

Nama & NPM	Topik:	Tanggal:
Ahmad Deedadz Asyaufi G1F024049	Operator Aritmatika	12-September2024
[Nomor Soal] Identifikasi Masalah:		
<p>1) Uraikan permasalahan dan variable</p> <p>Permasalahan yang diajukan adalah melakukan perhitungan pada ekspresi aritmatika ($2 * 3 + 6 / 2 - 4$) dan memahami urutan prioritas operator yang digunakan pada ekspresi tersebut.</p> <p>Variabel yang digunakan: Tidak ada variabel eksplisit untuk perhitungan, namun nilai ekspresi disimpan langsung dalam variabel <code>hasil</code>.</p> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>https://youtu.be/PzCMZOboxZM?si=DmK2CqNooDljXqQh</p>		
[Nomor Soal] Analisis dan Argumentasi		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Program menggunakan ekspresi aritmatika ($2 * 3 + 6 / 2 - 4$) dan menghitung hasilnya dengan mengikuti aturan prioritas operator. Hasilnya dicetak ke layar, dan setiap tahap perhitungan dijelaskan untuk memperjelas urutan eksekusi.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Urutan eksekusi dari ekspresi mengikuti aturan prioritas operator dalam Java: Operasi perkalian ($*$) dan pembagian ($/$) dikerjakan terlebih dahulu. Operasi penjumlahan ($+$) dan pengurangan ($-$) dikerjakan setelah operasi perkalian dan pembagian.</p>		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<p>1) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> Langkah 1: Deklarasi variabel <code>hasil</code> untuk menyimpan hasil dari ekspresi. Langkah 2: Hitung ekspresi ($2 * 3 + 6 / 2 - 4$) sesuai dengan urutan prioritas operator aritmatika. Langkah 3: Tampilkan hasil perhitungan dari ekspresi. Langkah 4: Tampilkan urutan eksekusi setiap operasi dalam ekspresi tersebut. <p>2) Tuliskan kode program dan luaran</p>		

```

1 public class OperatorAritmatika{
2     public static void main(String[] args) {
3         // ekspresi
4         int hasil = 2 * 3 + 6 / 2 - 4;
5
6         // output hasil
7         System.out.println();
8         System.out.println("Hasil dari ekspresi (2 * 3 + 6 / 2 - 4) adalah: " + hasil);
9         System.out.println();
10        System.out.println("Urutan Eksekusi dari Ekspresi (2 * 3 + 6 / 2 - 4) + (2 * 3 + 6 / 2 - 4));
11        System.out.println("2 * 3 = 6");
12        System.out.println("6 / 2 = 3");
13        System.out.println("6 + 3 = 9");
14        System.out.println("9 - 4 = 5");
15    }
16 }

```

```

PS C:\Users\didad> & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\didad\AppData\Local\Temp\vscodeses_40f8e\jdt_ws\jdt.ls-java-project\bin' 'OperatorAritmatika'

Hasil dari ekspresi (2 * 3 + 6 / 2 - 4) adalah: 5

Urutan Eksekusi dari Ekspresi (2 * 3 + 6 / 2 - 4)5
2 * 3 = 6
6 / 2 = 3
6 + 3 = 9
9 - 4 = 5
PS C:\Users\didad>

```

[Nomor Soal] Kesimpulan

1) Evaluasi

a) Apa konsekuensi dari skenario pemrograman ini?

Operator aritmatika memiliki prioritas yang menentukan urutan eksekusi perhitungan dalam suatu ekspresi. Operasi perkalian dan pembagian lebih prioritas dibandingkan operasi penjumlahan dan pengurangan.

Input berupa angka konstanta dalam ekspresi dihitung secara berurutan sesuai dengan aturan prioritas operator, yang menghasilkan luaran akhir 5.

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Ahmad Deedadz Asyaufi G1F024049	Operator Penugasan	12-September-2024
[Nomor Soal] Identifikasi Masalah:		
<p>3) Uraikan permasalahan dan variable</p> <p>Permasalahan yang diajukan adalah bagaimana menggunakan operator penugasan (penambahan <code>+=</code>, pengurangan <code>-=</code>, perkalian <code>*=</code>, pembagian <code>/=</code>, dan modulus <code>%=</code>) dalam perhitungan dengan variabel integer.</p> <p>Variabel yang digunakan: a: Inisialisasi dengan nilai 20. b: Inisialisasi dengan nilai 3.</p> <p>4) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>https://youtu.be/PzCMZOboxZM?si=WYJ76gwOET6KYeUX</p>		
[Nomor Soal] Analisis dan Argumentasi		
<p>3) Uraikan rancangan solusi yang diusulkan.</p> <p>Solusi ini melibatkan penggunaan operator penugasan untuk melakukan operasi aritmetika pada variabel b dengan operand a. Program ini menunjukkan hasil dari penggunaan operator <code>-=</code>, <code>*=</code>, <code>/=</code>, dan <code>%=</code>.</p> <p>4) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Operator penugasan menggabungkan operasi aritmetika dengan penugasan nilai, yang memudahkan pengeditan nilai variabel dalam satu pernyataan. Ini memberikan cara yang lebih ringkas dan efisien untuk melakukan perhitungan dan pembaruan nilai variabel.</p>		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<p>3) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none">Langkah 1: Inisialisasi variabel a dengan nilai 20 dan variabel b dengan nilai 3.Langkah 2: Gunakan operator penugasan <code>-=</code> untuk mengurangi nilai a dari b dan tampilkan hasilnya.Langkah 3: Reset nilai b dan gunakan operator penugasan <code>*=</code> untuk mengalikan nilai a dengan b dan tampilkan hasilnya.Langkah 4: Reset nilai b dan gunakan operator penugasan <code>/=</code> untuk membagi nilai b dengan a dan tampilkan hasilnya.Langkah 5: Reset nilai b dan gunakan operator penugasan <code>%=</code> untuk menghitung modulus dari b dengan a dan tampilkan hasilnya.		

4) Tuliskan kode program dan luaran

```
1 public class OperatorPenugasan {
2     public static void main(String[] args) {
3         // deklarasi nilai
4         int a = 20, b = 3;
5
6         System.out.println();
7
8         // operator penugasan dengan -=
9         System.out.println("Operator Penugasan Dengan -=");
10        b -= a;
11        System.out.println("Pengurangan (b -= a): " + b);
12        b = 3;
13        System.out.println("b = 3, a = 20");
14        System.out.println("b -= a -> b = b - a -> b = 3 - 20");
15        System.out.println("b = -17");
16        System.out.println();
17
18
19
20
21        // operator penugasan dengan *=
22        System.out.println("Operator Penugasan Dengan *=");
23        b *= a;
24        System.out.println("Perkalian (b *= a): " + b);
25        b = 3;
26        System.out.println("b = 3, a = 20");
27        System.out.println("b *= a -> b = b * a -> b = 3 * 20");
28        System.out.println("b = 60");
29        System.out.println();
30
31
32
33        // operator penugasan dengan /=
34        System.out.println("Operator Penugasan Dengan /=");
35        b /= a;
36        System.out.println("Pembagian (b /= a): " + b);
37        b = 3;
38        System.out.println("b = 3, a = 20");
39        System.out.println("b /= a -> b = b / a -> b = 3 / 20");
40        System.out.println("b = 0.15");
41        System.out.println();
42
43        // operator penugasan dengan %=
44        System.out.println("Operator Penugasan Dengan %=");
45        b %= a;
46        System.out.println("Modulus (b %= a): " + b);
47        b = 3;
48        System.out.println("b = 3, a = 20");
49        System.out.println("b %= a -> b = b % a -> b = 3 % 20");
50        System.out.println("b = 3");
51        System.out.println();
52    }
53 }
54
```

```
PS C:\Users\didad> & "C:\Program Files\Java\jre-1.8\bin\java.exe" "-cp" "C:\Users\didad\AppData\Local\Temp\vscodesw_c74a7\jdk_w\jdt.ls-java-project\bin" "OperatorPenugasan"

Operator Penugasan Dengan -=
Pengurangan (b -= a): -17
b = 3, a = 20
b -= a -> b = b - a -> b = 3 - 20
b = -17

Operator Penugasan Dengan *=
Perkalian (b *= a): 60
b = 3, a = 20
b *= a -> b = b * a -> b = 3 * 20
b = 60

Operator Penugasan Dengan /=
Pembagian (b /= a): 0
b = 3, a = 20
b /= a -> b = b / a -> b = 3 / 20
b = 0.15

Operator Penugasan Dengan %=
Modulus (b %= a): 3
b = 3, a = 20
b %= a -> b = b % a -> b = 3 % 20
b = 3

PS C:\Users\didad>
```

[Nomor Soal] Kesimpulan

2) Kreasi

- a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?

Menggunakan operator penugasan dapat menyederhanakan penulisan kode untuk manipulasi nilai variabel.

- b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

Operator penugasan memperlihatkan bagaimana operasi aritmetika dapat dilakukan dalam satu langkah sekaligus melakukan penugasan nilai baru ke variabel.

Nama & NPM	Topik:	Tanggal:
Ahmad Deedadz Asyaufi G1F024049	Operator Relasional	12-September-2024
[Nomor Soal] Identifikasi Masalah:		
<p>5) Uraikan permasalahan dan variable</p> <p>Permasalahan yang diajukan adalah bagaimana menggunakan operator relasional untuk membandingkan dua nilai dan menghasilkan luaran TRUE. Operator relasional yang digunakan meliputi >, <, >=, <=, ==, dan !=.</p> <p>Variabel yang digunakan: a: Inisialisasi dengan nilai 10. b: Inisialisasi dengan nilai 5.</p> <p>6) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>https://youtu.be/PzCMZObexZM?si=WYJ76gwOET6KYeUX</p>		
[Nomor Soal] Analisis dan Argumentasi		
<p>5) Uraikan rancangan solusi yang diusulkan.</p> <p>Solusi menggunakan operator relasional untuk memeriksa apakah nilai a lebih besar dari, lebih kecil dari, atau sama dengan nilai b. Operator seperti != memeriksa apakah dua nilai tidak sama.</p> <p>6) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Operator relasional penting dalam membuat keputusan logis berdasarkan perbandingan nilai. Dalam program ini, operator tersebut digunakan untuk memeriksa berbagai kondisi relasional antara a dan b, seperti apakah a lebih besar dari b, apakah a sama dengan 10, dan apakah b tidak sama dengan a.</p>		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<p>5) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> Langkah 1: Inisialisasi variabel a dengan nilai 10 dan variabel b dengan nilai 5. Langkah 2: Gunakan operator > untuk memeriksa apakah a lebih besar dari b. Langkah 3: Gunakan operator >= untuk memeriksa apakah a lebih besar atau sama dengan b. Langkah 4: Gunakan operator < dan <= untuk memeriksa apakah b lebih kecil dari a dan lebih kecil atau sama dengan a. Langkah 5: Gunakan operator == untuk m <p>6) Tuliskan kode program dan luaran</p>		

```

1 public class OperatorRelasional {
2     public static void main(String[] args) {
3         int a = 10;
4         int b = 5;
5         boolean hasil;
6
7         System.out.println("a = " + a + "\nb = " + b);
8
9         // apakah a lebih besar dari b?
10        hasil = a > b; // 10 > 5
11        System.out.println("\nApakah a lebih besar dari b? " + hasil);
12
13        // apakah a lebih besar atau sama dengan b?
14        hasil = a >= b; // 10 >= 5
15        System.out.println("\nApakah a lebih besar atau sama dengan b? " + hasil);
16
17        // apakah b lebih kecil dari a?
18        hasil = b < a; // 5 < 10
19        System.out.println("\nApakah b lebih kecil dari a? " + hasil);
20
21        // apakah b lebih kecil atau sama dengan a?
22        hasil = b <= a; // 5 <= 10
23        System.out.println("\nApakah b lebih kecil atau sama dengan a? " + hasil);
24
25        // apakah nilai a sama dengan 10?
26        hasil = a == 10; // a == 10
27        System.out.println("\nApakah nilai a sama dengan 10? " + hasil);
28
29        // apakah nilai b tidak sama dengan a?
30        hasil = b != a; // 5 != 10
31        System.out.println("\nApakah b tidak sama dengan a? " + hasil);
32    }
33 }
34

```

```

PS C:\Users\didad> & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\didad\AppData\Local\Temp\vscodesae_c74a7\jdk_ws\
dt.ls-java-project\bin' 'OperatorRelasional'
a = 10
b = 5

Apakah a lebih besar dari b? true
Apakah a lebih besar atau sama dengan b? true
Apakah b lebih kecil dari a? true
Apakah b lebih kecil atau sama dengan a? true
Apakah nilai a sama dengan 10? true
Apakah b tidak sama dengan a? true
PS C:\Users\didad>

```

[Nomor Soal] Kesimpulan

3) Evaluasi

b) Apa konsekuensi dari skenario pemrograman ini?

Menggunakan operator relasional mempermudah pembuatan keputusan dalam program, seperti pengecekan kondisi dalam struktur kontrol.

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

Input berupa nilai `a` dan `b` diproses melalui berbagai operator relasional untuk memeriksa apakah hasilnya `TRUE` atau `FALSE`, sesuai kondisi yang diuji.

Nama & NPM	Topik:	Tanggal:
Ahmad Deedadz Asyaufi G1F024049	Operator Increment dan Decrement	12-September-2024
[Nomor Soal] Identifikasi Masalah:		
<p>7) Uraikan permasalahan dan variable</p> <p>Permasalahan yang diajukan adalah bagaimana menggunakan operator pre-increment, post-increment, pre-decrement, dan post-decrement untuk menghasilkan keluaran yang sesuai, yaitu $a = 5$ dan $b = 6$.</p> <p>Variabel yang digunakan:</p> <p>8) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>https://youtu.be/PzCMZObexZM?si=WYJ76gwOET6KYeUX</p>		
[Nomor Soal] Analisis dan Argumentasi		
<p>7) Uraikan rancangan solusi yang diusulkan.</p> <p>Solusi yang diusulkan melibatkan penggunaan pre/post increment dan pre/post decrement untuk menunjukkan perbedaan cara operator ini bekerja. Post-increment mengembalikan nilai awal sebelum menaikkan variabel, sedangkan pre-increment menaikkan variabel sebelum mengembalikan nilainya.</p> <p>8) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Pada post-increment ($a++$), nilai b adalah nilai awal a sebelum a ditambah 1. Sedangkan pada pre-increment ($++a$), a bertambah terlebih dahulu, lalu hasilnya diberikan kepada b. Hal yang sama berlaku untuk pre-decrement dan post-decrement dalam penurunan nilai.</p>		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<p>7) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> Langkah 1: Inisialisasi variabel a dengan nilai 5. Langkah 2: Gunakan post-increment ($a++$) untuk menambahkan nilai a setelah penyimpanan ke b. Langkah 3: Reset nilai a menjadi 5, lalu gunakan pre-increment ($++a$) untuk menambahkan a sebelum menyimpan ke b. Langkah 4: Gunakan pre-decrement ($--a$) untuk menurunkan a sebelum menyimpan ke b. Langkah 5: Gunakan post-decrement ($a--$) untuk menurunkan a setelah penyimpanan ke b. <p>8) Tuliskan kode program dan luaran</p>		


```

1 public class OperatorIncrementDanDecrement {
2     public static void main(String[] args) {
3         // variable a awalnya bernilai 5
4         int a = 5;
5         int b;
6
7         // b = a++ -> b = a, a = a + 1
8         b = a++;
9         System.out.println("Post-increment: a = " + a + ", b = " + b);
10
11        // variable a di reset menjadi 5
12        a = 5;
13
14        // b = ++a -> a = a + 1, b = a
15        b = ++a;
16        System.out.println("Pre-increment: a = " + a + ", b = " + b);
17
18        // b = --a -> a = a - 1, b = a
19        b = --a;
20        System.out.println("Pre-decrement: a = " + a + ", b = " + b);
21
22        // b = a-- -> b = a, a = a - 1
23        b = a--;
24        System.out.println("Post-decrement: a = " + a + ", b = " + b);
25    }
26 }
27
28

```

```

b2 C:/n26a2/q1q9q9
p02f-q0c1eaw0f: 9 = 5* p = 2
p16-q0c1eaw0f: 9 = 2* p = 2
p16-t1c1eaw0f: 9 = e* p = e
p02f-t1c1eaw0f: 9 = e* p = 2
q1f-j2-j9a9-bu0f0c1f/p1u, ,0b0a101t1c1eaw0f0a0p0c1eaw0f,
b2 C:/n26a2/q1q9q9 g ,C:/p10b1u t1j02/j9a9/j16-j18/p1u/j9a9*0x6, ,.-cb, ,C:/n26a2/q1q9q9/bb0a1f9/10c9j/10a0b/1a20q0a0a2-c1a9a3/jq1m2/l

```

[Nomor Soal] Kesimpulan

4) Kreasi

- c) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?

Operator pre/post increment dan pre/post decrement memberikan fleksibilitas lebih dalam manipulasi nilai variabel dan memungkinkan kontrol yang lebih presisi pada perubahan nilai.

- d) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

Hubungan antara variabel a dan b dengan operator increment dan decrement memberikan pemahaman mendalam tentang bagaimana posisi operator memengaruhi hasil evaluasi variabel.

Nama & NPM	Topik:	Tanggal:
Ahmad Deedadz Asyaufi G1F024049	Operator Boolean	12-September-2024
[Nomor Soal] Identifikasi Masalah:		
<p>9) Uraikan permasalahan dan variable</p> <ul style="list-style-type: none"> Permasalahan yang diajukan adalah menggunakan operator logika <code>&&</code> (AND) dan <code> </code> (OR) untuk menghasilkan keluaran <code>true</code> dengan mengubah nilai variabel boolean <code>a</code> dan <code>b</code>. Variabel yang digunakan: <code>a = true</code> dan <code>b = true</code>: Untuk menghasilkan keluaran <code>true</code> pada operator <code>&&</code>. <code>a = false</code> dan <code>b = true</code>: Untuk menghasilkan keluaran <code>true</code> pada operator <code> </code>. <p>10) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>https://youtu.be/PzCMZObexZM?si=xj pzG-my8TFqDh_t</p>		
[Nomor Soal] Analisis dan Argumentasi		
<p>9) Uraikan rancangan solusi yang diusulkan.</p> <p>Solusi yang diusulkan melibatkan pengujian dua operator logika. Pertama, operator <code>&&</code> yang menguji apakah kedua operand <code>a</code> dan <code>b</code> bernilai <code>true</code>. Kedua, operator <code> </code> yang menguji apakah setidaknya salah satu dari <code>a</code> atau <code>b</code> bernilai <code>true</code>.</p> <p>10) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Dengan operator <code>&&</code>, hasil akan <code>true</code> jika kedua variabel <code>a</code> dan <code>b</code> bernilai <code>true</code>. Pada operator <code> </code>, cukup salah satu variabel bernilai <code>true</code> untuk mendapatkan keluaran <code>true</code>. Program ini memanfaatkan karakteristik dasar operator logika dalam pemrograman.</p>		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<p>9) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> Langkah 1: Inisialisasi variabel boolean <code>a</code> dan <code>b</code> dengan nilai <code>true</code> untuk operator <code>&&</code>. Langkah 2: Gunakan operator <code>&&</code> untuk mengevaluasi kondisi logika dari <code>a</code> dan <code>b</code>. Langkah 3: Ubah nilai variabel <code>a</code> menjadi <code>false</code> dan <code>b</code> tetap <code>true</code> untuk operator <code> </code>. Langkah 4: Gunakan operator <code> </code> untuk mengevaluasi apakah salah satu dari <code>a</code> atau <code>b</code> bernilai <code>true</code>. <p>10) Tuliskan kode program dan luaran</p>		

```

1 public class OperatorBooleanLogika {
2     public static void main(String[] args) {
3         boolean a = true;
4         boolean b = true;
5
6         // Menggunakan operator &&
7         System.out.println("Hasil logika (a && b): " + (a && b)); // true
8
9         a = false;
10        b = true;
11
12        // Menggunakan operator ||
13        System.out.println("Hasil logika (a || b): " + (a || b)); // true
14    }
15 }
16

```

```

PS C:\Users\didad> & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\didad\AppData\Local\Temp\vscodesws_c74a7\jdt_ws\j
dt.ls-java-project\bin' 'OperatorBooleanLogika'
Hasil logika (a && b): true
Hasil logika (a || b): true
PS C:\Users\didad>

```

[Nomor Soal] Kesimpulan

5) Analisa

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

- Operator `&&` menghasilkan `true` ketika kedua operand `a` dan `b` bernilai `true`.
- Operator `||` menghasilkan `true` ketika salah satu dari operand bernilai `true`, meskipun operand lainnya bernilai `false`.
- Dalam contoh kode, kedua kondisi logika ini menghasilkan keluaran `true` sesuai dengan aturan dasar operator logika dalam Java.

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

- Dalam contoh kode, kedua kondisi logika ini menghasilkan keluaran `true` sesuai dengan aturan dasar operator logika dalam Java.

Nama & NPM	Topik:	Tanggal:
Ahmad Deedadz Assyaufie G1F024049	Operator Kondisional	12-September-2024
[Nomor Soal] Identifikasi Masalah:		
<p>11) Uraikan permasalahan dan variable</p> <ul style="list-style-type: none"> Permasalahan yang diajukan adalah menentukan pesan selamat ("Selamat Pagi" atau "Selamat Malam") berdasarkan nilai variabel <code>Jam</code>. Apabila nilai <code>Jam</code> kurang dari 12, maka program akan menampilkan "Selamat Pagi", dan jika lebih dari 12, maka akan menampilkan "Selamat Malam". Variabel yang digunakan: <p><code>Jam = 12</code>: Digunakan sebagai patokan untuk menentukan waktu siang.</p> <p>12) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>https://youtu.be/PzCMZOboxZM?si=xjpzG-my8TFqDh_t</p>		
[Nomor Soal] Analisis dan Argumentasi		
<p>11) Uraikan rancangan solusi yang diusulkan.</p> <p>Solusi yang diusulkan melibatkan penggunaan operator ternary untuk memeriksa apakah <code>Jam</code> lebih kecil dari 12 atau lebih besar dari 12, dan berdasarkan hasil evaluasi tersebut, program akan menampilkan "Selamat Pagi" atau "Selamat Malam".</p> <p>12) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Solusi ini memecahkan masalah sederhana tentang penentuan waktu dengan cepat dan efisien, menggunakan operator ternary untuk membuat kode lebih singkat dibandingkan dengan penggunaan <code>if-else</code>. Hasilnya bergantung pada kondisi <code>Jam</code>, sehingga pesan yang ditampilkan bisa berbeda tergantung pada nilai variabel tersebut.</p>		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<p>11) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> Langkah 1: Inisialisasi variabel <code>Jam</code> dengan nilai tertentu. Langkah 2: Gunakan operator ternary untuk mengevaluasi apakah <code>Jam</code> lebih kecil dari 12. Langkah 3: Jika <code>Jam</code> kurang dari 12, tampilkan "Selamat Pagi"; jika tidak, tampilkan "Selamat Malam". 		

12) Tuliskan kode program dan luaran



```
1 public class OperatorKondisi {
2     public static void main(String[] args) {
3         int jam = 12;
4         String salam = (jam < 12) ? "Selamat Pagi" : (jam > 12) ? "Selamat Malam" : "Selamat Siang";
5         System.out.println(salam);
6     }
7 }
8
```

```
PS C:\Users\didad> & 'C:\Program Files\Java\jre-1.8\bin\java.exe' -cp 'C:\Users\didad\AppData\Local\Temp\vscodeses_c7da7\jdt_ws\jdt.ls-java-proj\
ect\bin' 'OperatorKondisi'
Selamat Siang
PS C:\Users\didad>
```

[Nomor Soal] Kesimpulan

6) Analisa

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

Penggunaan operator ternary terbukti efektif untuk menangani kondisi sederhana seperti ini, membuat kode lebih ringkas dan mudah dibaca. Evaluasi kondisi `Jam` menghasilkan pesan yang sesuai berdasarkan patokan waktu yang diberikan.

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

Keputusan untuk menggunakan operator ternary diambil karena efisiensinya dalam menangani alur logika yang sederhana, dan menghindari penggunaan struktur kontrol yang lebih rumit seperti `if-else`.

Nama & NPM	Topik:	Tanggal:
Ahmad Deedadz Assyaufie G1F024049	Operator Bitwise	12-September-2024
[Nomor Soal] Identifikasi Masalah:		
<p>13) Uraikan permasalahan dan variable</p> <ul style="list-style-type: none"> Permasalahan yang diajukan adalah menambahkan perhitungan menggunakan operator bitwise shift (>> dan <<) pada variabel <code>a</code> dan <code>b</code> dalam kode Java. Variabel yang digunakan: <ul style="list-style-type: none"> <code>a = 10</code> (1010 dalam biner) <code>b = 7</code> (0111 dalam biner) <code>hasil</code> digunakan untuk menyimpan hasil operasi bitwise. <p>14) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>https://youtu.be/PzCMZObexZM?si=xjpzG-my8TFqDh_t</p>		
[Nomor Soal] Analisis dan Argumentasi		
<p>13) Uraikan rancangan solusi yang diusulkan.</p> <p>Solusi melibatkan penggunaan operator bitwise tambahan, yaitu >> untuk pergeseran bit ke kanan dan << untuk pergeseran bit ke kiri. Setiap operasi bitwise akan ditambahkan ke dalam program, diikuti oleh perhitungan manual representasi biner.</p> <p>14) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Dengan menambahkan operator pergeseran bit, kita dapat melihat bagaimana nilai biner berubah saat bit digeser ke kanan atau kiri. Pergeseran ke kanan (>>) mengurangi nilai, sedangkan pergeseran ke kiri (<<) meningkatkan nilai. Penjelasan ini menghubungkan permasalahan pengenalan operator bitwise dengan dasar teori biner dan operasi logika.</p>		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<p>13) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> Langkah 1: Inisialisasi variabel <code>a</code> dan <code>b</code> dengan nilai tertentu. Langkah 2: Lakukan operasi bitwise AND, OR, XOR. Langkah 3: Lakukan operasi pergeseran bit ke kanan (>>) dan ke kiri (<<). Langkah 4: Tampilkan hasil operasi. <p>14) Tuliskan kode program dan luaran</p>		

```

1 public class OperatorBitwise {
2     public static void main(String[] args) {
3         int a = 10; // 1010 dalam biner
4         int b = 7;  // 0111 dalam biner
5         int hasil;
6
7         // Operator AND
8         hasil = a & b;
9         System.out.println("Hasil dari a & b : " + hasil); // Hasilnya adalah 2 (0010)
10
11        // Operator OR
12        hasil = a | b;
13        System.out.println("Hasil dari a | b : " + hasil); // Hasilnya adalah 15 (1111)
14
15        // Operator XOR
16        hasil = a ^ b;
17        System.out.println("Hasil dari a ^ b : " + hasil); // Hasilnya adalah 13 (1101)
18
19        // Operator pergeseran bit kanan >>
20        hasil = a >> 1;
21        System.out.println("Hasil dari a >> 1 : " + hasil); // Hasilnya adalah 5 (0101)
22
23        hasil = b >> 1;
24        System.out.println("Hasil dari b >> 1 : " + hasil); // Hasilnya adalah 3 (0011)
25
26        // Operator pergeseran bit kiri <<
27        hasil = a << 1;
28        System.out.println("Hasil dari a << 1 : " + hasil); // Hasilnya adalah 20 (10100)
29
30        hasil = b << 1;
31        System.out.println("Hasil dari b << 1 : " + hasil); // Hasilnya adalah 14 (1110)
32    }
33 }
34

```

```

PS C:\Users\didad> & "C:\Program Files\Java\jre-1.8\bin\java.exe" "-cp" "C:\Users\didad\AppData\Local\Temp\vscodesw_c\4a7\jdt_ws\jdt.ls-jav
a-project\bin" "OperatorBitwise"
Hasil dari a & b : 2
Hasil dari a | b : 15
Hasil dari a ^ b : 13
Hasil dari a >> 1 : 5
Hasil dari b >> 1 : 3
Hasil dari a << 1 : 20
Hasil dari b << 1 : 14
PS C:\Users\didad>

```

[Nomor Soal] Kesimpulan

- 7) Kreasi
- e) **Pengetahuan baru:** Dengan memahami pergeseran bit, kita bisa mengoptimalkan operasi sederhana dengan kecepatan komputasi yang lebih cepat.
- f) **Hubungan variabel dan konsep:** Hubungan antara pergeseran bit dan perubahan nilai desimal memungkinkan pemrogram memahami cara kerja manipulasi bit pada representasi bilangan biner.

