

## Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Syifa Ariqah Pajriyanti (G1F024009)	Operator Java	11 September 2024
<b>[nomor 1] Identifikasi Masalah:</b>		
<p>1) Uraikan permasalahan dan variabel</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><b>Latihan 1.</b></p> <p>1.1. Tambahkan baris <code>System.out.println("a + b = " + (a + b));</code> Ubahlah operator ( + ) dengan tanda ( -, *, /, %)</p> <p>1.2. Analisa perhitungan matematika yang terjadi!</p> <ul style="list-style-type: none"> <li>• Pengurangan (a - b):             <ul style="list-style-type: none"> <li>• Perhitungan: <math>20 - 3 = 17</math></li> <li>• Penambahan (<b>a + b</b>):</li> <li>• Perhitungan: <math>20 + 3 = 23</math></li> <li>• <b>Perkalian (a * b):</b></li> <li>• Perhitungan: <math>20 * 3 = 60</math></li> <li>• <b>Pembagian (a / b):</b></li> <li>• Perhitungan: <math>20 / 3</math> Hasil: 6 (pembagian integer membulatkan hasil ke bawah)</li> <li>• <b>Modulus (a % b):</b></li> <li>• Perhitungan: <math>20 \% 3 = \text{Hasil } 2</math> (sisa dari pembagian)</li> </ul> </li> <li>2) Rincikan sumber informasi yang relevan (buku / webpage)             <p><a href="https://www.youtube.com/channel/UC8B9rghd3dBIS6OKonLMylw">https://www.youtube.com/channel/UC8B9rghd3dBIS6OKonLMylw</a>  <b>Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada</b>  <b>laman</b> <a href="https://www.youtube.com/watch?v=PzCMZOboxZM">https://www.youtube.com/watch?v=PzCMZOboxZM</a>  <b>Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal</b>  <b>pada laman</b> <a href="https://www.youtube.com/watch?v=LcFgl0yrKEw">https://www.youtube.com/watch?v=LcFgl0yrKEw</a></p> </li> </ul>		
<b>[No.1] Analisis dan Argumentasi</b>		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Tujuan Program ini bertujuan untuk mendemonstrasikan penggunaan berbagai operator aritmatika dalam Java dan menampilkan hasilnya.</p> <p>Deskripsi program akan mendeklarasikan dua variabel integer (a dan b), kemudian menggunakan operator aritmatika dasar untuk melakukan operasi matematika pada variabel tersebut. Hasil dari setiap operasi akan dicetak ke layar.</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>[No.1] Penyusunan Algoritma dan Kode Program</b>		

- 1) Rancang desain solusi atau algoritma
  - a. Mulai Program
  - b. Deklarasi dan Inisialisasi Variabel Yaitu `int a= 20, int b=3`
  - c. Tampilkan Nilai Variabel
  - d. Operasi Aritmatika dan Tampilkan Hasil
  - e. Akhiri Program
- 2) Tuliskan kode program dan luaran
  - 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

The screenshot shows a Java IDE with the following code in the main editor:

```

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-    }
}

```

On the right side, the 'Input/Output' panel shows the output of the program:

```

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

```

### [Nomor.1] Kesimpulan

- 1) Evaluasi
  - a) Evaluasi input, proses, dan luaran yang dihasilkan!  
 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.

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Syifa Ariqah Pajriyanti (G1F024009)</b>	<b>Operator Java</b>	<b>10 September 2024</b>

### [nomor .2] Identifikasi Masalah:

1. 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);

        // 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
```

#### Latihan 2.

- 2.1. Bandingkan hasil Contoh 1 dengan Contoh 2!

Hasil contoh pada soal nomor 1 terdapat perbedaan dengan contoh soal nomor 2, perbedaan perbedaan luaran dan waktu eksekusi yaitu:

- operator aritmetika biasa hanya menampilkan hasil perhitungan tanpa mengubah nilai variabel secara permanen
- operator penugasan selain menampilkan hasil juga mengubah nilai variabel secara permanen.
- Pada soal variabel yang digunakan sudah jelas jelas. Kita disuruh untuk membuat operator penugasan dengan penjumlahan, pengurangan, perkalian, pembagian, dan sisa bagi. Kode tersebut menggunakan operator penugasan dengan benar, dan output yang dihasilkan sesuai dengan operasi yang dilakukan. Jika melihat output "Penambahan : 23" di layar itu adalah hasil yang benar berdasarkan operasi penambahan yang dilakukan oleh b+=a.
- Diketahui dari soal: variabel yang digunakan ada 2, yaitu int a = 20; int b = 3;. Variabel a diinisialisasi dengan nilai 20 dan variabel b diinisialisasi dengan nilai 3. Intinya terdapat perbedaan antara hasil pengurangan, pembagian, dan sisa bagi pada soal nomor 1 dan 2

- 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 2. 2. laman https://www.youtube.com/watch?v=PzCMZObexZM](https://www.youtube.com/watch?v=PzCMZObexZM)  
[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.  
Tujuan Program: Program ini bertujuan untuk mendemonstrasikan penggunaan operator penugasan dalam Java. Operator penugasan memungkinkan modifikasi nilai variabel dengan melakukan operasi aritmatika secara langsung pada nilai variabel tersebut.  
Deskripsi Program: ini akan mendeklarasikan dua variabel integer (a dan b) dan menggunakan operator penugasan untuk melakukan operasi penambahan, pengurangan, perkalian, pembagian, dan modulus pada variabel b. Hasil dari setiap operasi akan ditampilkan ke layar.
2. Analisis solusi, kaitkan dengan permasalahan.  
Dari segi efektivitas Solusi  
Permasalahan: Memahami bagaimana operator penugasan bekerja dan mempengaruhi nilai variabel dalam pemrograman.  
Solusi: Program ini secara efektif menunjukkan bagaimana operator penugasan memodifikasi nilai variabel melalui operasi aritmatika. Setiap operator (+, -, \*, /, %) mempengaruhi variabel b secara langsung, dan hasil dari setiap operasi ditampilkan.

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

- 3) Rancang desain solusi atau algoritma
  - a. Mulai
  - b. Deklarasi dan Inisialisasi Variabel a: 20, b:3
  - c. Operasi Aritmatika dengan Operator Penugasan
    - Operator Penugasan Penambahan
    - Operator Penugasan Pengurangan
    - Operator Penugasan Perkalian
    - Operator Penugasan Pembagian
    - Operator Penugasan sisabagi
  - d. Selesai
- 4) Tuliskan kode program dan luaran
  - d) 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, sisabagi
  - e) Uraikan luaran yang dihasilkan
    - Penambahan : 23
    - Pengurangan : 3
    - Perkalian : 60
    - Pembagian : 3
    - Sisa Bagi: 3
  - f) 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:

```

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

```

### [Nomor 2] Kesimpulan

#### 3. Evaluasi

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

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 .

- Kode ini mengilustrasikan perubahan bertahap dalam nilai variabel b dengan menggunakan berbagai operator penugasan.
- Hasil dari operasi terakhir seharusnya mencerminkan sisa pembagian yang benar, tetapi komentar akhir menunjukkan ketidakakuratan yang dapat membingungkan.
- Penting untuk memastikan bahwa komentar kode selalu diperbarui dan akurat agar kode mudah dipahami dan dipelihara.

### Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Syifa Ariqah Pajriyanti (G1F024009)	Operator Data	11 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.

#### [Nomor 3] Analisis dan Argumentasi

2. 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.

3. 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

5) Rancang desain solusi atau algoritma

- Mulai
- Deklarasi Variabel
- Input Nilai

- Perbandingan
- Output Hasil
- Selesai

6) Tuliskan kode program dan luaran

g) 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.

h) 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

i) Screenshot/ Capture potongan kode dan hasil luaran

```

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 + " dan 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 atau sama dengan B?
17        hasil = nilaiA >= nilaiB;
18        System.out.println("Hasil A >= B = " + hasil);
19
20        // apakah A lebih kecil atau sama dengan 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 sama dengan B?
29        hasil = nilaiA != nilaiB;
30        System.out.println("Hasil A != B = " + hasil);
31    }
32 }
  
```

Output:

```

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

4. Evaluasi

b) 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

Nama & NPM	Topik:	Tanggal:
Syifa Ariqah Pajriyanti (G1F024009)	Operator Data	11 September 2024

### [Nomor 4] Identifikasi Masalah:

4) 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):



<ul style="list-style-type: none"> <li>• Cetakan Pertama: Nilai variabel setelah increment (baru).</li> <li>• Cetakan Kedua: Nilai variabel yang sama (karena sudah diincrement).</li> </ul> <p>3. Post Decrement (c--):</p> <ul style="list-style-type: none"> <li>• Cetakan Pertama: Nilai variabel sebelum decrement (lama).</li> <li>• Cetakan Kedua: Nilai variabel setelah decrement (baru).</li> </ul> <p>4. Pre Decrement (--d):</p> <ul style="list-style-type: none"> <li>• Cetakan Pertama: Nilai variabel setelah decrement (baru).</li> <li>• Cetakan Kedua: Nilai variabel yang sama (karena sudah didecrement).</li> </ul> <p>Kesimpulan:</p> <ul style="list-style-type: none"> <li>• Post Increment/Decrement: Mencetak nilai sebelum perubahan (lama) lalu setelah perubahan (baru).</li> <li>• Pre Increment/Decrement: Mencetak nilai setelah perubahan (baru).</li> </ul>
<p><b>[Nomor 4] Analisis dan Argumentasi</b></p> <p>5. Uraikan rancangan solusi yang diusulkan.</p> <p>Rancangan Solusi:</p> <ul style="list-style-type: none"> <li>• Memahami perbedaan antara operator Post dan Pre dalam konteks Increment (++) dan Decrement (--) dalam bahasa pemrograman Java.</li> <li>• Operator Post Increment/Decrement (a++, c--) dan Pre Increment/Decrement (++b, --d) memiliki cara kerja yang berbeda dalam hal waktu perubahan nilai.</li> <li>• Bandingkan hasil yang diperoleh untuk Post dan Pre Increment/Decrement.</li> <li>• Pastikan hasil sesuai dengan ekspektasi bahwa Post mencetak nilai sebelum perubahan dan Pre mencetak nilai setelah perubahan.</li> </ul> <p>6. Analisis solusi, kaitkan dengan permasalahan.</p> <p>Solusi:</p> <ul style="list-style-type: none"> <li>• Post Increment/Decrement: Cetak nilai lama, lalu ubah variabel.</li> <li>• Pre Increment/Decrement: Ubah variabel dulu, lalu cetak nilai baru.</li> </ul> <p>Kaitan dengan Permasalahan:</p> <ul style="list-style-type: none"> <li>• Memahami Perbedaan: Program ini menunjukkan perbedaan jelas antara Post dan Pre, membantu memahami kapan nilai lama atau baru digunakan.</li> <li>• Pemilihan Operator: Memudahkan pemilihan operator yang tepat sesuai kebutuhan—nilai lama (Post) atau nilai baru (Pre).</li> </ul>
<p><b>[Nomor 4] Penyusunan Algoritma dan Kode Program</b></p> <p>7) Rancang desain solusi atau algoritma</p> <ol style="list-style-type: none"> <li>1. Inisialisasi Variabel</li> <li>2. Post Increment</li> <li>3. Pre Increment</li> <li>4. Post Decrement</li> <li>5. Pre Decrement</li> <li>6. Akhiri</li> </ol> <p>8) Tuliskan kode program dan luaran</p> <p>j) Beri komentar pada kode</p> <p>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.</p> <p>k) Uraikan luaran yang dihasilkan</p> <pre># Post Increment # ===== Isi variabel a: 10 Isi variabel a: 10 Isi variabel a: 11</pre>

#### # 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

I) Screenshot/ Capture potongan kode 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 defines a class 'operator' with a 'main' method that tests various increment and decrement operators on variables a, b, c, and d. The output panel shows the results of these operations, confirming the behavior of post-increment, pre-increment, post-decrement, and pre-decrement.

```
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
10        System.out.println();
11
12        int b = 10;
13        System.out.println("# Pre Increment #");
14        System.out.println("=====");
15        System.out.println("Isi variabel b: " + b);
16        System.out.println("Isi variabel b: " + ++b);
17        System.out.println("Isi variabel b: " + b);
18
19        System.out.println();
20
21        int c = 10;
22        System.out.println("# Post Decrement #");
23        System.out.println("=====");
24        System.out.println("Isi variabel c: " + c);
25        System.out.println("Isi variabel c: " + c--);
26        System.out.println("Isi variabel c: " + c);
27
28        System.out.println();
29
30        int d = 10;
31        System.out.println("# Pre Decrement #");
32        System.out.println("=====");
33        System.out.println("Isi variabel d: " + d);
34        System.out.println("Isi variabel d: " + --d);
35        System.out.println("Isi variabel d: " + d);
36    }
37 }
```

Output:

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

# Pre Increment #
=====
Isi variabel b: 10
Isi variabel b: 11
Isi variabel b: 11

# Post Decrement #
=====
Isi variabel c: 10
Isi variabel c: 10
Isi variabel c: 9

# Pre Decrement #
=====
Isi variabel d: 10
Isi variabel d: 9
Isi variabel d: 9
```

CPU Time: 0.05 secs | Memory: 2.122 MB  
Compiled and executed in 2.122 s

#### [Nomor 4] Kesimpulan

##### 7. Evaluasi

c) 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:
Syifa Ariqah Pajriyanti (G1F024009)	Operator Data	11 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;
    }
}

```

<pre> boolean c;  // AND operator (&amp;&amp;) c = a &amp;&amp; b; System.out.println("false &amp;&amp; false = " + c);  // OR operator (  ) c = a    b; System.out.println("false    false = " + c);  // Ekspresi logika a    b &amp;&amp; a    !b boolean result = a    b &amp;&amp; a    !b; System.out.println("false    false &amp;&amp; false    !false = " + result); } </pre> <p>Luaran:  false &amp;&amp; false = false  false    false = false  false    false &amp;&amp; false    !false = true</p> <p>uraikan urutan logika dan analisis tried an false :  operator AND (&amp;&amp;) memiliki prioritas lebih tinggi dibandingka dengan OR (  ).</p> <ul style="list-style-type: none"> <li>• b &amp;&amp; a menjadi false &amp;&amp; false hasilnya false.</li> <li>• a    (false) menjadi false    false, hasilnya false.</li> <li>• !b menjadi !false, hasilnya true.</li> <li>• Akhirnya, false    true, hasilnya adalah true.</li> </ul>
<p><b>[Nomor 5] Analisis dan Argumentasi</b></p> <ol style="list-style-type: none"> <li>Uraikan rancangan solusi yang diusulkan. <ul style="list-style-type: none"> <li>Program ini mendemonstrasikan penggunaan operator logika dasar (&amp;&amp;,   , dan !) dalam Java.</li> <li>Penilaian prioritas operator logika penting untuk memastikan hasil evaluasi yang benar.</li> <li>Output yang dihasilkan mencerminkan evaluasi dari ekspresi logika yang ditentukan dalam kode.</li> </ul> </li> <li>Analisis solusi, kaitkan dengan permasalahan. <ul style="list-style-type: none"> <li><b>Analisis Prioritas Operator:</b> Dalam ekspresi logika, operator &amp;&amp; memiliki prioritas lebih tinggi daripada   . Oleh karena itu, b &amp;&amp; a dihitung terlebih dahulu sebelum a    (result)    !b dievaluasi.</li> <li><b>Permasalahan:</b> 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.</li> </ul> </li> </ol>
<p><b>[Nomor 5] Penyusunan Algoritma dan Kode Program</b></p> <ol style="list-style-type: none"> <li>Rancang desain solusi atau algoritma <ul style="list-style-type: none"> <li>-inisialisasi variable</li> <li>-evaluasi operator AND (&amp;&amp;)</li> <li>-evaluasi operator OR (  )</li> <li>-evaluasi ekspresi logika kompleks</li> </ul> </li> <li>Tuliskan kode program dan luaran</li> </ol>

m) 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.

n) Uraikan luaran yang dihasilkan

`false && false = false`

`false || false = false`

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

o) Screenshot/ Capture potongan kode dan hasil luaran



The screenshot shows an IDE window titled 'New Project' and 'Online Java Compiler IDE'. The code editor contains the following Java code:

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

The right sidebar shows the 'Input/Output' panel with 'Language Version: JDK 21.0.0'. The 'Output' tab displays the following results:

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

## [Nomor 5] Kesimpulan

### 10. Evaluasi

d) 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.

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Syifa Ariqah Pajriyanti (G1F024009)</b>	<b>Oporator data</b>	<b>10 september 2024</b>
<b>[Nomor 6] Identifikasi Masalah:</b>		
<p>5) 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>		
<b>[Nomor 6] Analisis dan Argumentasi</b>		
<p>11. Uraikan rancangan solusi yang diusulkan. 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>12. Analisis solusi, kaitkan dengan permasalahan. 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>11) Rancang desain solusi atau algoritma</p> <p>Algoritmanya :</p> <ol style="list-style-type: none"> <li>1) Mulai</li> <li>2) inisialisasi variable <ul style="list-style-type: none"> <li>- Buat variable nilai dan set dengan nilai 60</li> </ul> </li> <li>3) Evaluasi kondisi menggunakan operator ternary <ul style="list-style-type: none"> <li>-Periksa apakah <i>nilai</i> lebih besar dari 60.</li> <li>-Jika benar, set <i>status</i> ke "Lulus".</li> <li>- Jika salah, set <i>status</i> ke "Gagal".</li> </ul> </li> <li>4) tampilkan output</li> <li>5) selesai</li> </ol>		

12) Tuliskan kode program dan 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: Gagal

- p) Screenshot/ Capture potongan kode dan hasil luaran
- q) 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
- r) Analisis luaran yang dihasilkan Luaran sudah sesuai dengan program yang disusun Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[Nomor 6] Kesimpulan**

2. Evaluasi
- e) Apa konsekuensi dari skenario pemrograman ini?
- Pada program ini saya mengkonversi menggunakan operator ternary unrtuk 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:
Syifa Ariqah Pajriyanti (G1F024009)	OPERATOR DATA	11 september 2024

**[Nomor 7] Identifikasi Masalah:**

- 6) 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 );
}

```

7.1 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  $a \wedge b$  : 13

Hasil dari  $\sim a$  : -11

Hasil dari  $a \gg 1$  : 5

Hasil dari  $b \ll 2$  : 28

7.2 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 =  $a \wedge b$ ;

System.out.println("Hasil dari  $a \wedge b$  : " + hasil);

- Perhitungan biner

$\&$	AND	$10 \& 12$	$1010 \& 1100$	1000	8
$ $	OR	$10   12$	$1010   1100$	1110	14
$\wedge$	XOR	$10 \wedge 12$	$1010 \wedge 1100$	0110	6

kesimpulan 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.
- $a \wedge b$  memberikan 6, yang merupakan bitwise XOR dari 10 dan 7.

#### [Nomor 7] Analisis dan Argumentasi

13. 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 ( $\wedge$ ): Operasi XOR membandingkan bit-bit dari dua bilangan dan menghasilkan 1 jika bit yang dibandingkan berbeda.

14. 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

13) Rancang desain solusi atau algoritma

Algoritmanya:

- 1) Mulai
- 2) Input data
- 3) Operasi Bitwise
  - o Operasi AND (&)
  - o Operasi OR (|)
  - o Operasi XOR (^)
- 4) Output hasil
- 5) selesai

14) Tuliskan kode program dan 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
```

Input/Output

Language Version: J

Input Arguments

Stdin inputs

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
```

s) Screenshot/ Capture potongan kode dan hasil luaran

t) Beri komentar pada kode

1. Input data

Bilangan bulat a diatur ke 10 dan b diatur ke 7.

2. Operasi Bitwise

- operasi AND (&) :  $a \& b$  untuk mendapatkan bitwise AND.
- operasi OR (|):  $a | b$  untuk mendapatkan bitwise OR.
- Operasi XOR(^):  $a \wedge b$  untuk mendapatkan bitwise XOR.

3. 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 (^):  $1010 \wedge 0111$  menghasilkan 1101 dalam biner, yang setara dengan 13 dalam decimal.

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

**[Nomor 7] Kesimpulan**

15. Evaluasi

- f) 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.
  - 2) Manipulasi Bit: Berguna untuk pengelolaan data bit dan pengaturan flag.
  - 3) Kompleksitas Kode: Memerlukan pemahaman yang baik agar kode mudah dipelihara dan dipahami.
  - 4) Keamanan: Penting dalam kriptografi dan algoritma keamanan untuk memastikan operasi dilakukan dengan benar.
  - 5) Portabilitas: Perlu diperhatikan agar kode tetap portabel di berbagai platform.
- g) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)