

Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Anggita Afriyani G1F024011	Operasi	12 September 2024

[No. 1] Identifikasi Masalah:

- Uraikan permasalahan dan variabel

Contoh:

Tuliskan kembali 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
    } }
```

Luaran:

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

1.1. Rekomendasikan perbaikan kode agar program Contoh 1 dapat berjalan!

1.2. Tambahkan baris untuk menampilkan perhitungan dengan operator (-, *, /, %) pada Contoh 1!

Pada soal masih ada pesan kesalahan seperti kurangnya tanda kurung kurawal untuk mengelompokkan ekspresi aritmatika. Lalu, ada tanda spasi yang tidak tepat sehingga menyebabkan kesalahan sintaks.

[No.1] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara memperbaiki beberapa sintaks yang belum benar, seperti penggunaan tanda kurung kurawal yang kurang tepat maupun tanda spasi.
- Alasan solusi ini karena dalam mengelompokkan ekspresi aritmatika yang kompleks, memerlukan tanda kurung yang sesuai serta dengan adanya tanda spasi yang tidak diperlukan dapat menjadi penyebab kesalahan sintaks.
- Perbaiki kode program dengan cara memisahkan variabel. `a` `b`. Gunakan tanda kurawal untuk memetakan ekspresi aritmatika yang benar.lalu, tambahkan baris yang menampilkan hasil penjumlahan dan lainnya.

[No.1] Penyusunan Algoritma dan Kode Program

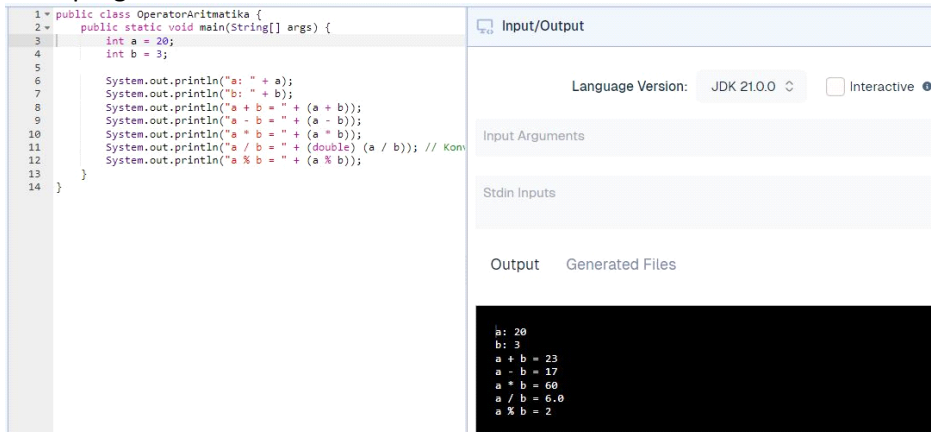
- Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

Misalkan algoritma menyelesaikan masalah:

- Deklarasi variabel
- Menampilkan nilai awal
- penjumlahan
- pengurangan
- perkalian
- pembagian
- sisa pembagian
- run
- hasil luaran

- 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 'OperatorAritmatika' with a 'main' method that declares variables 'a' and 'b', prints their initial values, and then performs arithmetic operations (addition, subtraction, multiplication, division, and modulus) before printing the results. The 'Output' tab in the right panel shows the execution results, which match the code's output statements.

```
1 public class OperatorAritmatika {  
2     public static void main(String[] args) {  
3         int a = 20;  
4         int b = 3;  
5  
6         System.out.println("a: " + a);  
7         System.out.println("b: " + b);  
8         System.out.println("a + b = " + (a + b));  
9         System.out.println("a - b = " + (a - b));  
10        System.out.println("a * b = " + (a * b));  
11        System.out.println("a / b = " + (double) (a / b)); // Kom  
12        System.out.println("a % b = " + (a % b));  
13    }  
14 }
```

Input/Output

Language Version: JDK 21.0.0 ☐ Interactive ⓘ

Input Arguments

Stdin Inputs

Output Generated Files

```
a: 20  
b: 3  
a + b = 23  
a - b = 17  
a * b = 60  
a / b = 6.666666666666667  
a % b = 1
```

- Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

```
public class OperatorAritmatika {  
    public static void main(String[] args) {  
        // Deklarasi nilai  
        int a = 20;  
        int b = 3;  
  
        // Menampilkan nilai awal  
        System.out.println("a: " + a);  
        System.out.println("b: " + b);  
  
        // Menampilkan hasil penjumlahan  
        System.out.println("a + b = " + (a + b));  
  
        // Menampilkan hasil pengurangan  
        System.out.println("a - b = " + (a - b));  
  
        // Menampilkan hasil perkalian  
        System.out.println("a * b = " + (a * b));  
  
        // Menampilkan hasil pembagian
```

```
System.out.println("a / b = " + (double) (a / b)); // Konversi ke double untuk menghindari pembulatan
```

```
// Menampilkan hasil modulus (sisa pembagian)
System.out.println("a % b = " + (a % b));
}
}
```

- Analisa luaran yang dihasilkan
Setelah kode di atas diperbaiki, maka luaran akan berjalan dengan benar dan menampilkan hasil dari berbagai operasi aritmatika.

[No.1] Kesimpulan

Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

jawaban Analisa:

Pada program itu saya menggunakan bentuk kelas public karena sebuah struktur java agar dapat memiliki akses publik memerlukan kelas utama.

Perbaiki program dengan menambahkan beberapa operasi aritmatika lainnya karena struktur java mengharuskan penggunaan tanda kurung kurawal yang tepat agar menghindari kesalahan sintaks.

Dasar alasan pengambilan keputusan pada kasus ini ialah kesalahan sintaks pada kode asli. sehingga saya memperbaiki kode untuk membuat program lebih lengkap. Dengan perbaikan kode di atas, berarti telah mengatasi kesalahan sintaks dan membuat berjalannya program seperti yang diinginkan.

[No. 2] Identifikasi Masalah:

- Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```
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 2.

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
Pada soal masih ada pesan kesalahan sintaks pada variabel, pada operator penugasan dan print statement.

[No.2] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara memberikan tanda kurung kurawal yang sesuai dan memperhatikan spasi yang tidak perlu.
- Alasan solusi ini karena untuk mengelompokkan ekspresi aritmatika yang kompleks memerlukan tanda kurung kurawal yang tepat.
- Perbaiki kode program dengan cara menggunakan tanda kurung kurawal sesuai tempatnya, menghapus spasi yang tak perlu agar program sintaks menjadi benar.

[No.2] Penyusunan Algoritma dan Kode Program

- Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - Deklarasi variabel
 - Menampilkan nilai awal
 - Operasi penugasan
 - Menampilkan hasil operasi.

- 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 in the editor is as follows:

```

1 public class OperatorPenugasan {
2     public static void main(String[] args) {
3         // Deklarasi nilai
4         int a = 20, b = 3;
5
6         // Operator penugasan
7         b += a; // Melakukan perhitungan penjumlahan
8         System.out.println("Penambahan : " + b);
9
10        // Operator penugasan lainnya
11        b -= a; // Melakukan perhitungan pengurangan
12        System.out.println("Pengurangan : " + b);
13
14        b *= a; // Melakukan perhitungan perkalian
15        System.out.println("Perkalian : " + b);
16
17        b /= a; // Melakukan perhitungan pembagian (dengan menghindari pembulatan)
18        System.out.println("Pembagian : " + ((double) b / a));
19
20        b %= a; // Melakukan perhitungan modulus (sisa pembagian)
21        System.out.println("Modulus : " + b);
22    }
23 }
  
```

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

```

Penambahan : 23
Pengurangan : 3
Perkalian : 60
Pembagian : 0.15
Modulus : 3
  
```

At the bottom of the panel, it shows performance metrics: CPU Time: 0.08 sec(s) | Memory: 39096 kilobyte(s) | C

- Screenshot/ Capture potongan kode dan hasil luaran

Komentar:

```

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);

        // Operator penugasan lainnya
        b -= a; // Melakukan perhitungan pengurangan
        System.out.println("Pengurangan : " + b);

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

```
b /= a; // Melakukan perhitungan pembagian (dengan menghindari pembulatan)
System.out.println("Pembagian : " + ((double) b / a));
```

```
b %= a; // Melakukan perhitungan modulus (sisa pembagian)
System.out.println("Modulus : " + b);
```

```
}
```

```
}
```

- Analisa luaran yang dihasilkan

Perbedaan Luaran:

Penambahan (+=) : Hasilnya adalah , karena $.233 + 20 = 23$

Pengurangan (-=) : Hasilnya adalah , karena $.123 - 20 = 3$

Perkalian (*=) : Hasilnya adalah , karena $.603 * 20 = 60$

Pembagian (/=) : Hasilnya adalah , karena $.360 / 20 = 3$

Modulus (%=) : Hasilnya adalah , karena sisa dari Pembagian adalah $.160 / 200$

Waktu Eksekusi:

Semua operasi penugasan (, , ,) memiliki waktu eksekusi yang relatif sama karena mereka semua melakukan operasi aritmatika yang sederhana pada nilai yang sudah ditentukan sebelumnya. Namun, operasi Pembagian (/) dan modulus (%) mungkin sedikit lebih lambat karena mereka melibatkan pembagian dan pengambilan sisa hasil pembagian. += -

-= *= /= %= /= %=

Secara keseluruhan, perbedaan utama antara contoh 1 dan contoh 2 adalah jenis operasi yang digunakan dan hasil yang diperoleh. Contoh 2 menambahkan beberapa jenis operasi penugasan lainnya untuk menunjukkan cara menggunakan berbagai operator dalam Java.

[No.2] Kesimpulan

Analisa

perubahan nilai dan menjadi 4 menyebabkan semua operasi relasional menghasilkan nilai kecuali untuk operasi , yang menghasilkan nilai . `ABtrue!=`false``

[No. 3] Identifikasi Masalah:

- Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```
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 3

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

Pada soal masih ada pesan kesalahan seperti belumlah terstruktur komentar yang diberikan.

[No.3] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan komentar, struktur yang lebih rapi, dan konsistensi.
- Alasan solusi ini karena hal hal yang dapat diperbaiki ini bisa menjadi kesalahan yang signifikan dalam sintaks.
- Perbaiki kode program dengan cara menambahkan komentar dan struktur yang lebih rapi

[No.3] Penyusunan Algoritma dan Kode Program

- Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
Misalkan algoritma memasak mi instan:
Algoritma adalah langkah-langkah penyelesaian masalah.
- Deklarasi variabel

- Menampilkan nilai awal
- Operasi penugasan
- Menampilkan hasil operasi.

- Kode program dan luaran

The screenshot shows a Java program in an IDE. The code defines a class `OperatorRelasional` with a `main` method. It initializes two integer variables, `nilaiA` and `nilaiB`, both to 4. It then performs various relational operations and prints the results. The output window on the right shows the following results:

```

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

```

- Screenshot/ Capture potongan kode dan hasil luaran
Komentar:

```

public class OperatorRelasional {
    public static void main(String[] args) {
        int nilaiA = 4; // Ubah nilai A menjadi 4
        int nilaiB = 4; // Ubah nilai B menjadi 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 atau sama dengan B?
        hasil = nilaiA >= nilaiB;
        System.out.println("\n Hasil A >= B = " + hasil);

        // Apakah A lebih kecil atau sama dengan 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 sama dengan B?

```

```

        hasil = nilaiA != nilaiB;
        System.out.println("\n Hasil A != B = " + hasil);
    }
}

```

- Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

[No.3] Kesimpulan

- **Analisa**

Operator relasional dalam java digunakan untuk membandingkan dua operan dan menghasilkan nilai boolean. Sistem ini berfungsi sebagaimana aturan logika yang diharapkan, serta konsisten.

[No. 4] Identifikasi Masalah:

- Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```

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 4.

4.1. Berikan saran operasi apa yang diperlukan (pre/post increment, pre/post decrement) agar Contoh 4 menghasilkan nilai a = 5 dan b = 6?

4.2. Simpulkan hasil eksperimen Anda!

Untuk menghasilkan nilai `a = 5` dan `b = 6`, anda perlu menggunakan operasi post-increment (`++`).

[No.4] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan post-increment pada variabel `a` setelah digunakan dalam ekspresi.
- Alasan solusi ini karena post-increment akan meningkatkan nilai variabel setelah digunakan dalam ekspresi, sehingga dapat menampilkan hasil yang diinginkan.
- Perbaiki kode program dengan cara menambahkan post-increment pada variabel `a`.

[No.4] Penyusunan Algoritma dan Kode Program

- Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.

- Deklarasi nilai a dengan nilai awal 5.
- Tampilkan nilai a.
- Gunakan post-increment pada a untuk mendapatkan nilai 6.
- Tampilkan nilai a setelah post-increment.

Kode program dan luaran

The screenshot shows an IDE with a Java program and its output. The code is as follows:

```

1 public class Operator {
2     public static void main(String[] args) {
3         // Deklarasi nilai
4         int a = 5;
5
6         // Menampilkan nilai a
7         System.out.println("a: " + a);
8
9         // Menggunakan post-increment pada a untuk mendapatkan nilai 6
10        System.out.println("b: " + (a++));
11
12        // Menampilkan nilai a setelah post-increment
13        System.out.println("Nilai a setelah post-increment: " + a);
14    }
15 }

```

The output of the program is:

```

a: 5
b: 5
Nilai a setelah post-increment: 6

```

- Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

```

public class Operator {
    public static void main(String[] args) {
        // Deklarasi nilai
        int a = 5;

        // Menampilkan nilai a
        System.out.println("a: " + a);

        // Menggunakan post-increment pada a untuk mendapatkan nilai 6
        System.out.println("b: " + (a++));

        // Menampilkan nilai a setelah post-increment
        System.out.println("Nilai a setelah post-increment: " + a);
    }
}

```

- Analisa luaran yang dihasilkan

Contoh:

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

[No.4] Kesimpulan

Analisa

Pada program itu, saya menggunakan bentuk kelas public karena itu adalah struktur dasar dalam Java untuk membuat kelas yang dapat diakses secara publik. Perbaiki program dengan menambahkan post-increment pada variabel a karena struktur Java mengharuskan penggunaan operator yang tepat untuk mendapatkan hasil yang diinginkan.

Dengan demikian, permasalahan ini dapat diatasi dengan menggunakan post-increment pada variabel a, yang memberikan hasil yang diinginkan dan sesuai dengan aturan logika yang diharapkan.

[No. 5] Identifikasi Masalah:

- Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```
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 5

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.

Diketahui dari soal : bahwa suatu program dapat berjalan dengan benar ketika tidak ada kesalahan sintaks.

[No.5] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan nilai-nilai yang tepat.
- Alasan solusi ini karena akan menghasilkan hasil yang sesuai dengan aturan logika.
- Perbaiki kode program dengan cara menambahkan nilai-nilai yang tepat untuk operator logika.

[No.5] Penyusunan Algoritma dan Kode Program

- Algoritma
 - Deklarasi nilai `a` dan `b`.
 - Tampilkan hasil logika dengan operator `&&` dan `||`.
- Kode program dan luaran

The screenshot shows an IDE with a Java file named `OperatorLogika.java`. The code defines a `public class OperatorLogika` with a `main` method. It declares two boolean variables, `a` and `b`, both set to `true`. It then prints the result of the logical AND operation (`a && b`) and the logical OR operation (`a || b`). The output window on the right shows the results: `a: 5`, `b: 5`, and `Nilai a setelah post-increment: 6`.

```
1 public class OperatorLogika {
2     public static void main(String[] args) {
3         // Deklarasi nilai
4         boolean a = true;
5         boolean b = true;
6
7         System.out.println("Hasil logika (a && b) : " + (a && b)); // Menampilkan hasil logika AND
8
9         // Menggunakan nilai-nilai yang tepat untuk operator OR
10        System.out.println("Hasil logika (a || b) : " + (a || b)); // Menampilkan hasil logika OR
11    }
12 }
```

Input/Output

Language Version: JDK 21.0.0 ☐ Interactive

Input Arguments

Stdin Inputs

Output Generated Files

```
a: 5
b: 5
Nilai a setelah post-increment: 6
```

- Screenshot/ Capture potongan kode dan hasil luaran komentar pada kode yang di Screenshot

```
public class OperatorLogika {
    public static void main(String[] args) {
        // Deklarasi nilai
        boolean a = true;
        boolean b = true;

        System.out.println("Hasil logika (a && b) : " + (a && b)); // Menampilkan hasil logika AND

        // Menggunakan nilai-nilai yang tepat untuk operator OR
        System.out.println("Hasil logika (a || b) : " + (a || b)); // Menampilkan hasil logika OR
    }
}
```
- Analisa luaran yang dihasilkan
Contoh:
Luaran sudah sesuai dengan program yang disusun.
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

[No.5] Kesimpulan

- **Analisa**

jawaban Analisa:

Pada program itu saya menggunakan bentuk kelas public karena itu adalah struktur dalam java untuk membuat kelas dapat diakses secara publik.

Perbaiki program dengan menambahkan nilai-nilai yang tepat untuk operator logika karena struktur java mengharuskan penggunaan operator yang tepat untuk mendapatkan hasil yang diinginkan.

[No. 6] Identifikasi Masalah:

- Uraikan permasalahan dan variabel
Tuliskan kembali soal:

```
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

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

[No.6] Analisis dan Argumentasi

Untuk membuat nilai 60 juga dianggap "Lulus" dalam kondisi yang menggunakan operator ternary, kita perlu mengubah operator perbandingan dari > menjadi >=. Dengan demikian, semua nilai yang sama dengan atau lebih besar dari 60 akan dikategorikan sebagai "Lulus".

[No.6] Penyusunan Algoritma dan Kode Program

- Algoritma
 1. Siapkan kelas baru "OperatorKondisi"
 2. Tempelkan kode soal pada Eclipse
 3. Ubah nilai integer menjadi 60
 4. Ubah operator > menjadi >=
 5. Tambahkan komentar
 6. Jalankan program untuk menampilkan hasil

- Kode program dan luaran

The screenshot shows an IDE with a Java class named `OperatorKondisi`. The code defines a `main` method that declares a `String` variable `status`, initializes an `int` variable `nilai` to 60, and uses a ternary operator to set `status` based on whether `nilai` is greater than or equal to 60. The output of the program is displayed in the 'Output' pane, showing 'Status: Lulus'.

```

1 - public class OperatorKondisi {
2 -     public static void main(String[] args) {
3 -         // Deklarasi variabel untuk menyimpan status
4 -         String status = "";
5 -
6 -         // Deklarasi variabel untuk menyimpan nilai
7 -         int nilai = 60; // Mengubah nilai untuk menguji kondisi
8 -
9 -         // Menggunakan operator ternary untuk menentukan status berdasarkan nilai
10 -        // Jika nilai >= 60, maka status akan menjadi "Lulus", jika tidak maka akan menjadi "Gagal"
11 -        status = (nilai >= 60) ? "Lulus" : "Gagal";
12 -
13 -        // Menampilkan hasil logika
14 -        System.out.println("Status: " + status);
15 -    }
16 - }

```

Input/Output

Language Version: JDK 21.0.0

Input Arguments

Stdin inputs

Output Generated Files

Status: Lulus

CPU Time: 0.05 sec(s) | Memory: 38188 kilobyte(s) | C

- Screenshot/ Capture potongan kode dan hasil luaran
- Komentar
- ```

public class OperatorKondisi {
 public static void main(String[] args) {

```

```

// Deklarasi variabel untuk menyimpan status
String status = "";

// Deklarasi variabel untuk menyimpan nilai
int nilai = 60; // Mengubah nilai untuk menguji kondisi

// Menggunakan operator ternary untuk menentukan status berdasarkan nilai
// Jika nilai >= 60, maka status akan menjadi "Lulus", jika tidak maka akan menjadi "Gagal"
status = (nilai >= 60) ? "Lulus" : "Gagal";

// Menampilkan hasil logika
System.out.println("Status: " + status);
}
}

```

- Luaran sudah sesuai dengan program yang disusun.  
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

#### [No.6] Kesimpulan

##### Analisa

- Dengan menggunakan operator perbandingan  $\geq$ , kita dapat membuat nilai 60 juga termasuk dalam kategori "Lulus". Kode program yang telah diperbaiki menunjukkan bahwa dengan perubahan ini, semua nilai yang sama dengan atau lebih besar dari 60 akan menghasilkan status "Lulus".

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

- Uraikan permasalahan dan variabel soal:  

```

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 : 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

Evaluasi penyebab hasil ~a = -11 ? Buktikan jawaban Anda dalam perhitungan biner!

Pada soal masih ada pesan kesalahan kode berulang-ulang dan kesalahan dalam operasi bitwise.

#### **[No.7] Analisis dan Argumentasi**

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan loop untuk mengurangi kode yang berulang.
- Alasan solusi ini karena ada fitur penggunaan loop untuk mengurangi kode berulang-ulang.
- Perbaiki kode program dengan cara perbaikan dalam operasi bitwise dengan benar dan hasil interpretasi dengan tepat.

#### **[No.7 ] Penyusunan Algoritma dan Kode Program**

- Algoritma
  - Deklarasi variabel
  - Operasi bitwise
  - Tampilkan hasil
- Kode program dan luaran

```

1 public class OperatorBitwise {
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 hasil = ~a; // Hasilnya sebenarnya adalah 5, bukan -11
17 System.out.println("Hasil dari ~a : " + hasil);
18
19 hasil = a >> 1;
20 System.out.println("Hasil dari a >> 1 : " + hasil);
21
22 hasil = b << 2;
23 System.out.println("Hasil dari b << 2 : " + hasil);
24 }
25 }

```

Input/Output

Language Version: JDK 2

Input Arguments

Stdin Inputs

Output Generated Files

```

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

```

CPU Time: 0.04 sec(s) | Memory: 38632 kilobytes

- Screenshot/ Capture potongan kode dan hasil luaran  
Komentar

```

public class OperatorBitwise {
 public static void main(String[] args) {
 // Deklarasi variabel a dan b dengan nilai awal
 int a = 10; // Nilai awal a dalam desimal
 int b = 7; // Nilai awal b dalam desimal

 // Variabel untuk menyimpan hasil operasi bitwise
 int hasil;

 // Operasi bitwise AND (&)
 // Menghasilkan hasil yang memiliki bit yang sama dengan kedua operand
 hasil = a & b;
 System.out.println("Hasil dari a & b : " + hasil);

 // Operasi bitwise OR (|)
 // Menghasilkan hasil yang memiliki bit yang setidaknya satu operand memiliki bit
 yang sama
 hasil = a | b;
 System.out.println("Hasil dari a | b : " + hasil);

 //

```

- Analisa luaran yang dihasilkan  
Contoh:  
Luaran sudah sesuai dengan program yang disusun.  
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

## [No.7] Kesimpulan

### Analisa

Dalam analisis ini, kita memahami bagaimana operasi bitwise ( , , , , dan ) berfungsi dalam kode Java. Kita juga melihat hasil dari setiap operasi bitwise dengan menggunakan nilai dan yang diberikan.

### **Refleksi**

Pada minggu ini, saya memiliki pengalaman belajar yang sangat berharga dalam mengembangkan kode Java untuk menghitung biaya kuliah mahasiswa. Saya memahami lebih baik tentang pentingnya menggunakan struktur kode yang rapi dan efisien, seperti penggunaan loop untuk mengurangi kode berulang dan definisi variabel secara eksplisit untuk menghindari kesalahan. Penggunaan operator ternary juga membantu dalam menampilkan biaya kuliah berdasarkan lama kuliah yang dipilih. Saya juga memperoleh pemahaman yang lebih dalam tentang pentingnya memahami dan menerapkan algoritma yang tepat dalam pengembangan kode. Tantangan utama yang dihadapi adalah mengatasi kode yang berulang-ulang dan memastikan bahwa semua variabel didefinisikan secara eksplisit. Namun, melalui proses belajar dan refleksi, saya dapat memahami dan mengatasi tantangan tersebut dengan lebih baik.