|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Meida Dinafani G1F024058** | **Operator Java** | **13 September 2024** |

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

**1. Operator Aritmatika: +, -, \*, /**

**Contoh 1:**Salin dan tempel kode program berikut 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 = 23

**Latihan 1**  
Susun kode Java untuk perhitungan dengan ekspresi (2\*3 + 6 / 2 -  4). Simpulkan urutan prioritas operator yang dijalankan ekspresi tersebut!

1. Operator Perkalian (\*) dan Pembagian (/): Operator ini memiliki prioritas lebih tinggi dibandingkan dengan operator penjumlahan (+) dan pengurangan (-). Jadi, ekspresi 2 \* 3 dan 6 / 2 dievaluasi terlebih dahulu.

2. Operator Penjumlahan (+) dan Pengurangan (-): Setelah perkalian dan pembagian dievaluasi, sisa ekspresi yang tersisa hanya melibatkan penjumlahan dan pengurangan.

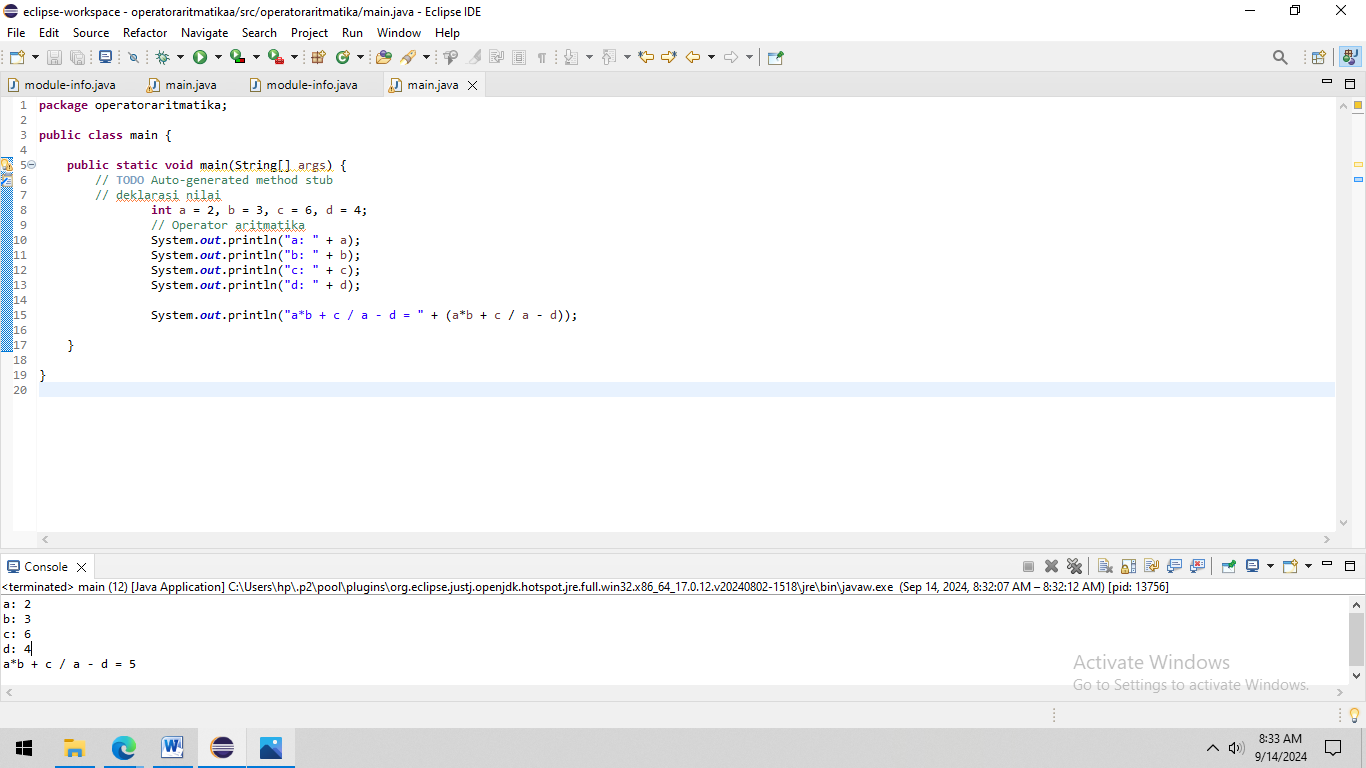
**[No.1] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode program pada Java yang melakukan perhitungan dengan ekspresi (2\*3 + 6 / 2 - 4) dan memahami urutan prioritas operator yang dijalankan.
2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan operator aritmatika.
3. Perbaikan kode program dengan cara memahami prioritas operator untuk menulis dan membaca ekspresi aritmatika dengan benar.

Serta mengubah atau memasukkan beberapa nilai yaitu a = 2, b = 3, c = 6, dan d = 4.

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

1. Algoritma
2. Mulai
3. Menentukan variabel
4. Deklarasikan variabel
5. Tampilkan hasil perhitungan
6. Selesai
7. Kode program dan luaran
8. Screenshot Kode program, hasil luaran, dan Komentar



1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta operator aritmatika yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.1] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

Pada program itu saya menggunakan bentuk operator untuk menulis dan membaca ekspresi aritmatika karena agar dapat memahami prioritas operator dengan benar.

Seperti :

1. Perkalian (\*) dan Pembagian (/) memiliki prioritas lebih tinggi dibandingkan dengan Penjumlahan (+) dan Pengurangan (-).
2. Operator di dalam tanda kurung () akan dievaluasi terlebih dahulu.

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

### ****2.    Operator Penugasan:  =****

Operator Penugasan adalah operator yang digunakan untuk memberikan nilai ke dalam variabel tertentu.

**Contoh 2:**Salin dan tempel kode program berikut ke Eclipse atau JDoodle

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);  
    }  
}

**Luaran:**  
Penambahan : 23

**Latihan 2.**  
Susun kode untuk menampilkan perhitungan dengan operator ( -=, \*=,  /=, %=)! Simpulkan hasilnya!

Operator penugasan melakukan operasi aritmatika dan memberikan hasilnya kembali ke operand di sebelah kiri:

* -= melakukan pengurangan,
* \*= melakukan perkalian,
* /= melakukan pembagian,
* %= melakukan modulus (sisa pembagian).

1. **Pengurangan (-=)**

* b -= a artinya b = b - a
* Nilai awal b adalah 3 dan a adalah 20
* Setelah operasi b -= a, b menjadi -17

1. **Perkalian (\*=)**

* b \*= a artinya b = b \* a
* Nilai awal b adalah 3 dan a adalah 20
* Setelah operasi b \*= a, b menjadi 60

1. **Pembagian (/=)**

* b /= a artinya b = b / a
* Nilai awal b adalah 3 dan a adalah 20
* Setelah operasi b /= a, b menjadi 0 (karena 3 dibagi 20 adalah 0 dalam pembagian integer)

1. **Modulus (%=)**

* b %= a artinya b = b % a
* Nilai awal b adalah 3 dan a adalah 20
* Setelah operasi b %= a, b menjadi 3 (karena sisa bagi 3 dibagi 20 adalah 3)

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

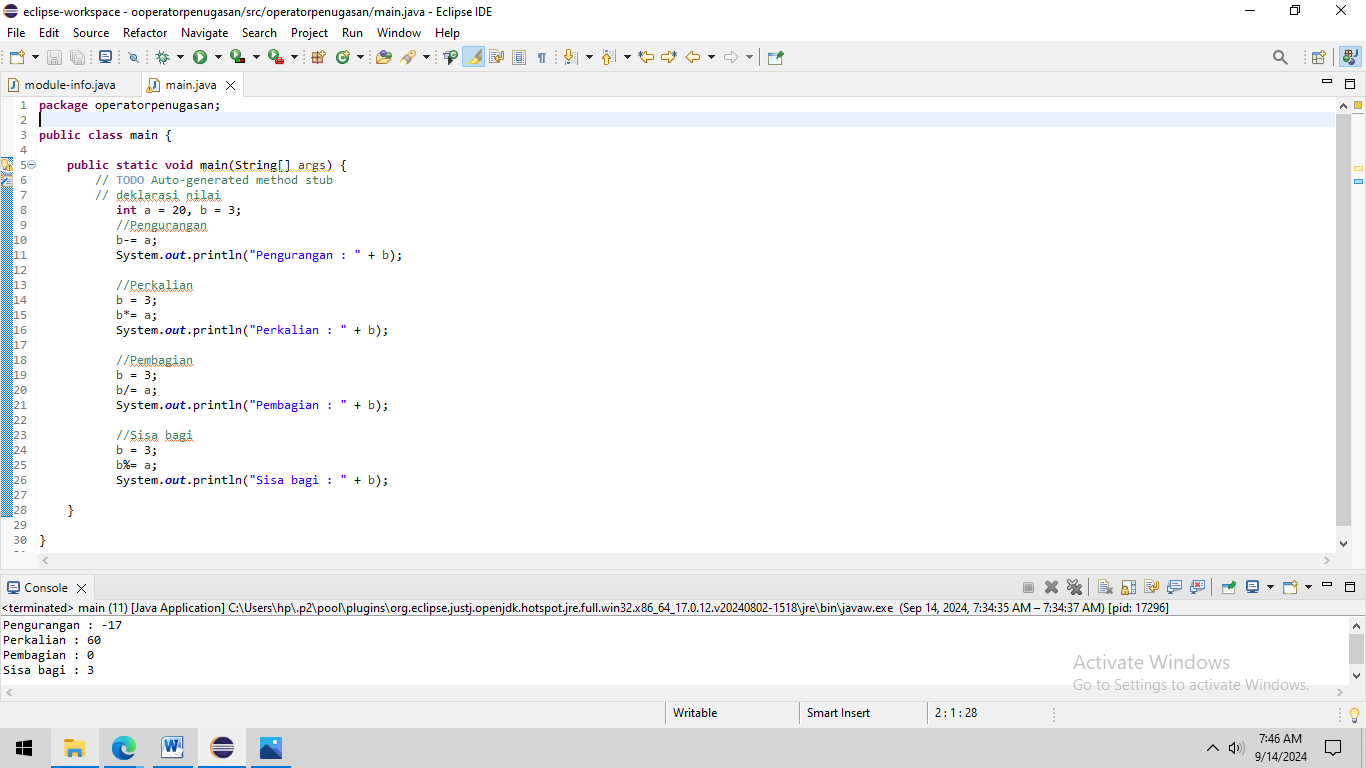
1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode program pada Java yang melakukan penggunaan operator penugasan dimulai dari pengurangan, perkalian, pembagian dan modulus (sisa pembagian).
2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan operator penugasan.
3. Perbaikan kode program dengan cara menambahkan beberapa contoh kode dengan berbagai operator penugasan (operator assignment) seperti -= (pengurangan), \*= (perkalian), /= (pembagian), dan %= (modulus).

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

1. Algoritma
2. Mulai
3. Menentukan variabel
4. Deklarasikan variabel
5. Mencetak hasil dengan menampilkan hasil

perhitungan

1. Selesai
2. Kode program dan luaran
   * 1. Screenshot Kode program, hasil luaran, dan Komentar



* + 1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta operator penugasan yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.2] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

* + 1. **Analisa**

Pada program itu saya menggunakan bentuk operator bentuk operator penugasan karena untuk memberikan nilai ke dalam suatu variabel.

Operator penugasan melakukan operasi aritmatika dan memberikan hasilnya kembali ke operand di sebelah kiri:

* -= melakukan pengurangan,
* \*= melakukan perkalian,
* /= melakukan pembagian,
* %= melakukan modulus (sisa pembagian).

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

### ****3.    Operator Relasional:   <, >, <=, >=, =, ==, !=****

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

**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

**Latihan 3**  
Susunlah perintah kode dengan operator relasional (<, >, <=, >=, =, ==, !=) untuk nilai a dan b yang menghasilkan luaran TRUE!

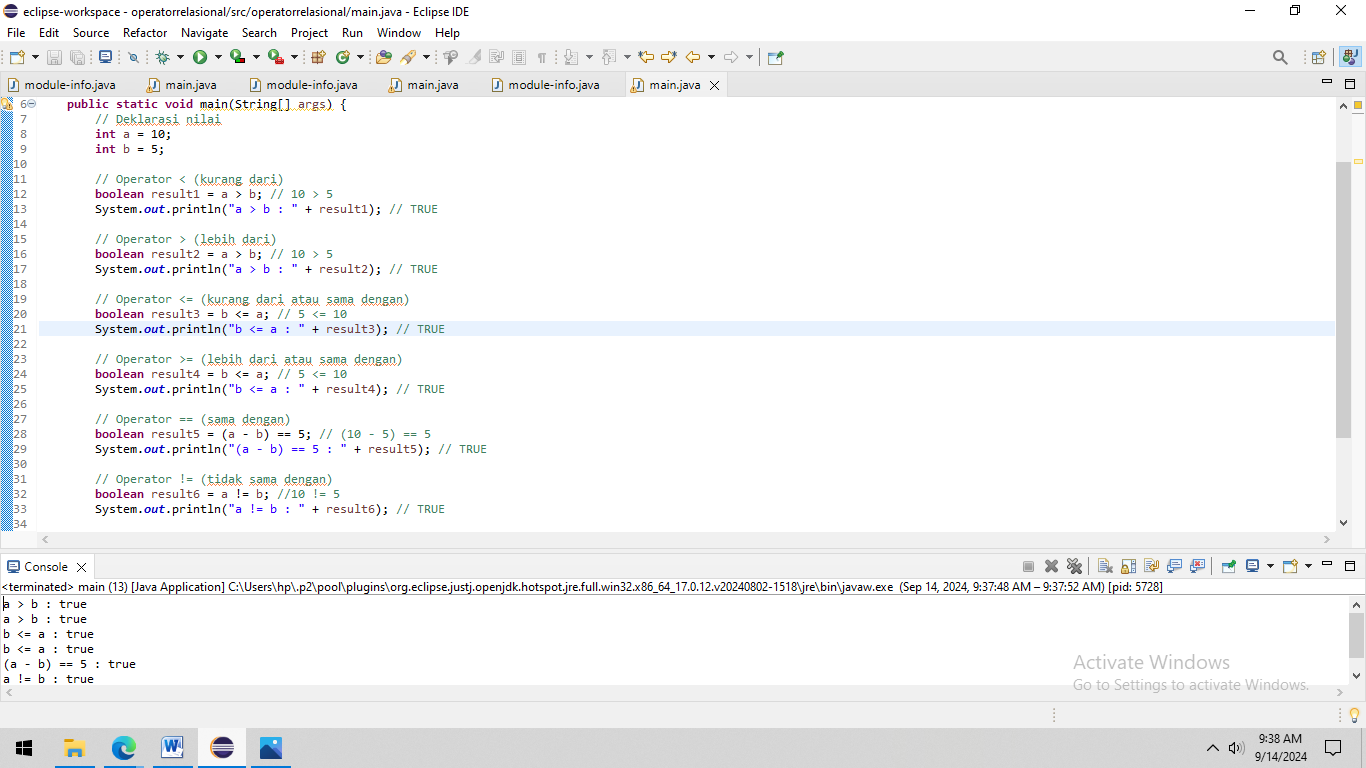
**[No.3] Analisis dan Argumentasi**

* + 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode program pada Java dengan cara memilih Operator Relasional yang Tepat yakni > (lebih besar dari), < (lebih kecil dari), >= (lebih besar dari atau sama dengan), <= (lebih kecil dari atau sama dengan), == (sama dengan), dan != (tidak sama dengan).
    2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan operator relasional.
    3. Perbaikan kode program dengan cara memastikan bahwa operator relasional yang digunakan sesuai dengan nilai-nilai yang akan dibandingkan.

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

1) Algoritma

1. Mulai
2. Menentukan nilai variabel
3. Tampilkan hasil
4. Selesai
   * 1. Kode program dan luaran
        1. Screenshot Kode program, hasil luaran, dan Komentar



* + - 1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta operator relasional yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.3] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

* + 1. **Analisa**

Pada program itu saya menggunakan bentuk operator relasional karena dapat menguji hubungan antara nilai dan atau variabel dan menghasilkan nilai true atau false.

Pilih Operator Relasional yang Tepat, operator relasional yang sesuai:

* > (lebih besar dari)
* < (lebih kecil dari)
* >= (lebih besar dari atau sama dengan)
* <= (lebih kecil dari atau sama dengan)
* == (sama dengan)
* != (tidak sama dengan)

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

### ****4.  Operator Increment dan Decrement: ++, --****

Kedua operator ini digunakan pada operand bertipe bilangan bulat. Operator penaikan digunakan untuk menaikan nilai variabel sebesar satu, sedangkan operator penurunan dipakai untuk menurunkan nilai variabel sebesar satu.

**Contoh 4:**Salin dan tempel kode program berikut ke Eclipse atau JDoodle

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

**Luaran:**  
a: 5  
b: 5

**Latihan 4.**  
4.1.  Susunlah kode program untuk menghasilkan luaran nilai a = 5 dan b = 6 dengan pre/post increment dan pre/post decrement. Untuk menghasilkan luaran di mana nilai a adalah 5 dan nilai b adalah 6, menggunakan operator post-increment (a++) dan pre-increment (++a). Dalam hal ini, a++ adalah post-increment yang menambahkan nilai a setelah nilai saat ini digunakan, sedangkan ++a adalah pre-increment yang menambahkan nilai a sebelum nilai digunakan.

4.2.  Simpulkan hasil perbandingan Anda (pre/post increment, pre/post decrement)!

1. **Post-Increment (a++)**:

* Nilai variabel digunakan terlebih dahulu, lalu nilai variabel di-increment.
* Misalnya, b = a++ berarti b mendapatkan nilai lama dari a, dan a di-increment setelah nilai tersebut digunakan.

2. **Pre-Increment (++a)**:

* Nilai variabel di-increment terlebih dahulu, lalu nilai baru variabel digunakan.
* Misalnya, b = ++a berarti a di-increment terlebih dahulu dan nilai baru dari a di-set ke b.

3. **Post-Decrement (a--)**:

* Nilai variabel digunakan terlebih dahulu, lalu nilai variabel dikurangi.
* Misalnya, b = a-- berarti b mendapatkan nilai lama dari a, dan a dikurangi setelah nilai tersebut digunakan.

4. **Pre-Decrement (--a)**:

* Nilai variabel dikurangi terlebih dahulu, lalu nilai baru variabel digunakan.
* Misalnya, b = --a berarti a dikurangi terlebih dahulu dan nilai baru dari a di-set ke b.

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

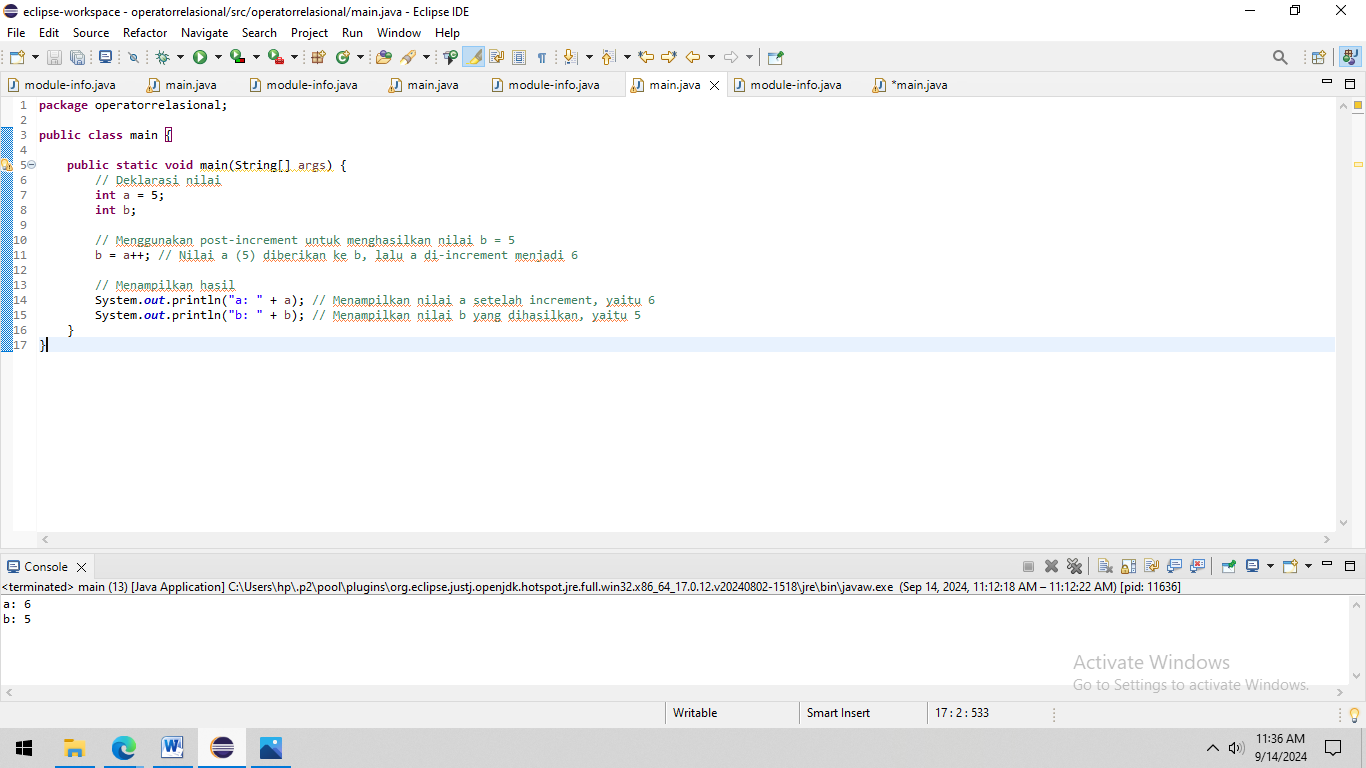
* + 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode program menggunakan operator post-increment (a++) dan pre-increment (++a)).
    2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan operator post-increment (a++) dan pre-increment (++a).
    3. Perbaikan kode program dengan cara mempelajari cara kerja pre dan post operator ini untuk pemrograman, terutama dalam urutan operasi karena dapat memengaruhi hasil akhir..

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

* 1. Algoritma

1. Mulai
2. Menentukan nilai variabel
3. Tampilkan hasil
4. Selesai

* + 1. Screenshot Kode program, hasil luaran, dan Komentar
       1. Screenshot Kode program, hasil luaran, dan Komentar



* + - 1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta operator post-increment (a++) dan pre-increment (++a) yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.4] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

* + 1. **Analisa**

Pada program itu saya menggunakan bentuk menggunakan operator post-increment (a++) dan pre-increment (++a). karena kedua operator ini dapat digunakan pada operand bertipe bilangan bulat.

* + Post-increment dan post-decrement mengembalikan nilai variabel sebelum operasi increment/decrement dilakukan.
  + Pre-increment dan pre-decrement mengembalikan nilai variabel setelah operasi increment/decrement dilakukan.

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

1. **Operator Boolean Logika:  &&, ||, !**

Contoh 5: Salin dan tempel kode program berikut ke Eclipse atau JDoodle

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));  
}    }

**Luaran:**  
Hasil logika (a && b) : false

**Latihan 5**  
Susun kode program dengan mengubah nilai a dan b untuk menghasilkan luaran true dengan operator && dan  operator | |.  Beri kesimpulan!

1. **Operator && (AND):**

Operator && menghasilkan true hanya jika kedua operand bernilai true.

1. **Operator || (OR):**

Operator || menghasilkan true jika salah satu dari operand bernilai true.

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

* 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode program menggunakan operator && dan ||.
  2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan operator && dan ||.
  3. Perbaikan kode program dengan cara memahami perbedaan antara operator && dan || agar dapat memanfaatkan logika boolean dengan tepat dalam berbagai situasi.

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

* 1. Algoritma

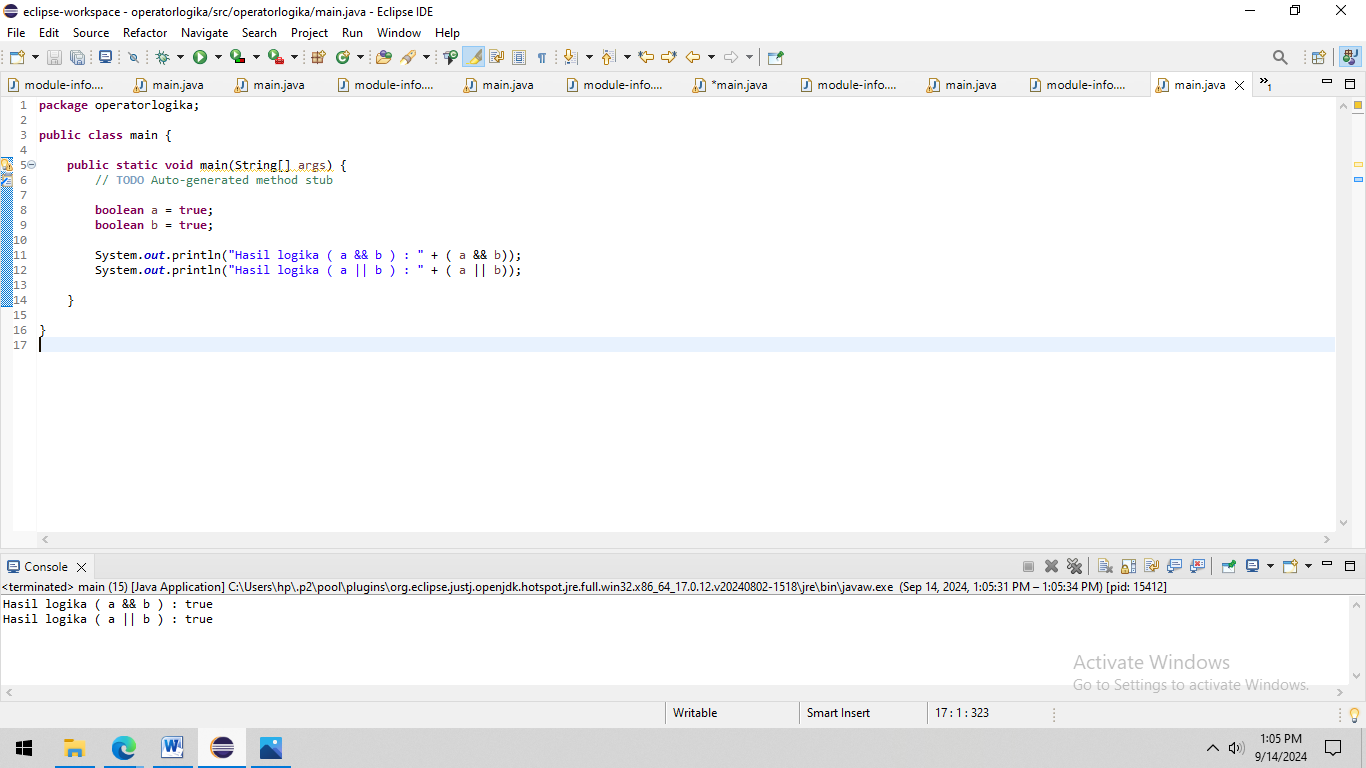
(a) Mulai

(b) Inisialisasi variabel

(c) Tampilkan hasil

(d) Selesai

* 1. Kode program dan luaran
     + 1. Screenshot Kode program, hasil luaran, dan Komentar



* + - 1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta operator operator && dan ||. yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.5] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

* + 1. **Analisa**

Pada program itu saya menggunakan bentuk Operator && dan Operator || karena ini merupakan operator logika untuk membandingkan dua nilai variabel atau lebih, yang hasilnya true atau false.

* + - 1. **Operator && (AND):**

Operator && menghasilkan true hanya jika kedua operand bernilai true.

* + - 1. **Operator || (OR):**

Operator || menghasilkan true jika salah satu dari operand bernilai true.

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

### ****6.    Operator Kondisional (Ternary):  ?:****

Operator Kondisi merupakan penyederhanaan dari bentuk if.else yang setiap blok dari if dan else hanya terdiri dari satu statement/perintah.

**Bentuk umum:  (ekspresi) ? (jika benar) : (jika salah);**

Contoh 6:   
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**  
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**

* 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode program menggunakan operator kondisional(Ternary).
  2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan operator ternary.
  3. Perbaikan kode program dengan cara menggunakan operator ternary untuk menentukan pesan berdasarkan nilai variabel jam.

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

1. Algoritma

(a) Mulai

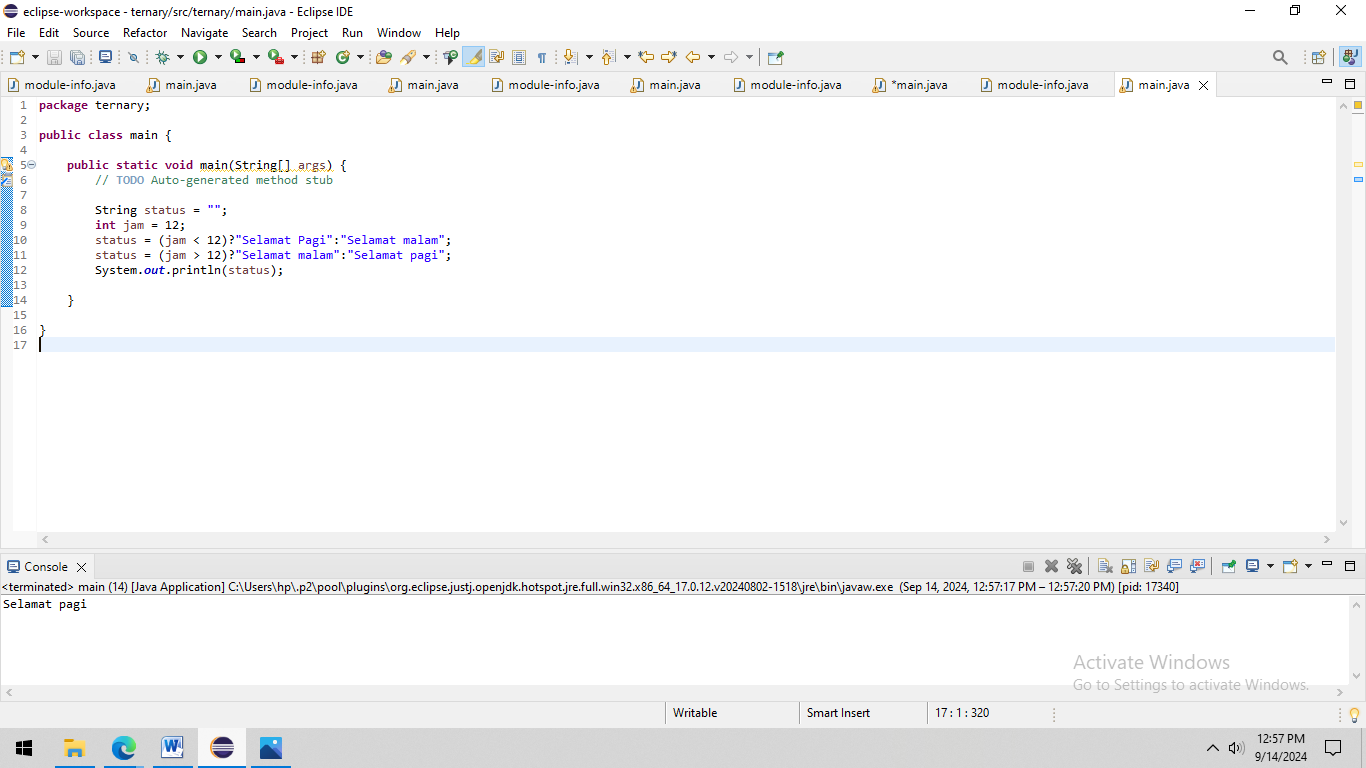
(b) Deklarasikan variabel

(c) Tampilkan hasil

(d) Selesai

2) Kode program dan luaran

* + - 1. Screenshot Kode program, hasil luaran, dan Komentar



* + - 1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta operator operator && dan ||. yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.6] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

* + - * 1. **Analisa**

Pada program itu saya menggunakan bentuk Operator Kondisional: Digunakan untuk memilih antara beberapa nilai berdasarkan kondisi yang dievaluasi. Dalam hal ini, operator ternary memeriksa beberapa kondisi untuk menentukan pesan yang akan ditampilkan. Dengan pendekatan ini, Anda dapat dengan mudah menyesuaikan nilai jam dan memeriksa bagaimana pesan berubah berdasarkan kondisi yang diberikan.

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

### ****7.    Operator Bitwise: &, |, ^, ~, <<, >>, >>>****

Operator bitwise merupakan operator untuk operasi bit (biner) dan berlaku untuk tipe data int, long, short, char, dan byte, karena akan menghitung dari bit-ke-bit.

**Contoh 7:**Salin dan tempel kode program berikut ke Eclipse atau JDoodle

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 );     
}   }

**Luaran:**  
Hasil dari a & b : 6  
Hasil dari a | b : 7  
Hasil dari a ^ b : 1

**Latihan 7**  
Susun kode tambahan dari Contoh 7 untuk melakukan perhitungan dengan operator (>>, <<). Hubungkan hasil luaran dengan perhitungan manual bilangan biner!

Hasil dari a & b : 2 (0110 dalam biner)

Hasil dari a | b : 15 (1111 dalam biner)

Hasil dari a ^ b : 13 (1101 dalam biner)

Hasil dari ~a : 5 (101 dalam biner)

Hasil dari a >> 2 : 2 (0110 dalam biner)

Hasil dari a << 2 : 2(0110 dalam biner)

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara membuat kode program yang melibatkan operasi bitwise (&, |, ^, ~,<<, >>).
2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan operator bitwise.
3. Perbaikan kode program dengan cara menggunakan operator bitwise serta penjelasan hasilnya dalam bilangan biner dan desimal.

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

1)Algoritma

(a) Mulai

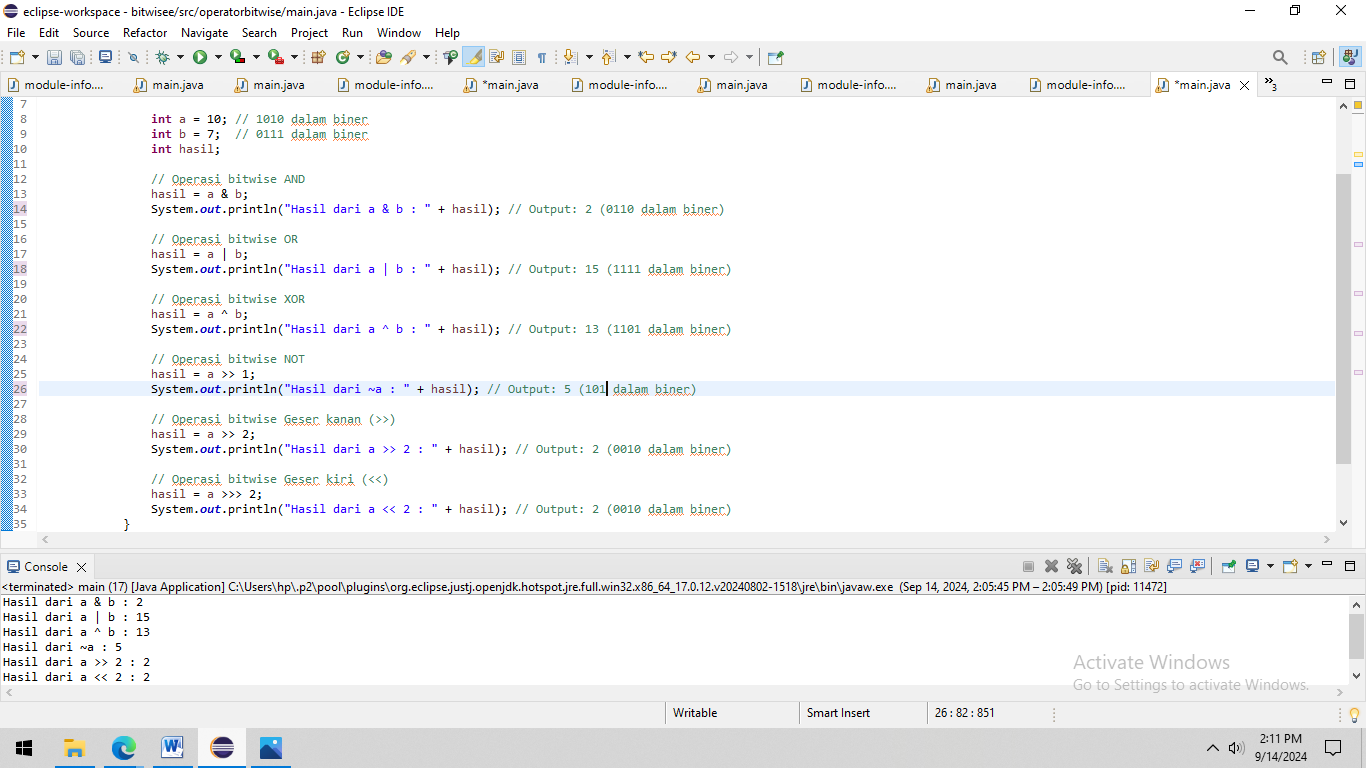
(b) Inisialisasi variabel

(c) Tampilkan hasil

(d) Selesai

2) Kode program dan luaran

* + - 1. Screenshot Kode program, hasil luaran, dan Komentar



* + - 1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta operator bitwise. yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.7] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

* + - 1. **Analisa**

Pada program itu saya menggunakan bentuk Operator bitwise. Karena operator ini digunakan untuk operasi bit (biner). Dan hasilnya dalam bilangan biner dan desimal

berlaku untuk tipe data int, long, short, char, dan byte, karena akan menghitung dari bit-ke-bit.

* + - 1. & untuk AND
      2. | untuk OR
      3. ^ untuk XOR
      4. ~ untuk NOT
      5. >> untuk Geser kanan
      6. << untuk Geser kiri

**Refleksi**

Pengalaman belajar yang didapat yakni mendapatkan pelajaran dan pemahaman mengenai pemrograman Operator yang ada di java meskipun ada yang susah dipahami. Pengetahuan yang di dapatkan yakni dapat mengetahui bagaimana cara kerja operator java. Serta mendapat pengetahuan juga mengenai Coding. Tantangan yang saya dihadapi yakni agak kesusahan dalam melakukan coding dan terkadang terjadi error, apalagi saya berasal dari sekolah Sma jadi belum terlalu menguasai bagaimana Coding.