

Nama & NPM	Topik:	Tanggal:
Khairul Aji Pangestu G1F024042	Operator Java	7 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) 		

Nama & NPM	Topik:	Tanggal:
Khairul Aji PAngestu G1F024042	Operator Java	7 September

[No. 1] Identifikasi Masalah:

Operator Aritmatika: +, -, *, /

Contoh 1: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
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!

- 1) Uraikan permasalahan dan variabel

Pada soal masih ada pesan kesalahan pada penulisan fungsi System.out.println.

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan tanda + untuk menggabungkan String "a+b" dengan hasil a+b pada fungsi System.out.println terakhir.
- 2) Alasan solusi ini karena pada Java, untuk menampilkan hasil operasi di dalam System.out.println, kita perlu menggabungkan teks dengan hasil operasi harus menggunakan tanda +.
- 3) Perbaikan kode program dengan cara menambahkan tanda + diantara "a+b=" dan a+b pada syntax System.out.println.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Mulai
- (b) Install Aplikasi Netbeans atau Eclipse
- (c) Buat Project OperatorJava
- (d) Deklarasi Class OperatorAritmatika
- (e) Deklarasi Method bersifat Public Static Void Main
- (f) Deklarasi nilai pada variabel yang akan dihitung
- (g) Print nilai variabel
- (h) Print hasil perhitungan
- (i) Selesai

2) Kode program dan luaran

a) Kode Program

```
va > E-Learning > OperatorAritmatika.java > ...
Codeium: Refactor | Explain
1 public class OperatorAritmatika {
  Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
2 public static void main(String[] args) {
3
4     // deklarasi nilai
5     int a = 20; // nilai awal a
6     int b = 3;  // nilai awal b
7
8     //operator aritmatika
9     System.out.println("a: " +a); // menampilkan nilai a
10    System.out.println("b: " +b); // menampilkan nilai b
11    System.out.println("a + b = " + (a + b)); // menampilkan hasil penjumlahan a dan b
12    System.out.println("a - b = " + (a - b)); // menampilkan pengurangan a dan b
13    System.out.println("a * b = " + (a * b)); // menampilkan perkalian a dan b
14    System.out.println("a / b = " + (a / b)); // menampilkan pembagian a dan b
15    System.out.println("a % b = " + (a % b)); // menampilkan pembagian sisa a dan b
16 } }
17
```

b) Luaran

```
C:\Users\user> java -cp . OperatorAritmatika
a: 20
b: 3
a + b = 23
a - b = 17
a * b = 60
a / b = 6
a % b = 2
PS D:\vscode>
```

c) Analisa uaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Kode program dibuat untuk membuat perhitungan +, -, *, /, % dari nilai 20 dan 3, hasil perhitungan sudah benar dan sesuai.

[No.1] Kesimpulan

1) Analisa

Masalah yang dihadapi pada soal ini adalah kesalahan penulisan sintaksis pada fungsi System.out.println. Kesimpulannya, solusi untuk masalah ini adalah menambahkan tanda + untuk menggabungkan string dan hasil perhitungan aritmatika dalam satu baris kode. Dengan demikian, perhitungan dapat ditampilkan dengan benar.

[No. 2] Identifikasi Masalah:

Operator Penugasan:

Contoh 2: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
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!

1) Uraikan permasalahan dan variabel

Pada soal kode program tersebut dibuat untuk menambahkan nilai variabel a ke variabel b dan langsung menyimpan hasilnya ke b.

[No.2] Analisis dan Argumentasi

- 1) Saya menganalisis kode program ini digunakan untuk menunjukkan penggunaan operator penugasan dalam melakukan operasi aritmatika dan menyimpan hasilnya dalam variabel.

- 2) Perbedaan luaran pada Contoh 1 dan Contoh 2 yaitu Contoh 1 hanya menjumlahkan $a + b$ dan menampilkan hasilnya, tanpa mengubah nilai b dengan luaran 23. Sedangkan pada Contoh 2 menjumlahkan a ke b dengan operator $+=$ dan memperbarui nilai b secara permanen dengan luaran Luaran: 23.
- 3) Perbedaan waktu eksekusi dari Contoh 1 dan Contoh 2 yaitu kedua program memiliki waktu eksekusi yang hampir sama karena operasi aritmatika sederhana. Perbedaan waktu sangat kecil dan tidak signifikan.

[No.2] Penyusunan Algoritma dan Kode Program

3) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Mulai
- (b) Install Aplikasi Netbeans atau Eclips
- (c) Buat Project OperatorJava
- (d) Deklarasi Class OperatorPenugasan
- (e) Deklarasi Method bersifat Public Static Void Main
- (f) Deklarasi nilai pada variabel yang akan dihitung
- (g) Lakukan operasi Operator Penugasan
- (h) Print hasil perhitungan
- (i) Selesai

4) Kode program dan luaran

a) Kode Program

```
ava > E-Learning > OperatorPenugasan.java > ...
Codeium: Refactor | Explain
1 public class OperatorPenugasan {
  Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
2   public static void main(String[] args) {
3       // deklarasi nilai
4       int a = 20, b = 3;
5       //operator penugasan
6       b += a; //melakukan perhitungan penjumlahan
7       System.out.println("Penambahan : " + b); // menampilkan hasil perhitungan penjumlahan
8       b -= a; //melakukan perhitungan pengurangan
9       System.out.println("Pengurangan : " + b); // menampilkan hasil perhitungan pengurangan
10      b *= a; //melakukan perhitungan pembagian
11      System.out.println("Perkalian : " + b); // menampilkan hasil perhitungan perkalian
12      b /= a; //melakukan perhitungan penjumlahan
13      System.out.println("Pembagian : " + b); // menampilkan hasil perhitungan pembagian
14      b %= a; //melakukan modulus
15      System.out.println("Modulus : " + b); // menampilkan hasil perhitungan modulus
16  }
17  }
18 }
```

b) Luaran

```
Penambahan : 23
Pengurangan : 3
Perkalian : 60
Pembagian : 3
Modulus : 3
PS D:\vscode>
```

c) Analisa uaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Kode program dibuat untuk membuat operator aritmatika penugasan +, -, *, /, % dari nilai 20 dan 3, hasil perhitungan sudah benar dan sesuai.

[No.2] Kesimpulan

1) Analisa

Operator penugasan digunakan untuk menghitung dan menyimpan hasil operasi ke variabel secara langsung. Dibandingkan dengan operator aritmatika biasa, operator penugasan memperbarui nilai variabel target. Waktu eksekusi relatif sama karena keduanya adalah operasi sederhana, dan hasilnya sudah sesuai dengan harapan.

[No. 3] Identifikasi Masalah:

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

Contoh 3: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
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!

1) Uraikan permasalahan dan variabel

Pada soal kode program tersebut dibuat untuk menentukan hubungan antara dua nilai dan menghasilkan hasil true atau false.

[No.3] Analisis dan Argumentasi

1) Saya menganalisis kode program ini digunakan untuk menentukan hubungan antara nilai A dan nilai B akan menghasilkan boolean true atau false.

- 2) Pada Contoh 3, jika nilai A=4 dan B=4 maka output akan berubah menjadi:

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

[No.3] Penyusunan Algoritma dan Kode Program

- 5) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Mulai
- (b) Install Aplikasi Netbeans atau Eclips
- (c) Buat Project OperatorJava
- (d) Deklarasi Class OperatorRasional
- (e) Deklarasi Method bersifat Public Static Void Main
- (f) Deklarasi nilai pada variabel yang akan dibandingkan
- (g) Print nilai variabel
- (h) Ptint hasil boolean
- (i) Selesai

- 6) Kode program dan luaran

- a) Kode Program


```
java > E-Learning > OperatorRelasional.java > ...
Codeium: Refactor | Explain
1 public class OperatorRelasional{
2     Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
3     public static void main(String[] args) {
4         int nilaiA = 4;
5         int nilaiB = 4;
6         boolean hasil;
7         System.out.println(" A = " + nilaiA + "\n B = " + nilaiB);
8         // apakah A lebih besar dari B?
9         hasil = nilaiA > nilaiB;
10        System.out.println("\n Hasil A > B = "+ hasil);
11        // apakah A lebih kecil dari B?
12        hasil = nilaiA < nilaiB;
13        System.out.println("\n Hasil A < B = "+ hasil);
14        // apakah A lebih besar samadengan B?
15        hasil = nilaiA >= nilaiB;
16        System.out.println("\n Hasil A >= B = "+ hasil);
17        // apakah A lebih kecil samadengan B?
18        hasil = nilaiA <= nilaiB;
19        System.out.println("\n Hasil A <= B = "+ hasil);
20        // apakah nilai A sama dengan B?
21        hasil = nilaiA == nilaiB;
22        System.out.println("\n Hasil A == B = "+ hasil);
23        // apakah nilai A tidak samadengan B?
24        hasil = nilaiA != nilaiB;
25        System.out.println("\n Hasil A != B = "+ hasil);
26    }
```

b) Luaran

```
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
PS D:\vscode>
```

c) Analisa uaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Kode program dibuat untuk membuat menentukan hubungan antara variabel A dan B memiliki nilai true atau false menggunakan Operator Rasional, hasil perhitungan sudah benar dan sesuai.

[No.3] Kesimpulan

1) Analisa

Operator relasional membandingkan dua nilai dan mengembalikan hasil boolean true atau false. Perubahan nilai variabel A dan B menghasilkan perubahan pada output perbandingan.

Kesimpulannya, jika nilai A dan B sama, maka hasil perbandingan tertentu akan berubah dari false menjadi true.

[No. 4] Identifikasi Masalah:

Operator Increment dan Decrement: ++, --

Contoh 4: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
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!

1) Uraikan permasalahan dan variabel

Pada soal kode program tersebut dibuat untuk menunjukkan fungsi operator Increment dan Decrement.

[No.4] Analisis dan Argumentasi

- 1) Saya menganalisis kode program ini digunakan untuk melakukan penjumlahan atau pengurangan sebesar 1 pada variabel a.
- 2) Untuk menghasilkan output a = 5 dan b = 6 maka saya menyarankan kode menggunakan operator Pre-Increment karena Pre- tidak perlu menggunakan fungsi Print()

- 3) Kesimpulannya jika kita menggunakan Pre- maka nilai variabel akan langsung naik atau turun pada output tanpa menggunakan fungsi Print() dahulu, sedangkan jika menggunakan Post- maka harus menggunakan fungsi Print() dahulu agar nilai variabel berubah pada output

[No.4] Penyusunan Algoritma dan Kode Program

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Mulai
- (b) Install Aplikasi Netbeans atau Eclipse
- (c) Buat Project OperatorJava
- (d) Deklarasi Class OperatorIncrementDecrement
- (e) Deklarasi Method bersifat Public Static Void Main
- (f) Deklarasi nilai pada variabel yang akan dinaikkan atau diturunkan
- (g) Print nilai variabel dengan menggunakan Operator Pre-Increment
- (h) Selesai

2) Kode program dan luaran

a) Kode Program

```
java > E-Learning > OperatorIncrDecrem.java > ...
Codeium: Refactor | Explain
1 public class OperatorIncrDecrem{
  Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
2   public static void main(String[] args) {
3       // deklarasi nilai
4       int a = 5;
5
6       System.out.println("a: " + a); //menampilkan nilai a
7       System.out.println("b: " + (++a)); //menggunakan operator pre-increment
8   }
9 }
```

b) Luaran

```
OperatorIncrDecrem
a: 5
b: 6
PS D:\vscode>
```

c) Analisa uaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Kode program dibuat untuk menaikkan nilai a dan menjadikan hasilnya menjadi nilai b menggunakan operator pre-increment, hasil perhitungan sudah benar dan sesuai.

[No.4] Kesimpulan

1) Analisa

Untuk menghasilkan output yang diinginkan pada contoh soal ini, penggunaan pre-increment lebih cocok dibandingkan post-increment. Pre-increment langsung memperbarui nilai variabel sebelum hasil dicetak, sedangkan post-increment memperbarui setelahnya.

[No. 5] Identifikasi Masalah:

Operator Logika: &&, ||, !

Contoh 5: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
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.

1) Uraikan permasalahan dan variabel

Pada soal kode program tersebut menggunakan Operator Logika AND.

[No.5] Analisis dan Argumentasi

- 1) Saya menganalisis kode program ini digunakan untuk menentukan Boolean dari Logika AND variabel a dan variabel b dan menghasilkan false.
- 2) Untuk menghasilkan luaran true dengan Logika AND maka saya menyarankan boolean kedua operator bernilai sama-sama true, singkatnya AND akan true jika kedua operator true. Dan Logika OR akan true cukup jika salah satu operator bernilai true, tetapi jika keduanya true maka akan tetap true
- 3) Kesimpulannya jika kita mendeklarasikan nilai dari salah satu operator false maka hasil luaran AND akan selalu false dan OR akan menghasilkan true. Sedangkan jika kita mendeklarasikan nilai dari kedua operator adalah true maka luaran dari Logika AND akan menjadi true dan OR akan tetap true.

[No.5] Penyusunan Algoritma dan Kode Program

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Mulai
- (b) Install Aplikasi Netbeans atau Eclipse
- (c) Buat Project OperatorJava
- (d) Deklarasi Class OperatorLogika
- (e) Deklarasi Method bersifat Public Static Void Main
- (f) Deklarasi boolean operator
- (g) Print hasil luaran dengan menggunakan Operator Logika AND dan OR
- (h) Selesai

2) Kode program dan luaran

a) Kode Program

```
java > E-Learning > OperatorLogika.java > ...
Codeium: Refactor | Explain
1 public class OperatorLogika {
  Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
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       System.out.println("Hasil logika (a || b) : " + (a || b)); //menampilkan hasil logika OR
9   } }
```

b) Luaran

```
Hasil logika (a && b) : true
Hasil logika (a || b) : true
PS D:\vscode>
```

c) Analisa uaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Kode program dibuat untuk menaikkan nilai a dan menjadikan hasilnya menjadi nilai b menggunakan operator pre-increment, hasil perhitungan sudah benar dan sesuai.

[No.5] Kesimpulan

1) Analisa

Operator logika AND (&&) akan menghasilkan true jika kedua operand bernilai true. Untuk menghasilkan true pada luaran, kedua operand harus bernilai true. Sebaliknya, jika salah satu bernilai false, hasil akan selalu false.

[No. 6] Identifikasi Masalah:

Operator Kondisional (Ternary): ?:

Bentuk umum: (ekspresi) ? (jika benar) : (jika salah);

Contoh 6: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
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 !

- 1) Uraikan permasalahan dan variabel

Pada soal kode program tersebut dibuat untuk menentukan lulus atau gagal menggunakan Operator Kondisional

[No.6] Analisis dan Argumentasi

- 1) Saya menganalisis kode program ini digunakan untuk menentukan Lulus atau Gagal dari nilai menggunakan Operator Kondisional.
- 2) Untuk membuat nilai = 60 memenuhi Lulus, saya merekomendasikan menggunakan tanda \geq .

[No.6] Penyusunan Algoritma dan Kode Program

- 1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Mulai
- (b) Install Aplikasi Netbeans atau Eclipse
- (c) Buat Project OperatorJava
- (d) Deklarasi Class OperatorKondisional

- (e) Deklarasi Method bersifat Public Static Void Main
- (f) Deklarasi String status dan int nilai
- (g) Deklarasi ekspresi dari status dengan nilai
- (h) Print status
- (i) Selesai

2) Kode program dan luaran

d) Kode Program

```

java > E-Learnring > OperatorKondisi.java > ...
Codeium: Refactor | Explain
1 public class OperatorKondisi{
  Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
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   }
8

```

e) Luaran

```

Lulus
PS D:\vscode>

```

f) Analisa uaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Kode program dibuat untuk menentukan agar nilai=60 bisa memenuhi Lulus, hasil perhitungan sudah benar dan sesuai.

[No.6] Kesimpulan

1) Analisa

Operator kondisional atau ternary digunakan untuk menentukan nilai berdasarkan kondisi.

Pada soal ini, untuk membuat nilai 60 memenuhi kriteria "lulus", operator >= digunakan.

Dengan ini, nilai 60 dan lebih besar akan menghasilkan status "lulus".

[No.7] Analisis dan Argumentasi

Operator Bitwise: &, |, ^, ~, <<, >>, >>>

Contoh 7: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```

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 $\sim a = -11$? Buktikan jawaban Anda dalam perhitungan biner!

- 1) Uraikan permasalahan dan variabel

Pada soal kode program tersebut dibuat menggunakan operator bitwise untuk operasi pada angka biner.

[No.7] Analisis dan Argumentasi

- 1) Saya menganalisis kode program ini digunakan untuk menghitung angka biner dengan operator bitwise.
- 2) Penyebab hasil $\sim a = -11$ yaitu karena angka 10 dalam angka biner adalah -11.

[No.7] Penyusunan Algoritma dan Kode Program

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Mulai
- (b) Install Aplikasi Netbeans atau Eclipse
- (c) Buat Project OperatorJava
- (d) Deklarasi Class OperatorBitwise
- (e) Deklarasi Method bersifat Public Static Void Main
- (f) Deklarasi nilai A dan nilai B
- (g) Gunakan Operator Bitwise
- (h) Print hasil
- (i) Selesai

2) Kode program dan luaran

a) Kode Program

```
java > E-Learning > OperatorBitwise.java > ...
Codeium: Refactor | Explain
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;
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    } }
```

b) Luaran

```
Bitwise
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
PS D:\vscode>
```

c) Analisa uaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Kode program dibuat untuk menghitung angka biner menggunakan operasi bitwise, hasil perhitungan sudah benar dan sesuai.

[No.6] Kesimpulan

1) Analisa

Operator bitwise melakukan operasi pada level bit. Nilai ~a menghasilkan -11 karena operasi bitwise not pada bilangan biner 10 (yang adalah 1010 dalam biner) mengubah semua bit, termasuk tanda, sehingga menghasilkan bilangan negatif.

Refleksi

Untuk materi Operator Java ini cukup susah untuk dipahami, tetapi saya masih bisa mengikutinya. Dalam pengerjaan masih ada kebingungan tetapi dengan usaha dan bantuan dari teman-teman saya bisa mengerjakannya.