**Template Lembar Kerja Individu**

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Resyaliana Esa Putri**  **G1A022038** | **Operator** | **9 September 2022** |
| **[Nomor Soal] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable 2. Rincikan sumber informasi yang relevan (buku / webpage) 3. Uraikan rancangan solusi yang diusulkan (jika ada). 4. Analisis susunan solusi, parameter solusi (jika ada). | | |
| **[Nomor Soal] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan. 2. Analisis solusi, kaitkan dengan permasalahan. | | |
| **[Nomor Soal] Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma 2. Tuliskan kode program dan luaran 3. Beri komentar pada kode 4. Uraikan luaran yang dihasilkan 5. Screenshot/ Capture potongan kode dan hasil luaran | | |
| **[Nomor Soal] Kesimpulan** | | |
| 1. Analisa 2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! 3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? 4. Evaluasi 5. Apa konsekuensi dari skenario pemprograman ini? 6. Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada) 7. Kreasi 8. Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi? 9. Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada) | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Resyaliana Esa Putri**  **G1A022038** | **Operator** | **9 September 2022** |

**[No. 1] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

Soal:

**Contoh 1:**Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle  
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));   
}   }  
Luaran:  
a: 20   
b: 3   
a - b = 17

1.1.  Tambahkan baris System.out.println("a + b = + (a + b));  Ubahlah operator ( + ) dengan tanda ( -, \*, /, %)  
1.2.  Analisa perhitungan matematika yang terjadi!

→ Pada soal terdapat kesalahan pada system out printnya dimana seharusnya itu tanda (+) karna yang ingin dihasilkan adalah penjumlahan maka saya mengganti operator (-) menjadi (+) agar sesuai dengan hasil luaran yang diketaui soal.

→ Pada soal kita disuruh untuk menambahkan baris System.out.println("a + b = " + (a + b)); dan mengganti operator (+) menjadi operator aritmatika yang ditentukan (-,\* ,/,%).

1. Rincikan sumber informasi yang relevan (buku / webpage)

→ <https://www.youtube.com/watch?v=PzCMZObexZM>

→ <https://www.youtube.com/watch?v=LcFgl0yrKEw>

**[No.1] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambah baris System.out.println("a + b = "  + (a + b)); dan memasukan operator aritmatika yang sudah ditentukan oleh soal.
2. Alasan solusi ini karena sesuai dengan permintaan soal dan kita dapat menganalisis hasil peritungan matimatika yang terjadi.

**[No.1 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma penambahan operator aritmatika:

1. Buka Eclipse atau Jdoodle yang ada dilaptop atau computer kamu.
2. Tambahkan class.
3. Masukan code yang sudah diberikan.

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));   
}   }

1. Lalu tambahkan  baris System.out.println("a + b = "  + (a + b));
2. Setelah itu ganti operator (+) menjadi (-,\*,/,%)

System.***out***.println("a + b = " + (a + b)); //tambahkan operator penjumlahan +

System.***out***.println("a - b = " + (a - b)); //tambahkan operator pengurangan -

System.***out***.println("a \* b = " + (a \* b)); //tambahkan operator perkalian \*

System.***out***.println("a / b = " + (a / b)); //tambahkan operator pembagian /

System.***out***.println("a % b = " + (a % b)); //tambahkan operator modulus untuk sisa bagi %

1. Apabila sudah sesuai runingkan system dan code kamu sudah selesai.

a + b = 23

a - b = 17

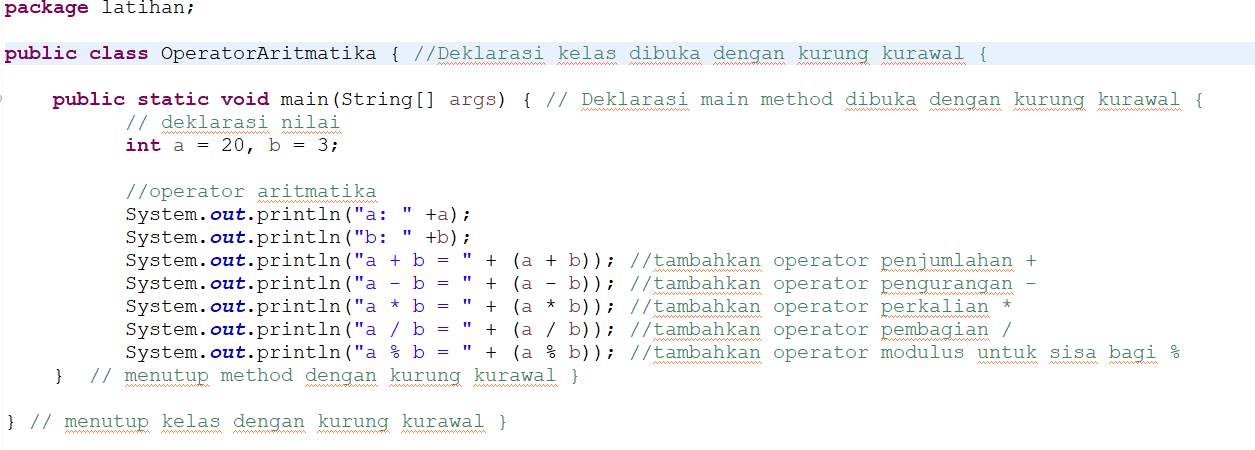
a \* b = 60

a / b = 6

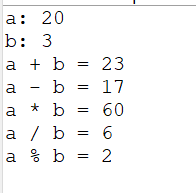
a % b = 2

1. Kode program dan luaran

Kode program :



Luaran :



1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

a adalah 20, b adalah 3 maka luaran yang dihasilkan adalah :

* a + b = 20
* a – b + 17
* a \* b = 60
* a / b = 6
* a % b = 2 (karena 2 merupakan hasil pembagian 20 / 3)

Jadi luaran yang dihasilkan sudah benar dan sesuai yang diinginkan

**[No.1] Kesimpulan** **Analisa**

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

Operator aritmatika merupakan operator yang digunakan untuk melakukan operasi aritmatika (perhitungan) dengan menggunakan operator penjumlahan (+), pengurangan (-), perkalian (\*), pembagian (/), dan sisa (%). Dari data terdebut kita masukan ke dalam soal dan dapat kita analisis hasil yang didapatkan.

Kode pemprograman :

**package** latihan;

**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 );

System.***out***.println("a - b = " + (a - b));

System.***out***.println("a \* b = " + (a \* b));

System.***out***.println("a / b = " + (a / b));

System.***out***.println("a % b = " + (a % b));

} }

Luaran :

a: 20

b: 3

a + b = 23

a - b = 17

a \* b = 60

a / b = 6

a % b = 2

**[No. 2] Identifikasi Masalah:**

1. Uraikan permasalahan dan variable

Soal :

public class OperatorPenugasan {  
    public static void main(String[] args) {  
      // deklarasi nilai  
       int a = 20, b = 3;  
       //operator penugasan   
        b += a;  
        System.out.println("Penambahan : " + b);

        // pengurangan  
        b -= a;  
        System.out.println("Pengurangan : " + b);

        // perkalian  
        b \*= a;  
        System.out.println("Perkalian : " + b);

        // Pembagian  
        b /= a;  
        System.out.println("Pembagian : " + b);

        // Sisa bagi  
        b %= a;  
        // sekarang b=0  
        System.out.println("Sisa Bagi: " + b);  
    }  
}

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

2.1.  Bandingkan hasil Contoh 1 dengan Contoh 2!

Pada contoh 1 di atas sudah terdapat dua variabel yaitu variabel (a) dan variabel (b) dan masing-masing variabel a dan variabel b sudah mempunyai nilai yaitu 20 dan 3 kemudian diterapkan dalam operator aritmatika dan mengolah dua nilai dari variabel a dan variabel b sesuia dengan perintah operator aritmatika masing-masing. Pada operator aritmatika menggunakan operator ( +, -, \*, /, % ).

Hasil pengolahan nilai operator aritmatika pada contoh 1:

|  |  |  |
| --- | --- | --- |
| **Ekspresi** | **Arti** | **Hasil** |
| a + b | ( penjumlahan ) | a + b → 20 + 3 = 20 |
| a – b | ( pengurangan ) | a – b → 20 - 3 = 17 |
| a \* b | ( perkalian ) | a \* b → 20 \* 3 = 60 |
| a / b | ( pembagian ) | a / b → 20 / 3 = 6 |
| a % b | ( modulus ) | a % b → 20 % 3 = 2 |

Pada contoh 2 diatas juga memiliki 2 variabel yaitu (a) dan (b) dengan nilai yang sama seperti contoh 1 yaitu 20 dan 3. Nilai tersebut diterapkan kedalam operator penugasan dan mengolah dua nilai dari variabel a dan variabel b sesuia dengan perintah operator penugasan masing-masing. Pada operator prnugasan menggunakan operator ( a += b, a -= b, a \*= b, a /= b, a %= b, a &= b, a ^= b, dan a <<= b ).

Hasil pengolahan nilai operator penugasan pada contoh 2:

|  |  |  |
| --- | --- | --- |
| **Ekspresi** | **Arti** | **Hasil** |
| b += a | b = b + a ( penjumlahan ) | b = b + a → b = 3 + 20 = 23 |
| b –= a | b = b – a ( pengurangan ) | b = b – a → b = 23 – 20 = 3 |
| b \*= a | b = b \* a ( perkalian ) | b = b \* a → b = 3 \* 20 = 60 |
| b /= a | b = b / a ( pembagian ) | b = b / a → b = 60 / 20 = 3 |
| b %= a | b = b % a ( modulus ) | b = b % a → b = 3 % 20 = 3 |

1. Rincikan sumber informasi yang relevan (buku / webpage)

→ <https://www.youtube.com/watch?v=PzCMZObexZM>

→ <https://www.youtube.com/watch?v=LcFgl0yrKEw>

**[No.2] Analisis dan Argumentasi**

Operator penugasan adalah operator yang menggunakan tanda sama dengan (=) untuk mengisi sebuah nilai dalam suatu variabel. Operator penugasan merupakan operator yang bergabung dengan operator aritmatika (compound assignment). Operator aritmatika adalah operator yang digunakan untuk melakukan operasi aritmatika (perhitungan ). Hasil operasi matematika operator aritmatika akan mengikuti tipe data operand.

Contoh 1 dan contoh 2 merupakan hasil dari pengolahan operator aritmatika dan operator penugasan. Dari nilai diatas terdapat beberapa hasil yang berbeda dikeduanya, itu dikarenakan pada operator penugasan nilai variabel ( b ) dihasilkan dari pengolahan nilai sebelumnya yang mana nilai variable ( b ) pada awalny adalah 20 dan nilai variable ( a ) adalah 3. Setelah nilai tersebut diterapkan kedalam operator penugasan didapatkan hasil seperti berikut :

|  |  |  |
| --- | --- | --- |
| **Ekspresi** | **Arti** | **Hasil** |
| b += a | b = b + a ( penjumlahan ) | b = b + a → b = 3 + 20 = 23 |
| b –= a | b = b – a ( pengurangan ) | b = b – a → b = 23 – 20 = 3 |
| b \*= a | b = b \* a ( perkalian ) | b = b \* a → b = 3 \* 20 = 60 |
| b /= a | b = b / a ( pembagian ) | b = b / a → b = 60 / 20 = 3 |
| b %= a | b = b % a ( modulus ) | b = b % a → b = 3 % 20 = 3 |

Jadi, nilai variable ( b ) didapatkan dari hasil pengolahan sebelumnya dan itulah yang menyebabkan perbedaan hasil yang didapatkan antara operator penugasan dan operator aritmatika.

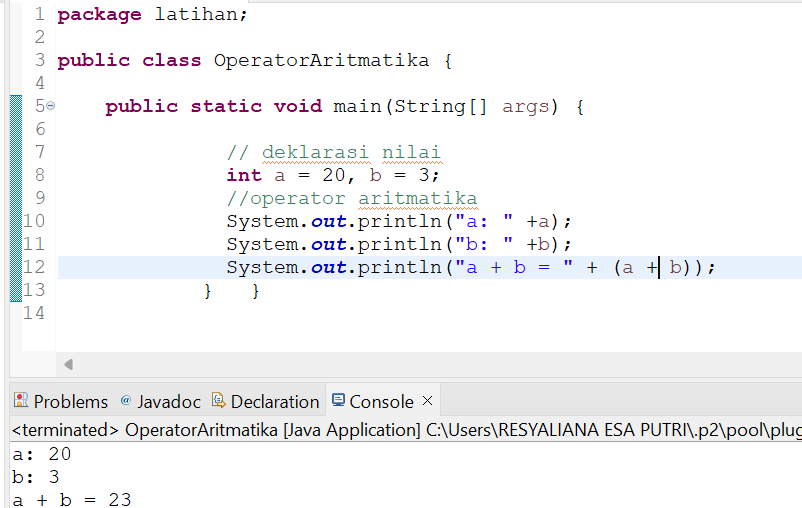
**[No.2] Penyusunan Algoritma dan Kode Program**

1. Algoritma

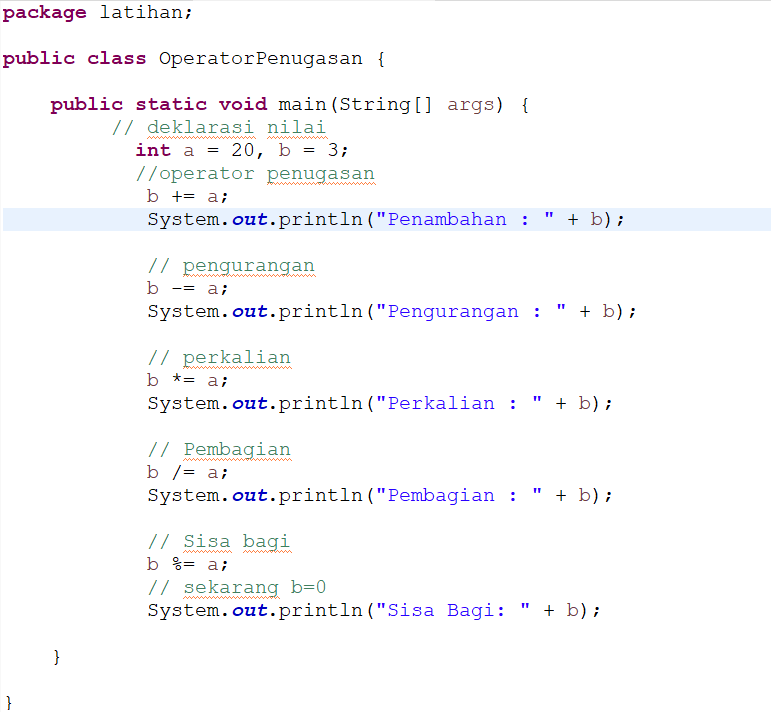
Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma penambahan operator penugasan :

1. Buka Eclipse atau Jdoodle yang ada dilaptop atau computer kamu.
2. Tambahkan class.
3. Masukan code yang sudah diberikan.
4. Setelah itu analisislah
5. Kode program dan luaran

Contoh 1:



Contoh 2:





1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.2] Kesimpulan** **Analisa**

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

Operator penugasan adalah operator yang menggunakan tanda sama dengan (=) untuk mengisi sebuah nilai dalam suatu variabel. Operator penugasan merupakan operator yang bergabung dengan operator aritmatika ( compound assignment ). Operator aritmatika adalah operator yang digunakan untuk melakukan operasi aritmatika ( perhitungan ).Hasil operasi matematika operator aritmatika akan mengikuti tipe data operand.

Contoh 1 dan contoh 2 merupakan hasil dari pengolahan operator aritmatika dan operator penugasan. Dari nilai diatas terdapat beberapa hasil yang berbeda dikeduanya, itu dikarenakan pada operator penugasan nilai variabel ( b ) dihasilkan dari pengolahan nilai sebelumnya yang mana nilai variable ( b ) pada awalny adalah 20 dan nilai variable ( a ) adalah 3. Sedangkan pada operator aritmatika hasil yang didapatkan sesuai dengan data operand. Maka, Itulah yang menyebabkan perbedaan hasil yang didapatkan antara operator penugasan dan operator aritmatika.

**[No. 3] Identifikasi Masalah:**

1. Uraikan permasalahan dan variable

Soal :

public class OperatorRealasional {  
    public static void main(String[] args) {  
        int nilaiA = 12;  
        int nilaiB = 4;  
        boolean hasil;

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

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

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

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

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

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

**Luaran:**   
A = 12  
B = 4

Hasil A > B = true  
Hasil A < B = false  
Hasil A >= B = true  
Hasil A <= B = false  
Hasil A == B = false  
Hasil A != B = true

**Latihan 3**  
3.1.  Ubahlah nilai A = 4 dan B = 4. Analisa perubahan yang terjadi!   
3.2   Bandingkan bagaimana perbedaan nilai A dan B mempengaruhi nilai luaran!

Jika nilai A diubah menjadi 4 maka luaran yang dihasilkan akan ikut berubah. Itu karena operator relasional digunakan untuk membandingkan dua variable. Operator yang digunakan adalah ( ==, !=, <, >, <=, >= ). Dari contoh diatas dapat kita bandingkan perubahan luaran setelah dan sebelum nilai A diubah.

Sebelum :

**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

Sesudah :

**Luaran:**

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

1. Rincikan sumber informasi yang relevan (buku / webpage)

→ <https://www.youtube.com/watch?v=PzCMZObexZM>

→ <https://www.youtube.com/watch?v=LcFgl0yrKEw>

**[No.3] Analisis dan Argumentasi**

Operator relasional adalah operator yang digunakan untuk menguji hubungan antara nilai atau variabel dan selalu menghasilkan nilai true atau false. Seperti pada contoh nilai luaran sesudah dan sebelum diubah menghasilkan pernyataan yang berbeda maka dari situlah dapat disimpulkan bahwa nilai luaran sangat dipengaruhi oleh nilai variable.

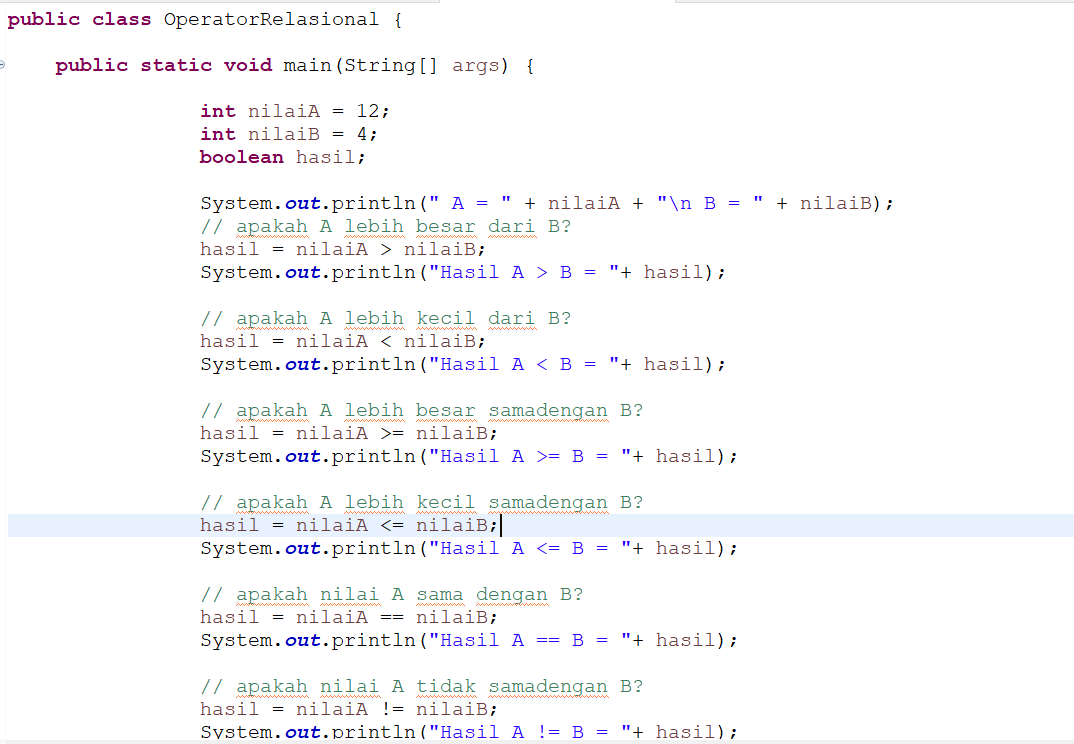
**[No.3] Penyusunan Algoritma dan Kode Program**

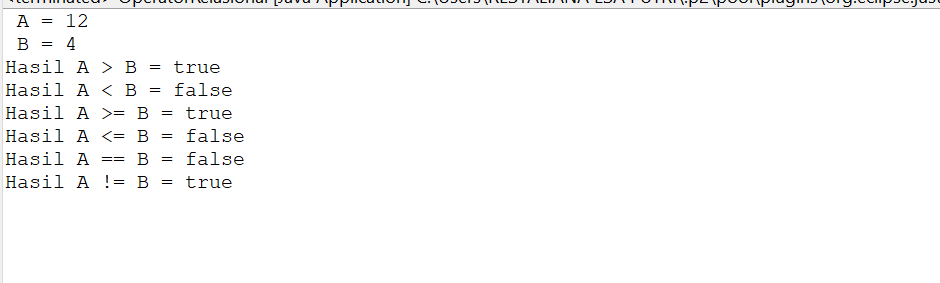
1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma penambahan operator relasional :

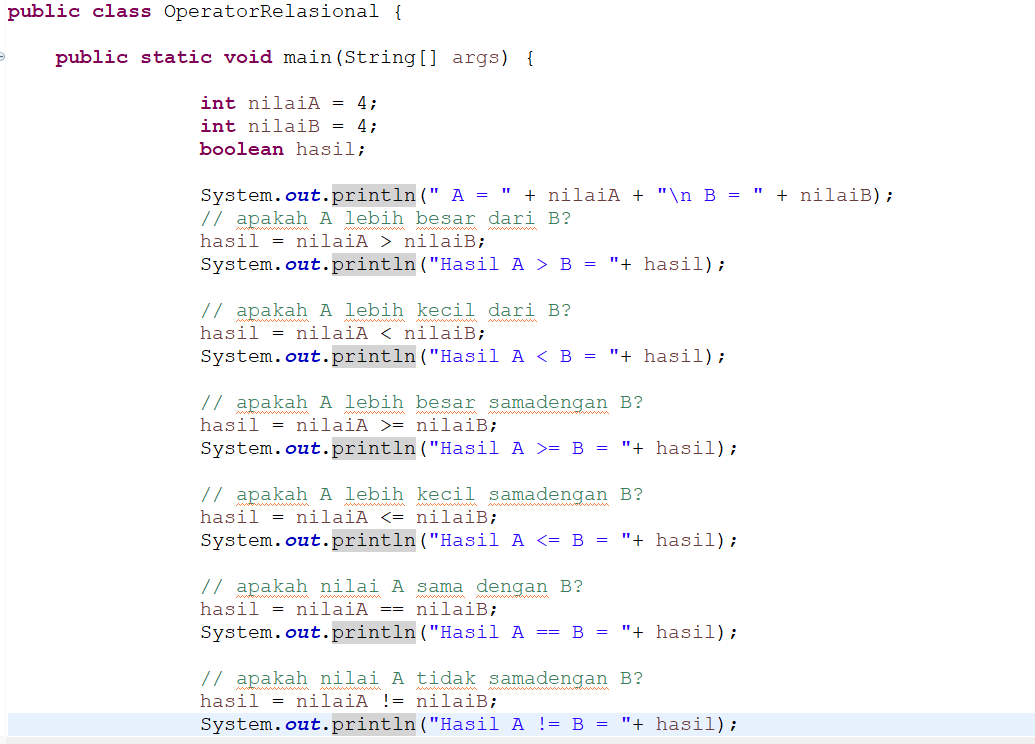
1. Buka Eclipse atau Jdoodle yang ada dilaptop atau computer kamu.
2. Tambahkan class.
3. Masukan code yang sudah diberikan.
4. Setelah itu, ubah nilai A menjadi 4
5. Lalu Bandingkan bagaimana perbedaan nilai A dan B mempengaruhi nilai luaran!
6. Kode program dan luaran

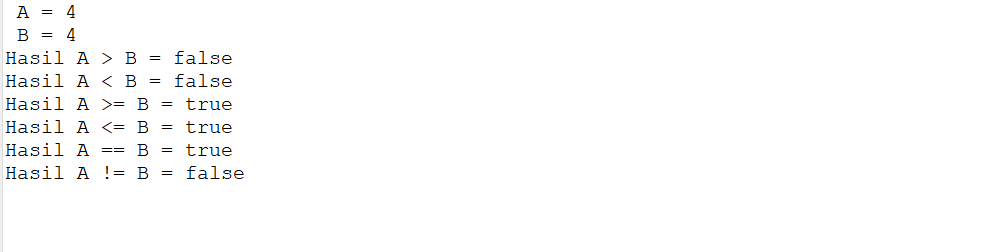
Sebelum :





Sesudah :





1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.3] Kesimpulan** **Analisa**

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

Operator relasional digunakan untuk menguji hubungan antara nilai atau variabel dan selalu menghasilkan nilai true atau false.

Operator yang digunakan :

|  |  |
| --- | --- |
| **Operator** | **Arti** |
| == | Sama dengan |
| != | Tidak sama dengan |
| < | Kurang dari |
| > | Lebih dari |
| <= | Kurang dari atau sama dengan |
| >= | Lebih dari atau sama dengan |

Pada permasalah pada soal telah didapatkan bahwa perbedaan nila A dan B dapat mempengaruhi nilai luaran. Dapat dilihat dari hasil luaran yang sudah didapatkan. pada contoh sebelumnya dimana nilai A adalah 12 dan nilai B adalah 4 menghasilkan luaran seperti berikut :

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

Dari data diatas dapat diuraikan :

|  |  |
| --- | --- |
| **Operasi Relasional** | **Arti** |
| A > B → 12 > 4 = True | Nilai A (12) lebih besar dari nilai B (4) |
| A < B → 12 < 4 = False | Nilai A (12) tidak lebih kecil dari nilai B (4) |
| A >= B → 12 >= 4 = True | Nilai A (12) lebih besar atau sama dengan nilai B (4) |
| A <= B → 12 <= 4 = False | Nilai A (12) lebih kecil atau sama dengan nilai B (4) |
| A == B → 12 == 4 = False | Nilai A (12) sama dengan nilai B (4) |
| A != B → 12 != 4 = True | Nilai A (12) tidak sama dengan nilai B (4) |

Setelah itu ubah nilai A menjadi 4 sehingga menghasilkan luaran seperti berikut :

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

Dari data diatas dapat diuraikan :

|  |  |
| --- | --- |
| **Operasi Relasional** | **Arti** |
| A > B → 4 > 4 = False | Nilai A (4) tidak besar dari nilai B (4) |
| A < B → 4 < 4 = False | Nilai A (4) tidak lebih kecil dari nilai B (4) |
| A >= B → 4 >= 4 = True | Nilai A (4) lebih besar atau sama dengan nilai B (4) |
| A <= B → 4 <= 4 = True | Nilai A (4) lebih kecil atau sema dengan nila B (4) |
| A == B → 4 == 4 = True | Nilai A (4) sama dengan nilai B (4) |
| A != B → 4 != 4 = False | Nilai A (4) tidak sama dengan nilai B (4) |

Jadi dapat disimpulkan bahwa nilai A dan nilai B sangat berpengaruh pada luaran yang dihasilkan pada pengolahan operator relasional.

**[No. 4] Identifikasi Masalah:**

1. Uraikan permasalahan dan variable

Soal :

public class operator {  
    public static void main(String[] args) {  
        int a = 10;   
          System.out.println("# Post Increment #");   
          System.out.println("==================");   
          System.out.println("Isi variabel a: " + a);   
          System.out.println("Isi variabel a: " + a++);  
          System.out.println("Isi variabel a: " + a);  
             
          System.out.println();  
             
          int b = 10;   
          System.out.println("# Pre Increment #");   
          System.out.println("==================");   
          System.out.println("Isi variabel b: " + b);   
          System.out.println("Isi variabel b: " + ++b);  
          System.out.println("Isi variabel b: " + b);   
          
          System.out.println();  
            
          int c = 10;   
          System.out.println("# Post Decrement #");   
          System.out.println("==================");   
          System.out.println("Isi variabel c: " + c);   
          System.out.println("Isi variabel c: " + c--);  
          System.out.println("Isi variabel c: " + c);  
             
          System.out.println();  
             
          int d = 10;   
          System.out.println("# Pre Decrement #");   
          System.out.println("==================");   
          System.out.println("Isi variabel d: " + d);   
          System.out.println("Isi variabel d: " + --d);  
          System.out.println("Isi variabel d: " + d);   
    }  
}  
**Luaran:**  
# Post Increment #  
==================  
Isi variabel a: 10  
Isi variabel a: 10  
Isi variabel a: 11

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

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

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

**Latihan 4.**  
4.1.  Berdasarkan luaran program Contoh 4, bandingkan hasil Post dan Pre untuk Increment dan Decrement!

Pada contoh terdapat 2 operator yaitu increment dan decrement. Increment digunakan untuk menambah nilai variable sebanyak 1 angka. Sedangkan Decrement digunakan untuk mengurangkan nilai variable sebanyak 1 angka.

Hasil yang didapatkan dari luaran operator Increment dan Decrement disesuaikan dengan penggunaan “Pre-“ dan “Post-“. Pre- adalah penambahan yang dilakukan sebelum proses lain dijalankan dengan pemberian tanda di depan ( ++a ). Post- adalah penambahan yang dilakukan setelah proses lain dijalankan dengan pemberian tanda di belakang ( a++ ).

Pada contoh dapat dilihat hasil dari Pre Increment dan Pre Decrement langsung ada sebelum dilakukannya proses lain.

   int b = 10;   
          System.out.println("# Pre Increment #");   
          System.out.println("==================");   
          System.out.println("Isi variabel b: " + b);   
          System.out.println("Isi variabel b: " + ++b);  
          System.out.println("Isi variabel b: " + b);

          int d = 10;   
          System.out.println("# Pre Decrement #");   
          System.out.println("==================");   
          System.out.println("Isi variabel d: " + d);   
          System.out.println("Isi variabel d: " + --d);  
          System.out.println("Isi variabel d: " + d);   
Luaran :

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

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

Sedangkan Post Increment dan Post Decrement dapat dilihat setelah dilakukannya proses lain.

int a = 10;

System.out.println("# Post Increment #");

System.out.println("==================");

System.out.println("Isi variabel a: " + a);

System.out.println("Isi variabel a: " + a++);

System.out.println("Isi variabel a: " + a);

int c = 10;

System.out.println("# Post Decrement #");

System.out.println("==================");

System.out.println("Isi variabel c: " + c);

System.out.println("Isi variabel c: " + c--);

System.out.println("Isi variabel c: " + c);

Luaran :

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

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

1. Rincikan sumber informasi yang relevan (buku / webpage)

→ <https://www.youtube.com/watch?v=PzCMZObexZM>

→ <https://www.youtube.com/watch?v=LcFgl0yrKEw>

**[No.4] Analisis dan Argumentasi**

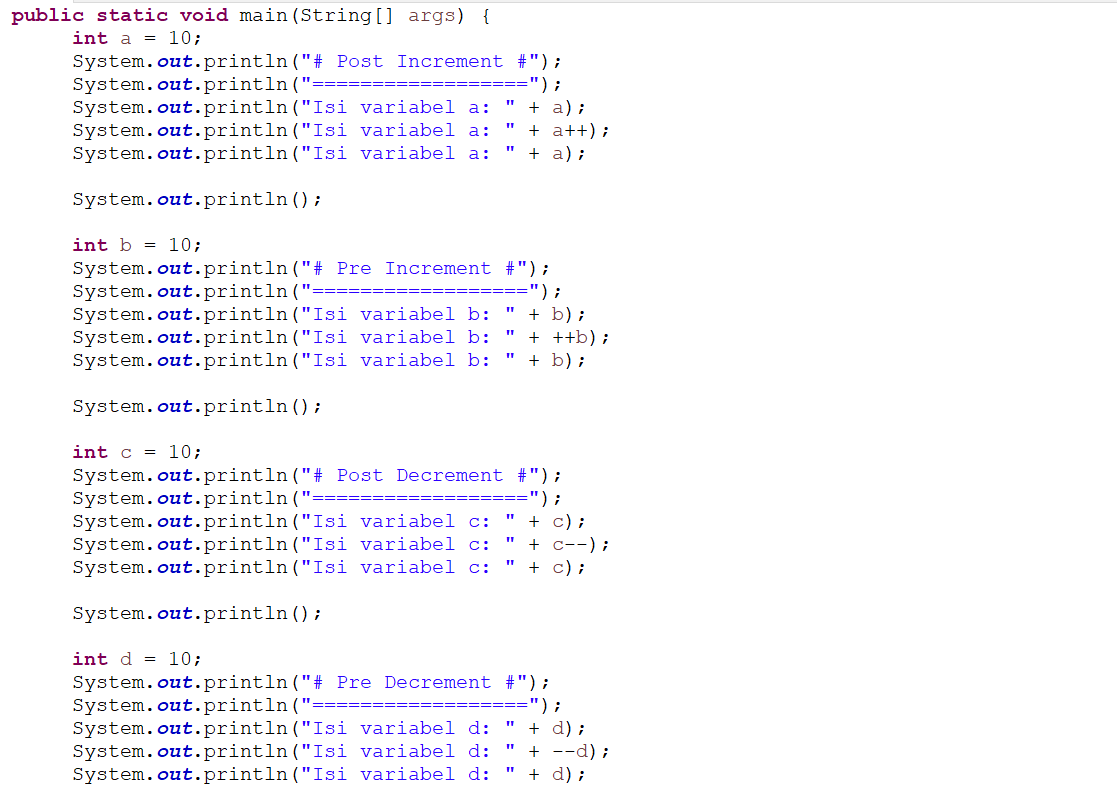
Operator Increment dan Decrement adalah operator yang berfungsi untuk menambahkan atau mengursngi nilai yang berada dalam variable. Penulisan code dengan penambahan yang berada didepan ( ++a ) itu disebut Pre- , hasil dari penggunaan Pre- Increment atau pun Pre- Decrement itu akan langsung ada tanpa harus melakukan proses yang lainnya. Sedangkan, penulisan code dengan penambahan yang berada di belakang itu disebut ( Post- ) pada code ini hasil yang didapatkan tidak dapat langsung keluar, jadi harus melakukan proses lainnya dahulu. Tetapi hasil yang diluarkan akan tetap sama hanya saja melalui tahapan yang berbeda.

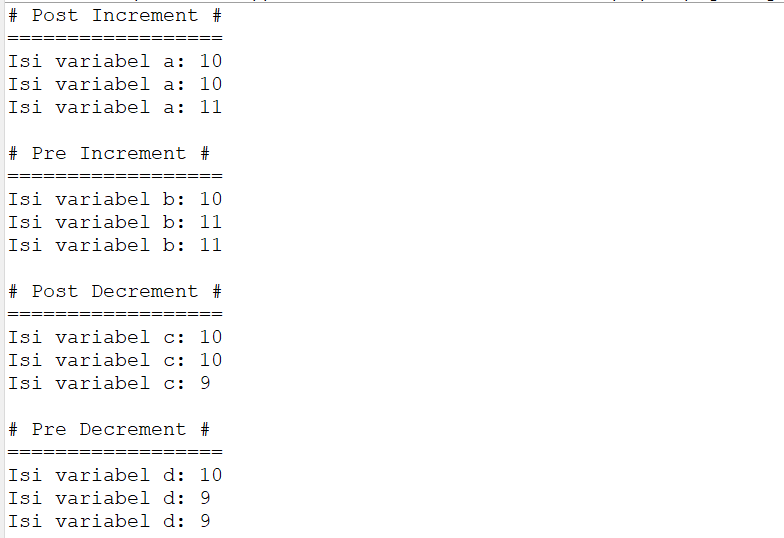
**[No.4] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma penambahan operator Increment dan Decrement.

1. Buka Eclipse atau Jdoodle yang ada dilaptop atau computer kamu.
2. Tambahkan class.
3. Masukan code yang sudah diberikan.
4. Setelah itu bandingkan hasilnya.
5. Kode program dan luaran





1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.4] Kesimpulan** **Analisa**

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

Operator Increment dan Decrement adalah operator yang berfungsi untuk menambahkan atau mengursngi nilai yang berada dalam variable. Kedua operator ini hanya dapat digunakan pada operand bilangan bulat. Penulisan code dengan penambahan atau pengurangan yang berada didepan ( ++a ) itu disebut Pre- , hasil dari penggunaan Pre- Increment atau pun Pre- Decrement itu akan langsung keluar tanpa harus melakukan proses yang lainnya. Sedangkan, penulisan code dengan penambahan atau pengurangan yang berada di belakang itu disebut ( Post- ) pada code ini hasil yang didapatkan tidak dapat langsung keluar, jadi harus melakukan proses lainnya dahulu seperti yang terdapat pada contoh. Tetapi untuk hasil yang diluarkan akan tetap sama hanya saja kedua operator ini melalui tahapan yang berbeda.

**[No. 5] Identifikasi Masalah:**

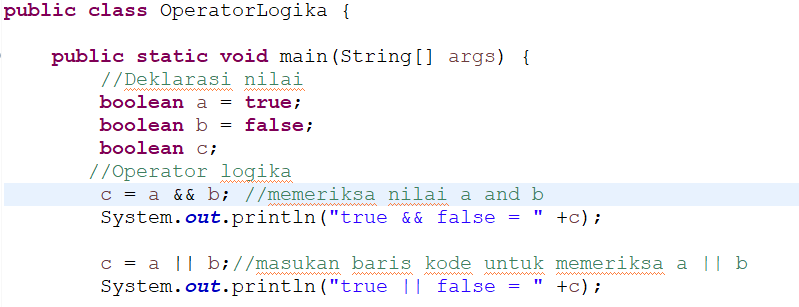
1. Uraikan permasalahan dan variable

Soal :

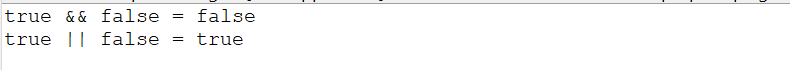
 public class OperatorLogika {  
     public static void main (String [] args) {  
          boolean a = true;  
          boolean b = false;  
          boolean c;  
          c = a && b;  
          System.out.println("true && false = " +c);  
}      }  
  
**Luaran:**  
true && false = false

**Latihan 5**  
5.1.  Tambahkan baris kode untuk memeriksa a || b.

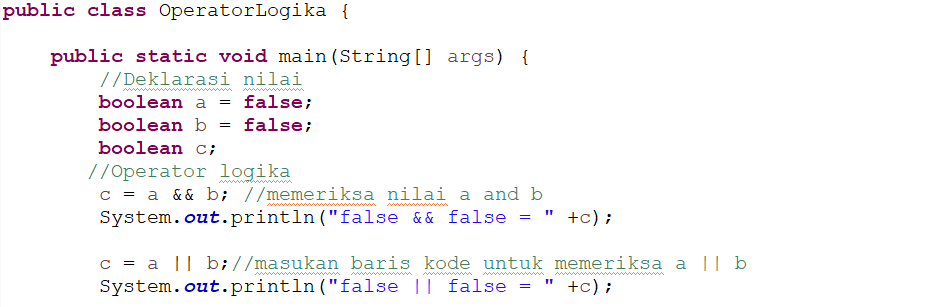
Kode program:



Luaran:

  
5.2.  Ubahlah nilai a = false dan b = false. Analisa perubahan dan perbedaan boolean yang terjadi!

Kode program:



Luaran:



Perbedaan yang tejadi setelah nilai (a) diubah menjadi false itu terdapat pada hasil luaran yang mana pada sebelumnya variable (a) adalah true dan variable (b) adalah false menghasilkan luaran sebagai berikut :

false && false = false

false || false = true

Dan setelah nilai variable (a) diubah menjadi false menghasilkan luaran sebagai berikut:

false && false = false

false || false = false

Perbedaan hasil luaran ini disebabkan karna operasi logika AND (&&) akan menghentikan evaluasi kondisi jika salah satu ekspresi bernilai salah. Jika salah satu kondisi ekspresi salah, dipastikan hasil operasi logika bernilai salah.

Contohnya :

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Y** |
| True | False | False |

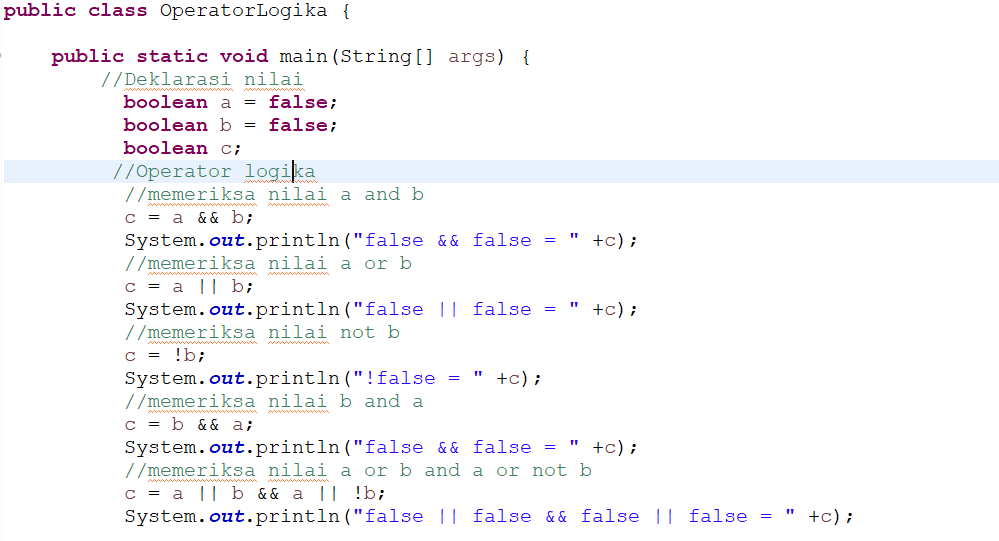
Sedangkan operasi logika OR (||) akan menghentikan evaluasi kondisi jika salah satu ekspresi bernilai benar. Jika salah satu kondisi ekspresi benar, dipastikan hasil operasi logika bernilai benar.

Contohnya:

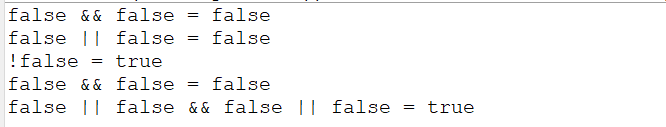
|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Y** |
| True | False | True |

5.3.  Apabila diketahui pernyataan a || b && a || !b. Uraikan urutan logika yang akan dikerjakan! Analisa luaran true atau false dari pernyataan tersebut!

Kode program:



Luaran:



Pada operasi logika a || b && a || !b dapat kita uraikan urutan logikanya sebagai berikut:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **a** | **b** | **a || b** | **a** | **a || b && a** | **!b** | **a || b && a !b** |
| False | False | False | False | False | True | True |

Luaran yang dihasilkan dari operasi logika a || b && a || !b adalah True.

1. Rincikan sumber informasi yang relevan (buku / webpage)

→ <https://www.youtube.com/watch?v=PzCMZObexZM>

→ <https://www.youtube.com/watch?v=LcFgl0yrKEw>

**[No.5] Analisis dan Argumentasi**

Operasi logika merupakan operasi yang menggunakan Boolean sebagai operand.

|  |  |
| --- | --- |
| **Operator** | **Arti** |
| && | AND (TRUE bila kedua operand TRUE) |
| || | OR (TRUE bila kedua operand dan salah satu operand TRUE) |
| ^ | XOR (TRUE bila salah satu operand TRUE) |
| ! | NOT (kebalikan dari input) |

Contoh operasi logika:

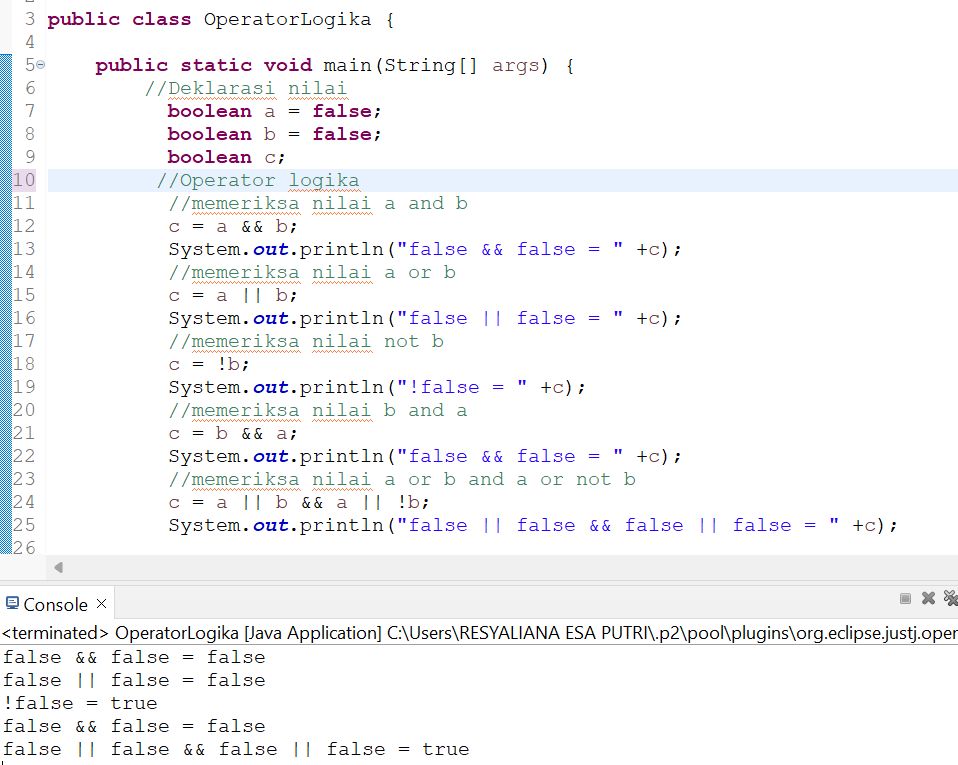
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **A && B** | **A || B** | **A ^ B** | **!A** | **!B** |
| FALSE | FALSE | FALSE | FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | TRUE | TRUE | TRUE | FALSE |
| TRUE | FALSE | FALSE | TRUE | TRUE | FALSE | TRUE |
| TRUE | TRUE | TRUE | TRUE | FALSE | FALSE | FALSE |

**[No.5] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma penambahan operator Increment dan Decrement.

1. Buka Eclipse atau Jdoodle yang ada dilaptop atau computer kamu.
2. Tambahkan class.
3. Masukan code yang sudah diberikan.
4. Tambahkan baris kode a || b dan periksalah.
5. Ubalah nilai a = false dan b = false, analisalah perubahannya.
6. Masukan nilai  a || b && a || !b dan uraikanlah urutannya.
7. Lalu  analisalah luaran true atau false dari pernyataan tersebut!
8. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

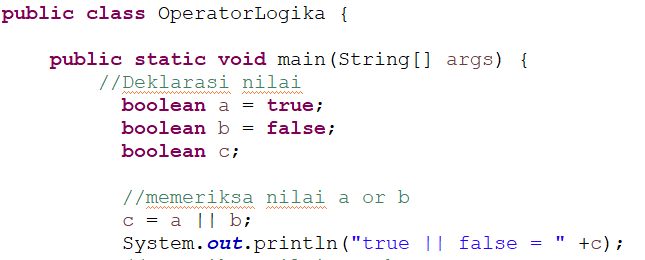
**[No.5] Kesimpulan** **Analisa**

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

Operasi logika merupakan opersasi yang menggunkan Boolean sebagai operand. Operasi logika menggunakan operator AND, OR, XOR, dan NOT. Hasil yang diluarkan oleh setiap operator akan berbeda-beda sesuai dengan artinya masing-masing.

Pada contoh untuk luaran yang dihasilkan oleh soal 5.1 dengan nilai variable a (TRUE) dan variable b (FALSE) digunakan untuk memeriksa a || b (a OR b) menghasilkan luaran TRUE. Sedangkan pada soal 5.2 dengan mengubah nilai variable a menjadi False menghasilkan luaran FALSE. Perbedaan hasil luaran pada operasi logika ini karena operasi logika OR (||) akan menghentikan evaluasi kondisi jika salah satu ekspresi bernilai benar. Jika salah satu kondisi ekspresi benar, dipastikan hasil operasi logika bernilai benar.

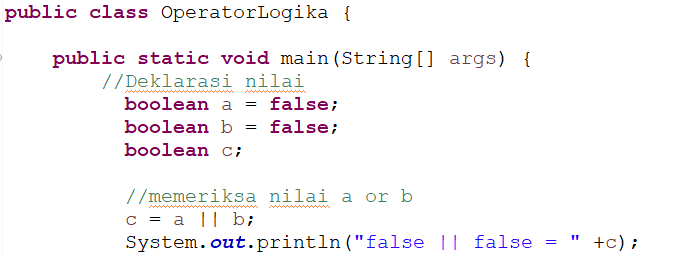
Contoh soal 5.1



Luaran:



Contoh soal 5.2



Luaran:



**[No. 6] Identifikasi Masalah:**

1. Uraikan permasalahan dan variable

Soal:

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

**Latihan 6**  
Berdasarkan Contoh 6, ubahlah nilai = 60. Analisis hasil dan proses yang terjadi!

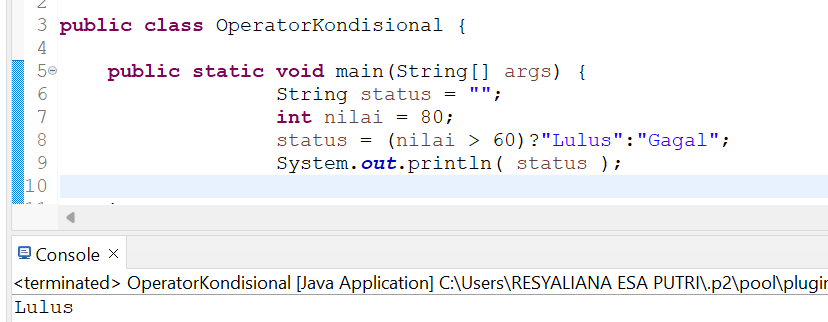
Setelah mengubah nilai int menjadi 60 luaran yang dihasilkan berubah menjadi “GAGAL” karena nilai 60 tidak lebih dari 60 untuk mendapatkan status lulus.

1. Rincikan sumber informasi yang relevan (buku / webpage)

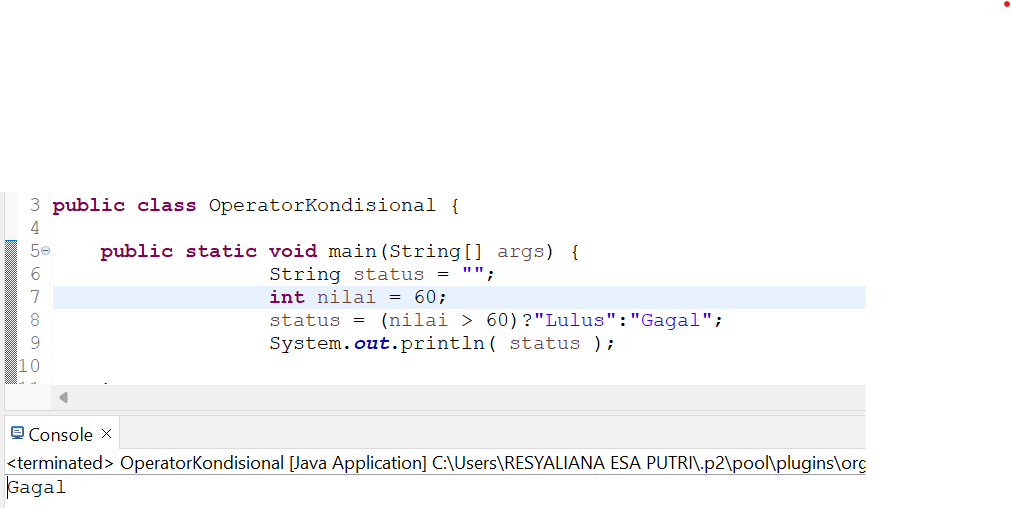
→ <https://www.youtube.com/watch?v=LcFgl0yrKEw>

**[No.6] Analisis dan Argumentasi**

Operasi Kondisional atau operasi Ternary merupakan satu-satunya operasi yang membutuhkan 3 operand. Operator ini identik dengan tanda tanya (?) dan titik dua (:). Pada soal ini saya mengubah nilai int 80 menjadi 60 dan saya analisis bahwa hasil yang diluarkan berubah status menjadi GAGAL karena nilai int 60 tidak lebih dari 60 untuk mendapatkan status lulus.



Hasil luaran setelah mengubah nilai int:

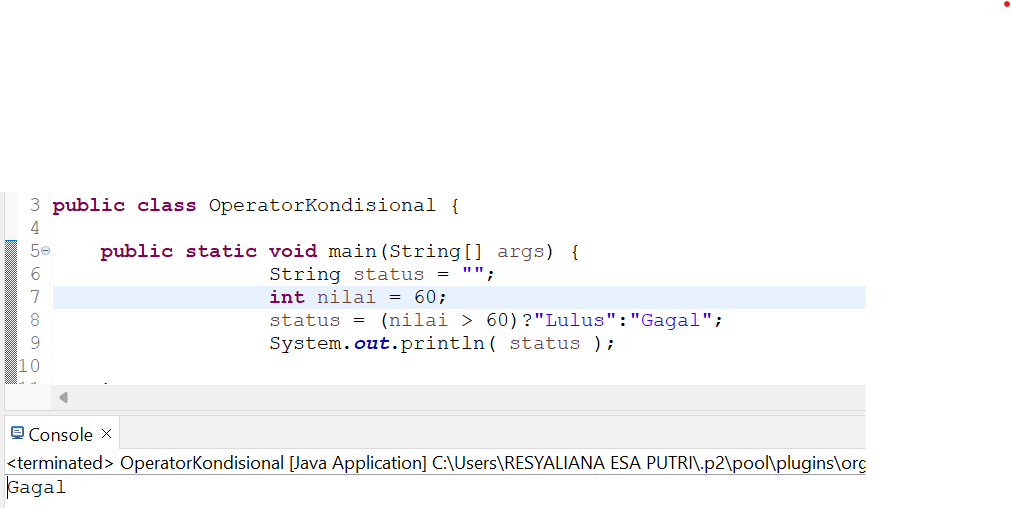


**[No.6] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma penambahan operator Kondisional:

1. Buka Eclipse atau Jdoodle yang ada dilaptop atau computer kamu.
2. Tambahkan class.
3. Masukan code yang sudah diberikan.
4. Ubahlah nilai int menjadi 60
5. Lalu analisislah hasil dan proses yang terjadi.
6. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.6] Kesimpulan** **Analisa**

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

Operator Kondisional atau operasi Ternary merupakan satu-satunya operasi yang membutuhkan 3 operand. Operator ini identik dengan tanda tanya (?) dan titik dua (:). Operator kondisional juga merupakan penyederhanaan dari bentuk if..else yang setiap blok dari if dan else hanya terdiri dari satu statement atau perintah.

Pada soal String status = "";

**int** nilai = 80;

status = (nilai > 60)?"Lulus":"Gagal";

saya mengubah nilai int 80 menjadi int 60 menghasilkan luaran yang berbeda dari pada sebelumnya karena nilai 60 tidak lebih besar dari 60.

Dapat disimpulkan bahwa hasil yang diluarkan akan berubah jika nilai yang dimasukan tidak mencukupi status.

**[No. 7] Identifikasi Masalah:**

1. Uraikan permasalahan dan variable

Soal:

public class operator {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 7;  
        int hasil;  
            
        hasil = a & b;  
        System.out.println("Hasil dari a & b : " + hasil );    
            
        hasil = a | b;  
        System.out.println("Hasil dari a | b : " + hasil );    
            
        hasil = a ^ b;  
        System.out.println("Hasil dari a ^ b : " + hasil );    
            
        hasil = ~a;  
        System.out.println("Hasil dari ~a : " + hasil );    
            
        hasil = a >> 1;  
        System.out.println("Hasil dari a >> 1 : " + hasil );    
            
        hasil = b << 2;  
        System.out.println("Hasil dari b << 2 : " + hasil );  
}   }

**Luaran:**  
Hasil dari a & b : 6  
Hasil dari a | b : 7  
Hasil dari a ^ b : 1  
Hasil dari ~a : -11  
Hasil dari a >> 1 : 3  
Hasil dari b << 2 : 28

**Latihan 7**  
Pilihlah 3 perhitungan Contoh 7, kemudian uraikan perhitungan biner! Simpulkan hasilnya!

1. Hasil dari a & b

* Konversi nilai a

a = 10/2 = 5 sisa 0

/2 = 2 sisa 1

/2 = 1 sisa 0

/2 = 0 sisa 1

a = 1010

Konversi nila b

b = 7/2 = 3 sisa 1

/2 = 1 sisa 1

/2 = 0 sisa 1

b = 111 atau 0111

* a & b (TRUE bila kedua operand TRUE)

a = 1010

b = 0111

Hasil a & b = 0010

Maka a & b = 2

1. Hasil dari a | b

* Konversi nilai a

a = 10/2 = 5 sisa 0

/2 = 2 sisa 1

/2 = 1 sisa 0

/2 = 0 sisa 1

a = 1010

Konversi nila b

b = 7/2 = 3 sisa 1

/2 = 1 sisa 1

/2 = 0 sisa 1

b = 111 atau 0111

* a | b (TRUE bila salah satu atau kedua operand TRUE)

a = 1010

b = 0111

Hasil a | b = 1111

Maka a | b = 15

1. Hasil dari a ^ b

* Konversi nilai a

a = 10/2 = 5 sisa 0

/2 = 2 sisa 1

/2 = 1 sisa 0

/2 = 0 sisa 1

a = 1010

Konversi nila b

b = 7/2 = 3 sisa 1

/2 = 1 sisa 1

/2 = 0 sisa 1

b = 111 atau 0111

* a ^ b (TRUE bila salah satu operand bernilai TRUE)

a = 1010

b = 0111

Hasil a ^ b = 1101

Maka a ^ b = 13

1. Rincikan sumber informasi yang relevan (buku / webpage)

→ <https://www.youtube.com/watch?v=LcFgl0yrKEw>

**[No.7] Analisis dan Argumentasi**

Operator Bitwise merupakan operator khusus yang menangani operasi bilangan biner secara langsung. Operator yang digunakan adalah AND(&), OR(|), EX OR(^), NOT(~), GESER KIRI(<<), GESER KANAN(>>), dan GESER KANAN TAK BERTANDA(>>>).

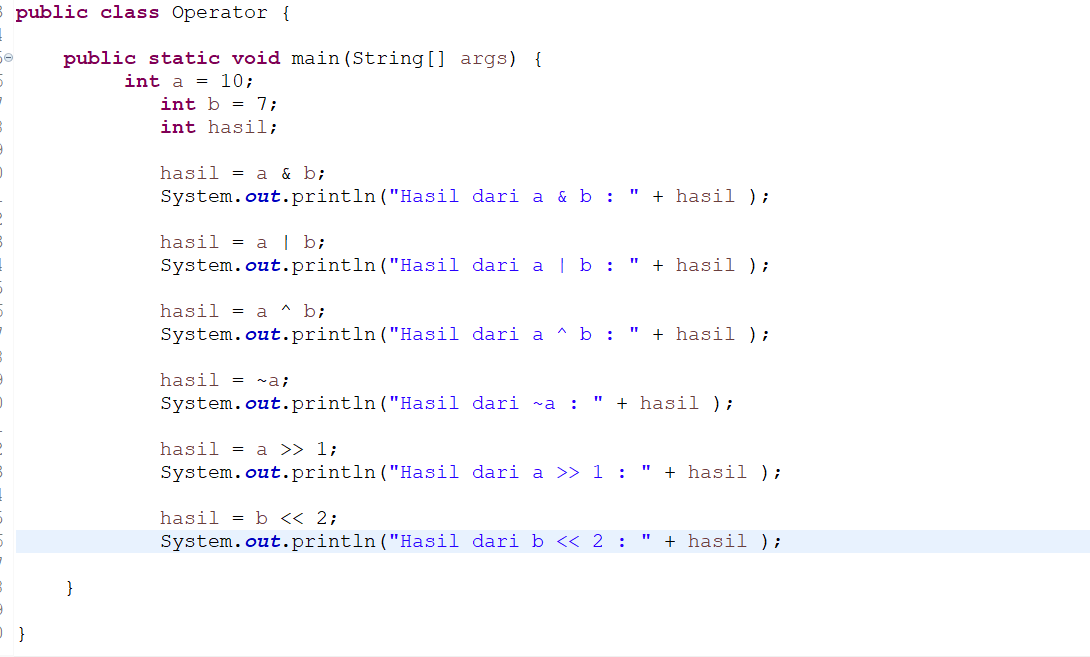
Pilih 3 perhitungan yang kemudian uraikan perhitungan binernya. Saya memilih perhitungan AND, OR, dan XOR karena mudah dimengerti. Untuk AND akan bernilai TRUE bila kedua operand TRUE, OR akan bernilai TRUE bila salah satu atau kedua operand TRUE, dan XOR akan bernilai TRUE bila salah satu operand bernilai TRUE.

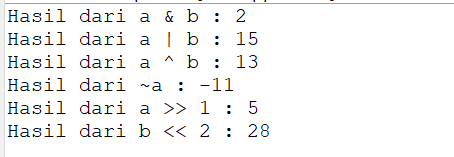
**[No.7] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma penambahan operator Kondisional:

1. Buka Eclipse atau Jdoodle yang ada dilaptop atau computer kamu.
2. Tambahkan class.
3. Masukan code yang sudah diberikan.
4. Pilihlah 3 perhitungan, kemudian uraikan perhitungan bilangan biner!
5. Kode program dan luaran

****

****

1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.7] Kesimpulan** **Analisa**

Operator Bitwise merupakan operator khusus yang menangani operasi bilangan biner secara langsung. Operator yang digunakan adalah AND (&), OR (|), EX OR (^), NOT (~), GESER KIRI (<<), GESER KANAN (>>), dan GESER KANAN TAK BERTANDA (>>>).

Pada soal ini saya memilih perhitungan AND, OR, dan XOR yang kemudian saya uraikan perhitungan binernya. Dari perhitungan tersebut saya mendapatkan hasil sebagai berikut:

* a & b (a AND b) = 2
* a | b (a OR b) = 15
* a ^ b (a XOR b) = 13

**Refleksi**

Operator adalah symbol dan karakter khusus suatu ekspresi. Operator terbagi 3 berdasarkan operand yaitu, Unary, Binary, dan Ternary. Jenis-jenis operator ada 7 yaitu, Operator Aritmatika, Operator Penugasan, Operator Relasional, Operator Increment dan Decrement, Operator Logika, Operator Kondisional, dan Operator Bitwise. Setiap operator tersebut memiliki fungsi dan syarat penggunaan masing-masing.