

Nama & NPM	Topik:	Tanggal:
Rizqi Nadhifah Setiani G1F024017	Operasi Aritmatika	08-09-2024

[NO.1] Identifikasi Masalah:

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

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

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

Pada kode program di atas masih terdapat pesan kesalahan (eror) karena tidak ada operator "+" untuk menggabungkan string "a + b =" dengan hasil penjumlahan (a + b).

Sumber informasi relevan yang dapat digunakan yakni **W3Schools** : sebuah website yang menyediakan tutorial dan referensi tentang berbagai teknologi web dan pemrograman, termasuk HTML, CSS, JavaScript, SQL, PHP, dan Java. Website ini dirancang untuk menjadi sumber belajar yang mudah diakses dan praktis bagi pengembang web dan pemrogram.

[No.1] Analisis dan Argumentasi

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!

Kesalahan kode program pada contoh 1 terjadi karena seharusnya operator + digunakan untuk menggabungkan string dengan hasil operasi aritmatika. Untuk itu, perbaikan kode program dapat dilakukan dengan menambahkan operator + di antara string dan hasil penjumlahan. Alasan penambahan "+" karena di dalam Java, operator + digunakan untuk menggabungkan (concatenate) string dengan nilai lainnya, termasuk hasil operasi aritmatika. Dalam konteks System.out.println, tanda + berfungsi untuk menggabungkan string dengan nilai atau ekspresi lainnya. Ketika menulis System.out.println("a: " + a);, Java akan menggabungkan string "a: " dengan nilai variabel a.

Untuk membuat baris yang akan menampilkan perhitungan dengan operator (-, *, / , %) pada Contoh 1, kita menggunakan operasi aritmatika. Operasi aritmatika merupakan fondasi dari banyak proses dalam pemrograman, dari perhitungan sederhana hingga algoritma kompleks.

Selanjutnya, ada beberapa operator yang digunakan dalam operasi aritmatika, yaitu :

1.Operator Penjumlahan (+)

Operator penjumlahan digunakan untuk menjumlahkan dua nilai. Dalam Java, operator ini dapat digunakan untuk bilangan bulat, bilangan desimal, dan bahkan untuk penggabungan string.

2.Operator Pengurangan (-)

Operator pengurangan digunakan untuk mengurangi satu nilai dari nilai lainnya. Sama seperti

operator penjumlahan, pengurangan juga dapat digunakan pada berbagai tipe data numerik.

3.Operator Perkalian (*)

Operator perkalian digunakan untuk mengalikan dua nilai. Ini sangat berguna dalam berbagai perhitungan matematika.

4.Operator Pembagian (/)

Operator pembagian digunakan untuk membagi satu nilai dengan nilai lainnya. Hasil dari operasi pembagian ini dapat berupa bilangan bulat atau desimal tergantung pada tipe data yang digunakan.

5.Operator Modulus (%)

Operator modulus memberikan sisa dari pembagian dua bilangan. Ini sering digunakan dalam berbagai aplikasi, seperti menentukan apakah suatu bilangan genap atau ganjil.

Dengan menggunakan operator-operator ini, kita dapat menambahkan baris untuk menampilkan perhitungan.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma

- Siapkan kelas baru "OperatorMatematika"
- Tempelkan kode soal pada eclipse
- Perbaiki eror dengan menambahkan "+"
- Buat operator matematika
- Jalankan program untuk menampilkan hasil

2) Kode program dan luaran

```
1 package Rizqi.Nadhifah;
2 public class OperatorAritmatika { // mendeklarasikan kelas
3     public static void main(String[] args) { // metode utama
4         // deklarasi nilai
5         int a = 20, b = 3;
6
7         // operator aritmatika
8         System.out.println("a: " + a);
9         System.out.println("b: " + b);
10
11        // menampilkan hasil penjumlahan
12        System.out.println("a + b = " + (a + b));
13
14        // menampilkan hasil pengurangan
15        System.out.println("a - b = " + (a - b));
16
17        // menampilkan hasil perkalian
18        System.out.println("a * b = " + (a * b));
19
20        // menampilkan hasil pembagian
21        System.out.println("a / b = " + (a / b));
22
23        // menampilkan hasil modulus (sisa bagi)
24        System.out.println("a % b = " + (a % b));
25    }
```

```
<terminated> OperatorAritmatika [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 9, 2
a: 20
b: 3
a + b = 23
a - b = 17
a * b = 60
a / b = 6
a % b = 2
```

Luaran sudah sesuai dengan program yang disusun.

[No.1] Kesimpulan

Analisa

Pada program itu saya melakukan perbaikan dengan menambahkan operator penjumlahan (+) untuk menampilkan hasil penjumlahan karena pada java operator tersebut digunakan untuk menggabungkan string dengan nilai lainnya, termasuk hasil operasi aritmatika. Sedangkan untuk pembuatan baris baru operasi aritmatika menggunakan operator lainnya, cukup mengikuti contoh yang sudah ada pada baris ppenjumlahan.

Nama & NPM	Topik:	Tanggal:
Rizqi Nadhifah Setiani G1F024017	Operator Penugasan	09-09-2024
[No.2] Identifikasi Masalah:		
<pre>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 } }</pre> <p>Luaran: Penambahan : 23</p> <p>Pada kode program tersebut ditampilkan perhitungan penjumlahan menggunakan operator penugasan dari variabel a = 20 dan b = 3, menampilkan luaran berupa hasil penjumlahannya yaitu 23. Selanjutnya diminta membuat perhitungan menggunakan operator lainnya.</p> <p>Sumber informasi relevan yang dapat digunakan yakni W3Schools : sebuah website yang menyediakan tutorial dan referensi tentang berbagai teknologi web dan pemrograman, termasuk HTML, CSS, JavaScript, SQL, PHP, dan Java. Website ini dirancang untuk menjadi sumber belajar yang mudah diakses dan praktis bagi pengembang web dan pemrogram.</p>		
[No.2] Analisis dan Argumentasi		
<p>Latihan 2.</p> <p>2.1. Tambahkan baris Contoh 2 untuk menampilkan perhitungan dengan operator (-=, *=, /=, %=)!</p> <p>2.2. Berikan argumentasi tentang perbedaan luaran dan waktu eksekusi Contoh 1 dan Contoh 2!</p> <p>Selain operator penjumlahan, terdapat 4 operator penugasan lainnya yang dapat digunakan pada pemrograman Java, yaitu :</p> <ol style="list-style-type: none"> Pengurangan (-=): Operator ini digunakan untuk mengurangi nilai variabel dengan nilai tertentu dan kemudian menyimpan hasilnya kembali ke variabel tersebut. Contohnya pada $b -= a$, ini setara dengan $b = b - a$. Perkalian (*=): Operator ini digunakan untuk mengalikan nilai variabel dengan nilai tertentu dan kemudian menyimpan hasilnya kembali ke variabel tersebut. Pembagian (/=): Operator ini digunakan untuk membagi nilai variabel dengan nilai tertentu dan kemudian menyimpan hasilnya kembali ke variabel tersebut. Modulus (%=): Operator ini digunakan untuk mendapatkan sisa hasil bagi dari pembagian nilai variabel dengan nilai tertentu dan kemudian menyimpan hasilnya kembali ke variabel tersebut. <p>Operator-operator ini memudahkan kita untuk melakukan operasi aritmatika dan langsung memperbarui nilai variabel tanpa harus menulis ulang variabel tersebut. Secara umum, luaran dari jenis operator aritmatika dan penugasan ini akan sama jika digunakan dengan cara yang benar. Namun, operator penugasan lebih ringkas dan langsung memperbarui nilai variabel, sedangkan operator aritmatika memerlukan langkah tambahan untuk memperbarui variabel. Dalam banyak kasus, perbedaan waktu eksekusi antara operator penugasan dan operator aritmatika sangat kecil dan mungkin tidak signifikan. Namun, operator penugasan bisa sedikit</p>		

lebih cepat karena mereka mengurangi jumlah operasi yang harus dilakukan.

[No.2] Penyusunan Algoritma dan Kode Program

Algoritma

- Siapkan kelas baru "OperatorPenugasan"
- Tempelkan kode soal pada eclipse
- Buat operator penugasan pengurangan, perkalian, pembagian, dan modulus
- Jalankan program untuk menampilkan hasil

Kode Program dan Luaran

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

Problems Javadoc Declaration Console x

<terminated> OperatorAritmatika [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 9, 2024, 5:49:09 PM -

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

Luaran sudah sesuai dengan program yang disusun.

[No.2] Kesimpulan

Analisa

Pada kode program tersebut ditampilkan perhitungan penjumlahan menggunakan operator penugasan dari variabel $a = 20$ dan $b = 3$, menampilkan luaran berupa hasil penjumlahannya yaitu 23. Selanjutnya, diminta membuat perhitungan menggunakan operator lainnya. Operator penugasan tersebut memudahkan kita untuk melakukan operasi aritmatika dan langsung memperbarui nilai variabel tanpa harus menulis ulang variabel tersebut. Secara umum, luaran dari jenis operator aritmatika dan penugasan akan sama jika digunakan dengan cara yang benar. Namun, operator penugasan lebih ringkas dan langsung memperbarui nilai variabel.

Nama & NPM	Topik:	Tanggal:
Rizqi Nadhifah Setiani G1F024017	Operator Relasional	09-09-2024
[No.3] Identifikasi Masalah:		
<pre> 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); } } </pre> <p>Luaran:</p> <pre> 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 </pre> <p>Kode program di atas menampilkan operator relasi untuk menguji hubungan antara nilai dan atau variabel dan selalu menghasilkan nilai true atau false. Pada program tersebut digunakan nilai A = 12 dan B = 4 yang menghasilkan nilai true dan false sesuai hubungan antara variabel yang ada. Selanjutnya, diminta untuk mengubah nilai A menjadi 12 dan melihat perubahan hubungan variabel yang terjadi.</p> <p>Sumber informasi relevan yang dapat digunakan yakni W3Schools : sebuah website yang menyediakan tutorial dan referensi tentang berbagai teknologi web dan pemrograman, termasuk HTML, CSS, JavaScript, SQL, PHP, dan Java. Website ini dirancang untuk menjadi sumber belajar yang mudah diakses dan praktis bagi pengembang web dan pemrogram.</p>		
[No.3] Analisis dan Argumentasi		
Latihan 3		

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

Operasi relasional adalah jenis operasi dalam pemrograman yang digunakan untuk membandingkan dua nilai atau variabel. Hasil dari operasi ini adalah nilai Boolean true atau false. Operasi relasional sering digunakan dalam pernyataan kondisi seperti if, while, dan for untuk membuat keputusan berdasarkan perbandingan nilai.

Berikut adalah beberapa operator relasional yang umum digunakan:

- a) **Lebih besar dari (>):**
Membandingkan apakah nilai di sebelah kiri lebih besar dari nilai di sebelah kanan.
Contoh: $a > b$ akan bernilai true jika a lebih besar dari b.
- b) **Lebih kecil dari (<):**
Membandingkan apakah nilai di sebelah kiri lebih kecil dari nilai di sebelah kanan.
Contoh: $a < b$ akan bernilai true jika a lebih kecil dari b.
- c) **Lebih besar atau sama dengan (>=):**
Membandingkan apakah nilai di sebelah kiri lebih besar atau sama dengan nilai di sebelah kanan.
Contoh: $a >= b$ akan bernilai true jika a lebih besar atau sama dengan b.
- d) **Lebih kecil atau sama dengan (<=):**
Membandingkan apakah nilai di sebelah kiri lebih kecil atau sama dengan nilai di sebelah kanan.
Contoh: $a <= b$ akan bernilai true jika a lebih kecil atau sama dengan b.
- e) **Sama dengan (==):**
Membandingkan apakah dua nilai sama.
Contoh: $a == b$ akan bernilai true jika a sama dengan b.
- f) **Tidak sama dengan (!=):**
Membandingkan apakah dua nilai tidak sama.
Contoh: $a != b$ akan bernilai true jika a tidak sama dengan b.

Operasi relasional sangat penting dalam pengambilan keputusan dan kontrol alur yang akan memungkinkan program untuk melakukan tindakan yang berbeda berdasarkan hasil perbandingan. Pada kode program yang terdapat pada soal, nilai A yang awalnya 12 diubah menjadi 4. Hal tersebut tentunya memberikan hasil luaran yang berbeda dengan luaran awalnya.

[No.3] Penyusunan Algoritma dan Kode Program

Algoritma

- a) Siapkan kelas baru "OperatorRelasional"
- b) Tempelkan kode soal pada eclipse
- c) Ubah nilai 12 menjadi 4
- d) Jalankan program untuk menampilkan hasil

Kode program dan luaran

```

1 package Rizqi.Nadhifah;
2 public class OperatorRelasional {
3     public static void main(String[] args) {
4         int nilaiA = 4;
5         int nilaiB = 4;
6         boolean hasil;
7
8         System.out.println("A = " + nilaiA + "\nB = " + nilaiB);
9         // apakah A lebih besar dari B?
10        hasil = nilaiA > nilaiB;
11        System.out.println("\nHasil A > B = " + hasil);
12
13        // apakah A lebih kecil dari B?
14        hasil = nilaiA < nilaiB;
15        System.out.println("\nHasil A < B = " + hasil);
16
17        // apakah A lebih besar sama dengan B?
18        hasil = nilaiA >= nilaiB;
19        System.out.println("\nHasil A >= B = " + hasil);
20
21        // apakah A lebih kecil sama dengan B?
22        hasil = nilaiA <= nilaiB;
23        System.out.println("\nHasil A <= B = " + hasil);
24
25        // apakah nilai A sama dengan B?
26        hasil = nilaiA == nilaiB;
27        System.out.println("\nHasil A == B = " + hasil);
28
29        // apakah nilai A tidak sama dengan B?
30        hasil = nilaiA != nilaiB;
31        System.out.println("\nHasil A != B = " + hasil);
32    }
33 }

```

<terminated> OperatorRelasional [Java Application] C:\Program Files\Java\jre1.8.

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

Luaran sudah sesuai dengan program yang disusun.

[No.3] Kesimpulan

Analisa

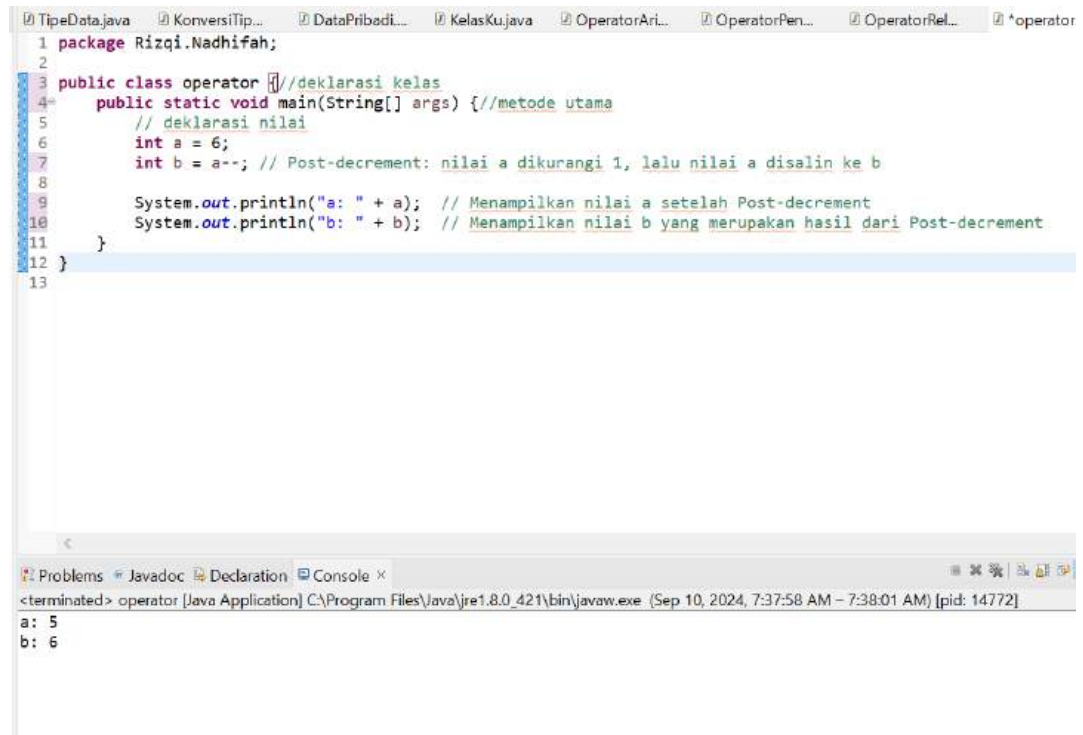
Pada kode program tersebut ditampilkan perbandingan nilai dari variable a dan b menggunakan operasi relasional. Adapun, penugasan dari variabel a = 12 dan b = 4, menampilkan luaran berupa nilai Boolean (true or false) sesuai relasi dari kedua nilai tersebut. Setelah variable a diubah dari 12 menjadi 4, luaran yang tadinya true, false, true, false, false, dan true menjadi false, false, true, true, true, dan false.

Nama & NPM	Topik:	Tanggal:
Rizqi Nadhifah Setiani G1F024017	Operator Increment dan Decrement	10-09-2024
[No.4] Identifikasi Masalah:		
<pre>public class operator { public static void main(String[] args) { // deklarasi nilai int a = 5; System.out.println("a: " + a); System.out.println("b: " + (a++)); } }</pre> <p>Luaran: a: 5 b: 5</p> <p>Pada kode program di atas mendeklarasikan sebuah variabel integer <i>a</i> dengan nilai awal 5. Kemudian, nilai <i>a</i> dicetak, diikuti oleh hasil dari ekspresi <i>a++</i>. Operasi <i>a++</i> adalah <i>post-increment</i>, artinya nilai <i>a</i> akan digunakan terlebih dahulu dalam ekspresi, baru kemudian nilainya ditingkatkan sebesar 1. Oleh karena itu, dalam mencetak nilai <i>b</i>, yang sebenarnya dicetak adalah nilai <i>a</i> sebelum ditingkatkan, yaitu 5.</p> <p>Sumber informasi relevan yang dapat digunakan yakni W3Schools : sebuah website yang menyediakan tutorial dan referensi tentang berbagai teknologi web dan pemrograman, termasuk HTML, CSS, JavaScript, SQL, PHP, dan Java. Website ini dirancang untuk menjadi sumber belajar yang mudah diakses dan praktis bagi pengembang web dan pemrogram.</p>		
[No.4] Analisis dan Argumentasi		
<p>Latihan 4.</p> <p>4.1. Berikan saran operasi apa yang diperlukan (pre/post increment, pre/post decrement) agar Contoh 4 menghasilkan nilai <i>a</i> = 5 dan <i>b</i> = 6?</p> <p>4.2. Simpulkan hasil eksperimen Anda!</p> <p>Untuk mendapatkan hasil luaran berupa <i>a</i> = 5 dan <i>b</i> = 6, kita perlu mengubah terlebih dahulu nilai <i>a</i> menjadi 6, kemudian gunakan operasi post-decrement.</p> <p>Dengan memahami konsep increment dan decrement, kita dapat mengontrol kapan nilai variabel ditingkatkan atau dikurangkan, dan mendapatkan hasil yang diinginkan dalam operasi aritmetika pada bahasa pemrograman.</p> <p>Penjelasan lebih lanjut :</p> <p>Post-increment: Digunakan ketika ingin menggunakan nilai yang lama terlebih dahulu, baru kemudian mengubah nilainya. Analoginya seperti mencetak nomor antrean sebelum nomor antrean bertambah.</p> <p>Pre-increment: Digunakan ketika ingin menggunakan nilai yang baru terlebih dahulu. Analoginya seperti langsung menggunakan nomor antrean yang baru untuk melakukan sesuatu.</p> <p>Pre-decrement: Digunakan ketika ingin menggunakan nilai yang sudah dikurangi 1 dalam ekspresi yang sama.</p> <p>Post-decrement: Digunakan ketika ingin menggunakan nilai yang lama terlebih dahulu, baru kemudian mengurangnya.</p> <p>Berdasarkan penjelasan tersebut, itulah mengapa luaran berupa <i>a</i> = 5 dan <i>b</i> = 6 hanya bias didapatkan jika nilai <i>a</i> diubah terlebih dahulu menjadi 6, karena saat menggunakan post-decrement program akan berjalan dengan prinsip “Pakai dulu, baru kurangi”, sehingga nilai <i>a</i> yang saat ini 6 digunakan untuk disimpan ke dalam variabel <i>b</i>. Setelah itu, baru nilai <i>a</i> dikurangi 1 menjadi 5.</p>		
[No.4] Penyusunan Algoritma dan Kode Program		

Algoritma :

- Siapkan kelas baru "Operator"
- Tempelkan kode soal pada eclipse
- Ubah nilai a menjadi 6
- Gunakan operator Post-decrement (a--)
- Jalankan program untuk menampilkan hasil

Kode program dan luaran



```
1 package Rizqi.Nadhifah;
2
3 public class operator { //deklarasi kelas
4     public static void main(String[] args) { //metode utama
5         // deklarasi nilai
6         int a = 6;
7         int b = a--; // Post-decrement: nilai a dikurangi 1, lalu nilai a disalin ke b
8
9         System.out.println("a: " + a); // Menampilkan nilai a setelah Post-decrement
10        System.out.println("b: " + b); // Menampilkan nilai b yang merupakan hasil dari Post-decrement
11    }
12 }
13
```

Problems Javadoc Declaration Console x

<terminated> operator [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 10, 2024, 7:37:58 AM – 7:38:01 AM) [pid: 14772]

a: 5
b: 6

Luaran yang dihasilkan sudah sesuai dengan kode program dan seperti yang diinginkan.

[No.4] Kesimpulan

Analisa

Berdasarkan eksperimen yang saya lakukan, dapat disimpulkan bahwa operator post-decrement (--) bekerja sesuai dengan definisinya. Ketika operator ini digunakan pada sebuah variabel, nilai variabel tersebut akan digunakan terlebih dahulu dalam ekspresi, baru kemudian nilainya dikurangi satu. Hal ini terbukti pada hasil percobaan di mana variabel b memperoleh nilai a sebelum a dikurangi satu.

Nama & NPM	Topik:	Tanggal:
Rizqi Nadhifah Setiani G1F024017	Operator Logika	10-09-2024
[No.5] Identifikasi Masalah:		
<pre> 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 } } </pre> <p>Luaran: Hasil logika (a && b) : false</p> <p>Pada kode program tersebut ditampilkan sebuah kode program yang menjalankan operator logika “and” dengan luaran berupa nilai boolean, yaitu false. Kemudian diminta untuk menentukan nilai a dan b sehingga dapat menghasilkan luaran “true” dari operator&& dan operator .</p> <p>Sumber informasi relevan yang dapat digunakan yakni W3Schools : sebuah website yang menyediakan tutorial dan referensi tentang berbagai teknologi web dan pemrograman, termasuk HTML, CSS, JavaScript, SQL, PHP, dan Java. Website ini dirancang untuk menjadi sumber belajar yang mudah diakses dan praktis bagi pengembang web dan pemrogram.</p>		
[No.5] Analisis dan Argumentasi		
<p>Latihan 5</p> <p>5.1. Rekomendasikan berapa nilai a dan b apabila ingin menghasilkan luaran <i>true</i> dengan operator && dan operator ?</p> <p>5.2. Berikan kesimpulan dari latihan 5.1.</p> <p>Luaran “true” dapat dihasilkan menggunakan operator AND dan OR. Operator && akan menghasilkan true hanya jika kedua operand bernilai true. Jika salah satu atau kedua operand bernilai false, hasilnya akan false.</p> <p>Contohnya :</p> <ol style="list-style-type: none"> a = true, b = true: a && b = true && true = true a = true, b = false: a && b = true && false = false <p>Sedangkan operator akan menghasilkan true jika salah satu atau kedua operand bernilai true. Hasilnya hanya akan false jika kedua operand bernilai false.</p> <p>Contohnya :</p> <ol style="list-style-type: none"> a = true, b = true: a b = true true = true a = true, b = false: a b = true false = true <p>Kesimpulannya yaitu, saat menggunakan operator && maka kedua nilai a dan b harus true untuk menghasilkan true. Sedangkan pada operator setidaknya salah satu nilai harus true untuk menghasilkan true, saya menggunakan nilai a sebagai true dan b sebagai false.</p>		

[No.5] Penyusunan Algoritma dan Kode Program

Algoritma

- Siapkan kelas baru "OperatorLogika"
- Tempelkan kode soal pada eclipse
- Ubah nilai b menjadi true
- Jalankan program untuk menampilkan luaran
- Ubah nilai b menjadi false kembali
- Ubah operator AND menjadi OR
- Tambahkan komentar
- Jalankan program untuk menampilkan luaran

Kode program dan luaran

```
1 package Rizqi.Nadhifah;
2 public class OperatorLogika {
3     public static void main(String[] args) {
4         // deklarasi nilai
5         boolean a = true;
6         boolean b = false; // atau bisa juga boolean a = false; boolean b = true;
7
8         System.out.println("Hasil logika (a || b) : " + (a || b)); // menampilkan hasil logika OR
9     }
10 }
11
```

Problems Javadoc Declaration Console ×
<terminated> OperatorLogika [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 10, 2024, 8:25:16 AM – 8:25:21 AM) [pid: 76...]
Hasil logika (a || b) : true

```
1 package Rizqi.Nadhifah;
2 public class OperatorLogika {
3     public static void main(String[] args) { //METODE UTAMA
4         // deklarasi nilai
5         boolean a = true;
6         boolean b = true;
7         System.out.println("Hasil logika (a && b) : " + (a && b)); // menampilkan hasil logika AND
8     }
9 }
10
```

Problems Javadoc Declaration Console ×
<terminated> OperatorLogika [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 10, 2024, 7:38:28 PM – 7:38:31 PM) [pid: 9164]
Hasil logika (a && b) : true

Luaran yang dihasilkan sudah sesuai dengan kode program yang dibuat dan diharapkan.

[No.5] Kesimpulan

Analisa

Pada kode program tersebut ditampilkan sebuah kode program yang menjalankan operator logika "and" dengan luaran berupa nilai boolean, yaitu false. Kemudian diminta untuk menentukan nilai a dan b sehingga dapat menghasilkan luaran "true" dari operator && dan operator |. Dari percobaan yang saya lakukan saat menggunakan operator && maka kedua nilai a dan b harus true untuk menghasilkan true. Sedangkan pada operator || setidaknya salah satu nilai harus true untuk menghasilkan true, saya menggunakan nilai a sebagai true dan b sebagai false.

Nama & NPM:	Topik:	Tanggal:
Rizqi Nadhifah Setiani G1F024017	Operator Kondisional (Ternary)	10-09-2024
[No.6] Identifikasi Masalah:		
<p>Contoh 6:</p> <pre>public class OperatorKondisi{ public static void main(String[] args){ String status = ""; int nilai = 80; status = (nilai > 60)?"Lulus":"Gagal"; System.out.println(status); } }</pre> <p>Luaran: Lulus</p> <p>Kode di atas menggunakan operator ternary (?:) untuk menentukan nilai status berdasarkan nilai "nilai". Jika "nilai" lebih besar dari 60, maka status akan bernilai "Lulus", jika tidak, maka "Gagal". Kemudian di soal diminta menyesuaikan kondisi agar nilai 60 juga termasuk dalam kategori "Lulus".</p>		
[No.6] Analisis dan Argumentasi		
<p>Latihan 6 Rekomendasikan apa bentuk tanda operator agar nilai = 60 memenuhi untuk Lulus !</p> <p>Operator ternary adalah sebuah operator kondisional dalam bahasa pemrograman yang memungkinkan kita mengevaluasi sebuah ekspresi dan memilih salah satu dari dua nilai berdasarkan hasil evaluasi tersebut.</p> <p>Penjelasan lebih lanjut : condition ? expression1 : expression2 condition: Sebuah ekspresi boolean yang menghasilkan nilai true atau false. expression1: Ekspresi yang akan dievaluasi jika condition bernilai true. expression2: Ekspresi yang akan dievaluasi jika condition bernilai false.</p> <p>Dalam contoh ini:</p> <ul style="list-style-type: none"> • nilai >= 60 adalah ekspresi boolean yang akan dievaluasi. • Jika nilai >= 60 bernilai true, maka status akan bernilai "Lulus". • Jika nilai >= 60 bernilai false, maka status akan bernilai "Gagal". <p>Pada soal nilai integer merupakan 80, kemudian diminta agar nilai 60 juga menghasilkan luaran berupa lulus. Untuk membuat nilai 60 juga dianggap "Lulus", kita perlu mengubah nilai integer menjadi 60 dan operator perbandingan dari > menjadi >=. Perubahan ini memastikan bahwa semua nilai yang sama dengan atau lebih besar dari 60 akan dikategorikan sebagai "Lulus".</p> <p>Terdapat operator lainnya dalam perbandingan kondisi, yaitu : < Lebih kecil dari, <= Lebih kecil dari atau sama dengan, == Sama dengan, dan != Tidak sama dengan. Berdasarkan operator-operator tersebut yang saya gunakan adalah operator >=. Dengan perubahan ini, kondisi (nilai >= 60) akan bernilai true ketika nilai adalah 60 atau lebih besar dari 60. Akibatnya, status akan ditetapkan sebagai "Lulus" dalam kasus tersebut.</p>		
[No.6] Penyusunan Algoritma dan Kode Program		
<p>Algoritma</p> <p>a) Siapkan kelas baru "OperatorKondisi"</p>		

- b) Tempelkan kode soal pada eclipse
- c) Ubah nilai integer menjadi 60
- d) Ubah operator > menjadi >=
- e) Tambahkan komentar
- f) Jalankan program untuk menampilkan luaran

Kode program dan luaran

```
1 package Rizqi.Nadhifah;
2
3 public class OperatorKondisi {
4     public static void main(String[] args) {
5         String status = "";
6         int nilai = 60; // Mengubah nilai untuk menguji kondisi
7
8         // Menggunakan operator >= agar nilai 60 termasuk dalam kategori "Lulus"
9         status = (nilai >= 60) ? "Lulus" : "Gagal";
10
11         System.out.println(status);
12     }
13 }
```

<terminated> OperatorKondisi [Java Application] C:\Program Files\Java\jre1.8.0_421\bin\javaw.exe (Sep 10, 2024, 9:08:40 f
Lulus

[No.6] Kesimpulan

Analisa

Kode di atas menggunakan operator ternary (?:) untuk menentukan nilai status berdasarkan nilai "nilai". Jika "nilai" lebih besar dari 60, maka status akan bernilai "Lulus", jika tidak, maka "Gagal". Kemudian di soal diminta menyesuaikan kondisi agar nilai 60 juga termasuk dalam kategori "Lulus".

Untuk membuat nilai 60 juga dianggap "Lulus", kita perlu mengubah nilai integer menjadi 60 dan operator perbandingan dari > menjadi >=. Perubahan ini memastikan bahwa semua nilai yang sama dengan atau lebih besar dari 60 akan dikategorikan sebagai "Lulus".

Nama & NPM	Topik:	Tanggal:
Rizqi Nadhifah Setiani G1F024017	Operator Bitwise	10-09-2024

[No.7] Identifikasi Masalah:

```

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

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

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

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

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

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

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

```

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

```

Operasi bitwise bekerja langsung pada representasi biner dari bilangan. Setiap bit (0 atau 1) dalam bilangan tersebut dioperasikan secara individual. Pada kode program di atas ditampilkan penggunaan operator bitwise yang terdiri dari & (AND), | (OR), ^ (XOR), ~ (NOT), >> (Shift Right), juga << (Shift Left). Kemudian, nilai dari integer a adalah 10. Hasil luaran dari operator not pada a menampilkan nilai -11. Diminta untuk mengevaluasi penyebab hasil ~a = -11 dengan membuktikan jawaban dalam perhitungan biner.

[No.7] Analisis dan Argumentasi

Latihan 7

Evaluasi penyebab hasil ~a = -11 ? Buktikan jawaban Anda dalam perhitungan biner!

Operator bitwise adalah alat yang sangat kuat dalam pemrograman. Operator ini bekerja langsung pada bit-bit individual dari suatu bilangan. Dengan memahami cara kerjanya, kita dapat melakukan manipulasi data tingkat rendah yang kompleks dan efisien. Namun, penggunaan operator bitwise membutuhkan pemahaman yang baik tentang representasi biner dan sistem bilangan.

Berikut ini adalah beberapa operator bitwise umum:

- **& (AND):** Kedua bit harus bernilai 1 agar hasilnya 1.

- **| (OR):** Salah satu atau kedua bit harus bernilai 1 agar hasilnya 1.
- **^ (XOR):** Kedua bit harus berbeda nilai agar hasilnya 1.
- **~ (NOT):** Membalikkan nilai setiap bit (0 menjadi 1, dan sebaliknya).
- **>> (Shift Right):** Menggeser semua bit ke kanan sejumlah posisi tertentu.
- **<< (Shift Left):** Menggeser semua bit ke kiri sejumlah posisi tertentu.

Kembali ke analisis kali ini, mengapa $\sim a = -11$?

Untuk menjawabnya, saya menggunakan analogi lampu. Bayangkan bilangan biner seperti sekumpulan lampu, yang mana setiap lampu mewakili satu bit, yang bisa menyala (1) atau mati (0). Bilangan desimal adalah cara kita menjumlahkan nilai dari lampu-lampu yang menyala, sedangkan operasi bitwise adalah cara kita untuk mengotak-atik lampu-lampu ini secara individu. Pada operator \sim (NOT), Semua lampu yang menyala kita matikan, dan semua yang mati kita nyalakan. Ini seperti membalik sakelar semua lampu sekaligus.

a (10) dalam biner adalah 00001010. Bayangkan 8 lampu, di mana lampu ke-2 dan ke-4 menyala.

$\sim a$ berarti kita membalik semua lampu. Jadi, sekarang lampu ke-2 dan ke-4 mati, sementara yang lain menyala.

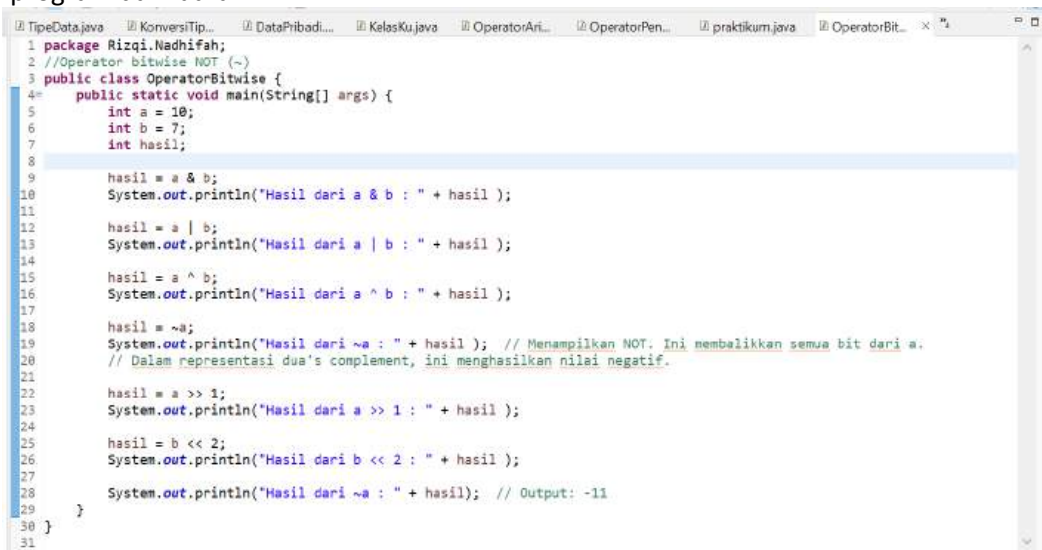
11110101 dalam bentuk desimal adalah -11. Ini karena komputer menggunakan sistem komplemen dua untuk merepresentasikan bilangan negatif. Hasil negative ini menunjukkan bahwa bilangan tersebut telah dibalik dan ditambahkan 1 angka.

[No.7] Penyusunan Algoritma dan Kode Program

Algoritma

- Siapkan kelas baru "OperatorBitwise"
- Tempelkan kode soal pada eclipse
- Tambahkan komentar
- Jalankan program untuk menampilkan luaran

Kode program dan luaran



```

1 package Rizqi.Nadhifah;
2 //Operator bitwise NOT (~)
3 public class OperatorBitwise {
4     public static void main(String[] args) {
5         int a = 10;
6         int b = 7;
7         int hasil;
8
9         hasil = a & b;
10        System.out.println("Hasil dari a & b : " + hasil);
11
12        hasil = a | b;
13        System.out.println("Hasil dari a | b : " + hasil);
14
15        hasil = a ^ b;
16        System.out.println("Hasil dari a ^ b : " + hasil);
17
18        hasil = ~a;
19        System.out.println("Hasil dari ~a : " + hasil); // Menampilkan NOT. Ini membalikkan semua bit dari a.
20        // Dalam representasi dua's complement, ini menghasilkan nilai negatif.
21
22        hasil = a >> 1;
23        System.out.println("Hasil dari a >> 1 : " + hasil);
24
25        hasil = b << 2;
26        System.out.println("Hasil dari b << 2 : " + hasil);
27
28        System.out.println("Hasil dari ~a : " + hasil); // Output: -11
29    }
30 }
31
  
```



```
Problems @ Javadoc Declaration Console x
<terminated> OperatorBitwise [Java Application] C:\Program Files\Java\j
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
Hasil dari ~a : 28
```

Luaran yang dihasilkan sudah sesuai dengan kode program yang dibuat.

[No.7] Kesimpulan

Analisa

Pada soal, nilai dari integer a adalah 10. Hasil luaran dari operator not pada a menampilkan nilai -11. Diminta untuk mengevaluasi penyebab hasil $\sim a = -11$ dengan membuktikan jawaban dalam perhitungan biner. Penggunaan operator bitwise membutuhkan pemahaman tentang representasi biner dan sistem bilangan. Untuk menjawabnya saya menggunakan analogi lampu. Jadi intinya operasi bitwise itu seperti bermain dengan sakelar lampu. Kita bisa menyalakan, mematikan, atau menggeser posisi lampu-lampu ini sesuai dengan operasi yang kita gunakan. Dan cara komputer menghitung nilai dari susunan lampu ini mengikuti aturan-aturan tertentu.

Refleksi

Pada materi ke-2 ini, yaitu operator Java, saya menjadi lebih mengenal mengenai berbagai operator yang ada dan implementasinya berkat dari 7 eksperimen yang diberikan. Merupakan tantangan yang lebih menantang lagi bagi saya pribadi, karena latihan ini hanya membutuhkan waktu 1 minggu untuk pemahamannya, juga penyelesaian coding dan laporan praktikumnya.