

### Template Lembar Kerja Individu

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Nafiisah Calista Hersa Agustine (G1F024077)</b>	<b>Operator Data</b>	<b>09 September 2024</b>
<b>[Nomor 1] Identifikasi Masalah:</b>		
<p>1) Uraikan permasalahan dan variable</p> <pre>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));     } }</pre> <p>Luaran:</p> <pre>a: 20 b: 3 a - b = 17</pre> <p>1.2. Analisa perhitungan matematika yang terjadi!</p> <ul style="list-style-type: none"> <li>• <b>Pengurangan (a - b):</b> Perhitungan: 20 – 3 Hasil: 17</li> <li>• <b>Penambahan (a + b):</b> Perhitungan: 20 + 3 Hasil: 23</li> <li>• <b>Perkalian (a * b):</b> Perhitungan: 20 * 3 Hasil: 60</li> <li>• <b>Pembagian (a / b):</b> Perhitungan: 20 / 3 Hasil: 6 (pembagian integer membulatkan hasil ke bawah)</li> <li>• <b>Modulus (a % b):</b> Perhitungan: 20 % 3 Hasil: 2 (sisa dari pembagian)</li> </ul> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p><a href="#">Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada laman https://www.youtube.com/watch?v=PzCMZOboxZM</a></p> <p><a href="#">Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman https://www.youtube.com/watch?v=LcFgl0yrKEw</a></p>		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Pada tugas ke 1 kali ini tidak terjadi permasalahan , hanya menambahkan baris <code>System.out.println("a + b = " + (a + b));</code> dan mengubah operator ( + ) dengan tanda ( -, *, /, %)</p>		
<b>[Nomor 1] Penyusunan Algoritma dan Kode Program</b>		
<p>1) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> <li>• Mulai Program</li> <li>• Deklarasi dan Inisialisasi Variabel Yaitu int a= 20, int b=3</li> <li>• Tampilkan Nilai Variabel</li> <li>• Operasi Aritmatika dan Tampilkan Hasil</li> <li>• Akhiri Program</li> </ul> <p>2) Tuliskan kode program dan luaran</p>		

a) Beri komentar pada kode

Kode ini untuk menghitung sebuah operasi aritmatika digunakan untuk melakukan perhitungan seperti penambahan, pengurangan, pembagian, perkalian, sisa bagi.

Program ini dimulai dengan mendeklarasikan sebuah kelas bernama OperatorAritmatika. Di dalam kelas tersebut, terdapat method main yang merupakan titik masuk utama eksekusi program

b) Uraikan luaran yang dihasilkan

a: 20

b: 3

a + b = 17

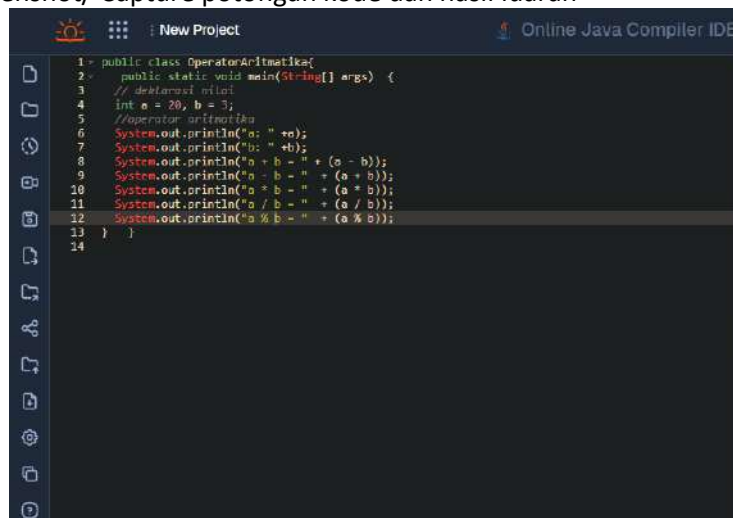
a - b = 23

a \* b = 60

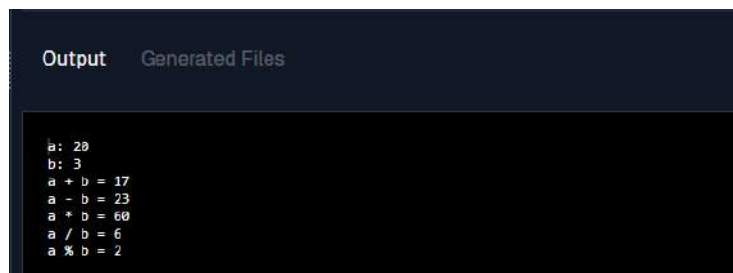
a / b = 6

a % b = 2

c) Screenshot/ Capture potongan kode dan hasil luaran



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



```
Output    Generated Files

a: 20
b: 3
a + b = 17
a - b = 23
a * b = 60
a / b = 6
a % b = 2
```

### [Nomor 1] Kesimpulan

1) Evaluasi

a) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)

Program Java ini mendeklarasikan dua variabel integer, a dan b, dengan nilai masing-masing 20 dan 3. Program kemudian melakukan beberapa operasi aritmatika pada kedua variabel tersebut, yaitu pengurangan, penjumlahan, perkalian, pembagian, dan modulus. Hasil dari setiap operasi aritmatika dicetak ke layar bersama dengan deskripsi yang sesuai.

## Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Nafiisah Calista Hersa Agustine (G1F024077)	Operator Data	09 September 2024

### [Nomor 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

```
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);
    }
}
```

#### Luaran:

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

#### 2.1. Bandingkan hasil Contoh 1 dengan Contoh 2!

Contoh 1 hanya menunjukkan hasil penjumlahan dari dua angka, yaitu 20 dan 3, yang menghasilkan 23. Sedangkan contoh 2 menunjukkan bagaimana satu variabel (b) diubah melalui beberapa operasi berturut-turut. Dimulai dari 3, nilai b berubah menjadi 23 setelah penambahan, kembali ke 3 setelah pengurangan, menjadi 60 setelah perkalian, 3 lagi setelah pembagian, dan tetap 3 setelah operasi sisa bagi.

- 2) Rincikan sumber informasi yang relevan (buku / webpage)

[Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada](#)

[laman https://www.youtube.com/watch?v=PzCMZOObexZM](https://www.youtube.com/watch?v=PzCMZOObexZM)

[Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada](#)  
[laman https://www.youtube.com/watch?v=LcFgl0yrKEw](https://www.youtube.com/watch?v=LcFgl0yrKEw)

### [Nomor 2] Analisis dan Argumentasi

- 1) Uraikan rancangan solusi yang diusulkan.

Program ini menggunakan variabel a dan b, di mana a diatur ke 20 dan b ke 3. Dengan menggunakan berbagai operator penugasan (+, -, \*, /, %), program melakukan operasi matematika pada variabel b dan mencetak hasilnya setelah setiap operasi.

- 2) Analisis solusi, kaitkan dengan permasalahan.

Program menunjukkan bagaimana nilai variabel b berubah setelah operasi penambahan, pengurangan, perkalian, pembagian, dan sisa bagi. Hasil yang ditampilkan sesuai dengan

operasi matematika yang dilakukan, sehingga membantu memahami bagaimana operator penugasan bekerja dalam Java.

#### [Nomor 2] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
  - Mulai
  - Deklarasi dan Inisialisasi Variabel:
  - Penambahan:
  - Pengurangan:
  - Perkalian:
  - Pembagian:
  - Sisa Bagi:
  - Selesai
- 2) Tuliskan kode program dan luaran
  - a) Beri komentar pada kode

Program ini menunjukkan bagaimana operator penugasan digunakan untuk memodifikasikan nilai variabel b dengan melakukan berbagai operasi aritmatika menggunakan variabel a

Inisialisasi: Variabel a diinisialisasi dengan nilai 20, dan variabel b diinisialisasi dengan nilai 3.

Operasi Penugasan: penambahan, pengurangan, perkalian, pembagian, sisa bagi
  - b) Uraikan luaran yang dihasilkan

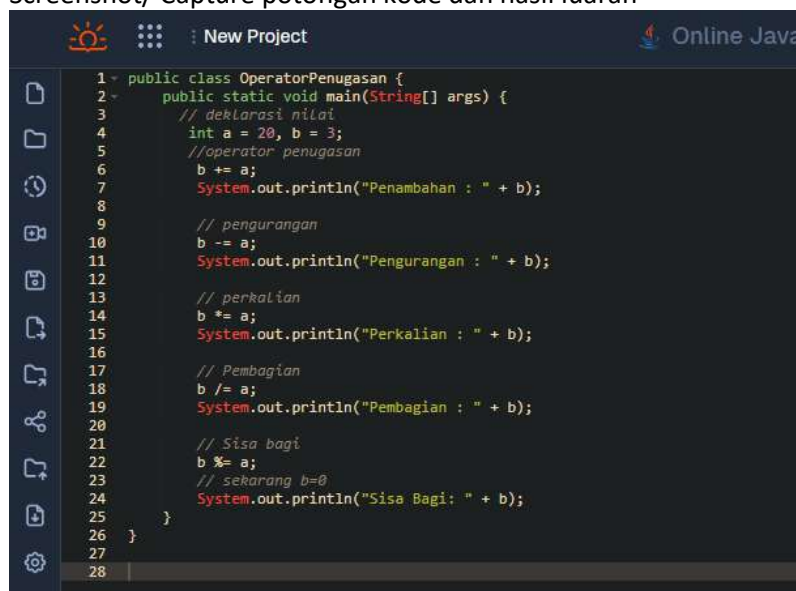
Penambahan : 23

Pengurangan : 3

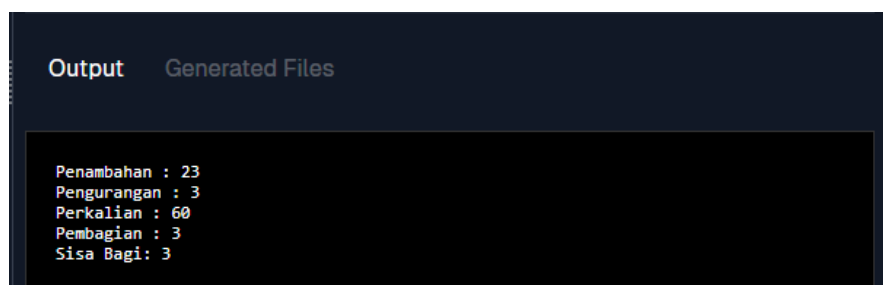
Perkalian : 60

Pembagian : 3

Sisa Bagi: 3
  - c) Screenshot/ Capture potongan kode dan hasil luaran



```
1 public class OperatorPenugasan {
2     public static void main(String[] args) {
3         // deklarasi nilai
4         int a = 20, b = 3;
5         //operator penugasan
6         b += a;
7         System.out.println("Penambahan : " + b);
8
9         // pengurangan
10        b -= a;
11        System.out.println("Pengurangan : " + b);
12
13        // perkalian
14        b *= a;
15        System.out.println("Perkalian : " + b);
16
17        // Pembagian
18        b /= a;
19        System.out.println("Pembagian : " + b);
20
21        // Sisa bagi
22        b %= a;
23        // sekarang b=0
24        System.out.println("Sisa Bagi: " + b);
25    }
26 }
27
28
```



```
Output    Generated Files

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

<b>[Nomor 2] Kesimpulan</b>
<div data-bbox="229 228 391 264" data-label="Section-Header"><p>1) Evaluasi</p></div> <div data-bbox="229 264 900 302" data-label="Section-Header"><p>a) Apa konsekuensi dari skenario pemrograman ini?</p></div> <div data-bbox="181 300 1353 407" data-label="Text"><p>Konsekuensi dari skenario dalam kode operasi penugasan dapat dipahami dalam beberapa cara: Perubahan Nilai Variabel, Output yang Tidak Diharapkan, Pemahaman Kode, Debugging dan Maintenance, Kesalahan Penulisan atau Pemahaman .</p></div> <div data-bbox="181 407 1382 622" data-label="List-Group"><ul style="list-style-type: none"><li>- Kode ini mengilustrasikan perubahan bertahap dalam nilai variabel b dengan menggunakan berbagai operator penugasan.</li><li>- Hasil dari operasi terakhir seharusnya mencerminkan sisa pembagian yang benar, tetapi komentar akhir menunjukkan ketidakakuratan yang dapat membingungkan.</li><li>- Penting untuk memastikan bahwa komentar kode selalu diperbarui dan akurat agar kode mudah dipahami dan dipelihara.</li></ul></div>

### Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Nafiisah Calista Hersa Agustine (G1F024077)	Operator Data	09 September 2024

#### [Nomor 3] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

```
public class OperatorRealasional {  
    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("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);  
    }  
}
```

#### 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
```

3.1. Ubahlah nilai A = 4 dan B = 4. Analisa perubahan yang terjadi!

Perubahan yan terjadi adalah

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

3.2 Bandingkan bagaimana perbedaan nilai A dan B mempengaruhi nilai luaran!

Jika A dan B sama, operasi >, <, dan != menghasilkan false, sedangkan operasi >=, <=, dan ==

menghasilkan true.

Jika A dan B berbeda, hasil dari  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $==$ , dan  $!=$  akan tergantung pada nilai relatif A dan B.

2) Rincikan sumber informasi yang relevan (buku / webpage)

[Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada](#)

[laman https://www.youtube.com/watch?v=PzCMZOboxZM](https://www.youtube.com/watch?v=PzCMZOboxZM)

[Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada](#)  
[laman https://www.youtube.com/watch?v=LcFgl0yrKEw](https://www.youtube.com/watch?v=LcFgl0yrKEw)

### **[Nomor 3] Analisis dan Argumentasi**

1) Uraikan rancangan solusi yang diusulkan.

Untuk membandingkan dua angka, mulai dengan menentukan dua angka yang akan dibandingkan. Kemudian, menggunakan beberapa tanda perbandingan (seperti lebih besar dari, lebih kecil dari, atau sama dengan) untuk melihat bagaimana angka-angka itu berhubungan. Setelah itu, mencetak hasil perbandingan untuk memeriksa apakah hasilnya sesuai dengan yang diharapkan. Juga melakukan beberapa uji coba dengan nilai yang berbeda untuk memastikan semuanya bekerja dengan benar.

2) Analisis solusi, kaitkan dengan permasalahan.

Dengan perbandingan ini, kita bisa mengetahui apakah satu angka lebih besar, lebih kecil, atau sama dengan angka lainnya. Program ini sederhana dan efektif untuk memberikan informasi yang jelas tentang hubungan antara dua angka. Jadi, solusi ini membantu kita memahami bagaimana dua angka saling berhubungan dengan cara yang mudah dan andal.

### **[Nomor 3] Penyusunan Algoritma dan Kode Program**

1) Rancang desain solusi atau algoritma

- Mulai
- Deklarasi Variabel
- Input Nilai
- Perbandingan
- Output Hasil
- Selesai

2) Tuliskan kode program dan luaran

a) Beri komentar pada kode

Kode ini membandingkan dua angka menggunakan operator perbandingan ( $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $==$ ,  $!=$ ) dan menampilkan hasilnya. Program memeriksa berbagai kondisi hubungan antara dua nilai dan menampilkan apakah setiap kondisi benar (true) atau salah (false). Ini membantu memahami bagaimana dua angka berhubungan satu sama lain secara jelas dan sederhana.

b) Uraikan luaran yang dihasilkan

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

c) Screenshot/ Capture potongan kode dan hasil luaran

```
Online Java Compiler IDE
New Project

1 public class OperatorRelasional {
2     public static void main(String[] args) {
3         int nilaiA = 4;
4         int nilaiB = 4;
5         boolean hasil;
6
7         System.out.println(" A = " + nilaiA + "\n B = " + nilaiB);
8         // apakah A lebih besar dari B?
9         hasil = nilaiA > nilaiB;
10        System.out.println("Hasil A > B = " + hasil);
11
12        // apakah A lebih kecil dari B?
13        hasil = nilaiA < nilaiB;
14        System.out.println("Hasil A < B = " + hasil);
15
16        // apakah A lebih besar samadengan B?
17        hasil = nilaiA >= nilaiB;
18        System.out.println("Hasil A >= B = " + hasil);
19
20        // apakah A lebih kecil samadengan B?
21        hasil = nilaiA <= nilaiB;
22        System.out.println("Hasil A <= B = " + hasil);
23
24        // apakah nilai A sama dengan B?
25        hasil = nilaiA == nilaiB;
26        System.out.println("Hasil A == B = " + hasil);
27
28        // apakah nilai A tidak samadengan B?
29        hasil = nilaiA != nilaiB;
30        System.out.println("Hasil A != B = " + hasil);
31    }
32 }
```

#### 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
```

### [Nomor 3] Kesimpulan

- 1) Evaluasi
  - a) Apa konsekuensi dari skenario pemrograman ini?

Program ini membandingkan dua angka dan menampilkan hasil perbandingan. Ini akurat, sederhana, dan berguna untuk belajar logika perbandingan. Program mungkin perlu penanganan tambahan untuk input yang salah.



## Template Lembar Kerja Individu

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Nafiisah Calista Hersa Agustine (G1F0240770)</b>	<b>Operator Data</b>	<b>09 September 2024</b>

### [Nomor 4] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

```
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);
    }
}
```

### Luaran:

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

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

1. Post Increment (a++):

- Cetakan Pertama: Nilai variabel sebelum increment (lama).
- Cetakan Kedua: Nilai variabel setelah increment (baru).

2. Pre Increment (++b):

- Cetakan Pertama: Nilai variabel setelah increment (baru).
- Cetakan Kedua: Nilai variabel yang sama (karena sudah diincrement).

3. Post Decrement (c--):

- Cetakan Pertama: Nilai variabel sebelum decrement (lama).
- Cetakan Kedua: Nilai variabel setelah decrement (baru).

4. Pre Decrement (--d):

- Cetakan Pertama: Nilai variabel setelah decrement (baru).
- Cetakan Kedua: Nilai variabel yang sama (karena sudah didecrement).

Kesimpulan:

- Post Increment/Decrement: Mencetak nilai sebelum perubahan (lama) lalu setelah perubahan (baru).
- Pre Increment/Decrement: Mencetak nilai setelah perubahan (baru).

2) Rincikan sumber informasi yang relevan (buku / webpage)

[Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada](#)

[laman https://www.youtube.com/watch?v=PzCMZOboxZM](https://www.youtube.com/watch?v=PzCMZOboxZM)

[Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada](#)  
[laman https://www.youtube.com/watch?v=LcFgl0yrKEw](https://www.youtube.com/watch?v=LcFgl0yrKEw)

#### [Nomor 4] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

Rancangan Solusi:

- Memahami perbedaan antara operator Post dan Pre dalam konteks Increment (++) dan Decrement (--) dalam bahasa pemrograman Java.
- Operator Post Increment/Decrement (a++, c--) dan Pre Increment/Decrement (++b, --d) memiliki cara kerja yang berbeda dalam hal waktu perubahan nilai.
- Bandingkan hasil yang diperoleh untuk Post dan Pre Increment/Decrement.
- Pastikan hasil sesuai dengan ekspektasi bahwa Post mencetak nilai sebelum perubahan dan Pre mencetak nilai setelah perubahan.

2) Analisis solusi, kaitkan dengan permasalahan.

Solusi:

- Post Increment/Decrement: Cetak nilai lama, lalu ubah variabel.
- Pre Increment/Decrement: Ubah variabel dulu, lalu cetak nilai baru.

Kaitan dengan Permasalahan:

- Memahami Perbedaan: Program ini menunjukkan perbedaan jelas antara Post dan Pre, membantu memahami kapan nilai lama atau baru digunakan.
- Pemilihan Operator: Memudahkan pemilihan operator yang tepat sesuai kebutuhan—nilai lama (Post) atau nilai baru (Pre).

#### [Nomor 4] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma

1. Inisialisasi Variabel
2. Post Increment
3. Pre Increment
4. Post Decrement
5. Pre Decrement

## 6. Akhiri

### 2) Tuliskan kode program dan luaran

#### a) Beri komentar pada kode

Post Increment/Decrement (a++, c--) mencetak nilai lama sebelum perubahan dan kemudian mengubah nilai variabel, sedangkan Pre Increment/Decrement (++b, --d) mengubah nilai variabel terlebih dahulu dan kemudian mencetak nilai baru.

#### b) Uraikan luaran yang dihasilkan

# Post Increment #

=====

Isi variabel a: 10

Isi variabel a: 10

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 #

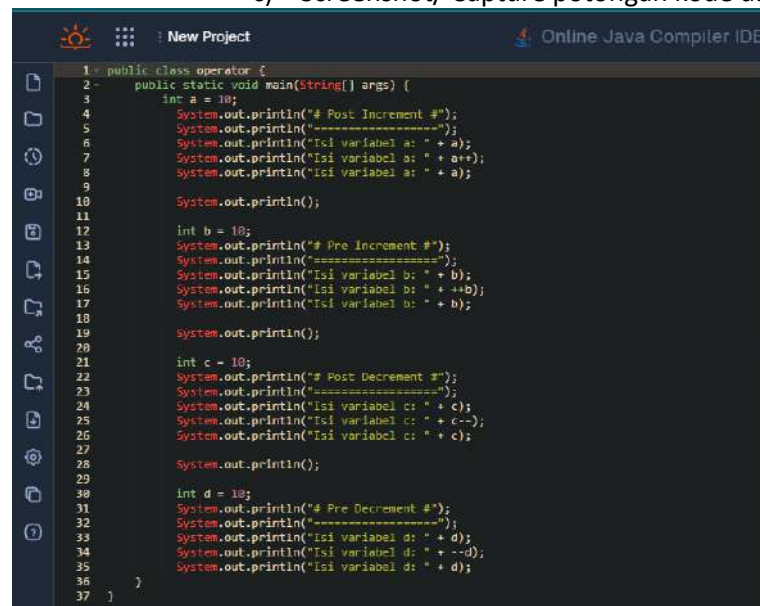
=====

Isi variabel d: 10

Isi variabel d: 9

Isi variabel d: 9

#### c) Screenshot/ Capture potongan kode dan hasil luaran



```
1 public class operator {
2     public static void main(String[] args) {
3         int a = 10;
4         System.out.println("# Post Increment #");
5         System.out.println("=====");
6         System.out.println("Isi variabel a: " + a);
7         System.out.println("Isi variabel a: " + a++);
8         System.out.println("Isi variabel a: " + a);
9         System.out.println();
10
11         int b = 10;
12         System.out.println("# Pre Increment #");
13         System.out.println("=====");
14         System.out.println("Isi variabel b: " + b);
15         System.out.println("Isi variabel b: " + ++b);
16         System.out.println("Isi variabel b: " + b);
17         System.out.println();
18
19         int c = 10;
20         System.out.println("# Post Decrement #");
21         System.out.println("=====");
22         System.out.println("Isi variabel c: " + c);
23         System.out.println("Isi variabel c: " + c--);
24         System.out.println("Isi variabel c: " + c);
25         System.out.println();
26
27         int d = 10;
28         System.out.println("# Pre Decrement #");
29         System.out.println("=====");
30         System.out.println("Isi variabel d: " + d);
31         System.out.println("Isi variabel d: " + --d);
32         System.out.println("Isi variabel d: " + d);
33     }
34 }
```

## Output    Generated Files

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

### [Nomor 4] Kesimpulan

#### 1) Evaluasi

a) Apa konsekuensi dari skenario pemrograman ini?

Post Increment/Decrement (a++, c--) menggunakan nilai lama sebelum mengubah variabel, sementara Pre Increment/Decrement (++b, --d) mengubah nilai variabel terlebih dahulu. Ini mempengaruhi urutan dan hasil operasi dalam kode, jadi penting untuk memilih yang sesuai dengan kebutuhan logika program.

### Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Nafiisah Calista Hersa Agustine (G1F024077)	Operator Data	10 September 2024

#### [Nomor 5] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

```
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);  
    }  
}
```

Luaran:

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

5.1. Tambahkan baris kode untuk memeriksa a || b.

```
public class OperatorLogika {  
    public static void main (String [] args) {  
        boolean a = false;  
        boolean b = false;  
        boolean c;  
  
        // AND operator (&&)  
        c = a && b;  
        System.out.println("false && false = " +c);  
  
        // OR operator (||)  
        c = a || b;  
        System.out.println("false || false = " +c);  
    }  
}
```

5.2. Ubahlah nilai a = false dan b = false. Analisa perubahan dan perbedaan boolean yang terjadi!

```
public class OperatorLogika {  
    public static void main (String [] args) {  
        boolean a = false;  
        boolean b = false;  
        boolean c;  
  
        // AND operator (&&)  
        c = a && b;  
        System.out.println("false && false = " +c);  
  
        // OR operator (||)  
        c = a || b;  
        System.out.println("false || false = " +c);  
    }  
}
```

Luaran:

```
false && false = false
```

```
false || false = false
```

analisis perubahan dan perbedaan yang terjadi:

Perubahan yang terjadi adalah ketika kedua nilai a dan b bernilai false, maka hasil dari a && b

adalah false karena operator AND memerlukan kedua operand untuk bernilai true. Sedangkan hasil dari `a || b` juga false karena operator OR memerlukan setidaknya satu operand bernilai true.

5.3. Apabila diketahui pernyataan `a || b && a || !b`. Uraikan urutan logika yang akan dikerjakan! Analisa luaran true atau false dari pernyataan tersebut!

```
public class OperatorLogika {  
    public static void main (String [] args) {  
        boolean a = false;  
        boolean b = false;  
        boolean c;  
  
        // AND operator (&&)  
        c = a && b;  
        System.out.println("false && false = " + c);  
  
        // OR operator (||)  
        c = a || b;  
        System.out.println("false || false = " + c);  
  
        // Ekspresi logika a || b && a || !b  
        boolean result = a || b && a || !b;  
        System.out.println("false || false && false || !false = " + result);  
    }  
}
```

Luaran:

false && false = false

false || false = false

false || false && false || !false = true

uraikan urutan logika dan analisis terdapat false :

operator AND (&&) memiliki prioritas lebih tinggi dibandingkan dengan OR (||).

- `b && a` menjadi false && false hasilnya false.
- `a || (false)` menjadi false || false, hasilnya false.
- `!b` menjadi !false, hasilnya true.
- Akhirnya, false || true, hasilnya adalah true.

2) Rincikan sumber informasi yang relevan (buku / webpage)

[Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada](#)

[laman https://www.youtube.com/watch?v=PzCMZOboxZM](https://www.youtube.com/watch?v=PzCMZOboxZM)

[Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada](#)

[laman https://www.youtube.com/watch?v=LcFgl0yrKEw](https://www.youtube.com/watch?v=LcFgl0yrKEw)

#### [Nomor 5] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

- Program ini mendemonstrasikan penggunaan operator logika dasar (&&, ||, dan !) dalam Java.
- Penilaian prioritas operator logika penting untuk memastikan hasil evaluasi yang benar.
- Output yang dihasilkan mencerminkan evaluasi dari ekspresi logika yang ditentukan dalam kode.

2) Analisis solusi, kaitkan dengan permasalahan.

- **Analisis Prioritas Operator:** Dalam ekspresi logika, operator && memiliki prioritas lebih tinggi daripada ||. Oleh karena itu, `b && a` dihitung terlebih dahulu sebelum `a || (result) || !b` dievaluasi.

- **Permasalahan:** Ketika menyusun ekspresi logika yang kompleks, penting untuk memahami prioritas operator untuk mendapatkan hasil yang benar. Mengabaikan prioritas operator bisa menghasilkan hasil yang tidak terduga.

#### [Nomor 5] Penyusunan Algoritma dan Kode Program

##### 1) Rancang desain solusi atau algoritma

- -inisialisasi variable
- -evaluasi operator AND (&&)
- -evaluasi operator OR (||)
- -evaluasi ekspresi logika kompleks

##### 2) Tuliskan kode program dan luaran

###### a) Beri komentar pada kode

- baris ini menjelaskan bahwa variabel a dan b diinisialisasi dengan nilai false, dan variabel c disiapkan untuk menyimpan hasil sementara dari operasi logika.
- Komentar operasi AND menjelaskan bahwa operasi a && b adalah operasi AND yang memerlukan kedua operand bernilai true untuk menghasilkan true. Karena kedua operand false, hasilnya adalah false.
- Hasil dari operasi ini disimpan di variabel c dan dicetak.
- Komentar operasi OR menjelaskan bahwa operasi a || b adalah operasi OR yang memerlukan salah satu operand bernilai true untuk menghasilkan true. Karena kedua operand false, hasilnya adalah false.
- Hasil dari operasi ini disimpan di variabel c dan dicetak.
- Komentar evaluasi ekspresi logika kompleks menjelaskan bahwa ekspresi a || b && a || !b melibatkan prioritas operator. && memiliki prioritas lebih tinggi daripada ||, sehingga b && a dievaluasi terlebih dahulu. !b dievaluasi sebagai true karena b adalah false.
- Setelah evaluasi, ekspresi menjadi a || (b && a) || !b. Dengan a = false, b = false, dan !b = true, hasil akhir adalah true.
- Hasil dari ekspresi ini disimpan di variabel result dan dicetak.

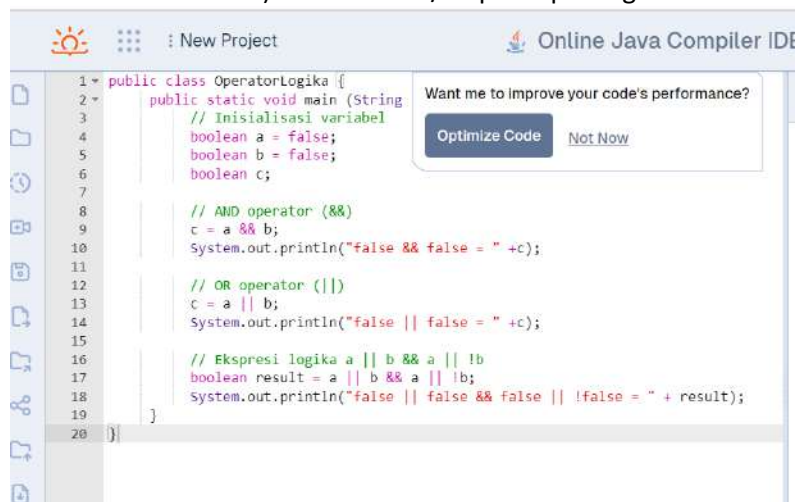
###### b) Uraikan luaran yang dihasilkan

false && false = false

false || false = false

false || false && false || !false = true

###### c) Screenshot/ Capture potongan kode dan hasil luaran



```

1 public class OperatorLogika {
2     public static void main (String
3         // Inisialisasi variabel
4         boolean a = false;
5         boolean b = false;
6         boolean c;
7
8         // AND operator (&&)
9         c = a && b;
10        System.out.println("false && false = " + c);
11
12        // OR operator (||)
13        c = a || b;
14        System.out.println("false || false = " + c);
15
16        // Ekspresi logika a || b && a || !b
17        boolean result = a || b && a || !b;
18        System.out.println("false || false && false || !false = " + result);
19    }
20 }
  
```

Want me to improve your code's performance?  
[Optimize Code](#) [Not Now](#)

Output    Generated Files

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

#### [Nomor 5] Kesimpulan

##### 1) Evaluasi

##### a) Apa konsekuensi dari skenario pemrograman ini?

Memahami prioritas operator seperti `&&` (AND) dan `||` (OR) penting untuk hasil yang benar. Salah dalam pemahaman dapat menyebabkan hasil yang tidak diinginkan. Evaluasi ekspresi logika kompleks memerlukan perhatian terhadap urutan dan prioritas operator. Hasil akhir dari ekspresi dapat mempengaruhi alur program.



### Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Nafiisah Calista Hersa Agustine (G1F024077)	Oporator data	10 september 2024
<b>[Nomor 6] Identifikasi Masalah:</b>		
<p>1) Uraikan permasalahan dan variable</p> <pre>public class OperatorKondisi{     public static void main( String[] args ){         String status = "";         int nilai = 80;         status = (nilai &gt; 60)?"Lulus":"Gagal";         System.out.println( status );     } }</pre> <p>1.1 Analisis hasil dan proses yang terjadi!</p> <pre>public class OperatorKondisi{      public static void main( String[] args ){          String status = "";          int nilai = 60;          status = (nilai &gt; 60)?"Lulus":"Gagal";          System.out.println( status );      } }</pre> <p>Luaran :</p> <p>Gagal</p> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p><a href="https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMylw">https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMylw</a>  <a href="#">Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada</a>  <a href="#">laman https://www.youtube.com/watch?v=PzCMZOboxZM</a>  <a href="#">Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada</a>  <a href="#">laman https://www.youtube.com/watch?v=LcFgl0yrKEw</a></p>		
<b>[Nomor 6] Analisis dan Argumentasi</b>		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Kode ternary yang diberikan untuk menentukan status kelulusan berdasarkan nilai. Namun, kondisi <code>nilai &gt; 60</code> tidak termasuk nilai 60 dalam kategori "Lulus". Oleh karena itu, jika nilai tepat 60, hasilnya adalah "Gagal", padahal bisa jadi Anda ingin menyertakan nilai 60 sebagai nilai yang lulus.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Solusi mengubah kondisi menjadi nilai <code>&gt;= 60</code>, sehingga nilai 60 juga termasuk dalam kategori "Lulus". Ini sesuai dengan kriteria kelulusan yang lebih umum dan sering diterapkan.</p>		
<b>[Nomor Soal] Penyusunan Algoritma dan Kode Program</b>		
<p>1) Rancang desain solusi atau algoritma</p> <p>Algoritmanya :</p> <p>1) Mulai</p>		

- 2) inialisasi variable
  - Buat variable nilai dan set dengan nilai 60
- 3) Evaluasi kondisi menggunakan operator ternary
  - Periksa apakah *nilai* lebih besar dari 60.
  - Jika benar, set *status* ke "Lulus".
  - Jika salah, set *status* ke "Gagal".
- 4) tampilkan output
- 5) selesai
- 2) Tuliskan kode program dan luaran
  - a) Beri komentar pada kode
    - 1) Kode ini mendeklarasikan sebuah kelas java bernama OperatorKondisi.
    - 2) Metode *main* adalah metode utam yang dijalankan ketika program ini dijalankan. *String[] args*, adalah parameter untuk argument baris perintah meskipun tidak digunakan dalam kode ini.
    - 3) Variable nilai dideklarasikan sebagai tipe *int* (integer) = nilai 60
    - 4) Nilai > 60 adalah kondisi yang diperiksa, jika kondisi benar (nilai lebih dari 60) maka status diatur menjadi " Lulus ". Jika kondisi salah (nilai 60 atau kurang) maka status diatur menjadi "Gagal".
    - 5) Tampilan output
  - b) Uraikan luaran yang dihasilkan  
Gagal
  - c) Screenshot/ Capture potongan kode dan hasil luaran

```

1 public class OperatorKondisi{
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 }
}

```

Output    Generated Files

Gagal

## [Nomor 6] Kesimpulan

- 1) Evaluasi
  - a) Apa konsekuensi dari skenario pemrograman ini?
 

Pada program ini saya mengkonversi menggunakan operator ternary untuk menetapkan status kelulusan berdasarkan nilai. Konsekuensi utama adalah bahwa nilai persis 60 tidak dianggap sebagai kelulusan. Jika aturan kelulusan melibatkan nilai 60 sebagai kelulusan, maka kondisi harus diperbaiki. Selain itu, jika logika menjadi lebih kompleks, pertimbangkan untuk menggunakan struktur kontrol alur yang lebih jelas.

### Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Nafiisah Calista Hersa Agustine (G1F024077)	OPERATOR DATA	11 september 2024

#### [Nomor 7] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

```
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 );  
    }  
}
```

1.1 analisis hasil dan proses yang terjadi.

```
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 );

} }

```

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

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

- Perhitungan yang saya pilih adalah:

```

    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 );

```

- Perhitungan biner

&	AND	10 & 12	1010 & 1100	1000	8
	OR	10   12	1010   1100	1110	14
^	XOR	10 ^ 12	1010 ^ 1100	0110	6

kesimpulkan hasilnya :

- a & b memberikan 2, yang merupakan bitwise AND dari 10 dan 7.
- a | b memberikan 15, yang merupakan bitwise OR dari 10 dan 7.
- a ^ b memberikan 13, yang merupakan bitwise XOR dari 10 dan 7.

3) Rincikan sumber informasi yang relevan (buku / webpage)

<https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMylw>

[Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada](#)

[laman https://www.youtube.com/watch?v=PzCMZOboxZM](#)

[Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman https://www.youtube.com/watch?v=LcFgl0yrKEw](#)

#### **[Nomor 7] Analisis dan Argumentasi**

- 1) Uraikan rancangan solusi yang diusulkan.
  - Bitwise AND (&): Operasi AND membandingkan bit-bit dari dua bilangan dan menghasilkan 1 hanya jika kedua bit yang dibandingkan adalah 1.
  - Bitwise OR (|): Operasi OR membandingkan bit-bit dari dua bilangan dan menghasilkan 1 jika salah satu atau kedua bit yang dibandingkan adalah 1.
  - Bitwise XOR (^): Operasi XOR membandingkan bit-bit dari dua bilangan dan menghasilkan 1 jika bit yang dibandingkan berbeda.
- 2) Analisis solusi, kaitkan dengan permasalahan.  
Pemahaman Operasi Bitwise: Solusi ini memberikan pemahaman dasar tentang bagaimana operasi bitwise bekerja dan bagaimana mereka dapat diterapkan dalam pemrograman. Menggunakan contoh bilangan 10 dan 7 memudahkan visualisasi dan pemahaman.

#### **[Nomor 7] Penyusunan Algoritma dan Kode Program**

- 1) Rancang desain solusi atau algoritma  
Algoritmanya:
  - 2) Mulai
  - 3) Input data
  - 4) Operasi Bitwise
    - a. Operasi AND (&)
    - b. Operasi OR (|)
    - c. Operasi XOR (^)
  - 5) Output hasil
  - 6) selesai
- 2) Tuliskan kode program dan luaran
  - a) Beri komentar pada kode  
Input data  
Bilangan bulat a diatur ke 10 dan b diatur ke 7.  
Operasi Bitwise  
operasi AND (&) : a & b untuk mendapatkan bitwise AND.  
operasi OR (|): a | b untuk mendapatkan bitwise OR.  
Operasi XOR(^): a ^ b untuk mendapatkan bitwise XOR.  
Output hasil  
Konversi ke biner:  
a (10) dalam biner adalah 1010.  
  
b (7) dalam biner adalah 111.

Hasil Operasi Bitwise:

AND (&): 1010 & 0111 menghasilkan 0010 dalam biner, yang setara dengan 2 dalam desimal.

OR (|): 1010 | 0111 menghasilkan 1111 dalam biner, yang setara dengan 15 dalam desimal.

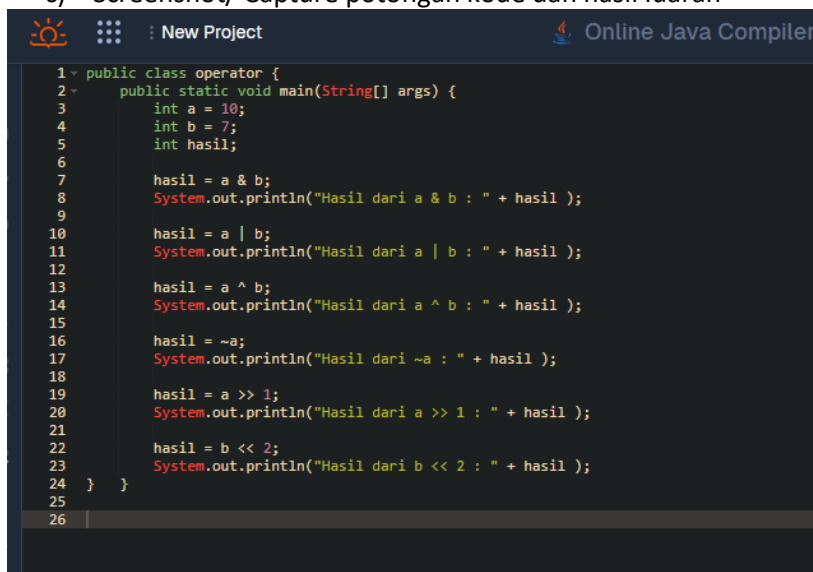
XOR (^): 1010 ^ 0111 menghasilkan 1101 dalam biner, yang setara dengan 13 dalam decimal.

b) Uraikan luaran yang dihasilkan

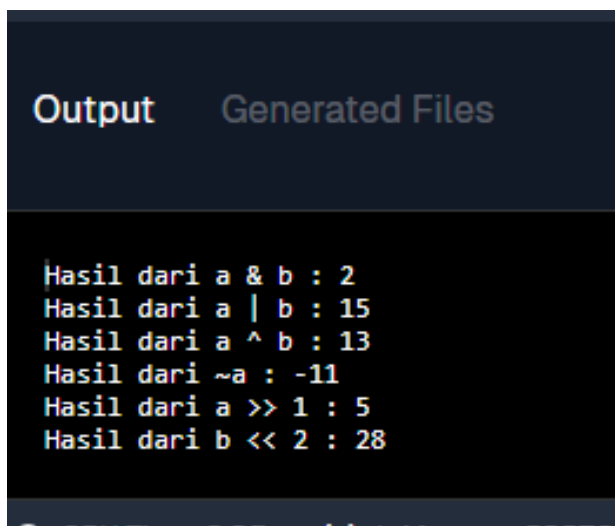
Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data

c) Screenshot/ Capture potongan kode dan hasil luaran



```
1 public class operator {
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    }
25 }
26
```



```
Output    Generated Files

Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
Hasil dari ~a : -11
Hasil dari a >> 1 : 5
Hasil dari b << 2 : 28
```

#### [Nomor 7] Kesimpulan

1) Evaluasi

a) Apa konsekuensi dari skenario pemrograman ini?

Pada program ini saya mengkonversi bahwa menggunakan operasi bitwise dalam pemrograman memiliki beberapa konsekuensi dan pertimbangan , yaitu :

1) Efisiensi: Menyediakan cara yang sangat efisien untuk manipulasi bit.

- |  |
|--|
| <ol style="list-style-type: none"><li>2) Manipulasi Bit: Berguna untuk pengelolaan data bit dan pengaturan flag.</li><li>3) Kompleksitas Kode: Memerlukan pemahaman yang baik agar kode mudah dipelihara dan dipahami.</li><li>4) Keamanan: Penting dalam kriptografi dan algoritma keamanan untuk memastikan operasi dilakukan dengan benar.</li><li>5) Portabilitas: Perlu diperhatikan agar kode tetap portabel di berbagai platform.</li></ol> |
|  |

