

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
[Nomor Soal] Identifikasi Masalah:		
<ol style="list-style-type: none">1) Uraikan permasalahan dan variabel2) Rincikan sumber informasi yang relevan (buku / webpage)3) Uraikan rancangan solusi yang diusulkan (jika ada).4) Analisis susunan solusi, parameter solusi (jika ada).		
[Nomor Soal] Analisis dan Argumentasi		
<ol style="list-style-type: none">1) Uraikan rancangan solusi yang diusulkan.2) Analisis solusi, kaitkan dengan permasalahan.		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<ol style="list-style-type: none">1) Rancang desain solusi atau algoritma2) Tuliskan kode program dan luaran<ol style="list-style-type: none">a) Beri komentar pada kodeb) Uraikan luaran yang dihasilkanc) Screenshot/ Capture potongan kode dan hasil luaran		
[Nomor Soal] Kesimpulan		
<ol style="list-style-type: none">1) Analisa<ol style="list-style-type: none">a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?2) Evaluasi<ol style="list-style-type: none">a) Apa konsekuensi dari skenario pemrograman ini?b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)3) Kreasi<ol style="list-style-type: none">a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)		

Contoh Jawaban:

Nama & NPM	Topik:	Tanggal:
Vivilia Regita Pramesti G1F024060	Operator Java	13/09/2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

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

        //operator aritmatika
        System.out.println("a: " +a);
        System.out.println("b: " +b);
        System.out.println("a + b = " (a + b)); //menampilkan hasil penjumlahan
    } }
```

Luaran:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
Syntax error on token ""a + b = "" , AssignmentOperator expected after this token
The left-hand side of an assignment must be a variable
```

Latihan 1

- 1.1. Rekomendasikan perbaikan kode agar program Contoh 1 dapat berjalan!
- 1.2. Tambahkan baris untuk menampilkan perhitungan dengan operator (-, *, / , %) pada Contoh 1!

Jawab :

- 1.1 kode dapat diperbaiki dengan cara menambahkan operator + diantara string dan
System.out.println
- 1.2 Saya sudah menambahkan operator (-,*,/,%) pada kode program yang sudah saya buat.

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan kode program + diantara string dan System.out.println
- 2) Alasan solusi ini karena dapat membantu memperbaiki kode program dan tidak error
- 3) Perbaikan kode program dengan cara menambahkan kode program + diantara string dan System.out.println

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - (a) Mulai
 - (b) Deklarasikan dua variabel a dan b dengan tipe data int dan berikan nilai awal.Masukkan mie
 - (c) Perhitungan Aritmatika: Hitung hasil penjumlahan a + b, hasil pengurangan a - b, hasil perkalian a * b, hasil pembagian a / b, dan hasil sisa bagi a % b.
 - (d) Tampilkan hasil perhitungan selain penjelasan.
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran
Beri komentar pada kode yang di Screenshot

The screenshot displays the Eclipse IDE interface. The top editor shows a Java file named `main.java` with the following code:

```
1 package Kelasku;
2
3 public class main {
4
5     public static void main(String[] args) {
6         // Deklarasi nilai
7         int a = 20, b = 3;
8
9         // Menampilkan hasil operasi aritmatika
10        System.out.println("a: " + a);
11        System.out.println("b: " + b);
12        System.out.println("a + b = " + (a + b)); // Menampilkan hasil penjumlahan
13        System.out.println("a - b = " + (a - b)); // Menampilkan hasil pengurangan
14        System.out.println("a * b = " + (a * b)); // Menampilkan hasil perkalian
15        System.out.println("a / b = " + (a / b)); // Menampilkan hasil pembagian
16        System.out.println("a % b = " + (a % b)); // Menampilkan hasil modulus
17    }
18 }
19
```

The bottom console window shows the output of the program:

```
<terminated> main (3) [Java Application] C:\Users\ASUS\p2\pool\plugins\org.eclipse.justj.openjdk.
a: 20
b: 3
a + b = 23
a - b = 17
a * b = 60
a / b = 6
a % b = 2
```

b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.1] Kesimpulan

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab :

- Dengan menambahkan operator + untuk menggabungkan string dengan hasil perhitungan, kode program diperbarui dan menampilkan hasil dari operasi aritmatika yang diminta: penjumlahan, pengurangan, perkalian, pembagian, dan modulo.
- Perbaikan ini diperlukan untuk menampilkan hasil perhitungan yang sebenarnya, bukan hanya string literal. Ini memastikan bahwa output program sesuai dengan yang diharapkan dan memberikan informasi yang akurat tentang hasil perhitungan.

[No. 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; //melakukan perhitungan penjumlahan  
        System.out.println("Penambahan : " + b); // menampilkan hasil  
        perhitungan penjumlahan  
    }  
}
```

Luaran:

Penambahan : 23

Latihan 2.

- 2.1. Tambahkan baris Contoh 2 untuk menampilkan perhitungan dengan operator (-=, *=, /=, %=)!
- 2.2. Berikan argumentasi tentang perbedaan luaran dan waktu eksekusi Contoh 1 dan Contoh 2!

Jawab :

2.1.

```
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("Modulus : " + b);
```

2.2 Perbedaan Luaran:

Contoh 1: Kode ini menunjukkan hasil operasi aritmatika dasar (+, -, *, /, %) antara dua variabel.

Output menampilkan hasil dari masing-masing operasi dengan penjelasan yang sesuai.

Contoh 2: Kode ini menggunakan operator penugasan (+=, -=, *=, /=, %=) untuk melakukan operasi aritmatika dan menyimpan hasilnya langsung ke dalam variabel yang sama (b). Setelah mengubah nilai b dengan operator penugasan, output akan mencetak hasil dari setiap operasi aritmatika.

Diferensiasi Waktu Eksekusi:

Contoh 1: Kode ini melakukan operasi aritmatika secara independen. Hasil dari setiap operasi dicetak tanpa mengubah nilai awal variabel a dan b, misalnya, a tetap 10 dan b tetap 5. Karena variabel tidak diubah antara operasi, waktu eksekusi biasanya lebih konsisten.

Contoh 2: Operator penugasan yang digunakan dalam kode ini mengubah nilai variabel b secara bertahap dengan setiap operasi. Nilai ini dipengaruhi oleh setiap operasi.

[No.2] Analisis dan Argumentasi

- 2) Saya mengusulkan permasalahan ini dapat diatasi dengan cara memberikan penjelasan yang jelas tentang bagaimana operator penugasan bekerja, dengan contoh yang menunjukkan bagaimana nilai variabel berubah setelah setiap operasi.
- 3) Alasan solusi ini karena penjelasan akan membantu pemula memahami konsekuensi dari menggunakan operator penugasan. Dengan melihat bagaimana nilai variabel berubah secara langsung, mereka akan lebih mudah memahami efek dari setiap operator.

- 4) Perbaiki kode program dengan cara Menambahkan komentar yang menjelaskan setiap langkah kode, terutama bagaimana nilai variabel berubah setelah setiap operasi.

[No.2] Penyusunan Algoritma dan Kode Program

3) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

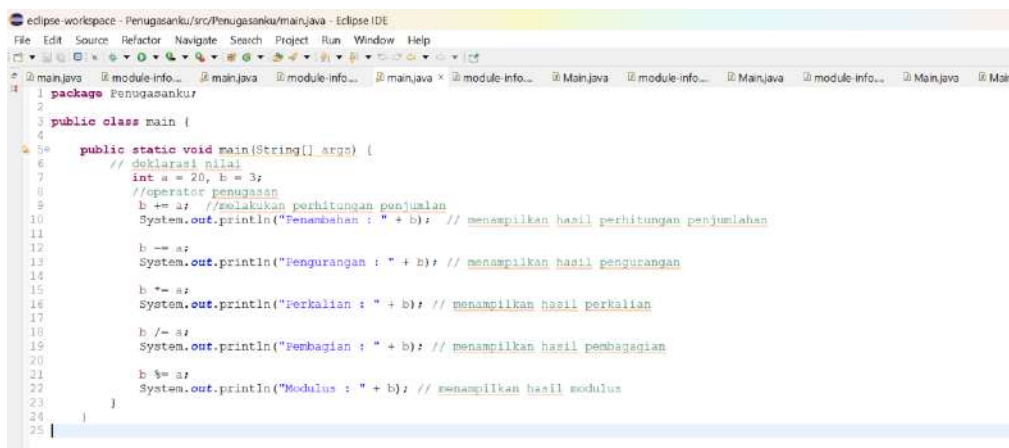
- a) Deklarasikan dan inisialisasi dua variabel, a dan b, dengan nilai awal tertentu.
- b) Hitung hasil dari operasi aritmatika dasar (+, -, *, /, %) antara variabel a dan b.
- c) Gunakan operator penugasan (+=, -=, *=, /=, %=) untuk melakukan operasi aritmatika dan menyimpan hasilnya dalam variabel yang sama.
- d) Tampilkan hasil setiap operasi setelah perubahan nilai variabel.

2) Kode program dan luaran

3)

- a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



```
1 package Penugasankur;
2
3 public class main {
4
5     public static void main(String[] args) {
6         // deklarasi nilai
7         int a = 20, b = 3;
8         //operator penugasan
9         b += a; //melakukan perhitungan penjumlahan
10        System.out.println("Penambahan : " + b); // menampilkan hasil perhitungan penjumlahan
11
12        b -= a;
13        System.out.println("Pengurangan : " + b); // menampilkan hasil pengurangan
14
15        b *= a;
16        System.out.println("Perkalian : " + b); // menampilkan hasil perkalian
17
18        b /= a;
19        System.out.println("Pembagian : " + b); // menampilkan hasil pembagian
20
21        b %= a;
22        System.out.println("Modulus : " + b); // menampilkan hasil modulus
23    }
24 }
25
```



```
<terminated> Main [6] [Java Application] C:\Users\ASUS\AppData\Local\Temp\org.eclipse.jdt.launcher\openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1518\jre\bin\javaw.exe (17 Sep 2024 12.24.53 - 12.24.54) [pid: 4436]
Pengurangan : 3
Perkalian : 60
Pembagian : 3
Modulus : 3
```

- b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.2] Kesimpulan

Analisa

- c) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Permasalahan: Problemnnya adalah memahami dan menerapkan penggunaan operator penugasan dalam operasi aritmatika pada variabel. Dalam kasus ini, operator penugasan seperti +=, -=, *=, /=, dan %= digunakan untuk secara langsung mengubah nilai variabel. Ini berbeda dengan

Algoritma: Hitung hasil dari operasi aritmatika dasar (+, -, *, /, %) antara variabel a dan b. Gunakan operator penugasan (+=, -=, *=, /=, %=) untuk melakukan operasi aritmatika dan menyimpan hasilnya dalam variabel yang sama.

- c) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab :

Keputusan untuk menggunakan operator penugasan ($+=$, $-=$, $*=$, $/=$, $\%=$) atau operasi aritmatika dasar dalam situasi ini didasarkan pada dua pertimbangan utama:

Kejelasan Kode: Operator penugasan memungkinkan penulisan kode yang lebih sederhana dan efektif, sehingga tidak ada variabel tambahan yang diperlukan. Namun, ini memerlukan pemahaman yang baik tentang bagaimana setiap operasi mempengaruhi nilai variabel karena setiap perubahan nilai dapat mempengaruhi operasi berikutnya.

Keterbacaan dan Konsistensi: Operasi aritmatika dasar memberikan hasil yang konsisten tanpa mengubah nilai variabel yang digunakan. Ini membuat hasil lebih prediktif dan lebih mudah untuk di-debug.

[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

3.1. Ubahlah nilai A = 4 dan B = 4 pada Contoh 3. Simpulkan perubahan yang terjadi!

Jawab:

```
int nilaiA = 4;  
int nilaiB = 4;  
boolean hasil;
```

```
System.out.println(" A = " + nilaiA + "\n B = " + nilaiB);  
hasil = nilaiA > nilaiB; // apakah A lebih besar dari B?  
System.out.println("\n Hasil A > B = "+ hasil);
```

```
hasil = nilaiA < nilaiB; // apakah A lebih kecil dari B?  
System.out.println("\n Hasil A < B = "+ hasil);
```

```
hasil = nilaiA >= nilaiB; // apakah A lebih besar samadengan B?
```

```
System.out.println("\n Hasil A >= B = "+ hasil);
```

```
hasil = nilaiA <= nilaiB; // apakah A lebih kecil samadengan B?
```

```
System.out.println("\n Hasil A <= B = "+ hasil);
```

```
hasil = nilaiA == nilaiB; // apakah nilai A sama dengan B?
```

```
System.out.println("\n Hasil A == B = "+ hasil);
```

```
hasil = nilaiA != nilaiB; // apakah nilai A tidak samadengan B?
```

```
System.out.println("\n Hasil A != B = "+ hasil);
```

Jika nilai A dan B berubah 4 atau menjadi sama, hasil perbandingan yang mengukur kesamaan menjadi benar, sedangkan hasil perbandingan yang mengukur ketidaksamaan menjadi salah. Ini menunjukkan bahwa operator relasional efektif dalam membandingkan nilai numerik dan memberikan hasil yang konsisten berdasarkan hubungan antara nilai-nilai tersebut.

[No.3] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menunjukkan cara setiap operator relasional bekerja dan bagaimana hasil perbandingan dipengaruhi oleh nilai-nilai variabel yang sama. Di antaranya adalah penjelasan tentang bagaimana hasil perbandingan untuk kesamaan (==) dan ketidaksamaan (!=) berbeda, serta operator untuk hubungan lebih besar (>, >=) dan lebih kecil (<, <=).
- 2) Alasan solusi ini karena penjelasan yang jelas dan mendetail akan membantu pemahaman tentang bagaimana operator relasional bekerja dalam berbagai kondisi.

[No.3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
Misalkan algoritma memasak mi instan:
 - a) deklarasikan dua variabel integer nilaiA dan nilaiB lalu inisialisasikan dengan nilai 4
 - b) perbandingan
 - hasil = (nilaiA > nilaiB) → Print hasil
 - hasil = (nilaiA < nilaiB) → Print hasil
 - hasil = (nilaiA <= nilaiB) → Print hasil
 - hasil = (nilaiA >= nilaiB) → Print hasil
 - hasil = (nilaiA == nilaiB) → Print hasil
 - hasil = (nilaiA != nilaiB) → Print hasil
 - c) selesai
- 4) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran
Beri komentar pada kode yang di Screenshot


```

eclipse-workspace - Tugaslafi/src/Tugaslafi/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
package Tugaslafi;

public class Main {

    public static void main(String[] args) {
        int nilaiA = 4; // nilai A diubah menjadi 4
        int nilaiB = 4; // nilai B diubah menjadi 4
        boolean hasil;

        System.out.println("A = " + nilaiA + " B = " + nilaiB);
        hasil = nilaiA > nilaiB; // apakah A lebih besar dari B?
        System.out.println("Hasil A > B = " + hasil);

        hasil = nilaiA < nilaiB; // apakah A lebih kecil dari B?
        System.out.println("Hasil A < B = " + hasil);

        hasil = nilaiA >= nilaiB; // apakah A lebih besar atau sama dengan B?
        System.out.println("Hasil A >= B = " + hasil);

        hasil = nilaiA <= nilaiB; // apakah A lebih kecil atau sama dengan B?
        System.out.println("Hasil A <= B = " + hasil);

        hasil = nilaiA == nilaiB; // apakah nilai A sama dengan B?
        System.out.println("Hasil A == B = " + hasil);

        hasil = nilaiA != nilaiB; // apakah nilai A tidak sama dengan B?
        System.out.println("Hasil A != B = " + hasil);
    }
}

```

```

eclipse-workspace - Tugaslafi/src/Tugaslafi/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
<terminated> main [Java Application] C:\Users\ASUS\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre1
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) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.3] Kesimpulan

5) Analisa

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

Permasalahan: Permasalahan utama adalah memahami bagaimana hasil dari operator relasional berubah ketika nilai variabel A dan B diubah menjadi sama. Kode ini menguji perbandingan antara dua variabel integer menggunakan berbagai operator relasional.

Algoritma :

Mendeklarasikan dan menginisialisasi variabel.

Melakukan perbandingan antara variabel menggunakan operator relasional.

Menampilkan hasil dari setiap perbandingan.

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

Untuk memastikan bahwa operator relasional memberikan hasil yang konsisten dan sesuai dengan ekspektasi dasar perbandingan, diputuskan untuk menguji variabel dengan nilai sama. Dengan menetapkan nilai A dan B yang sama, pengujian ini menunjukkan bagaimana operator relasional menangani kesamaan dan ketidaksetaraan, memberikan gambaran tentang bagaimana operator ini bekerja dalam situasi nyata.

[No. 4] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

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

Luaran:

a: 5
b: 5

Latihan 4.

- 4.1. Berikan saran operasi apa yang diperlukan (pre/post increment, pre/post decrement) agar Contoh 4 menghasilkan nilai a = 5 dan b = 6?
- 4.2. Simpulkan hasil eksperimen Anda!

Jawab :

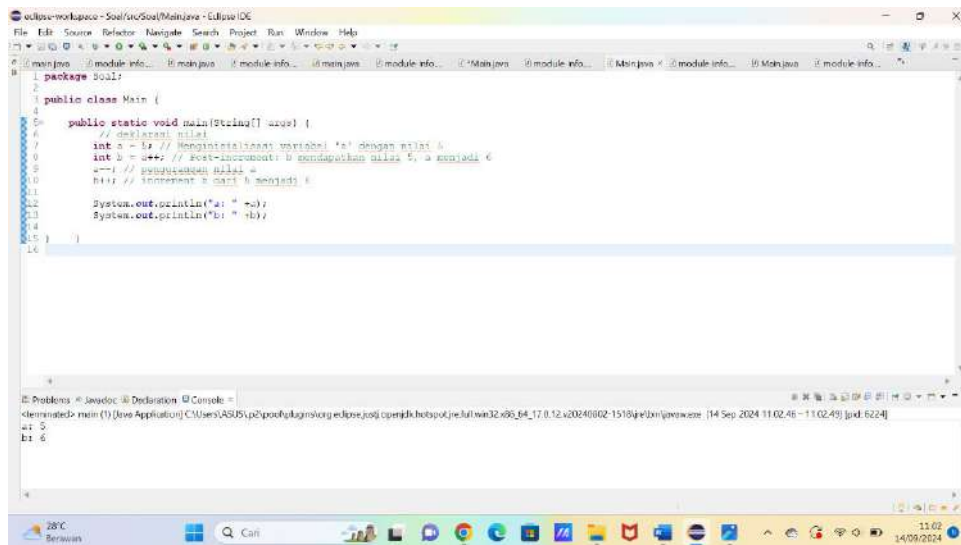
- 4.1. saya menggunakan Pre-Increment Operator (++a) pada variabel b sehingga a menjadi 6. Tambahkan a--; sehingga Nilai a dikurangi 1, mengembalikannya ke nilai 5. Tambahkan b++; sehingga Nilai b diincrement dan menjadi 6.
- 4.2. Eksperimen menunjukkan bahwa dengan menggunakan operator post-increment (a++) untuk menginisialisasi variabel b dan kemudian melakukan decrement (a--) dan increment (b++), kita dapat mencapai hasil yang diinginkan di mana nilai a tetap 5 dan b menjadi 6. Kode yang tepat memastikan bahwa a diatur kembali ke nilai awal setelah increment, sementara b diincrement untuk mencapai nilai yang sesuai. Ini membuktikan bahwa penggunaan operator increment dan decrement secara efektif dapat mengatur nilai variabel dengan presisi sesuai kebutuhan.

[No.4] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan kombinasi operator post-increment untuk b, decrement untuk a, dan increment untuk b, ini dapat memastikan hasil akhir sesuai dengan yang diinginkan. Ini menyederhanakan manipulasi nilai variabel secara efektif dan menghindari kesalahan logika dalam operasi aritmatika.
- 2) Alasan solusi ini karena dapat memastikan hasil sesuai yang diinginkan

[No.4] Penyusunan Algoritma dan Kode Program

- 4) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
Misalkan algoritma memasak mi instan:
 - a) Deklarasikan variabel a dan inisialisasi dengan nilai 5.
 - b) Post-Increment untuk Variabel b
 - c) Decrement Variabel a
 - d) Increment Variabel b
 - e) Tampilkan output
2. Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran
Berikan komentar pada kode yang di Screenshot



```
package Soal1;

public class Main {

    public static void main(String[] args) {
        // deklarasi nilai
        int a = 5; // Menginisialisasi variabel 'a' dengan nilai 5
        int b = a++; // Post-Increment: b mendapatkan nilai 5, a menjadi 6
        --a; // Mengurangi nilai a
        b++; // Increment: a dari 5 menjadi 6
        System.out.println("a: " + a);
        System.out.println("b: " + b);
    }
}
```

Problems | Javadoc | Declaration | Console |
<terminated> main() (Jave Application) C:\Users\ASUS\p2\poo\p2\src\main\java\Soal1\Main.java [14 Sep 2024 11:02:46 - 11:02:49] [jdk: 6224]
a: 5
b: 6

b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.4] Kesimpulan

3. Analisa

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

Permasalahan yang dihadapi adalah bagaimana mengatur nilai variabel a dan b sedemikian rupa sehingga setelah serangkaian operasi, nilai a tetap 5 dan b menjadi 6. Permasalahan ini dipecahkan dengan menggunakan operator increment dan decrement secara strategis.

Algoritma yang diusulkan melibatkan penggunaan operator post-increment (a++) untuk inisialisasi b, decrement (a--) untuk mengembalikan nilai a ke 5, dan increment (b++) untuk mencapai nilai akhir b yang diinginkan. Dengan mengikuti langkah-langkah algoritma ini, hasil akhir yang diinginkan dapat dicapai dengan efektif.

Kode yang diimplementasikan sesuai dengan algoritma yang ditetapkan dan menghasilkan output yang diinginkan: Variabel a diatur untuk tetap 5. Variabel b diatur untuk mencapai nilai 6.

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

Strategi ini memastikan bahwa a dan b mendapatkan nilai akhir yang diinginkan dengan memanfaatkan sifat dari operator post-increment dan pre-increment/decrement, yang memungkinkan penggunaan operator increment dan decrement secara sistematis dan terstruktur untuk mengubah nilai variabel dengan tepat sesuai kebutuhan. Ini adalah dasar dari pengambilan keputusan ini.

[No. 5] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

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

Luaran:

Hasil logika (a && b) : false

Latihan 5

5.1. Rekomendasikan berapa nilai a dan b apabila ingin menghasilkan luaran *true* dengan operator && dan operator | ?

5.2. Berikan kesimpulan dari latihan 5.1.

Jawab

5.1. untuk operator AND && agar (a && b) menghasilkan true, kedua operand a dan b harus bernilai true. untuk operator OR || agar (a || b) menghasilkan true, kedua operand a dan b harus bernilai true.

5.2. &&: Memerlukan kedua operand untuk true agar hasilnya true.

||: Memerlukan minimal satu operand true agar hasilnya true.

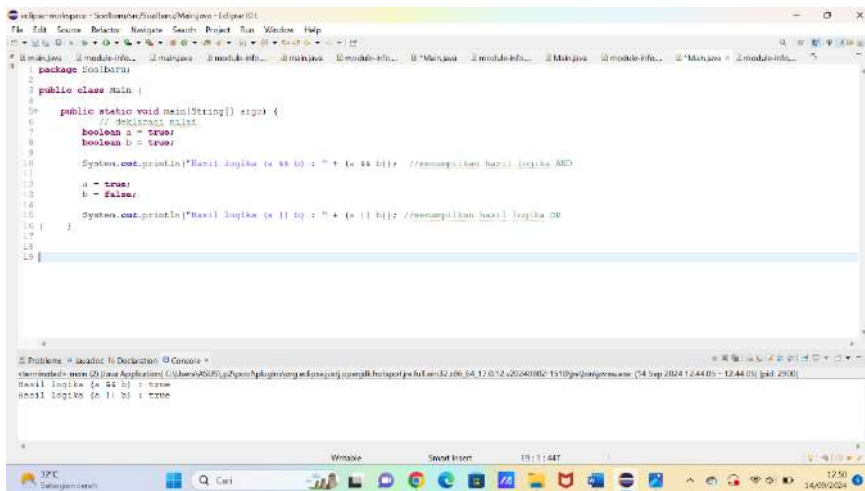
Latihan ini menegaskan bahwa pemahaman yang baik tentang operator logika sangat penting untuk membuat keputusan dalam pemrograman yang melibatkan kondisi boolean. Penggunaan yang benar dari && dan || dapat membantu dalam mengontrol alur logika program dengan akurat.

[No.5] Analisis dan Argumentasi

- a) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menerapkan pemahaman operator, untuk memastikan penggunaan operator logika && dan || menghasilkan hasil yang diinginkan, penting untuk memahami bagaimana kedua operator tersebut bekerja dalam berbagai situasi.
- b) Alasan solusi ini karena pemahaman yang benar tentang operator logika membantu dalam merancang dan mengimplementasikan kondisi yang tepat dalam kode program.

[No.5] Penyusunan Algoritma dan Kode Program

- 5) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - a) Deklarasikan dua variabel boolean, a dan b.
 - b) Inisialisasi variabel a dan b dengan nilai boolean yang sesuai (misalnya true atau false).
 - c) Terapkan operator logika && pada variabel a dan b.
 - d) ubah nilai dari variabel a dan b untuk menciptakan berbagai kombinasi nilai.
 - e) Terapkan operator logika || pada variabel yang sudah diperbarui.
 - f) Tampilkan output
2. Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran
Berikan komentar pada kode yang di Screenshot



```
1 package com.kelompok5;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // Deklarasi variabel
7         boolean a = true;
8         boolean b = true;
9
10        System.out.println("Hasil logika (a && b) : " + (a && b)); //menampilkan hasil logika AND
11
12        a = true;
13        b = false;
14
15        System.out.println("Hasil logika (a || b) : " + (a || b)); //menampilkan hasil logika OR
16    }
17
18 }
19
```

Output Console:

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

b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.5] Kesimpulan

3. Analisa

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

Permasalahan: Dalam Latihan 5, kita perlu menggunakan operator logika && (AND) dan || (OR) untuk menghasilkan hasil boolean true berdasarkan nilai variabel a dan b.

Algoritma: Untuk &&, pastikan kedua variabel a dan b bernilai true agar hasilnya true. Untuk ||, setidaknya satu dari variabel a atau b harus bernilai true agar hasilnya true.

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

[No. 6] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

```
public class OperatorKondisi{
    public static void main( String[] args ){
        String status = "";
        int nilai = 80;
        status = (nilai > 60)?"Lulus":"Gagal";
        System.out.println( status );
    }
}
```

Luaran:

Lulus

Latihan 6

Rekomendasikan apa bentuk tanda operator agar nilai = 60 memenuhi untuk Lulus !

Jawab :

Agar nilai 60 tetap memenuhi untuk lulus perlu untuk menambahkan operator \geq (lebih besar dari atau sama dengan).

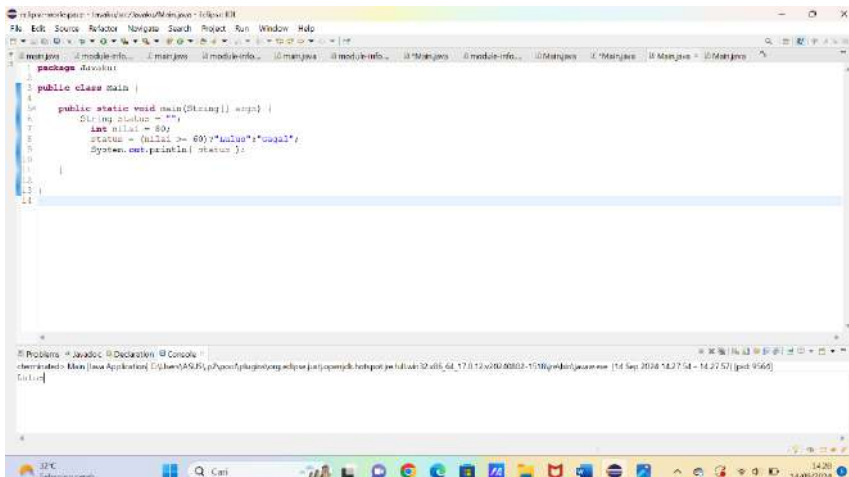
[No.6] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan atau menggantinya dengan operator \geq (lebih besar dari atau sama dengan)
- 2) Alasan solusi ini karena dapat membuat nilai 60 bisa dianggap lulus
- 3) Perbaiki kode program dengan cara menambahkan \geq

[No.6] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - a) Deklarasikan variabel nilai dan inisialisasi
 - b) Ubah operator dari > menjadi >= untuk memastikan nilai 60 juga memenuhi syarat.
 - c) Jika nilai lebih besar dari atau sama dengan 60, setel status ke "Lulus". Jika tidak, setel status ke "Gagal".
 - d) Tampilkan output

- 1) Kode program dan luaran
- a) Screenshot/ Capture potongan kode dan hasil luaran
Berikan komentar pada kode yang di Screenshot



- b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.6] Kesimpulan

6) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! utama dalam contoh kode adalah bahwa nilai 60 tidak memenuhi kriteria untuk dianggap "Lulus" karena operator perbandingan yang digunakan adalah $>$, yang hanya memeriksa apakah nilai lebih besar dari 60. Oleh karena itu, nilai 60 dianggap "Gagal". Untuk mengatasi masalah ini, algoritma dan kode program diperbarui dengan menggunakan operator perbandingan \geq (lebih besar dari atau sama dengan). Algoritma diperbarui untuk membandingkan nilai dengan 60 menggunakan \geq , memastikan bahwa nilai 60 juga memenuhi kriteria kelulusan. Dengan perubahan ini, kode program yang baru akan memeriksa apakah nilai lebih besar dari atau sama dengan 60, dan jika benar, maka status akan diatur ke "Lulus". Sebaliknya, jika nilai kurang dari 60, maka status akan diatur ke "Gagal". Kesimpulannya, dengan menggunakan operator \geq , nilai 60 kini memenuhi syarat untuk "Lulus", seperti yang diharapkan.
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Dasar alasan pengambilan keputusan adalah untuk memastikan bahwa nilai batas (60) juga memenuhi syarat yang diinginkan. Operator \geq digunakan untuk membandingkan nilai, sehingga nilai batas dapat termasuk dalam kategori "Lulus". Penggunaan operator \geq secara efektif memperbaiki masalah yang ada dengan kode awal dan memastikan bahwa kriteria kelulusan yang ditetapkan dapat dicapai dengan benar, termasuk nilai batas tersebut.

[No. 7] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

```
public class OperatorBitwise {
    public static void main(String[] args) {
        int a = 10;
        int b = 7;
        int hasil;

        hasil = a & b;
        System.out.println("Hasil dari a & b : " + hasil );

        hasil = a | b;
        System.out.println("Hasil dari a | b : " + hasil );

        hasil = a ^ b;
        System.out.println("Hasil dari a ^ b : " + hasil );

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

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

        hasil = b << 2;
        System.out.println("Hasil dari b << 2 : " + hasil );
    }
}
```

Luaran:

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

Latihan 7

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

Jawab :

Operasi bitwise NOT membalikkan setiap bit representasi biner angka. Sebagai representasi biner dari $a=10$, representasi binernya adalah 0000 0000 0000 0000 0000 0000 1010. setelah menerapkan \sim akan mendapatkan 1111 1111 1111 1111 1111 1111 1111 1111 0101. Dengan menginterpretasikan ini dalam format dua's complement, hasilnya adalah -11. Karena operasi bitwise NOT pada 10, representasi binernya diubah menjadi representasi -11 dalam format dua's complement. Akibatnya, hasil dari ~ 10 adalah -11.

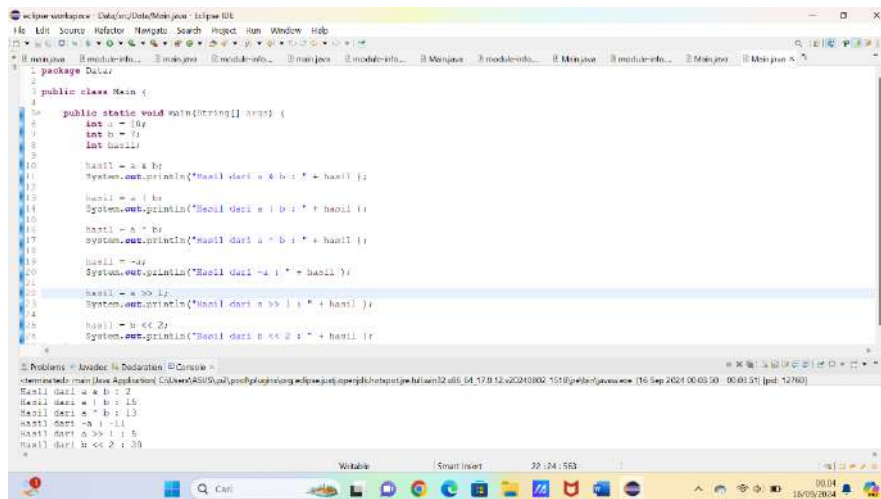
[No.7] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menjelaskan cara kerja operasi bitwise NOT dan bagaimana mengonversi hasilnya dalam format dua's complement.
- 2) Alasan solusi ini karena solusi ini menguraikan bagaimana operasi bitwise NOT membalikkan setiap bit dan kemudian menjelaskan bagaimana hasil tersebut diinterpretasikan dalam format dua's complement.

[No.7] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
Misalkan algoritma memasak mi instan:
 - a) Mulai
 - b) Masukkan Bilangan Positif

- a) Screenshot/ Capture potongan kode dan hasil luaran
Berikan komentar pada kode yang di Screenshot



- Luaran sudah sesuai dengan program yang disusun.

3. Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Kami menghadapi masalah dengan operasi bitwise NOT (\sim) pada bilangan bulat dalam format dua tambah. Dalam keadaan ini, kami menggunakan NOT bitwise pada angka 10, dan hasilnya adalah -11. Untuk memahami mengapa hasilnya demikian, kita harus mempelajari operasi bitwise NOT dan format komplementer dua. Kita pertama-tama mengubah angka positif 10 menjadi representasi biner 32-bit dari angka tersebut. Setelah itu, kita membalikkan setiap bit dari representasi biner tersebut dengan operasi bitwise NOT. Untuk menginterpretasikan hasil ini dalam format dua plus dua, kita perlu melakukan konversi tambahan. Bilangan negatif disimpan dalam format dua kali dua, yang memungkinkan kita menggunakan bit paling kiri sebagai tanda. Untuk mendapatkan bentuk positif, langkah selanjutnya adalah mengubah hasil bitwise NOT.
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Contoh jawaban Analisa:
Kami menghadapi masalah dengan operasi bitwise NOT (\sim) pada bilangan bulat dalam format dua tambah. Dalam keadaan ini, kami menggunakan NOT bitwise pada angka 10, dan hasilnya adalah -11. Untuk memahami mengapa hasilnya demikian, kita harus mempelajari operasi bitwise NOT dan format komplementer dua. Kita pertama-tama mengubah angka positif 10 menjadi representasi biner 32-bit dari angka tersebut. Setelah itu, kita membalikkan setiap bit dari representasi biner tersebut dengan operasi bitwise NOT. Untuk

menginterpretasikan hasil ini dalam format dua plus dua, kita perlu melakukan konversi tambahan. Bilangan negatif disimpan dalam format dua kali dua, yang memungkinkan kita menggunakan bit paling kiri sebagai tanda.

e) Untuk mendapatkan bentuk positif, langkah selanjutnya adalah mengubah hasil bitwise NOT.

Refleksi

Minggu ini, saya mempelajari berbagai jenis operator Java, termasuk aritmatika, relasional, logika, bitwise, dan penugasan. Pengalaman ini memperluas pemahaman saya tentang bagaimana berbagai operator berfungsi untuk melakukan operasi matematika, membandingkan nilai, dan mengubah bit dalam data.

Saya mendapatkan pemahaman baru tentang penggunaan operator bitwise dan operator logika, yang membuka mata saya pada cara yang lebih baik untuk mengelola dan memproses data pada tingkat yang lebih tinggi. Untuk menulis kode yang lebih efisien dan optimal, sangat penting untuk memahami cara operator ini bekerja. Mengingat dan menerapkan berbagai jenis operator dalam konteks yang berbeda adalah masalah utama saya. Memahami perbedaan halus antara operator, terutama ketika menggunakan operator bitwise daripada aritmatika, membutuhkan banyak latihan.