

**Template Lembar Kerja Individu dan Kelompok**

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Delta Setyawan G1F024056</b>	<b>Operator</b>	<b>08 September 2024</b>
<b>[Nomor Soal] Identifikasi Masalah:</b>		
<ol style="list-style-type: none"><li>1) Uraikan permasalahan dan variabel</li><li>2) Rincikan sumber informasi yang relevan (buku / webpage)</li><li>3) Uraikan rancangan solusi yang diusulkan (jika ada).</li><li>4) Analisis susunan solusi, parameter solusi (jika ada).</li></ol>		
<b>[Nomor Soal] Analisis dan Argumentasi</b>		
<ol style="list-style-type: none"><li>1) Uraikan rancangan solusi yang diusulkan.</li><li>2) Analisis solusi, kaitkan dengan permasalahan.</li></ol>		
<b>[Nomor Soal] Penyusunan Algoritma dan Kode Program</b>		
<ol style="list-style-type: none"><li>1) Rancang desain solusi atau algoritma</li><li>2) Tuliskan kode program dan luaran<ol style="list-style-type: none"><li>a) Beri komentar pada kode</li><li>b) Uraikan luaran yang dihasilkan</li><li>c) Screenshot/ Capture potongan kode dan hasil luaran</li></ol></li></ol>		
<b>[Nomor Soal] Kesimpulan</b>		
<ol style="list-style-type: none"><li>1) Analisa<ol style="list-style-type: none"><li>a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!</li><li>b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?</li></ol></li><li>2) Evaluasi<ol style="list-style-type: none"><li>a) Apa konsekuensi dari skenario pemrograman ini?</li><li>b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)</li></ol></li><li>3) Kreasi<ol style="list-style-type: none"><li>a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?</li><li>b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)</li></ol></li></ol>		

**[No. 1] Identifikasi Masalah:**

- 1) Uraikan permasalahan dan variabel

```
public class OperatorAritmatika{
    public static void main(String[] args) {
        // deklarasi nilai
        int a = 20, b = 3;
        //operator aritmatika
        System.out.println("a: " +a);
        System.out.println("b: " +b);
        System.out.println("a + b = " + (a + b));
    } }
```

**Luaran:**

```
a: 20
b: 3
a + b = 23
```

**Latihan 1**

Susun kode Java untuk perhitungan dengan ekspresi  $(2*3 + 6 / 2 - 4)$ . Simpulkan urutan prioritas operator yang dijalankan ekspresi tersebut!

**[No.1] Analisis dan Argumentasi**

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara
  - a) Menggunakan tipe data int, agar bisa menentukan bilangan bulat/decimal
  - b) Menambahkan deklarasi nilai int c= 6, dan int d=4
  - c) `System.out.println("a + b = " + (a + b));` menjadi `System.out.println("Hasil  $2*3 + 6 / 2 - 4 =$  " + hasil);`
- 2) Alasan solusi ini karena saya harus
  - a) Menggunakan variabel bertipe int yang dideklarasikan dengan nilai
    - $a = 2$
    - $b = 3$
    - $c = 6$
    - $d = 4$
  - b) Menghitung ekspresi dengan aritmatika
    - $a * b = 2*3 = 6$
    - $c / a = 6/2 = 3$
    - $2 / 6 + 3 = 9$
    - $3 / 9 - 4 = 5$

Perbaiki kode program dengan cara

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

        // deklarasi variabel
        int a = 2;
        int b = 3;
        int c = 6;
        int d = 4;
        int hasil;

        // menampilkan variabel
        System.out.println("a: " +a);
        System.out.println("b: " +b);
        System.out.println("c: " +c);
        System.out.println("d: " +d);
```

```

// menampilkan hasil perhitungan
hasil = a * b + c / a - d;
System.out.println("Hasil 2*3 + 6 / 2 - 4 = " + hasil);
}

}

```

## [No.1] Penyusunan Algoritma dan Kode Program

### 1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- Mulai
- Cari permasalahan
- Selesaikan permasalahan
- Selesai

### 2) Kode program dan luaran

#### a. Kode program

Codeium: Refactor | Explain

```

public class OperatorAritmatika {
    Run main | Debug main | Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
    public static void main(String[] args) {
        // Deklarasi variabel
        int a = 2; // nilai a adalah 2
        int b = 3; // nilai b adalah 3
        int c = 6; // nilai c adalah 6
        int d = 4; // nilai d adalah 4
        int hasil; // hasil perhitungan

        // Menampilkan variabel
        System.out.println("a: " +a); // menampilkan nilai a
        System.out.println("b: " +b); // menampilkan nilai b
        System.out.println("c: " +c); // menampilkan nilai c
        System.out.println("d: " +d); // menampilkan nilai d

        // Menampilkan hasil perhitungan
        hasil = a * b + c / a - d; // perhitungan
        System.out.println("Hasil 2*3 + 6 / 2 - 4 = " + hasil); // menampilkan hasil
    }
}

```

#### b. Luaran

```

a: 2
b: 3
c: 6
d: 4
Hasil 2*3 + 6 / 2 - 4 = 5

```

#### a) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan kode program yang disusun.

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

## [No.1] Kesimpulan

### Analisa

Pada program itu saya disuruh untuk mencari luaran menggunakan operator aritmatika, Perbaiki program dengan cara menambahkan deklarasi variabel menggunakan tipe data int a, int b, int c, int d. jadi Outputnya Hasil  $2 * 3 + 6 / 2 - 4 = 5$

**[No. 2] Identifikasi Masalah:**

- 2) Uraikan permasalahan dan variabel

```
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);
    }
}
```

**Luaran:**

Penambahan : 23

**Latihan 2.**

Susun kode untuk menampilkan perhitungan dengan operator ( -=, \*=, /=, %=)!  
Simpulkan hasilnya!

**[No.2] Analisis dan Argumentasi**

- 3) Saya mengusulkan permasalahan ini dapat diatasi dengan cara
- Tidak ada permasalahan dari kode program tersebut karena luaranya ada
  - Operator += sering digunakan untuk memperpendek penulisan kode. Daripada menulis  $b = b + a$ ;, Anda bisa langsung menggunakan  $b += a$ ;
- 4) Alasan solusi ini karena dari kode program sudah memiliki permasalahan, oleh karena itu saya membuat perhitungan dengan operator ( -=, \*=, /=, %=)!.  
5) Perbaiki kode program dengan cara

```
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);

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

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

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

        b %= a;
    }
}
```

```

        System.out.println("Sisa Bagi : " + b);
    }
}

```

## [No.2 ] Penyusunan Algoritma dan Kode Program

### 3) Algoritma

- Mulai
- Cari permasalahan
- Selesaikan permasalahan
- Selesai

### 4) Kode program dan luaran

#### a. Kode program

```

public class OperatorPenugasan {
    public static void main(String[] args) {
        // deklarasi nilai
        int a = 20, b = 3;
        // penjumlahan
        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;
        System.out.println("Sisa Bagi : " + b);
    }
}

```

#### b. Luaran

```

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

```

### b) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan kode program yang disusun.

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

## [No.2] Kesimpulan

### Analisa

Pada program di atas, operator penugasan seperti +=, -=, \*=, /=, dan %= digunakan untuk memodifikasi nilai variabel b secara langsung dengan melibatkan variabel a. Menggunakan penambahan, pengurangan, perkalian, pembagian, dan sisa bagi.

### [No.3] Identifikasi Masalah:

#### 1) Uraikan permasalahan dan variabel

```
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

Susunlah perintah kode dengan operator relasional (<, >, <=, >=, =, ==, !=) untuk nilai a dan b yang menghasilkan luaran TRUE!

### [No.3] Analisis dan Argumentasi

#### 6) Saya mengusulkan permasalahan ini dapat diatasi dengan cara

- a. Tidak ada permasalahannya, karena luaranya sudah ada
- b. Menggunakan kode perintah operator relasional untuk menentukan luaran TRUE semua

Alasan solusi ini karena agar bisa menentukan operator relasional Hasil A > B = true Hasil A < B = false, Hasil A >= B = true, Hasil A <= B = false, Hasil A == B = false, Hasil A != B = true. Jadi luaran yang false harus dirubah ke true

- 7) Perbaiki kode program dengan cara

```
public class OperatorRelasional {
    public static void main(String[] args) {
        int nilaiA = 12; // nilai A di set 12
        int nilaiB = 4; // nilai B di set 4
        boolean hasil; // deklarasi variable boolean

        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] Penyusunan Algoritma dan Kode Program

- 5) Algoritma
  - Mulai
  - Cari permasalahan
  - Selesaikan permasalahan
  - Selesai
- 6) Kode program dan luaran
  - a. Kode program

```

public class OperatorRelasional {
    public static void main(String[] args) {
        int nilaiA = 12; // nilai A di set 12
        int nilaiB = 4; // nilai B di set 4
        boolean hasil; // deklarasi variable boolean

        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);
    }
}

```

b. luaran

```

A = 12
B = 4

Hasil A > B = true

Hasil A < B = true

Hasil A >= B = true

Hasil A <= B = true

Hasil A == B = true

Hasil A != B = true

```

c) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan kode program yang disusun.

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

### [No.3] Kesimpulan

#### Analisa

Pada program itu saya menggunakan bentuk public kelas operator relasional untuk menentukan boolean true semua. Untuk menampilkan hasil true, setiap hasil false akan diubah menjadi true dan saya menggunakan tanda seru, dan tanda kurung. Contohnya !(nilaiA < nilaiB);.

### [No.4] Identifikasi Masalah:

3) Uraikan permasalahan dan variabel



```

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. Susunlah kode program untuk menghasilkan luaran nilai a = 5 dan b = 6 dengan pre/post increment dan pre/post decrement.
- 4.2. Simpulkan hasil perbandingan Anda (pre/post increment, pre/post decrement)!

**[No.4] Analisis dan Argumentasi**

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara
  - a. Tidak ada permasalahan karena sudah memiliki luarannya
  - b. Kode program untuk menghasilkan luaran b = 6
- 2) Alasan solusi ini karena untuk menampilkan luaran a= 5 dan b=6, agar bisa luarannya b di ganti 6, maka saya harus menggunakan ++ di awal sebelum angka (++a).
- 3) Perbaiki kode program dengan cara

```

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] Penyusunan Algoritma dan Kode Program**

- 7) Algoritma
  - Mulai
  - Cari permasalahan
  - Selesaikan permasalahan
  - Selesai
- 8) Kode program dan luaran
  - a. Kode program

```

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

        System.out.println("a: " + a); // menampilkan nilai a
        System.out.println("b: " + (++a)); // increment nilai a dan menampilkannya
    }
}

```

b. Luaran

```

a: 5
b: 6

```

d) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan kode program yang disusun.

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

#### 1) [No.4] Kesimpulan

##### Analisa

Variabel a dideklarasikan dan diinisialisasi dengan nilai 5. Ini menunjukkan cara sederhana untuk mengatur nilai awal pada variabel tipe int. Kode ini menggunakan operator increment (++) dengan gaya prefix (++a). Gaya ini menambah nilai variabel sebelum digunakan dalam ekspresi. Dalam contoh ini, nilai a bertambah dari 5 menjadi 6 sebelum dicetak.

#### [No. 5] Identifikasi Masalah:

4) Uraikan permasalahan dan variable

```

public class operator {
    public static void main(String[] args) {
        // deklarasi nilai
        boolean a = true;
        boolean b = false;

        System.out.println("Hasil logika (a && b) : " + (a && b));
    }
}

```

5) Luaran:

Hasil logika (a && b) : false

## 6) Latihan 5

Susun kode program dengan mengubah nilai a dan b untuk menghasilkan luaran true dengan operator && dan operator | |. Beri kesimpulan!

### [No.5] Analisis dan Argumentasi

- 7) Saya mengusulkan permasalahan ini dapat diatasi dengan cara
  - a. keluaran akan true jika kedua operand bernilai true
  - b. keluaran akan true jika salah satu atau kedua operand bernilai true
- 8) Alasan solusi ini karena untuk operator && kedua nilai a dan b harus true. Untuk operator | |, cukup salah satu dari nilai a atau b yang true agar hasilnya juga true
- 9) Perbaiki kode program dengan cara

```
public class Operator {  
    public static void main(String[] args) {  
        // deklarasi nilai  
        boolean a = true;  
        boolean b = true; // Ubah b menjadi true  
  
        // Operator AND (&&)  
        System.out.println("Hasil logika (a && b) : " + (a && b)); // true && true = true  
  
        // Operator OR (| |)  
        System.out.println("Hasil logika (a | | b) : " + (a | | b)); // true | | true = true  
    }  
}
```

### [No.5] Penyusunan Algoritma dan Kode Program

- 9) Algoritma
  - Mulai
  - Cari permasalahan
  - Selesaikan permasalahan
  - Selesai

## 10) Kode program dan luaran

- a. Kode program

```
public class Operator {  
    public static void main(String[] args) {  
        // deklarasi nilai  
        boolean a = true;  
        boolean b = true; // Ubah b menjadi true  
  
        // Operator AND (&&)  
        System.out.println("Hasil logika (a && b) : " + (a && b)); // true && true = true  
  
        // Operator OR (| |)  
        System.out.println("Hasil logika (a | | b) : " + (a | | b)); // true | | true = true  
    }  
}
```

b. Luaran

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

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

#### [No.5] Kesimpulan

##### 2) Analisa

Operator AND (&&) lebih ketat karena membutuhkan kedua operand bernilai `true` untuk menghasilkan `true`.

Operator OR (||) lebih longgar karena hanya memerlukan satu operand bernilai `true` untuk menghasilkan `true`.

#### [No. 6 Identifikasi Masalah:

- 4) Uraikan permasalahan dan variabel

```
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**

Susun kode program! Dengan informasi berikut:

Diketahui nama variabel Jam = 12

Apabila jam < 12 maka tampil "Selamat Pagi", apabila jam > 12 maka tampil "Selamat Malam".

#### [No.6] Analisis dan Argumentasi

- 8) Saya mengusulkan permasalahan ini dapat diatasi dengan cara
  - a. Menggunakan tipe data `int` untuk menenrukan angka
  - b. Menggunakan tipe data `string` untuk menentukan status
- 9) Alasan solusi ini karena untuk menampilkan luaran selamat malam, maka kita harus (<) dan (>) dengan menggunakan `status = (jam < 12)?"Selamat Pagi":"Selamat Malam";`.
- 10) Perbaiki kode program dengan cara

```

public class kondisional {
    public static void main(String[] args) {
        int jam = 12;
        String status;

        status = (jam < 12)?"Selamat Pagi":"Selamat Malam";
        System.out.println(status);
    }
}

```

#### [No.6] Penyusunan Algoritma dan Kode Program

##### 11) Algoritma

- Mulai
- Cari permasalahan
- Selesaikan permasalahan
- Selesai

##### 12) Kode program dan luaran

###### a. Kode progrsm

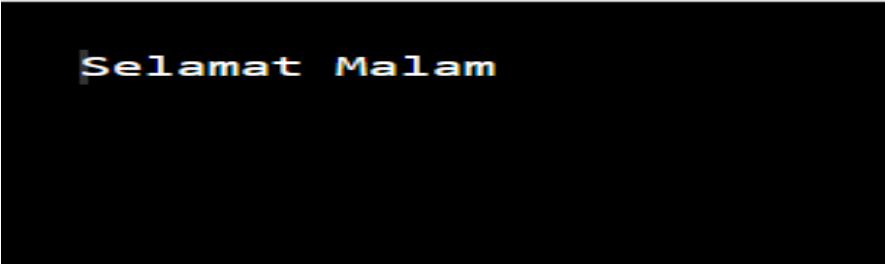
```

public class kondisional {
    public static void main(String[] args) {
        int jam = 12;
        String status;

        status = (jam < 12)?"Selamat Pagi":"Selamat Malam";
        System.out.println(status);
    }
}

```

###### b. Luaran



```

Selamat Malam

```

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

## [No.6 ] Kesimpulan

### 3) Analisa

Variabel jam adalah sebuah bilangan bulat yang mewakili waktu dalam format 24 jam.  
Dalam contoh ini, jam diinisialisasi dengan nilai 12.

Variabel status adalah sebuah string yang akan digunakan untuk menyimpan hasil dari pengecekan kondisi waktu, yaitu apakah akan menampilkan "Selamat Pagi", "Selamat Siang", atau "Selamat Malam".

## [No. 7] Identifikasi Masalah:

- 5) Uraikan permasalahan dan variabel

```
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 );  
    }  
}
```

### Luaran:

```
Hasil dari a & b : 6  
Hasil dari a | b : 7  
Hasil dari a ^ b : 1
```

### Latihan 7

Susun kode tambahan dari Contoh 7 untuk melakukan perhitungan dengan operator (>>, <<). Hubungkan hasil luaran dengan perhitungan manual bilangan biner!

## [No.7] Analisis dan Argumentasi

- 11) Saya mengusulkan permasalahan ini dapat diatasi dengan cara
- a >> 1: Geser bit 1010 satu tempat ke kanan menjadi 0101 (yaitu 5 dalam desimal).
  - b >> 1: Geser bit 0111 satu tempat ke kanan menjadi 0011 (yaitu 3 dalam desimal).
  - Operator geser kiri (<<)
  - Geser bit ke kiri, menggeser bit ke kiri dan menambahkan bit 0 di tempat paling kanan.
  - a << 1: Geser bit 1010 satu tempat ke kiri menjadi 10100 (yaitu 20 dalam desimal).
  - b << 1: Geser bit 0111 satu tempat ke kiri menjadi 1110 (yaitu 14 dalam desimal).
- 12) Alasan solusi ini karena saya tinggal menghitung biner di setiap angka bit
- 13) Perbaiki kode program dengan cara

```

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 );

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

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

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

        hasil = b << 1;
        System.out.println("Hasil dari b << 1 : " + hasil );
    }
}

```

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

##### **13) Algoritma**

- Mulai
- Cari permasalahan
- Selesaikan permasalahan
- Selesai

##### **14) Kode program dan luaran**

- a. Kode program

```

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 );
        // Operator geser
        hasil = a >> 1;
        System.out.println("Hasil dari a >> 1 : " + hasil );
        hasil = b >> 1;
        System.out.println("Hasil dari b >> 1 : " + hasil );
        hasil = a << 1;
        System.out.println("Hasil dari a << 1 : " + hasil );
        hasil = b << 1;
        System.out.println("Hasil dari b << 1 : " + hasil );
    }
}

```

b. Luaran

```

Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
Hasil dari a >> 1 : 5
Hasil dari b >> 1 : 3
Hasil dari a << 1 : 20
Hasil dari b << 1 : 14

```

g) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan kode program yang disusun.

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

## [No.7] Kesimpulan

### Analisa

Operasi bitwise didasarkan pada manipulasi bit dari bilangan biner. Setiap operasi dapat diuraikan dalam langkah perbit, sehingga hasilnya bisa diverifikasi dengan cara perhitungan manual bilangan biner. Misalnya:

- Operasi & antara bit 1010 dan 0111 menghasilkan 0010, yang merupakan 2 dalam desimal.
- Operasi >> pada 1010 menggeser semua bit ke kanan dan menghasilkan 0101, yang merupakan 5 dalam desimal, setara dengan membagi 10 dengan 2.

### Refleksi

Praktikum ini dilakukan secara individu, namun saya sering berdiskusi dengan teman sekelas untuk bertukar ide dan memecahkan masalah bersama. Diskusi ini sangat membantu dalam memahami berbagai pendekatan untuk menyelesaikan masalah dan memperbaiki desain aplikasi. Kolaborasi dalam diskusi ini juga membantu saya melihat cara orang lain menghadapi tantangan serupa.