

Nama & NPM	Topik:	Tanggal:
Bagas Satrio Winata	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)); //menampilkan hasil penjumlahan
    } }
```

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan tanda + pada System.out.println("a+b= " (a+b)); menjadi System.out.println("a+b= " + (a+b));,dan juga menambahkan fungsi (-,*,/,%)
- 2) Alasan solusi ini karena Ketika menggabungkan teks perlu menambahkan tanda (+). pada setiap operator aritmatika memiliki fungsinya masing-masing dan akan menghasilkan hasil yang sesuai dengan operator
- 3) Perbaikan kode program dapat dilakukan dengan menambahkan tanda (+) dan menambahkan baris dengan mengganti setiap fungsi masing-masing(-,*,/,%)

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma

- Mulai program
- Mendeklarasikan public class operatorAritmatika
- Mendeklarasikan public static void main(String[] args) didalam public class operatorAritmatika
- Mendeklarasikan variable int a = 20, b = 3 didalam public static void main(String[] args)
- Mendeklarasikan Sistem.out.println() untuk menampilkan hasil operator aritmatika yang sesuai dengan fungsinya masing-masing didalam public static void main(String[] args)
- Program selesai

2) Kode program dan luaran

```
public class OperatorAritmatika {
    public static void main(String[] args) {
        // deklarasi nilai variabel
        int a = 20, b = 3; //variabel a untuk nilai 20 dan variabel b untuk nilai 3

        // operator aritmatika penjumlahan
        System.out.println("a: " + a);
        System.out.println("b: " + b);
        System.out.println("a + b = " + (a + b)); // menampilkan hasil penjumlahan
        System.out.println("\n");
        // operator aritmatika pengurangan
        System.out.println("a: " + a);
        System.out.println("b: " + b);
        System.out.println("a - b = " + (a - b)); // menampilkan hasil pengurangan
        System.out.println("\n");
        // operator aritmatika perkalian
        System.out.println("a: " + a);
        System.out.println("b: " + b);
        System.out.println("a * b = " + (a * b)); // menampilkan hasil perkalian
        System.out.println("\n");
        // operator aritmatika pembagian
        System.out.println("a: " + a);
        System.out.println("b: " + b);
        System.out.println("a / b = " + (a / b)); // menampilkan hasil pembagian
        System.out.println("\n");
        // operator aritmatika persen
        System.out.println("a: " + a);
        System.out.println("b: " + b);
        System.out.println("a % b = " + (a % b)); // menampilkan hasil persen
        System.out.println("\n");
    }
}
```

```
Input/Output
Output  Generated Files

a: 20
b: 3
a + b = 23

a: 20
b: 3
a - b = 17

a: 20
b: 3
a * b = 60

a: 20
b: 3
a / b = 6

a: 20
b: 3
a % b = 2

CPU Time: 0.06 secs | Memory: 38776 kilobytes | Compiled and executed in 1.665 secs
```

Analisa luaran yang dihasilkan:

perbaikan pada kode program dan menambahkan fungsi (-,*,/,%,) pada kode program sudah dilakukan. luaran sudah menunjukkan hasil yang sesuai dengan kebutuhan atau permintaan.

[No.1] Kesimpulan dan Analisa

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

Berdasarkan permasalahan yang ada, yaitu kurangnya tanda (+) pada `system.out.println` ("a+b= " (a+b)); sehingga program tidak dapat dijalankan. Dengan melakukan langkah2 yang ada di Algoritma program dapat dilakukann= dan menghasilkan luaran yang sesuai dengan permintaan.

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

Pada program itu saya menggunakan bentuk kelas `public` karena operator aritmatika dapat dilakukan di kelas `public`. Perbaikan program dengan menambahkan tanda (+) pada `system.out.println()` dan menambahkan fungsi (-,*,/,%,) sesuai dengan permintaan. karena struktur java mengharuskan penggabungan teks di `system.out.println()` menggunakan tanda (+).

[No.2] Identifikasi Masalah

```
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  
    }  
}
```

[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan masalah ini dapat diatasi hanya dengan menambah operator(=,*,/,%,/=) dan baris `System.out.println()` untuk menampilkan hasil operator
- 2) Alasan solusi ini karena setiap fungsi masing-masing mempunyai operator yang berbeda
- 3) Argumentasi mengenai perbedaan contoh 1 dan 2:
 - Contoh1 menggunakan operator aritmatika dasar secara langsung tanpa memodifikasi variabel asli. Setiap operasi hanya menghasilkan hasil tanpa mengubah nilai variabel a atau b.
 - Contoh2 menggunakan operator penugasan yang memodifikasi nilai variabel b selama eksekusi program. Nilai b berubah setiap kali dilakukan operasi aritmatika dengan operator penugasan.

[No.2] Menyusun Algoritma dan kode Program

- 1) Algoritma
 - Mulai program
 - Mendeklarasikan `public class operatorPenugasan`
 - Mendeklarasikan `public static void main(String[] args)` didalam `public class operatorPenugasan`
 - Mendeklarasikan variable `int a = 20, b = 3` didalam `public static void main(String[] args)`

- Mendeklarasikan operatorPenugasan (+,=, -=, *=, /=, %=) pada variable didalam public static void main(String[] args)
- Mendeklarasikan Sistem.out.println() untuk menampilkan hasil operator aritmatika yang sesuai dengan fungsinya masing-masing didalam public static void main(String[] args)
- Program selesai

2) Kode Program dan Luaran

```
public class OperatorPenugasan {
    public static void main(String[] args) {
        // deklarasi nilai
        int a = 20, b = 3;

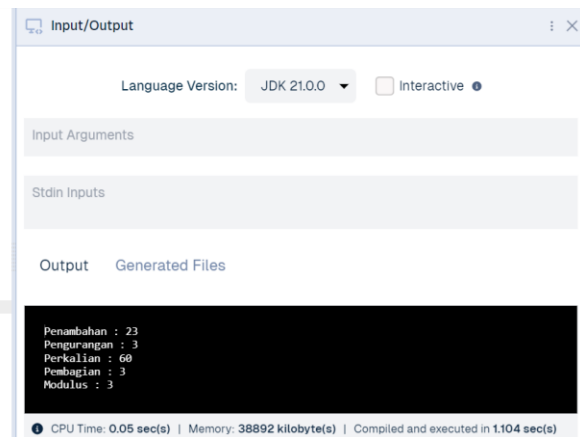
        // operator penugasan
        b += a; // penjumlahan
        System.out.println("Penambahan : " + b); // Menampilkan hasil penjumlahan

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

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

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

        b %= a; // persen
        System.out.println("Modulus : " + b); // Menampilkan hasil persen
    }
}
```



Analisa luaran yang di hasilkan:

Dengan menambahkan operator penugasan (+=, -=, *=, /=, %=) pada kode program. Luaran menghasilkan hasil sesuai dengan kebutuhan dan permintaan

[No.2] Kesimpulan dan Analisa

a) Kesimpulan berdasarkan permasalahan, algoritma, dan kode program:

Program ini menggunakan operator penugasan untuk memodifikasi nilai variabel b dengan operasi aritmatika yang berbeda. Setiap operasi dilakukan dengan menggabungkan operator aritmatika dengan operator penugasan (+,=, -=, *=, /=, %=), sehingga nilai b akan terus berubah selama program dijalankan. Nilai b diubah secara bertahap menggunakan operator penugasan. Setelah setiap operasi, nilai b yang baru ditampilkan. Ini memungkinkan program untuk melakukan modifikasi nilai variabel secara langsung melalui operasi yang lebih singkat. Program tidak menghadapi kesalahan sintaks atau logika. Kode sudah sesuai dengan aturan Java, sehingga tidak ada error atau masalah.

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

Penggunaan kelas public: Pada program ini, saya menggunakan kelas public karena sifatnya yang bisa diakses oleh semua kelas di proyek Java tersebut. public memastikan bahwa kelas tersebut dapat dijalankan oleh JVM saat program dijalankan. Saya menambahkan operator penugasan (+,=, -=, *=, /=, %=) karena struktur Java mengharuskan pendekatan yang lebih efisien dalam memodifikasi variabel selama operasi aritmatika. Ini memberikan cara yang lebih ringkas untuk melakukan operasi aritmatika dan memberikan hasil yang sama seperti penggunaan operator aritmatika biasa dengan penugasan tambahan.

[No.3] Identifikasi Masalah

```
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);
}
}

```

[No.3] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengubah nilai a menjadi 4
- 2) Alasan solusi ini karena ingin melihat perubahan yang akan terjadi
- 3) Perbaiki kode program dengan cara a=12 diganti menjadi a=4

[No.3] Algoritma dan Kode Program

- 1) Algoritma
 - Mulai program
 - Mendeklarasikan public class operatorRelasional
 - Mendeklarasikan public static void main(String[] args) didalam public class operatorRelasional
 - Mendeklarasikan variable int nilai a = 4, int nilai b = 4 didalam public static void main(String[] args)
 - Mendeklarasikan operatorRelasional (<, >, <=, >=, =, ==, !=) pada variable didalam public static void main(String[] args)
 - Mendeklarasikan Sistem.out.println() untuk menampilkan hasil operator Relasional yang sesuai dengan fungsinya masing-masing didalam public static void main(String[] args)
 - Program selesai

2) 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("\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);
    }
}
```

Want me to debug your code?
Debug Code Not Now

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
```

Analisa luaran yang dihasilkan

Ketika nilai dari variabel a di ganti dengan 4.kode program berhasil dijalankan dengan hasil yang berbeda dari yang awal. Luaran menghasilkan hasil yang sesuai dengan kebutuhan dan permintaan.

[no.3] Kesimpulan dan Analisa

Perubahan nilai variabel a=4 dan b =4 menghasilkan :

- Hasil a > b akan menjadi = false
- Hasil a < b akan menjadi = false
- Hasil a >= b akan menjadi = true
- Hasil a <= b akan menjadi = true
- Hasil a == b akan menjadi = true
- Hasil a != b akan menjadi = false

kesimpulan:

- Karena nilai A dan B sama (4), operator Relasional > dan < menghasilkan nilai false.
- Operator >= dan <= menghasilkan nilai true karena A sama dengan B.
- Operator == juga menghasilkan true karena A sama dengan B.
- Operator != akan menghasilkan false karena A dan B sama.

Pada program itu saya menggunakan bentuk kelas public karena didalam Java, kelas public diperlukan untuk memastikan bahwa kelas dapat diakses dari mana saja, baik di dalam program itu sendiri maupun dari kelas atau paket lain. Kelas ini juga memiliki metode main, yang berfungsi sebagai titik awal eksekusi program.

[No.4] Identifikasi Masalah

```
public class operator {
    public static void main(String[] args) {
        // deklarasi nilai
        int a = 5;

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

[No.4] Analisis dan argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan operator pre-increment yaitu ++a, karena jika ingin menampilkan hasil setelah nilai dari variable a harus ditambahkan terlebih dahulu.
- 2) Alasan solusi ini karena operator post-increment a++ akan menampilkan nilai asli dari variable a. sehingga, nilai dari variable a yang akan bertambah. Jadi, untuk menampilkan nilai a yang sudah bertambah secara langsung kita perlu menggunakan pre-increment ++a.
- 3) Perbaiki kode program dengan cara mengubah System.out.println("b: " + (a++)); menjadi System.out.println("b: " + (++a)); agar nilai dari variable a ditampilkan setelah ditambah.

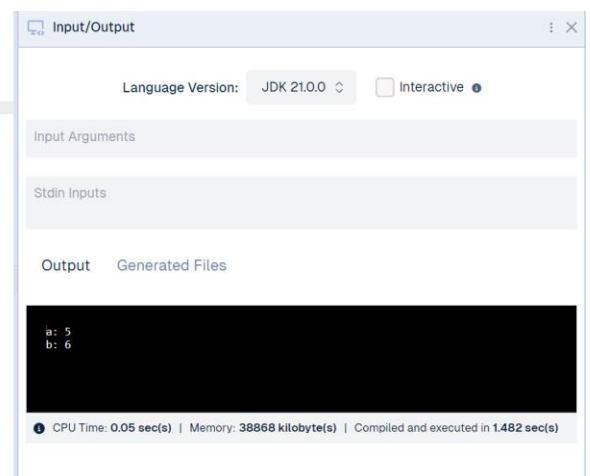
[No.4] Menyusun Algoritma dan Kode Program

1. Algoritma

- Mulai Program
- Mendeklarasikan public class operator
- Mendeklarasikan public static void main(String[] args) didalam public class operator
- Mendeklarasikan tipe data int a untuk 5 didalam public static void main(String[] args).
- Mendeklarasikan System.out.println() untuk menampilkan hasil dari operator didalam public static void main(String[] args).
- Program selesai.

2. Kode program dan luaran

```
public class operator {  
    public static void main(String[] args) {  
        // deklarasi nilai dari variabel  
        int a = 5; // tipe data int a untuk nilai 5  
  
        System.out.println("a: " + ++a); // Mendeklarasikan hasil dari operator  
        System.out.println("b: " + (++a)); // Mendeklarasikan hasil dari operator  
    }  
}
```



Analisa kode program dan hasil luarannya

Kode program ini menghasilkan nilai variable a = 5, b = 6 dapat dilakukan dengan cara mengganti System.out.println("b: " + (a++)); menjadi System.out.println("b: " + (++a));, dengan demikian kode program menghasilkan luaran yang sesuai dengan kebutuhan dan permintaan.

[No.4] Kesimpulan

kesimpulan

Kode ini menggunakan operator increment (a++) pada variabel a = 5, b = 5. Bagaimana operator tersebut bekerja pada variabel sebelum dan sesudah dievaluasi sehingga, menghasilkan nilai variable a = 5, b = 6. Yaitu dengan cara, mengubah operator dari operator post-increment ke operator pre-increment

[No.5] Identifikasi Masalah

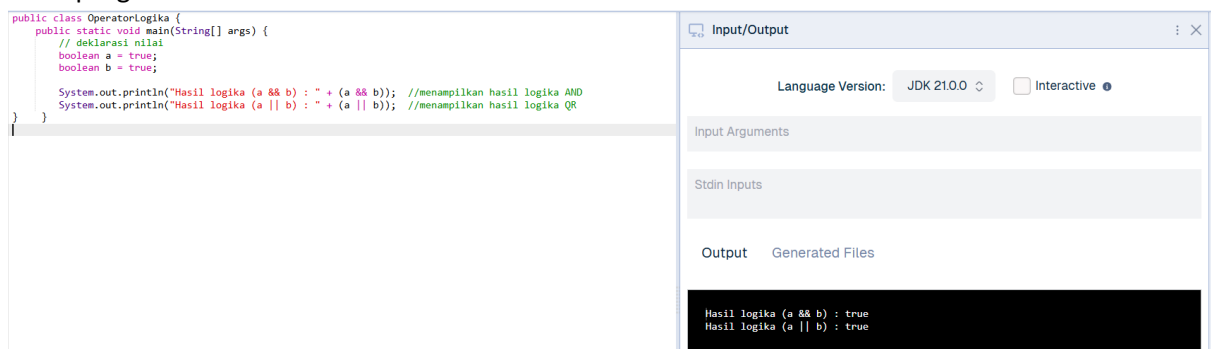
```
public class OperatorLogika {  
    public static void main(String[] args) {  
        // deklarasi nilai  
        boolean a = true;  
        boolean b = false;  
  
        System.out.println("Hasil logika (a && b) : " + (a && b)); //menampilkan hasil logika AND  
    } }  
}
```

[No.5] Analisis dan Argumentasi

- 1) Menurut saya permasalahan ini dapat diatasi dengan cara mengganti nilai Boolean b = false menjadi Boolean b = true. Dan menambahkan System.out.println() untuk operator logika QR
- 2) Alasan saya memilih usulan ini karena jika ingin menghasilkan nilai true maka nilai Booleannya harus di rubah b menjadi true, karena jika nilai b false maka akan menghasilkan false
- 3) Kode program dapat di perbaiki dengan cara mengganti nilai b dan menambah baris system. Out.println() pada program.

[No.5] Menyusun Algoritma dan kode program

1. Algoritma
 - Mulai program
 - Mendeklarasikan public class operatorlogika
 - Mendeklarasikan public static void main(String []args) didalam public class operatorlogika
 - Mendeklarasikan tipe data boolean untuk nilai variable a = true dan nilai variable b = true didalam public static void main(String []args).
 - Mendeklarasikan system.out.println() untuk menampilkan hasil dari operatorlogika AND dan QR didalam public static void main(String []args).
 - Program selesai
2. Kode program dan hasil 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:

```
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  
        System.out.println("Hasil logika (a || b) : " + (a || b)); //menampilkan hasil logika QR  
    } }  
}
```

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

```
Hasil logika (a && b) : true  
Hasil logika (a || b) : true
```

Analisa hasil luaran

Kode program sudah di sesuaikan dengn soal 5.1 dan hasil luaran sudah sesusai dengan kebutuhan atau permintaan.

[No.5] Kesimpulan

Kode program yang sebelum di perbaiki menghasilkan operatorlogika AND = false dan operatorlogika QR = true, sehingga di lakukan perubahan nilai tipe data Boolean b =false ke Boolean b =true dengan demikian, kode program berhasil menampilkan output yang sesuai dengan permintaan

[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 );  
    } }
```

[No.6] Analisis dan Argumentasi

- 1) Menurut saya permasalahan pada kode tersebut dapat diperbaiki dengan cara mengubah tanda lebih besar pada operator kondisi status(nilai >60) menjadi lebih besar sama dengan status (nilai >= 60).
- 2) Alasan saya memilih usulan ini karena pada operator kondisi status (nilai >60) berarti yang lulus hanya nilai yang lebih besar dari 60. Sehingga, nilai 60 it belum memenuhi kriteria nilai yang lulus.
- 3) Kode program ini dapat di perbaiki dengan mengubah tanda lebih besar (>) menjadi lebih besar sama dengan (>=).dan mengubah nilai int nilai=80 menjadi int nilai=60 untuk membuktikan kode program tersebut

[No.6] Menyusun Algoritma dan kode program

1. Algoritma
 - Mulai Program
 - Mendeklarasikan public class operatorKondisi
 - Mendeklarasikan Public static void main (String []args) didalam public class operatorKondisi.
 - Mendeklarasikan tipe data int untuk variable nilai =60 didalam Public static void main (String []args).
 - Mendeklarasikan status = (nilai >=60)?"lulus":"gagal"; didalam Public static void main (String []args).
 - Mendeklarasikan System.out.println() untuk menampilkan hasil dari operator kondisi didalam Public static void main (String []args).
 - Program selesai
2. Kode program dan hasil luaran


```
public class OperatorKondisi{
    public static void main( String[] args ){
        // mendeklarasikan nilai
        String status = "";
        int nilai = 60; // variabel nilai untuk 60
        status = (nilai >= 60)? "lulus": "Gagal"; // status operator kondisi
        System.out.println( status); // mendeklarasikan hasil operator kondisi
    }
}
```

Clipboard

Input/Output

Language Version: JDK 21.0.0 ☐ Interactive

Input Arguments

Stdin Inputs

Output Generated Files

lulus

Analisa hasil luaran

Kode program sudah berhasil melakukan pemograman dengan sedikit perubahan. Alhasil luaran sudah berhasil mengeluarkan output yang sesuai dengan kebutuha dan permintaan

[No.7] Identifikasi Masalah

```
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 );
    } }
```

[No.7] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diselesaikan dengan cara mencari tahu kenapa nilai hasil dari operatorbitwe $\sim a = -11$.

- 2) Alasannya karena, operator bitwise not(~) akan membalikkan setiap bit dari bilangan tersebut dan hasilnya dalam format komplement dua mewakili bilangan negatif.
- 3) Operatorbitwe :yang awalnya nilai a= 10 ketika dipresentasikan ke biner adalah 32-bit, maka 10 adalah: 00000000 00000000 00000000 00001010. Hasil dari ~a =11111111 11111111 11111111 11110101. Karena dimulai dengan tanda bit 1 menunjukkan negatif,maka hasil akhirnya adalah -11.

[No.7] 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
```

[No.7] Kesimpulan

hasil dari ~10 adalah -11 karena operasi bitwise not (~) membalikkan setiap bit dari bilangan, dan hasilnya dalam format komplement dua mewakili bilangan negatif -11.