

Nama & NPM	Topik:	Tanggal:
M. Bagas Arjuna	Operator aritmatika	9 September 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

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));
    } }
```

Pada soal yang di atas ada terdapat perbedaan tanda teks yang di cetak dan operasi aritmatika yang dilakukan. Teks yang di cetak ("a + b = ") operasi aritmatika yang di input adalah (a - b) artinya pengurangan selanjutnya ditugaskan untuk menambahkan lagi baris dengan kode .

```
System.out.println("a + b = " + (a + b));
```

dengan mengubah operator dengan tanda (-, *, /, %)

[No.1] Analisis dan Argumentasi

Pada hasil analisa yang saya lakukan adalah pada kode yang di berikan terdapat sedikit perbedaan pada teks yang di cetak dan operasi aritmatika yang di lakukan tetapi tidak pengaruh karna yang di baca adalah operasinya hanya saja kode tersebut menjadi menggajal tidak enak di lihat dan selajutnya adalah saya meng analisa perhitungan matematika/Aritmatika yang dilakukan dengan ada nya tugas yang diberikan yaitu menganalisa matematika dengan tanda (-, *, /, %).

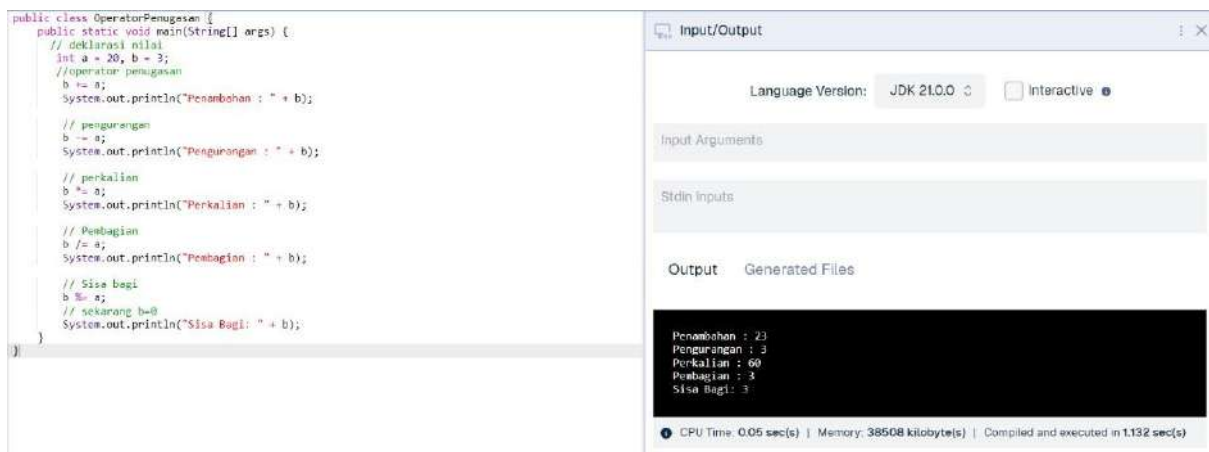
- **Pengurangan:**
Perhitungan : $20 - 3 = 17$ Output: $a - b = 17$
- **Penjumlahan:**
Perhitungan: $20 + 3 = 23$ Output: $a + b = 23$
- **Pembagian:**
Perhitungan: $20 / 3$ menggunakan pembagian integer, jadi hasilnya adalah 6 (karena sisa dibuang dalam pembagian integer). Output: $a / b = 6$
- **Perkalian:**
Perhitungan: $20 * 3 = 60$ Output: $a * b = 60$
- **Sisa Bagi:**
Perhitungan: $20 * 3 = 60$ Output: $a * b = 60$

- 1) Alasan solusi ini karena terbuktinya dengan tidak ada kode yang error di tampilan output dan juga solusi ini menjadi efektif karena dengan telitinya mengerjakan codingan atau meng input kode kita dapat mengurangi terjadinya error dalam output

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Penyusunan Algoritma dan Kode Program
 - (a) Mulai
 - (b) Mendeklarasi dan inisialisai variabel
 - (c) MengOutput nilai variable yang ingin dilakukan
 - (d) Melakukan pengoperasian aritmatika
 - (e) Selesai
- 2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran



The screenshot displays a Java IDE with a code editor on the left and an 'Input/Output' window on the right. The code defines a class 'OperatorPenugasan' with a 'main' method. It initializes variables 'a' (20) and 'b' (3), then performs addition, subtraction, multiplication, division, and modulus operations, printing the results. The output window shows the following results:

```
Penambahan : 23
Pengurangan : 3
Perkalian : 60
Pembagian : 4
Sisa Bagi : 3
```

At the bottom of the output window, it states: CPU Time: 0.05 sec(s) | Memory: 38508 kilobyte(s) | Compiled and executed in 1.132 sec(s).

b) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun dan pada nilai pengoperasian matematika suda benanr

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

[No.1] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Kode program ini bertujuan untuk mendemonstrasikan penggunaan operator aritmatika dasar dalam Bahasa pemograman java.

Operator-operator ini digunakan untuk melakukan operasi matematika sederhana seperti perjumlahan,pengurangan,perkallian,pembagian, dan modulus pada dua buah bilangan bulat.

[No. 2] Identifikasi Masalah:

2) 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;  
        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);  
    }  
}
```

Pada kode yang di atas adalah penggunaan operator penugasan secara benar dengan hasil yang sesuai.

[No.2] Analisis dan Argumentasi

Pada hasil analisa yang saya lakukan adalah saya dapat membandingkan bahwa kode programing dari contoh 1 dan 2 adalah:

Pada Contoh 1 :

- (a) dicetak sebagai 20.
- (b) dicetak sebagai 3.
- Operasi $(a - b)$ menghasilkan 17, sehingga hasil yang dicetak adalah 17.

Sedangkan pada contoh 2:

- **Penambahan:** $(b += a)$ menjadikan b menjadi 23 $(3 + 20)$.
- **Pengurangan:** $(b -= a)$ menjadikan b kembali ke 3 $(23 - 20)$.
- **Perkalian:** $(b *= a)$ menjadikan b menjadi 60 $(3 * 20)$.
- **Pembagian:** $(b /= a)$ menjadikan b menjadi 3 $(60 / 20)$.
- **Sisa Bagi:** $(b \%= a)$ menghasilkan sisa bagi dari 3 dibagi 20, yang tetap 3.

Kesimpulan dari perbandingan contoh 1 dan 2 yang telah saya lakukan adalah saya dapat menyimpulkan bahwa contoh 1 itu menggunakan operator aritmatika biasa sedangkan pada contoh 2 menggunakan operator penugasan yang mengubah nilai variable secara langsung.

[No.2] Penyusunan Algoritma dan Kode Program

3) Kode program dan luaran

```

public class OperatorRelasional {
    public static void main(String[] args) {
        int nilaiA = 4;
        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);
    }
}

```

Input/Output

Language Version: JDK 21.0.0 ☐ Interactive

Input Arguments

Stdin Inputs

Output Generated Files

```

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

```

CPU Time: 0.06 sec(s) | Memory: 39988 kilobyte(s) | Compiled and executed in 1.151 sec(s)

- c) Screenshot/ Capture potongan kode dan hasil luaran
Kode yang di input dan luaran yang di tampilkan hanya untuk membandingkan hasil contoh codingan tugas 1 dan 2.
- d) Analisa luaran yang dihasilkan
Luaran sudah sesuai dengan program yang disusun dan pada nilai pengoperasian matematika suda benar

[No.2] Kesimpulan

2) Analisa

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

Contoh jawaban Analisa:

Pada program tugas 2 saya di tugaskan untuk membandingkan contoh codingan tugas 1 dan 2 sehingga saya dapat menyimpulkan bahwa contoh 1 **lebih sederhana dan hanya menunjukkan hasil dari suatu operasi aritmatika.**

Pada contoh 2 **lebih kompleks, menunjukkan penggunaan berbagai operator penugasan secara berurutan, dan memberikan hasil akhir setelah setiap operasi.**

Dasar alasan pengambilan keputusan untuk kasus ini adalah untuk mengetahui hasil perbandingan dari contoh 1 dan 2 serta untuk mengetahui kode atau codingan mana yang lebih efektif untuk suatu penghitungan aritmatika.

[No. 3] Identifikasi Masalah:

- 3) Uraikan permasalahan dan variabel
- 4)

```
public class OperatorRealasional {
    public static void main(String[] args) {
        int nilaiA = 12;
        int nilaiB = 4;
        boolean hasil;
```
- 5)

```
System.out.println(" A = " + nilaiA + "\n B = " + nilaiB);
// apakah A lebih besar dari B?
hasil = nilaiA > nilaiB;
System.out.println("Hasil A > B = " + hasil);
```
- 6)

```
// apakah A lebih kecil dari B?
hasil = nilaiA < nilaiB;
System.out.println("Hasil A < B = " + hasil);
```
- 7)

```
// apakah A lebih besar samadengan B?
hasil = nilaiA >= nilaiB;
System.out.println("Hasil A >= B = " + hasil);
```
- 8)

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

```

9)          // apakah nilai A sama dengan B?
           hasil = nilaiA == nilaiB;
           System.out.println("Hasil A == B = " + hasil);
10)         // apakah nilai A tidak samadengan B?
           hasil = nilaiA != nilaiB;
           System.out.println("Hasil A != B = " + hasil);
           }
       }

```

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

Pada hasil analisa yang saya lakukan saya mendapatkan ada beberapa perubahan yaitu yang awalnya :

```

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

```

Setelah nilai A = 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

```

[No.3] Penyusunan Algoritma dan Kode Program

Algoritma

Penyusunan Algoritma dan Kode Program

1. Mulai
2. Mendeklarasi dan inisialisai variabel
3. Melakukan perubahan nilai dari awal nilai a = 12 menjadi a =4
4. elakukan pengoperasian aritmatika
5. Selesai

Kode program dan luaran

```

public class OperatorRealasional {
    public static void main(String[] args) {
        int nilaiA = 4;
        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);
    }
}

```

Input/Output

Language Version: JDK 21.0.0 ☐ Interactive

Input Arguments

Stdin Inputs

Output: Generated Files

```

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

```

CPU Time: 0.06 sec(s) | Memory: 39988 kilobyte(s) | Compiled and executed in 1.151 sec(s)

e) Screenshot/ Capture potongan kode dan hasil luaran

f) Analisa luaran yang dihasilkan

Luaran yang di dihasilkan

```

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

```

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

Pada hasil pertama memberikan luaran false yang dimana A tidak lebih dari B

Pada hasil kedua memberikan luaran false karena A tidak kurang dari B

Pada hasil ketiga memberikan luaran true karena A lebih dari sama dengan B

Pada hasil keempat memberikan luaran true karena A kurang dari sama dengan B

Pada hasil kelima memberikan luaran true karena nilai A sama dengan nilai B

Pada hasil keenam memberikan luaran false karena nilai A sama dengan nilai B

[No.3] Kesimpulan

3) Analisa

e) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

f) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Pada program itu saya mengganti nilai A yang awalnya 12 menjadi A = 4 sehingga saya dapat menyimpulkan bahwa yang tadinya kode Boolean tersebut berubah ketika nilai variable A di ganti menjadi A = 4

Dasar saya mengambil keputusan kasus ini karena saya harus menyelesaikan hingga harus mendapatkan hasil yang berbeda dari nilai variable sebelumnya. Dan tidak mendapatkan hasil kode yang error.

[No. 4] Identifikasi Masalah:

11) Uraikan permasalahan dan variabel

```
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);  
    }  
}
```

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

[No.4] Analisis dan Argumentasi

Pada hasil analisa yang saya lakukan saya mendapatkan perbandingan hasil dari Post dan Pre increment dan Decrement.

1. Post Increment:

- **Isi variabel a:** 10
- **Isi variabel a:** 10 (nilai a sebelum increment karena a++ menggunakan nilai a saat ini sebelum melakukan increment)
- **Isi variabel a:** 11 (nilai a setelah increment)

2. Pre increment:

- **Isi variabel b:** 10
- **Isi variabel b:** 11 (nilai b setelah increment karena ++b melakukan increment terlebih dahulu baru kemudian menggunakan nilai baru)
- **Isi variabel b:** 11 (nilai b setelah increment)

3. Post decrement:

- **Isi variabel c:** 10

- **Isi variabel c:** 10 (nilai c sebelum decrement karena `c--` menggunakan nilai c saat ini sebelum melakukan decrement)
- **Isi variabel c:** 9 (nilai c setelah decrement)

4. Pre decrement:

- **Isi variabel d:** 10
- **Isi variabel d:** 9 (nilai d setelah decrement karena `--d` melakukan decrement terlebih dahulu baru kemudian menggunakan nilai baru)
- **Isi variabel d:** 9 (nilai d setelah decrement)

[No.4] Penyusunan Algoritma dan Kode Program

Algoritma

Penyusunan Algoritma dan Kode Program

1. Mulai
2. Mendeklarasi dan inialisai variabel
3. Melakukan pengecekan kembali terhadap kode
4. Melakukan Running
5. Selesai

Kode program dan luaran

```

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);
    }
}

```

Input/Output

Stdin Inputs

Output Generated Files

```

# Post Increment #
=====
Isi variabel a: 10
Isi variabel a: 11
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

```

ⓘ CPU Time: 0.06 sec(s) | Memory: 38892 kilobyte(s) | Compiled and executed in 1.214 sec(s)

g) Screenshot/ Capture potongan kode dan hasil luaran

h) Analisa luaran yang dihasilkan

Luaran yang di dihasilkan

Pada hasil pertama memberikan luaran false yang dimana A tidak lebih dari B

Pada hasil kedua memberikan luaran false karena A tidak kurang dari B

Pada hasil ketiga memberikan luaran true karena A lebih dari sama dengan B

Pada hasil keempat memberikan luaran true karena A kurang dari sama dengan B

Pada hasil kelima memberikan luaran true karena nilai A sama dengan nilai B

Pada hasil keenam memberikan luaran false karena nilai A sama dengan nilai B

[No.4] Kesimpulan

4) Analisa

g) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Pada program itu permasalahan yang di lakukan adalah melakukan perbandingan antara post increment dan decrement serta pre increment dan decrement saya mengetahui bahwa

- **Post-increment/decrement:**

Nilai variabel digunakan dalam ekspresi terlebih dahulu, kemudian diperbarui. Ini memungkinkan kita untuk menggunakan nilai lama untuk keperluan seperti pencetakan atau perhitungan, sementara nilai variabel diperbarui untuk penggunaan berikutnya.

- **Pre-increment/decrement:**

Variabel diperbarui terlebih dahulu, kemudian nilai baru digunakan dalam ekspresi. Ini berguna ketika perubahan pada variabel perlu diterapkan sebelum digunakan.

- h) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Dasar saya mengambil keputusan kasus ini karena saya harus menyelesaikan hingga harus mendapatkan hasil yang berbeda dari nilai variable sebelumnya. Dan tidak mendapatkan hasil kode yang error.

[No. 5] Identifikasi Masalah:

Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```
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);  
        //baris kode boolean ||  
        boolean A = false;  
        boolean B = false;  
        boolean C;  
        C = a || b;  
        System.out.println(" false || false = " +C);  
    }  
}
```

No.5] Analisis dan Argumentasi

- 1) Saya menganalisis Ketika menggunakan baris kode || yang dimana Boolean A = false dan B = false hasil print yang di hasilkan adalah true sedangkan saat menggunakan kode && dengan variable a = true dan b = false hasil print nya adalah false.

[No.5] Penyusunan Algoritma dan Kode Program

- 4) Algoritma
 - (f) Mulai
 - (g) Mendeklarasikan bari kode baru
 - (h) Melakukan perubahan nilai variable A dan B
 - (i) Membuat System.out.println untuk ke output
 - (j) Melakukan runnning
 - (k) Selesai

- 5) Kode program dan luaran

```
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);  
  
        boolean A = false;  
        boolean B = false;  
        boolean C;  
        C = a || b;  
        System.out.println(" false || false = " +C);  
    }  
}
```

(gambar program)

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

 CPU Time: 0.07 sec(s) | Memory: 38896 kilobyte(s) | Compiled and executed in 1.711 sec(s)

(gambar luaran)

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

[No.5] Kesimpulan

Analisa

- j) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Saya menganalisis Ketika menggunakan baris kode `||` yang dimana Boolean A = false dan B = false hasil print yang di hasilkan adalah true sedangkan saat menggunakan kode `&&` dengan variable a = true dan b = false hasil print nya adalah false.
- k) dasar alasan pengambilan keputusan Anda untuk kasus ini?
Dasar pengambilan keputusan ini adalah untuk memahami bagaimana kerja system method berubah Ketika ada suatu nilai atau data berubah

[No. 6] Identifikasi Masalah:

Uraikan permasalahan dan variable

- 1) Tuliskan kembali soal:

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

Pada soal menggantikan int nilai dari 80 menjadi 60

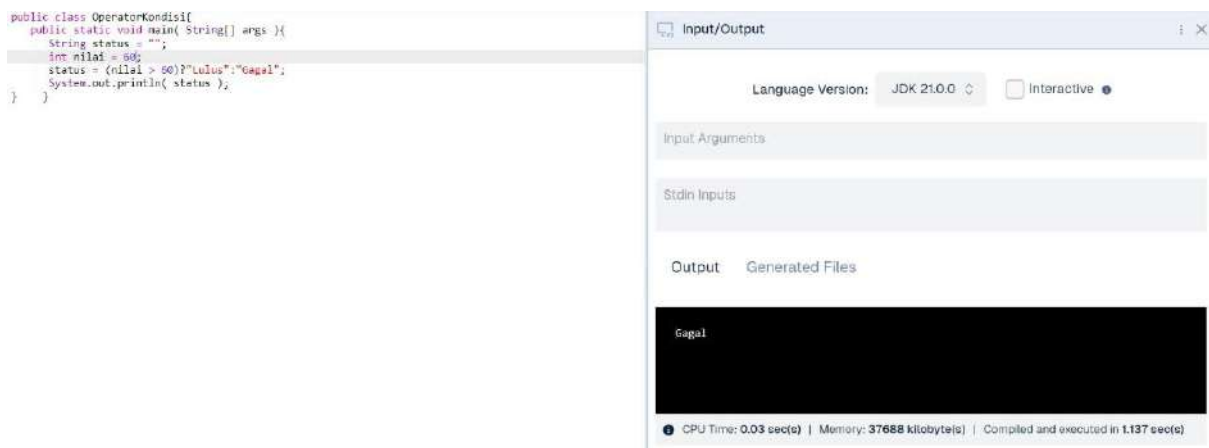
[No.6] Analisis dan Argumentasi

- 2) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambah tanda sama dengan pada baris status = (nilai > 60)?"Lulus":"Gagal"; menjadi status = (nilai >= 60) ? "Lulus" : "Gagal";
- 3) Alasan solusi ini karena untuk mengetahui perubahan yang akan dihasilkan Ketika nilai int dari 80 menjadi 60 dan apakah hasil akhir yang tapil yaitu lulus atau gagal

[No.6] Penyusunan Algoritma dan Kode Program

Algoritma adalah langkah-langkah penyelesaian masalah

- 1) Mulai program.
- 2) Deklarasikan variabel status sebagai string kosong.
- 3) Deklarasikan variabel nilai dengan nilai awal 60.
- 4) Periksa kondisi apakah nilai lebih besar atau sama dengan 60:
Jika benar (nilai >= 60), set status menjadi "Lulus".
Jika salah, set status menjadi "Gagal".
- 4) Selesai



- 5) Kode program dan luaran
Screenshot/ Capture potongan kode dan hasil luaran

- 1) Analisa luaran yang dihasilkan :

- **Deklarasi dan Inisialisasi Variabel:**
String status = ""; Variabel status dideklarasikan sebagai string kosong.
- int nilai = 60; Variabel nilai dideklarasikan dan diinisialisasi dengan nilai 60.
- **Operator Ter nary:**
- status = (nilai >= 60) ? "Lulus" : "Gagal";
- Ini adalah operator ternary yang bekerja seperti pernyataan if-else.
- Kondisi yang diuji adalah (nilai >= 60). Jika kondisi ini benar (true), maka nilai status akan menjadi "Lulus". Jika kondisi ini salah (false), maka nilai status akan menjadi "Gagal".

- Dalam hal ini, karena nilai 60 sama dengan 60, kondisi nilai ≥ 60 adalah true. Oleh karena itu, status akan diatur menjadi "Lulus".
- **Output:**
- `System.out.println(status);` : Baris ini akan mencetak nilai dari variabel status ke layar.
- Mengingat hasil dari operator ternary adalah "Lulus", maka output yang dihasilkan adalah:
- Lulus

2) Luaran sudah sesuai dengan program yang disusun.

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

[No.6] Kesimpulan

2) Analisa

Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Logika Program:

- 3) Kode menggunakan operator kondisional (ternary) untuk menetapkan nilai `status` berdasarkan hasil perbandingan `nilai` dengan 60.
- 4) Dalam kode ini, nilai 60 tidak memenuhi kondisi `nilai > 60`, sehingga hasil `status` adalah "Gagal".

Output:

Karena nilai 60 tidak lebih besar dari 60, maka hasil yang dicetak adalah "Gagal".

Perbaikan:

Jika tujuan adalah untuk menganggap nilai 60 sebagai kelulusan, maka operator perbandingan seharusnya \geq (lebih besar dari atau sama dengan) alih-alih $>$. Dengan perbaikan ini, kode akan menetapkan `status` sebagai "Lulus" ketika nilai sama dengan atau lebih besar dari 60.

Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Dasar saya dalam pengambilan keputusan ini adalah untuk mengetahui kondisi perbandingan yang terjadi dalam codingan yang menggunakan `if` dan `else` dan juga sebagai pembuatan kriteria penilaian jika nilai lebih besar dari 60 (`nilai > 60`) maka hasil yang di tampilkan lulus tetapi jika (`nilai < 60`) maka hasil yang akan di tampilkan gagal.

[No. 7] Identifikasi Masalah:

Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```
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 );  
    }  
}
```

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

[No.7] Analisis dan Argumentasi

- 1). Saya mengusulkan permasalahan ini dapat diselesaikan dengan cara mencari tahu kenapa nilai hasil dari operator bitwise $\sim a = -11$.
- 2). Alasannya karena, operator bitwise not (\sim) akan membalikkan setiap bit dari bilangan tersebut dan hasilnya dalam format komplement dua mewakili bilangan negatif.
- 3). Operator bitwise : yang awalnya nilai $a = 10$ ketika dipresentasikan ke biner adalah 32-bit, maka 10 adalah: 00000000 00000000 00000000 00001010. Hasil dari $\sim a = 11111111 11111111 11111111 11110101$. Karena dimulai dengan tanda bit 1 menunjukkan negatif, maka hasil akhirnya adalah -11.


[No.7] Penyusunan Algoritma dan Kode Program

Kode program dan luaran

```
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 );  
    }  
}
```

(GAMBAR CODING)

```
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: 0.07 sec(s) | Memory: 38892 kilobyte(s) | Compiled and executed in 1.83 sec(s)

(HASIL OUTPUT)

l) Screenshot/ Capture potongan kode dan hasil luaran

m) Analisa luaran yang dihasilkan:

Luaran yang di hasilkan adalah

- 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
1. Menggunakan operator dan bitwise
 2. Menggunakan operator or bitwise
 3. Menggunakan operator XOR bitwise
 4. Menggunakan operato NOT bitwise
 5. Menggunakan operator right shift
 6. Menggunakan operator left shift

Luaran yang di buat sudah sesuai dengan yang di perintahkan atau luaran sudah benar

No.7] Kesimpulan

Kode ini memberikan contoh sederhana tentang bagaimana menggunakan operator bitwise dalam Java. Operator-operator ini sangat berguna ketika kita ingin melakukan manipulasi data pada tingkat bit, seperti misalnya dalam operasi bitmasking, perhitungan checksum, atau implementasi algoritma kriptografi.

Refleksi

Pengalaman singkat tentang pengalaman belajar adalah sedikit membingungkan tentang materi yang di hadapi pada minggu ini karena materi yang di pelajari lebih banyak lagi tentang method yang di jalankan sehingga membuat harus lebih mengulang Kembali materi yang di jelaskan sehingga nanti tidak bingung lagi Ketika menghadapi berbagai permasalahan yang diberikan.