah biner ke decimal adalah dalam sistem biner, setiap bit memiliki bobot yang merupakan pangkat dari 2, berdasarkan posisinya dari kanan ke kiri, dimulai dari 0. Sekarang kita akan menghitung nilai desimal dengan mengalikan setiap bit dengan bobotnya, lalu hasil dari perkalian di jumlahkan dan hasil penjumlahan itulah hasil dari konversi biner ke decimal.

[No.7] Kesimpulan

7) Analisa

Operator bitwise adalah alat yang kuat dalam pemrograman yang memungkinkan untuk melakukan manipulasi data dengan efisien. Memahami cara kerja operator ini penting untuk pengembangan perangkat lunak yang optimal dan efektif, operator bitwise juga dapat memberikan cara yang efisien untuk menyelesaikan berbagai masalah dalam pengolahan data dan algoritma.