

| | | |
|---|----------------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| HANIFAH AZIZAH G1F024037 | OPERATOR JAVA | 05/09/2024 |
| [No.1] Identifikasi Masalah: | | |
| <p>OPERATOR ARITMATIKA.</p> <p>Contoh 1: Salin dan tempel kode program berikut ke Eclipse atau JDoodle</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: a: 20 b: 3 a + b = 23</p> <p>Berdasarkan kode tersebut, diberikan pertanyaan berupa:</p> <ol style="list-style-type: none"> 1. Susun kode Java untuk perhitungan dengan ekspresi $(2*3 + 6 / 2 - 4)$. 2. Simpulkan urutan prioritas operator yang dijalankan ekspresi tersebut! | | |
| [No.1] Analisis dan Argumentasi | | |
| <ol style="list-style-type: none"> 1) Untuk nomor satu, saya menyusun kode baru yang dapat menampilkan ekspresi $(2*3 + 6 / 2 - 4)$. di eclipse dengan mengubah kode yang sudah ada dengan deklarasi dan inisialisasi kode baru yang sesuai dengan perintah yang diberikan. Karena ekspresi yang diminta tidak dapat ditampilkan jika menggunakan susunan kode yang lama. Ekspresi tersebut juga tidak dapat ditampilkan langsung, tetapi perlu dideklarasikan dan diinisialisasi terlebih dahulu. 2) Untuk soal nomor dua, saya menganalisa cara kerja java dalam penyelesaian operasi hitung yang telah dideklarasikan dan diinisialisasi melalui hasil dan urutan operasi matematika. Diketahui bahwa hasil operasi dengan ekspresi $(2*3 + 6 / 2 - 4)$. adalah '5' sehingga diketahui bahwa java mengurutkan operasi aritmatika dengan hirarki penghitungan itu sendiri. Di mana perkalian dan pembagian dikerjakan terlebih dahulu, serta penambahan dan pengurangan dikerjakan kemudian dari bagian yang paling kiri. | | |
| [No.1] Penyusunan Algoritma dan Kode Program | | |
| <ol style="list-style-type: none"> 1) Algoritma <ol style="list-style-type: none"> a) Mulai b) Buat nama kelas java 'Praktikumsatu'. c) Tambahkan main method 'public static void main(String[] args){ d) Deklarasikan dan inisialisasi nilai a, b, c, dan d pada tipe data int. e) Deklarasikan dan inisialisasi perhitungan $(a * b + c / a - d)$ pada tipe data int. f) Tampilkan hasil perhitungan. g) Akhiri program. 2) Tuliskan kode program dan luaran | | |

```
Praktikumsatu.java x
1 package operatorAritmatika;
2
3 public class Praktikumsatu { //menentukan nama kelas
4     public static void main( String[] args ){ //menentukan main method
5
6         //menggunakan tipe data int untuk bilangan bulat
7         int a = 2, b = 3; //deklarasi variabel dengan nilai yang diinginkan
8         int c = 6, d = 4; //deklarasi variabel dengan nilai yang diinginkan
9
10        int hasil = (a * b + c / a - d); //menyimpan data perhitungan
11        System.out.println("Hasil dari perhitungan: " + hasil); //menampilkan hasil perhitungan
12    }
13 }
14 }
```

```
Console x
<terminated> Praktikumsatu [Java Application] C:\Users\HP\p2\poo\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1
Hasil dari perhitungan: 5
```

Analisa luaran: kode berhasil diubah mengikuti perintah yang diberikan pada soal Latihan 1. Luaran yang dihasilkan berhasil menampilkan hasil perhitungan tanpa adanya error.

[No.1] Kesimpulan

1) Analisa

a) Kesimpulan

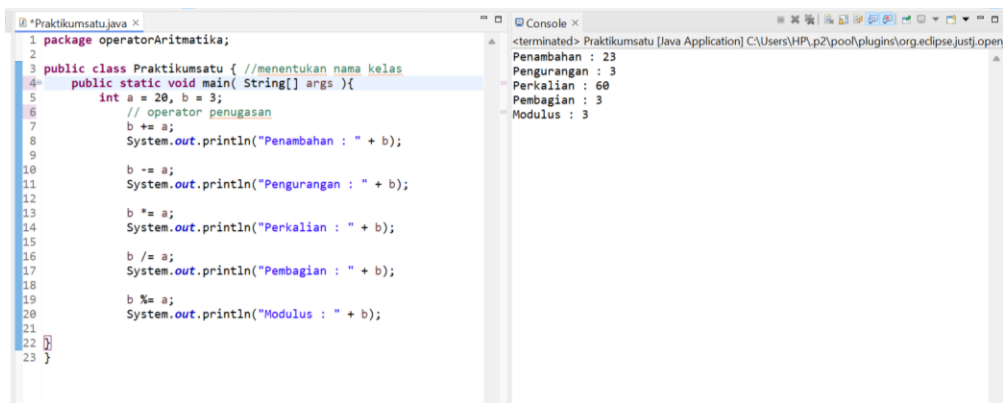
Kode java yang diberikan tidak memiliki masalah yang signifikan, tetapi diperlukan beberapa perubahan pada kode sesuai dengan arahan yang diberikan oleh soal. Untuk itu saya mengubah kode semula awalnya menjadi kode yang bisa mengerjakan perhitungan operasi aritmatika dan menampilkan hasil sesuai dengan urutan perhitungan yang benar yaitu perhitungan perkalian dan pembagian didahulukan, dilanjutkan dengan perhitungan penambahan dan pengurangan dengan catatan pengerjaan dilakukan mulai dari bagian yang paling kiri.

b) Dasar Pengambilan Keputusan

Dasar pengambilan keputusan saya untuk mengubah kode awal adalah agar ekspresi yang ingin dikerjakan dan ditampilkan dapat dikerjakan oleh java tanpa adanya error. Perlu adanya penyusunan yang tepat untuk penempatan nilai data pada variable yang tepat agar menghasilkan luaran yang tidak keliru.

| | | |
|--|----------------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| HANIFAH AZIZAH G1F024037 | OPERATOR JAVA | 05/09/2024 |
| [No.2] Identifikasi Masalah: | | |
| <p>OPERATOR PENUGASAN.</p> <p>Contoh 2: Salin dan tempel kode program berikut ke Eclipse atau JDoodle</p> <pre>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); } }</pre> <p>Luaran: Penambahan : 23</p> <p>Berdasarkan kode tersebut, diberikan pertanyaan berupa:</p> <ol style="list-style-type: none"> 1. Susunlah kode untuk menampilkan perhitungan dengan operator (-=, *=, /=, %=)! 2. Simpulkan hasil dari kode tersebut! | | |
| [No.2] Analisis dan Argumentasi | | |
| <ol style="list-style-type: none"> 1. Untuk nomor satu, saya menyusun kode baru yang dapat menampilkan ekspresi (-=, *=, /=, %=) di eclipse dengan menambah kode yang sudah ada dengan deklarasi dan inisialisasi kode baru yang sesuai dengan perintah yang diberikan. Kode terbaru yang ingin ditampilkan berupa operator penugasan. 2. Untuk soal nomor dua, saya menganalisa kode yang akan dibuat dengan topik utama operator penugasan. Di mana susunan-susunan kode tersebut menampilkan perhitungan dengan operator pengurangan, perkalian, pembagian, dan modulus sehingga didapatkan luaran berupa: Penambahan : 23 Pengurangan : 3 Perkalian : 60 Pembagian : 3 Modulus : 3 | | |
| [No.2] Penyusunan Algoritma dan Kode Program | | |
| <ol style="list-style-type: none"> 1. Algoritma <ol style="list-style-type: none"> h) Mulai i) Buat nama kelas java 'Praktikumsatu'. j) Tambahkan main method 'public static void main(String[] args){ '. k) Deklarasikan dan inisialisasi nilai a = 20 dan b = 3. l) Deklarasikan operasikan penambahan. m) Lanjutkan deklarasi untuk operasi lainnya n) Tampilkan hasil perhitungan. o) Akhiri program. | | |

2. Tuliskan kode program dan luaran



The screenshot shows the Eclipse IDE with a Java file named `*Praktikumsatu.java` and a console window. The Java code defines a package `operatorAritmatika` and a class `Praktikumsatu` with a `main` method. The code initializes `a = 20` and `b = 3`, then performs five operations: addition, subtraction, multiplication, division, and modulus, printing the results. The console output shows the results of these operations: `Penambahan : 23`, `Pengurangan : 3`, `Perkalian : 60`, `Pembagian : 3`, and `Modulus : 3`.

```
1 package operatorAritmatika;
2
3 public class Praktikumsatu { //menentukan nama kelas
4     public static void main( String[] args ){
5         int a = 20, b = 3;
6         // operator penugasan
7         b += a;
8         System.out.println("Penambahan : " + b);
9
10        b -= a;
11        System.out.println("Pengurangan : " + b);
12
13        b *= a;
14        System.out.println("Perkalian : " + b);
15
16        b /= a;
17        System.out.println("Pembagian : " + b);
18
19        b %= a;
20        System.out.println("Modulus : " + b);
21
22    }
23 }
```

<terminated> Praktikumsatu [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.open
Penambahan : 23
Pengurangan : 3
Perkalian : 60
Pembagian : 3
Modulus : 3

Analisa luaran: kode berhasil ditambahkan dengan mengikuti perintah operasi yang diinginkan pada soal Latihan 2. Luaran yang dihasilkan berhasil menampilkan hasil perhitungan operator penugasan penambahan, pengurangan, perkalian, pembagian, dan modulus tanpa adanya error.

[No.2] Kesimpulan

Analisa

a) Kesimpulan

Kode yang ditampilkan dari soal tidak memiliki masalah tetapi membutuhkan penambahan kode baru sehingga menampilkan hasil luaran yang terdapat di analisis dan argumentasi. Operator penugasan yang ada berupa simbol pengurangan, perkalian, penambahan, pembagian, dan modulus.

b) Dasar Pengambilan Keputusan

Dasar pengambilan keputusan saya untuk menambah kode awal adalah agar ekspresi yang ingin ditampilkan dapat diluncurkan oleh java tanpa adanya error dan sesuai dengan kode yang diinginkan. Perlu adanya penyusunan yang tepat untuk penempatan nilai data pada variable yang tepat agar menghasilkan luaran yang tidak keliru.

| | | |
|-------------------------------------|----------------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| HANIFAH AZIZAH G1F024037 | OPERATOR JAVA | 12/09/2024 |

[No.3] Identifikasi Masalah:

OPERATOR RELASIONAL.

Contoh 3: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

```
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

Berdasarkan kode tersebut, diberikan pertanyaan berupa:

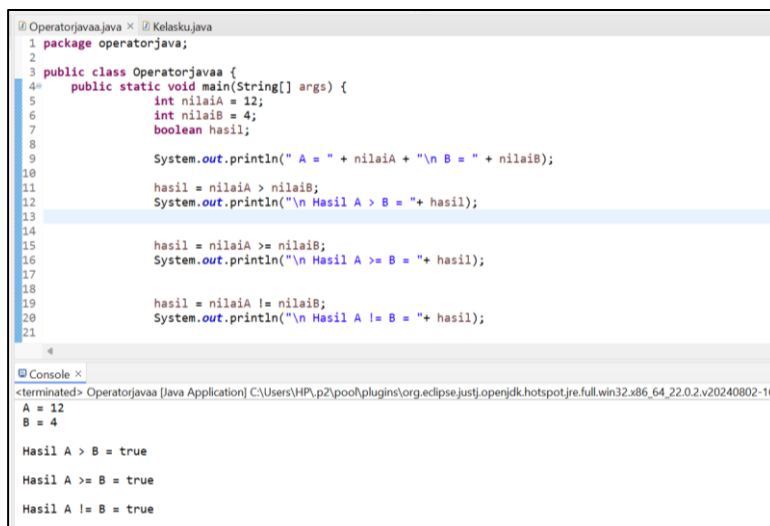
1. Susunlah perintah kode dengan operator relasional (<, >, <=, >=, =, ==, !=) untuk nilai a dan b yang menghasilkan luaran TRUE!

[No.3] Analisis dan Argumentasi

1. Soal nomor tiga menampilkan kode operator relasional sehingga saya menyusun kode yang hanya menampilkan kode relasional berluaran True. Ketika salah satu operator relasional berluaran True digunakan, maka operator relasional lainnya akan mengeluarkan luaran True berdasarkan nilai data pada variable yang dimasukkan.

[No.3] Penyusunan Algoritma dan Kode Program

1. Algoritma
 - a) Mulai
 - b) Buat nama kelas java 'Operatorjavaa'.
 - c) Tambahkan main method '**public static void** main(String[] args){ '.
 - d) Deklarasikan variabel nilaiA dan nilaiB dengan nilai masing-masing 12 dan 4.
 - e) Deklarasikan variabel boolean hasil.
 - f) ampilkan nilai dari nilaiA dan nilaiB di layar.
 - g) Hitung nilaiA > nilaiB.
 - h) Simpan hasil perhitungan ke dalam variabel hasil.
 - i) Tampilkan hasil hasil dengan pesan "Hasil A > B = ".
 - j) Hitung nilaiA > nilaiB.
 - k) Simpan hasil perhitungan ke dalam variabel hasil.
 - l) Tampilkan hasil hasil dengan pesan "Hasil A > B = ".
 - m) Hitung nilaiA > nilaiB.
 - n) Simpan hasil perhitungan ke dalam variabel hasil.
 - o) Tampilkan hasil hasil dengan pesan "Hasil A > B = ".
 - p) Akhiri program.
2. Tuliskan kode program dan luaran



```
1 package operatorjava;
2
3 public class Operatorjavaa {
4     public static void main(String[] args) {
5         int nilaiA = 12;
6         int nilaiB = 4;
7         boolean hasil;
8
9         System.out.println(" A = " + nilaiA + "\n B = " + nilaiB);
10
11         hasil = nilaiA > nilaiB;
12         System.out.println("\n Hasil A > B = " + hasil);
13
14
15         hasil = nilaiA >= nilaiB;
16         System.out.println("\n Hasil A >= B = " + hasil);
17
18
19         hasil = nilaiA != nilaiB;
20         System.out.println("\n Hasil A != B = " + hasil);
21     }
22 }
```

Console Output:

```
<terminated> Operatorjavaa [Java Application] C:\Users\HP\p2\poo\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-16
A = 12
B = 4

Hasil A > B = true
Hasil A >= B = true
Hasil A != B = true
```

Analisa luaran: luaran yang dihasilkan menampilkan True karena sesuai dengan perintah operator relasional. Luarannya yang dihasilkan berhasil ditampilkan tanpa menunjukkan error syntax.

[No.3] Kesimpulan

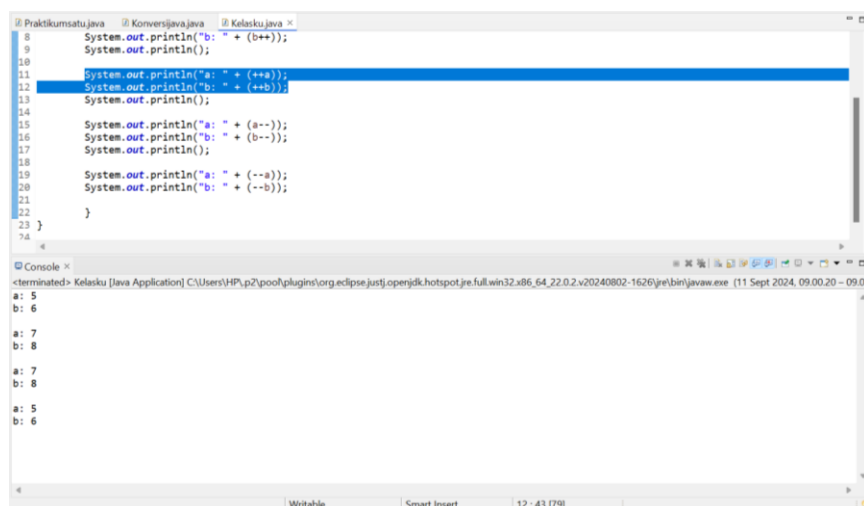
- Analisa
 - a) Kesimpulan
Kode yang ditampilkan merupakan kode operator relasional dengan hasil luaran perbandingan True atau False tergantung dengan nilai variabel dan operator relasional yang digunakan.
 - b) Dasar Pengambilan Keputusan
Dasar pengambilan keputusan saya untuk membuat kode tersebut adalah sesuai dengan permintaan soal yang hanya ingin menampilkan luaran True dengan operator relasional.

| | | |
|---|----------------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| HANIFAH AZIZAH G1F024037 | OPERATOR JAVA | 11/09/2024 |
| [No.4] Identifikasi Masalah: | | |
| <p>OPERATOR INCREMENT DECREMENT.</p> <p>Contoh 4: Salin dan tempel kode program berikut ke Eclipse atau JDoodle</p> <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>Berdasarkan kode tersebut, diberikan pertanyaan berupa:</p> <ol style="list-style-type: none"> 1. Susunlah kode program untuk menghasilkan luaran nilai a = 5 dan b = 6 dengan pre/post increment dan pre/post decrement. 2. Simpulkan hasil perbandingan Anda (pre/post increment, pre/post decrement)! | | |
| [No.4] Analisis dan Argumentasi | | |
| <ol style="list-style-type: none"> 1. Untuk nomor satu, saya menyusun kode yang dapat menampilkan luaran berupa a=5 dan b=6 menggunakan pre/post decrement dan pre/post increment. Pada kode di baris 7 dan 8 saya membuat kode yang menampilkan post-increment untuk variable a dan b sehingga menghasilkan luaran berupa a: 5 dan b: 6. <p>Selanjutnya pada baris 11 dan 12 saya mengubah posisi operator increment menjadi pre-increment sehingga menghasilkan variable a: 7 dan b: 8. Perubahan luaran ini terjadi karena ketika mencetak variable a dengan post-increment maka nilai asli a akan dicetak terlebih dahulu baru ditambah 1. Variable selanjutnya akan menyimpan nilai a=6.</p> <p>Sehingga ketika kode berikutnya diberikan berupa pre-increment, maka nilai a yang telah ditambah 1 (a=6) akan ditambahkan satu lagi dan menghasilkan luaran a: 7.</p> <p>Begitupula dengan variable b yang pertamakali diberikan kode post-increment maka akan ditampilkan nilai aslinya terlebih dahulu. Kemudian variable akan menyimpan nilai b yang telah ditambah satu (b=7). Selanjutnya pada penempatan kode pre-increment, luaran akan menampilkan nilai variable yang sebelumnya disimpan sebagai b=7 ditambahkan lagi dengan satu, menghasilkan b=8.</p> <p>Pada baris 15 dan 16, saya membuat kode post-decrement sehingga luaran yang ditampilkan berupa variable yang belum dikurangi 1 (a= 7 dan b= 8) hasil dari kode post-increment yang dilakukan sebelumnya.</p> <p>Setelahnya variable akan otomatis menyimpan nilai a dan b yang telah dikurangi satu (a=6 dan b=7). Sehingga ketika saya memasukkan kode pre-increment di baris 19 dan 20 maka luaran akan kembali menghasilkan nilai a=5 dan b=6 karena itu adalah nilai yang telah dikurangi satu.</p> | | |

2. Hasil perbandingan yang saya lakukan memberikan kesimpulan bahwa post-increment menampilkan nilai asli pada luaran sebelum ditambahkan satu dan menyimpan secara otomatis nilai yang telah ditambah, sedangkan pre-increment menampilkan nilai variable setelah ditambahkan satu pada luaran.
- Dan post-decrement menampilkan nilai awal variable sebelum dikurang satu dan menyimpan secara otomatis nilai yang telah dikurang, sedangkan pre-decrement menampilkan luaran berupa variable yang telah dikurangi satu.

[No.4] Penyusunan Algoritma dan Kode Program

1. Algoritma
 - a) Mulai program.
 - b) Buat nama kelas java 'Kelaskujava'
 - c) Tambahkan main method '**public static void main(String[] args)**'.
 - d) Masukkan variable a dengan nilai 5 dan variable b dengan nilai 6 dengan tipe data int.
 - e) Tampilkan luaran post-increment untuk variabel a.
 - f) Tampilkan luaran post-increment untuk variable b
 - g) Tampilkan luaran spasi.
 - h) Tampilkan luaran pre-increment untuk variable a.
 - i) Tampilkan luaran pre-increment untuk variabel b.
 - j) Tampilkan luaran spasi.
 - k) Tampilkan luaran post-decrement untuk variabel a.
 - l) Tampilkan luaran post-decrement untuk variabel b
 - m) Tampilkan luaran spasi.
 - n) Tampilkan luaran pre-decrement untuk variable a
 - o) Tampilkan luaran pre-decrement untuk variabel b.
 - p) Akhiri program.
2. Tuliskan kode program dan luaran



```
8      System.out.println("b: " + (b++));
9      System.out.println();
10
11     System.out.println("a: " + (++a));
12     System.out.println("b: " + (++b));
13     System.out.println();
14
15     System.out.println("a: " + (a--));
16     System.out.println("b: " + (b--));
17     System.out.println();
18
19     System.out.println("a: " + (--a));
20     System.out.println("b: " + (--b));
21
22 }
23
24
```

Console:

```
<terminated> Kelasku [Java Application] C:\Users\HP\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\java.exe (11 Sept 2024, 09:00:20 - 09:00)
a: 5
b: 6

a: 7
b: 8

a: 7
b: 8

a: 5
b: 6
```

Analisa luaran: kode berhasil disusun dengan mengikuti perintah operasi yang diinginkan pada soal Latihan 4 berupa materi pre/post increment dan pre/post decrement. Luaran yang dihasilkan berhasil menampilkan nilai variable hasil perhitungan operator post-increment, pre-increment, post-decrement, dan pre-decrement sesuai dengan karakteristik operator itu sendiri tanpa adanya error syntax di bagian luaran.

[No.4] Kesimpulan

Analisa

a) Kesimpulan

Kode yang ditampilkan dari soal memungkinkan saya untuk melanjutkan kode menjadi susunan kode yang menampilkan susunan operator pre/post increment dan pre/post decrement. Diketahui bahwa hasil luaran yang dihasilkan pre-increment berbeda dengan post-increment karena perintah yang berikan berbeda. Begitupula pre-decrement dan post-decrement sebagai pengurangan.

b) Dasar Pengambilan Keputusan

Dasar pengambilan keputusan saya untuk membuat susunan kode tersebut adalah karakteristik penyimpanan nilai pre/post increment dan pre/post decrement yang menghasilkan luaran berbeda sesuai dengan urutan penempatan operator.

| | | |
|--|----------------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| HANIFAH AZIZAH G1F024037 | OPERATOR JAVA | 11/09/2024 |
| [No.5] Identifikasi Masalah: | | |
| <p>OPERATOR BOOLEAN LOGIKA.</p> <p>Contoh 5: Salin dan tempel kode program berikut ke Eclipse atau JDoodle</p> <pre>public class operator { public static void main(String[] args) { // deklarasi nilai boolean a = true; boolean b = false; System.out.println("Hasil logika (a && b) : " + (a && b)); } }</pre> <p>Luaran: Hasil logika (a && b) : false</p> <p>Berdasarkan kode tersebut, diberikan pertanyaan berupa:</p> <ol style="list-style-type: none"> 1. Susun kode program dengan mengubah nilai a dan b untuk menghasilkan luaran true dengan operator && dan operator . 2. Simpulkan hasil kode tersebut! | | |
| [No.5] Analisis dan Argumentasi | | |
| <ol style="list-style-type: none"> 1. Kode awal yang diberikan pada soal menunjukkan kode dengan luaran 'false' untuk operator logika &&. Sehingga untuk menampilkan luaran 'true' dengan operator logika && perlu mengganti data pada variable b dengan 'true'. Karena operator logika && akan berhenti mengevaluasi kode jika salah satu dari kodenya menampilkan 'false'. Otomatis luaran akan menghasilkan 'false', jadi variable a dan b harus berisi 'true' agar menghasilkan luaran true. <p>Kode dengan operator logika , untuk menghasilkan luaran 'true' mempunyai dua cara. Yaitu: dengan menetapkan nilai variable a sebagai true dan b sebagai true; selanjutnya menetapkan variable a sebagai true dan b sebagai false. Ketika kedua variable bernilai true maka luaran dengan operator akan menghasilkan true. Namun ketika hanya salah satu variable yang bernilai true, maka hasil luaran akan tetap menampilkan true karena operator logika akan menghentikan evaluasi kondisi jika salah satu ekspresi bernilai benar. Jika salah satu kondisi ekspresi benar, dipastikan hasil operasi logika bernilai benar.</p> <ol style="list-style-type: none"> 2. Operator logika && dan menghasilkan luaran logika yang berbeda tergantung dengan nilai yang dimasukkan ke variable itu sendiri dan juga operator logika yang digunakan. | | |
| [No.5] Penyusunan Algoritma dan Kode Program | | |
| <ol style="list-style-type: none"> 1. Algoritma <ol style="list-style-type: none"> a) Mulai program. b) Buat nama kelas java 'Kelaskujava' c) Tambahkan main method public static void main(String[] args){'. d) Deklarasikan dan inisialisasi tipe data Boolean dengan variable a dengan data true. e) Deklarasikan dan inisialisasi tipe data Boolean dengan variable b dengan data true. f) Tampilkan nilai hasil logika &&. g) Deklarasikan dan inisialisasi tipe data Boolean dengan variable a dengan data true. h) Deklarasikan dan inisialisasi tipe data Boolean dengan variable b dengan data true. i) Tampilkan nilai hasil logika . | | |

- j) Deklarasikan dan inialisasi tipe data Boolean dengan variable a dengan data true.
- k) Deklarasikan dan inialisasi tipe data Boolean dengan variable b dengan data false.
- l) Tampilkan nilai hasil logika `||`.
- m) Akhiri program.

2. Tuliskan kode program dan luaran

```
1 public class Kelasku { //nama kelas dan nama file Kelasku.java
2     public static void main(String[] args) { //deklarasi method utama
3         boolean a = true;
4         boolean b = true;
5
6         System.out.println("Hasil logika (a && b) : " + (a && b));
7
8     }
9 }
10
11
12
13
```

Console x

<terminated> Kelasku [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-162

Hasil logika (a && b) : true

```
1 public class Kelasku {
2     public static void main(String[] args) {
3         boolean a = true;
4         boolean b = true;
5
6         System.out.println("Hasil logika (a && b) : " + (a || b));
7
8     }
9 }
10
11
12
13
```

Console x

<terminated> Kelasku [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-162

Hasil logika (a && b) : true

```
1 public class Kelasku {
2     public static void main(String[] args) {
3         boolean a = true;
4         boolean b = false;
5
6         System.out.println("Hasil logika (a && b) : " + (a || b));
7
8     }
9 }
10
11
12
13
```

Console x

<terminated> Kelasku [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-162

Hasil logika (a && b) : true

Analisa luaran: kode berhasil disusun dengan mengikuti perintah operasi yang diinginkan pada soal Latihan 5 dengan menggunakan operator logika `&&` dan `||`. Serta menghasilkan luaran true dengan nilai variable yang berbeda agar menghasilkan true. Luarannya berhasil menampilkan true tanpa adanya error.

| |
|---|
| [No.5] Kesimpulan |
| <p>Analisa</p> <p>a) Kesimpulan</p> <p>Operator logika && dan memiliki karakteristik yang berbeda dalam menentukan luaran. Operator logika && tidak akan menghasilkan luaran true jika salah satu variable berisi nilai false. Sedangkan operator logika akan menghasilkan luaran true walaupun salah satu dari variable berisi nilai false.</p> <p>b) Dasar Pengambilan Keputusan</p> <p>Dasar pengambilan keputusan saya untuk membuat susunan kode tersebut adalah perbedaan karakteristik dari masing-masing operator logika sehingga harus adanya perubahan dari nilai variable.</p> |

| | | |
|--|----------------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| HANIFAH AZIZAH G1F024037 | OPERATOR JAVA | 11/09/2024 |
| [No.6] Identifikasi Masalah: | | |
| <p>OPERATOR KONDISIONAL/TERNARY.</p> <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>Berdasarkan kode tersebut, diberikan pertanyaan berupa:</p> <ol style="list-style-type: none"> 1. Susun kode program! Dengan informasi berikut: Diketahui nama variabel Jam = 12 Apabila jam < 12 maka tampil "Selamat Pagi", apabila jam > 12 maka tampil "Selamat Malam". | | |
| [No.6] Analisis dan Argumentasi | | |
| <ol style="list-style-type: none"> 1. Diketahui bahwa soal meminta saya untuk menyusun kode program dengan operator ternary (kondisional) di mana pada kasus ini apabila jam < 12 maka tampil "Selamat Pagi", dan apabila jam > 12 maka tampil "Selamat Malam". Sehingga saya mengubah kode yang telah deprogram sebelumnya menjadi kode program yang menampilkan hasil dari kondisi sesuai dengan yang diinginkan karena ternary merupakan fungsi if, else yang telah disederhanakan dengan hanya melakukan satu perintah. | | |
| [No.6] Penyusunan Algoritma dan Kode Program | | |
| <ol style="list-style-type: none"> 1. Algoritma <ol style="list-style-type: none"> a) Mulai program. b) Buat nama kelas java 'Operatorjavaa' c) Tambahkan main method 'public static void main(String[] args){ d) Deklarasikan variabel status bertipe String dan inialisasi dengan string kosong (""). e) Deklarasikan variabel jam bertipe int dan inialisasi dengan nilai 12. f) Gunakan operator kondisional ternary untuk menentukan nilai dari status: g) Jika jam lebih besar dari 12, maka set status menjadi "Selamat Pagi". h) Jika jam tidak lebih besar dari 12, maka set status menjadi "Selamat Malam". i) Tampilkan hasil luaran. j) Akhiri program. | | |

2. Tuliskan kode program dan luaran



```
1 package operatorjava;  
2  
3 public class Operatorjava {  
4     public static void main(String[] args) {  
5         String status = "";  
6         int jam = 12;  
7         status = (jam > 12)?"Selamat Pagi":"Selamat Malam";  
8         System.out.println(status);  
9     }  
10 }  
11
```

Console x
<terminated> Operatorjava [Java Application] C:\Users\HP\p2\poo\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe
Selamat Malam

Analisa luaran: kode program berhasil disusun untuk perintah kondisional (ternary) dengan melihat jam yang disimpan pada tipe data int. Ketika status menunjukka jam > 12 maka luaran akan menghasilkan “Selamat Malam”, dan begitupula sebaliknya. Maka didapatkan hasil sesuai dengan perintah soal yang diberikan tanpa adanya error syntax pada program yang disusun.

[No.6] Kesimpulan

Analisa

a) Kesimpulan

Operator ternary atau kondisional memungkinkan program untuk melaksanakan perintah berupa hanya satu statement yang merupakan bentuk penyederhanaan dari fungsi if, else. Kode operator ternary tidak hanya bisa untuk membuat hasil lulus atau gagal sesuai nilai variable patokan, tetapi juga dapat diubah menjadi bentuk ekspresi lain sesuai kebutuhan.

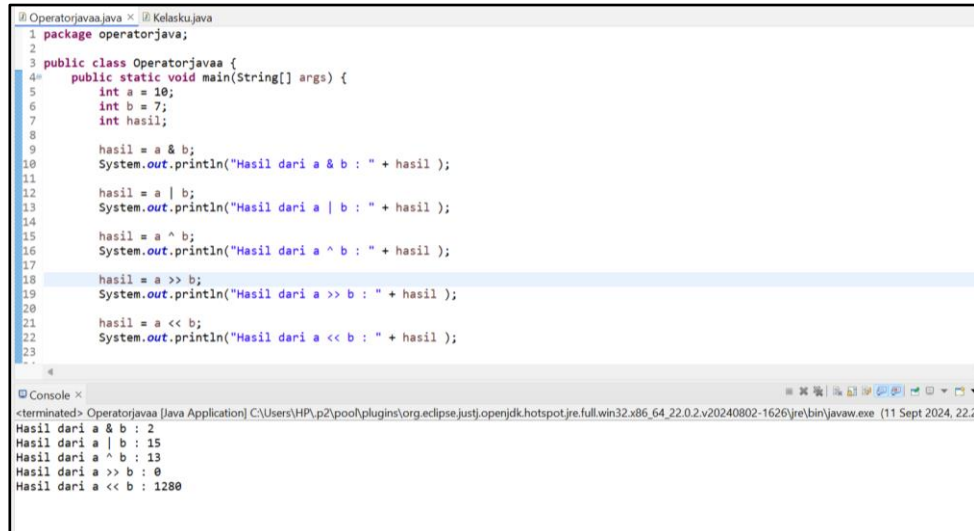
c) Dasar Pengambilan Keputusan

Dasar pengambilan keputusan saya untuk membuat susunan kode tersebut adalah berupa penggunaan ternary dalam kode java memungkinkan penulisan ekspresi kondisional dalam bentuk singkat dan lebih mudah dibaca.

| | | |
|---|----------------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| HANIFAH AZIZAH G1F024037 | OPERATOR JAVA | 11/09/2024 |
| [No.7] Identifikasi Masalah: | | |
| <p>OPERATOR BITWISE.</p> <p>Contoh 7: Salin dan tempel kode program berikut ke Eclipse atau JDoodle</p> <pre>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); } }</pre> <p>Luaran:</p> <pre>Hasil dari a & b : 6 Hasil dari a b : 7 Hasil dari a ^ b : 1</pre> <p>Berdasarkan kode tersebut, diberikan pertanyaan berupa:</p> <ol style="list-style-type: none"> 1. Susun kode tambahan dari Contoh 7 untuk melakukan perhitungan dengan operator (>>, <<). 2. Hubungkan hasil luaran dengan perhitungan manual bilangan biner! | | |
| [No.7] Analisis dan Argumentasi | | |
| <ol style="list-style-type: none"> 1. Soal yang diberikan mengharuskan saya untuk menambahkan kode dengan kode operator bitwise right and left shift berupa (>> dan <<). Untuk itu saya menambahkan baris-baris kode baru sesuai permintaan soal dan didapatkan hasil luaran untuk operator bitwise right shift (>>) berupa 0, dan hasil luaran untuk operator bitwise left shift (<<) berupa 1280. 1280 dalam bilangan biner berupa 10100000000. 2. Hal ini dapat terjadi karena right shift menggeser bit-bit dari variable a ke kanan sebanyak b posisi sehingga bit yang terbuang di sebelah kanan akan dihapus, dan bit-bit baru di sebelah kiri akan diisi dengan bit tanda (0 untuk positif, 1 untuk negatif). Kemudian left shift menggeser bit-bit dari variable a ke kiri sebanyak b posisi sehingga bit-bit baru yang muncul di sebelah kanan akan diisi dengan 0, sedangkan bit-bit yang terbuang di sebelah kiri akan dihapus. | | |
| [No.7] Penyusunan Algoritma dan Kode Program | | |
| <ol style="list-style-type: none"> 1. Algoritma <ol style="list-style-type: none"> a) Mulai program. b) Buat nama kelas java 'Operatorjavaa' c) Tambahkan main method 'public static void main(String[] args){ '. d) Deklarasikan int a dan inisialisasi dengan nilai 10. e) Deklarasikan int b dan inisialisasi dengan nilai 7. f) Hitung a & b dan tampilkan hasilnya. g) Hitung a b dan tampilkan hasilnya. h) Hitung a ^ b dan tampilkan hasilnya. | | |

- i) Hitung $a \gg b$ dan tampilkan hasilnya.
- j) Hitung $a \ll b$ dan tampilkan hasilnya.
- k) Akhiri program.

2. Tuliskan kode program dan luaran



```
1 package operatorjava;
2
3 public class Operatorjava {
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 >> b;
19        System.out.println("Hasil dari a >> b : " + hasil);
20
21        hasil = a << b;
22        System.out.println("Hasil dari a << b : " + hasil);
23    }
24 }
```

Console Output:

```
<terminated> Operatorjava [Java Application] C:\Users\HP\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\java.exe (11 Sept 2024, 22.2)
Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
Hasil dari a >> b : 0
Hasil dari a << b : 1280
```

Analisa luaran: Kode yang ada pada soal yang diberikan tambahan operator bitwise right shift dan operator bitwise left shift sehingga terjadi pergeseran bit-bit dari sebuah bilangan ke arah kanan atau kiri tergantung operator bitwise shift mana yang digunakan. Luarannya yang dihasilkan menampilkan nilai 0 untuk right shift karena variabel a tergeser ke kanan sebanyak b , begitupun dengan nilai 1280 untuk left shift karena variabel a tergeser ke kiri sebanyak b .

[No.7] Kesimpulan

Analisa

a) Kesimpulan

Operator bitwise memiliki operator right shift dan left shift yang berfungsi untuk menggeser bit-bit data ke kanan ataupun ke kiri dan operator bitwise right shift left shift ini sangat berguna dalam situasi jika memerlukan pembagian cepat, manipulasi bit, atau bekerja pada level rendah dengan data.

d) Dasar Pengambilan Keputusan

Dasar pengambilan keputusan saya untuk membuat susunan kode tersebut adalah karena operator bitwise right shift dan left shift mampu menggeser posisi bit-bit dalam biner secara efektif.

Refleksi

Pengalaman belajar yang saya dapatkan selama minggu materi operator ini adalah banyaknya ilmu baru yang berguna dalam proses saya untuk mengerjakan kode program java dan susunan-susunan kode program tersebut memungkinkan saya untuk mengeksplor lebih banyak pengetahuan baru. Walaupun awalnya saya merasa bingung karena operator-operator tersebut baru saya pahami ketika mengerjakan soal-soal yang diberikan dan tentunya dengan bantuan teman-teman dan sumber yang relevan yang membantu saya dalam menentukan jawaban yang tepat.

Tantangan yang saya hadapi pada minggu ini adalah saya menargetkan kepada diri saya untuk tidak menunda-nunda sehingga saya memaksa diri saya untuk memahami soal-soal dan materi yang diberikan. Walau begitu, akhirnya saya dapat memahami soal yang diberikan namun tidak serratus persen. Saya bertekad untuk lebih memahami kembali materi dan soal-soal yang diberikan.

Pada minggu ini saya mendapatkan soal tentang operator aritmatika, operator penugasan, operator relasional, operator increment dan decrement, operator Boolean logika, operator kondisional (ternary), dan operator bitwise.

)