|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **MOZA VIOLETHA SALSABILLA**  **G1F024028** | **Operator Java** | **12 September 2024** |
| **[Nomor Soal] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variabel 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) | | |
|  | | |

Contoh Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **MOZA VIOLETHA SALSABILLA G1F024028** | **Operator Java** | **12 September 2024** |

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

1. Uraikan permasalahan dan variabel

**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. **Latihan 1.**  
   1.1.  Tambahkan baris System.out.println("a + b = "  + (a + b));  Ubahlah operator ( + ) dengan tanda ( -, \*, /, %)  
   1.2.  Analisa perhitungan matematika yang terjadi!

**Jawaban:** Pada kode program masih terdapat permasalahan, yaitu pada *line* ke-8. Dimana dibuat *systemprintout* untuk menampilkan operasi a+b. Namun, pada saat pemanggilan fungsi a dan b operasi yang dikerjakan operasi pengurangan. Seharusnya, operasi yang dikerjakan adalah penambahan dari variabel a dan b.

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

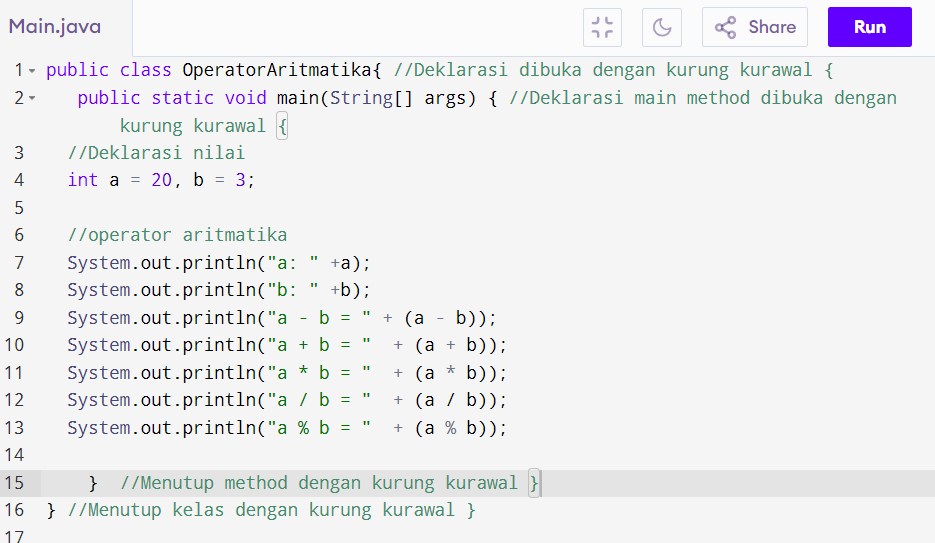
1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengganti tanda kurang (-) pada *line* ke-8 dengan tanda tambah sesuai dengan *output* yang diminta yaitu penambahan. Untuk soal nomor 2 yang meminta untuk menganalisis perhitungan, pada operasi penambahan, pengurangan, perkalian, pembagian, dan modulus (sisa bagi) telah menghasilkan *output* atau luaran yang diminta, tidak terjadi *error*. Pada operasi penambahan, *system* menambah variabel a dan variabel b dan menghasilkan *output* berupa nilai 23, pada operasi pengurangan, *system* mengurangi variabel a dengan variabel b sehingga menghasilkan *output* berupa 17, pada operasi perkalian, system mengalikan variabel a dan variabel b sehingga menghasilkan *output* berupa nilai 60, untuk operasi pembagian *system* membagi variabel a dengan variabel b dan menghasilkan nilai 6. Lalu kemana sisa bagi dari variabel a dan b? sisa bagi nya akan masuk ke operasi module (%) yaitu berupa nilai 2 karena 20 dibagi 3 hasilnya 6 dan sisanya 2.
2. Alasan solusi ini karena operasi yang diminta adalah penambahan variabel a dan variabel b sedangkan pada kode program, system diminta untuk menyelesaikan operasi pengurangan variabel a dan variabel b. Lalu untuk perhitungan matematika lainnya tidak mengalami permasalahan.
3. Perbaikan kode program dengan cara mengganti atau mengubah operasi perhitungan pada line ke-8 yang seharusnya pengurangan menjadi penambahan.

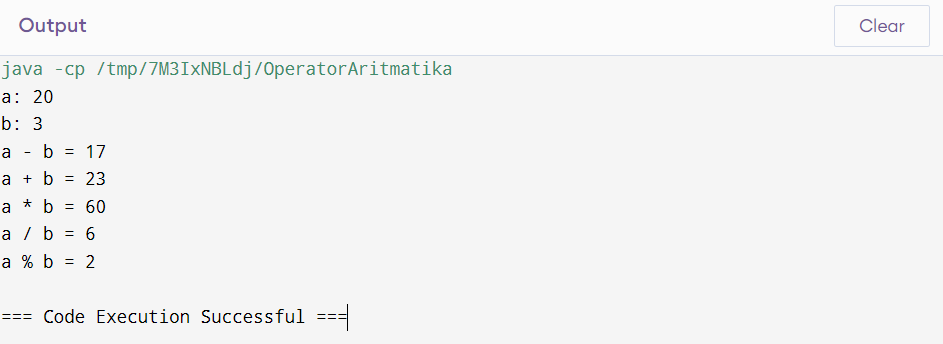
**[No.1 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai
2. Deklarasi nilai dari variabel a dan b
3. Jalankan operasi matematika
4. Menampilkan output ke layar
5. Selesai
6. Kode program dan luaran
7. Screenshot/ Capture potongan kode dan hasil luaran





1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Hasil operasi matematika yang ditampilkan mulai dari penambahan, pengurangan, perkalian, pembagian, dan modulus telah sesuai dengan kebutuhan dan permintaan.

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

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

Jawaban Analisa:

1. Pada program itu saya mengubah operasi pada *line* ke-8 dari operasi pengurangan menjadi penambahan karena *system* meminta *print out* operasi penambahan variabel a ditambah variabel b, sedangkan fungsi memanggil variabel a dikurang dengan variabel b.
2. Pada kasus ini saya mengambil dasar dari materi *e-learning* menganai *system* operasi bahwa apabila fungsi yang dipanggil harus sesuai dengan operasi yang diminta, karena apabila tidak sesuai maka nilai yang keluar tidak akan sesuai dengan luaran atau output yang diminta.

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

1. Uraikan permasalahan dan variabel

**Contoh 2**: Salin dan tempel potongan kode ini 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);

        // 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=3  
        System.out.println("Sisa Bagi: " + b);  
    }  
}

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

1. **Latihan 2.**

2.1.  Bandingkan hasil Contoh 1 dengan Contoh 2!

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

1. Pada contoh 1 kode program menggunakan operasi aritmatika berupa penambahan, pengurangan, perkalian, pembagian, dan modulus. Di contoh 1 diberikan input nilai a = 20, dan b = 3. Lalu, program langsung menhitung nilai dari variabel a+b, a-b, a  
   \*b, a/b, a%b dengan nilai a dan b tetap (maksudnya disini adalah bahwa nilai dari variabel a tetap 20 dan nilai dari variabel b tetap 3 tanpa adanya perubahan nilai setiap operasi diselesaikan) dan luaran yang dihasilkan adalah;

a – b = 17

a + b = 23

a \* b = 60

a / b = 6

a % b = 2

Sedangkan, pada contoh 2 kode program menggunakan operasi penugasan berupa penambahan, pengurangan, perkalian, pembagian, dan modulus. Sama seperti contoh 1, pada contoh 2 juga diberikan input nilai pada variabel a dan b yaitu a = 20 dan b = 3. Pada contoh 2 program diminta untuk memodifikasi nilai variabel b dari operasi aritmatika yang diberikan secara berturut-turut. Jadi, pada setiap operasi aritmatika nilai dari variabel b akan berubah sesuai dengan nilai b yang menjadi hasil operasi sebelumnya. Luaran yang dihasilkan adalah;

b + a = 23

b – a = 3

b \* a = 60

b / a = 3

b % a = 3

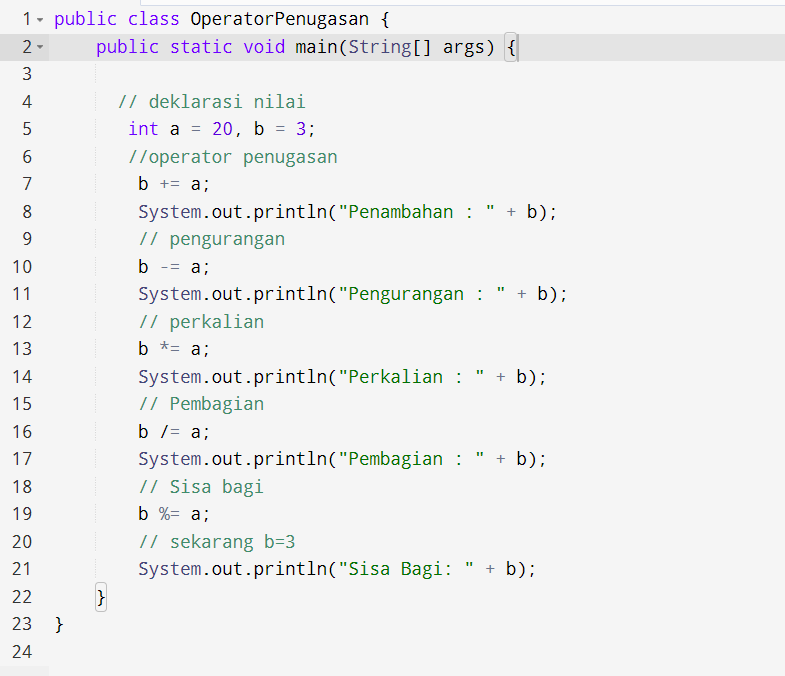
Untuk operasi modulus didapatkan hasil berupa nilai 3, kenapa bukan 0? Hal ini karena nilai variabel b < nilai variabel a maka, operasi modulus akan menghasilkan b yang sama dengan nilai awalnya yaitu 3.

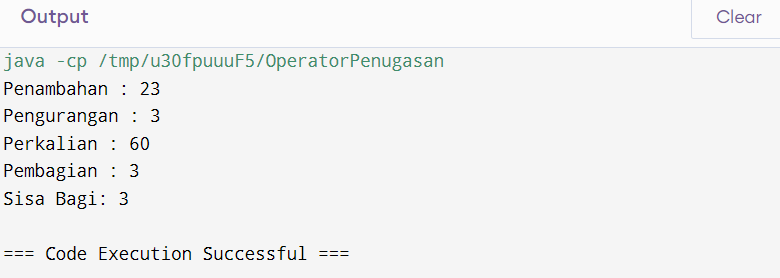
**[No.2] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai
2. Deklarasi nilai dari variabel a dan b
3. Jalankan operasi penugasan
4. Menampilkan output ke layar
5. Selesai
6. Kode program dan luaran
7. Screenshot/ Capture potongan kode dan hasil luaran





1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Hasil operasi penugasan yang ditampilkan mulai dari penambahan, pengurangan, perkalian, pembagian, dan modulus telah sesuai dengan kebutuhan dan permintaan.

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

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

Jawaban Analisa:

1. Pada contoh 1 lebih fokus atau mengedepankan pada operasi aritmatika dasar seperti penambahan, pengurangan, perkalian, pembagian, dan modulus tanpa memodifikasi nilai dari variabelnya. Sedangkan, pada contoh 2 menunjukkan cara menggunakan operator penugasan dalam operasi aritmatika yang berbeda secara berurutan untuk memodifikasi variabel b pada program.
2. Pada kasus ini saya mengambil dasar dari materi *e-learning* mengenai *system* operasi bahwa pada operator aritmatika menunjukkan hasil sederhana (tidak memodifikasi variabel), sedangkan pada operator penugasan memodifikasi nilai dari variabel secara berurutan atau bertahap.

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

1. Uraikan permasalahan dan variabel

**Contoh 3:** Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle  
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!

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

1. Pada saat nilai A dan B diubah, terjadi perbedaan saat luaran dihasilkan yaitu sebagai berikut;

Dengan nilai variabel A = 12, luaran yang dihasilkan adalah

* 1. Hasil A > B = *true*
  2. Hasil A < B = *false*
  3. Hasil A >= B = *true*
  4. Hasil A <= B = *false*
  5. Hasil A == B = *false*
  6. Hasil A != B = *true*

Analisis ;

1. Pada bagian (a) bernilai *true*, karena 12 lebih besar daripada 4.
2. Pada bagian (b) bernilai *false*, karena 12 tidak lebih kecil daripada 4. Melainkan lebih besar daripada 4.
3. Pada bagian (c) bernilai *true*, karena 12 lebih besar atau sama dengan 4
4. Pada bagian (d) bernilai *false*, karena 12 tidak lebih kecil atau sama dengan 4
5. Pada bagian (e) bernilai *false*, karena 12 tidak sama dengan 4
6. Pada bagian (f) bernilai *true*, karena 12 berbeda dengan 4

Sedangkan, jika nilai variabel B diganti menjadi 4, maka keluaran yang dhasilkan adalah;

1. Hasil A > B = *false*
2. Hasil A < B = *false*
3. Hasil A >= B = *true*
4. Hasil A <= B = *true*
5. Hasil A == B = *true*
6. Hasil A != B = *false*

Analisis ;

1. Pada bagian (a) bernilai *false*, karena 4 tidak lebih besar daripada 4. Melainkan 4 sama dengan 4.
2. Pada bagian (b) bernilai *false*, karena 4 tidak lebih kecil daripada 4. Melainkan 4 sama dengan 4.
3. Pada bagian (c) bernilai *true*, karena 4 lebih besar atau sama dengan 4
4. Pada bagian (d) bernilai *true*, karena 4 lebih kecil atau sama dengan 4
5. Pada bagian (e) bernilai *true*, karena 4 sama dengan 4
6. Pada bagian (f) bernilai false, karena 4 tidak berbeda dengan 4

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

a) Mulai

b) Deklarasi nilai dari variabel a dan b

c) Tampilkan *output* nilai awal

d) Periksa relasi

e) Tampilkan *output* atau luaran ke layar

f) Selesai

1. Kode program dan luaran
   * 1. Screenshot/ Capture potongan kode dan hasil luaran



* + 1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Hasil operasi relasional yang ditampilkan telah sesuai dengan kebutuhan dan permintaan.

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

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

Jawaban Analisa:

1. Pada saat variabel A dan B memiliki nilai yang sama yaitu 4, hasil keluaran dari operasi relasional yang memeriksa fungsi sama dengan menjadi bernilai *true*, sedangkan, fungsi tidak sama dengan menjadi bernilai *false*. Kemudian, Operator yang memeriksa hubungan besar,kecil, besar dari sama dengan, dan kurang dari sama dengan (>, <, >=, <=) akan menilai kebenaran berdasarkan perbandingan dari nilai-nilainya.
2. Pada kasus ini saya mengambil dasar dari materi *e-learning* mengenai *system* operasi yaitu operator relasional bahwa pada operator relasional akan membandingkan antara dua nilai yang dideklarasikan dan menghasilkan hasil luaran berupa boolean (*true* atau *false*).

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

1. Uraikan permasalahan dan variabel

**Contoh 4:**Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle  
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!

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

Pada increment hasil luaran post dan pre sebagai berikut;

|  |  |
| --- | --- |
| Post increment | Pre increment |
| Isi variabel a: 10 Isi variabel a: 10 Isi variabel a: 11 | Isi variabel b: 10 Isi variabel b: 11 Isi variabel b: 11 |

Pada post increment variabel a awalnya bernilai 10, kemudian diberikan fungsi post increment (a++) ke system. Lalu system akan menambahkan otomatis nilai 10 ditambahkan 1. Akan tetapi, karena merupakan fungsi post maka nilai 10 akan ditampilkan terlebih dahulu sebelum ditambahkan dengan 1. Setelah fungsi a++, nilai a menjadi 11 lalu ditampilkan ke layar. Sedangkan, pada pre increment variabel b awalnya bernilai 10. Kemudian diberikan fungsi pre increment (++b) maka nilai variabel b akan ditambahkan dengan 1 terlebih dahulu sehingga menjadi 11. Kemudian nilai 11 akan ditampilkan setelah penambahan.

|  |  |
| --- | --- |
| Post decrement | Pre decrement |
| Isi variabel c: 10 Isi variabel c: 10 Isi variabel c: 9 | Isi variabel d: 10 Isi variabel d: 9 Isi variabel d: 9 |

Pada post decrement variabel c awalnya bernilai 10, kemudian diberikan fungsi post decrement(c--) ke system. Lalu system akan mengurangi otomatis nilai 10 dikurangi 1. Akan tetapi, karena merupakan fungsi post maka nilai 10 akan ditampilkan terlebih dahulu ke layar sebelum dikurangi 1. Setelah fungsi c--, nilai c menjadi 9 lalu ditampilkan ke layar. Sedangkan, pada pre decrement variabel d awalnya bernilai 10. Kemudian, diberikan fungsi pre decrement (--d) maka nilai variabel d akan dikurangi dengan 1 terlebih dahulu sehingga menjadi 9. Lalu, nilai 9 akan ditampilkan setelah pengurangan.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

a) Mulai

b) Deklarasi nilai dari variabel a, b, c, dan d

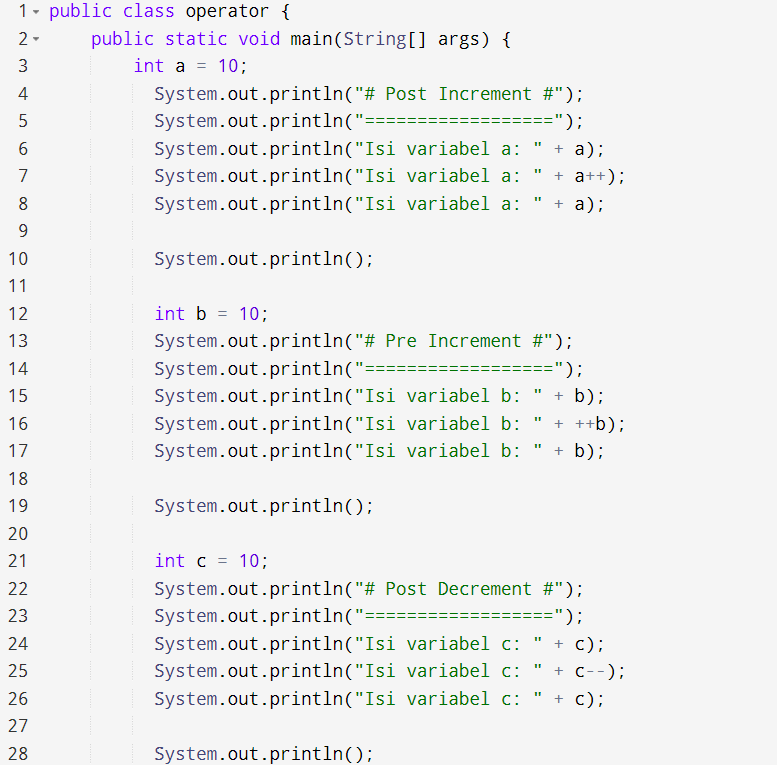
c) Jalankan post increment dan pre increment

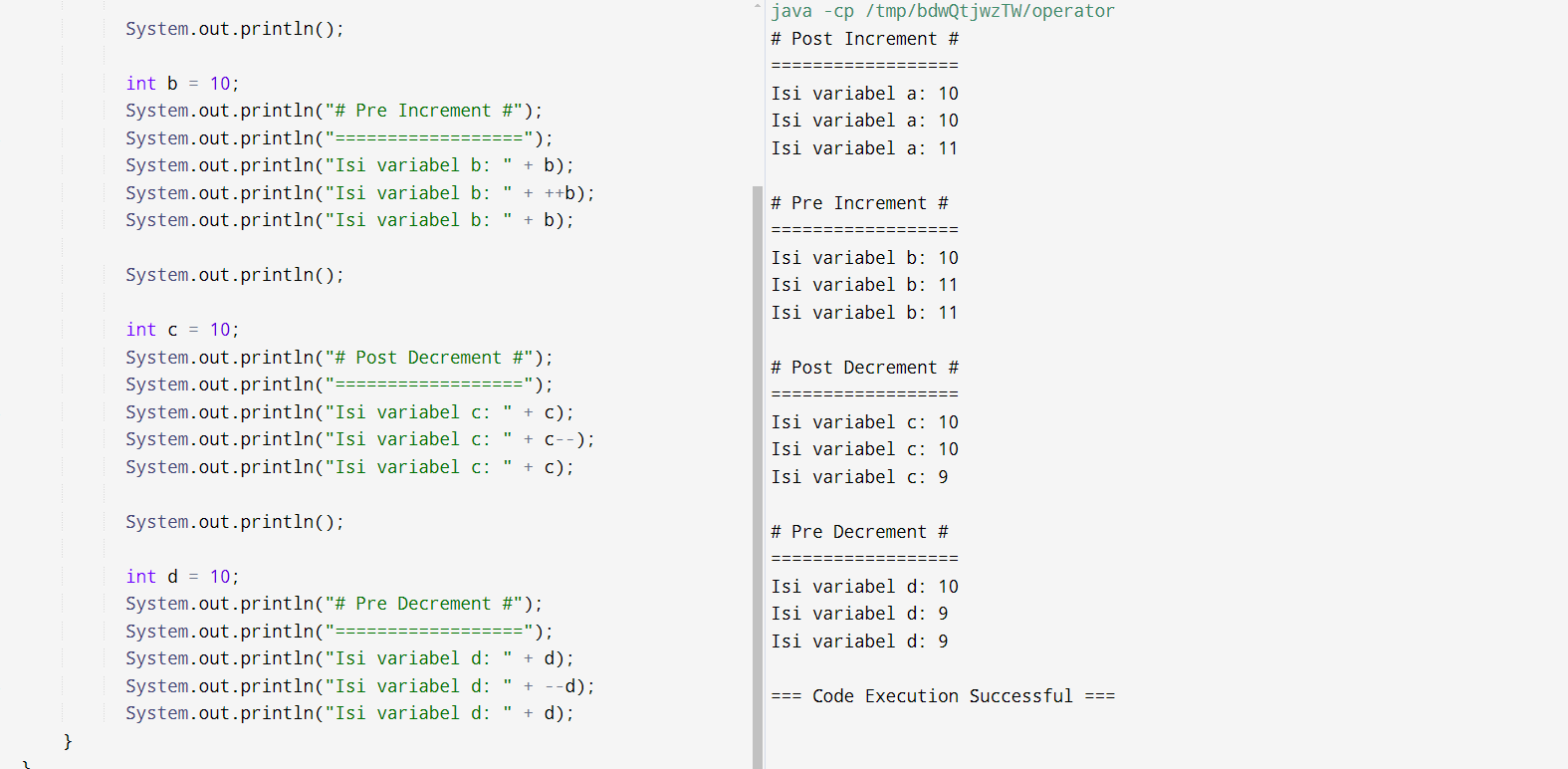
d) Jalankan post decrement dan pre decrement

e) Tampilkan *output* atau luaran ke layar

f) Selesai

1. Kode program dan luaran
2. Screenshot/ Capture potongan kode dan hasil luaran





1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Hasil operator increment dan decrement yang ditampilkan telah sesuai dengan kebutuhan dan permintaan.

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

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

Jawaban Analisa:

1. Pada dasarnya post dan pre increment maupun decrement itu sama saja, hanya berbeda dalam urutan operasinya dalam pengubahan nilai dan penampilannya. Post increment atau decrement akan mengubah nilai dari variabel setelah menampilkan hasil, sementara pre-increment atau decrement mengubah nilai sebelum menampilkan hasil luarannya.
2. Pada kasus ini saya mengambil dasar dari materi *e-learning* mengenai *system* operasi yaitu operator increment dan decrement. Dimana pada operator increment adalah operasi aritmetika yang digunakan untuk menambah nilai variabel dengan 1. Sedangkan, operator decrement adalah operasi aritmatika yang digunakan untuk mengurangi nilai variabel dengan 1. Lalu untuk post itu ditampilkan dulu nilai sebelumnya baru di eksekusi, sedangkan pre itu di eksekusi terlebih dahulu baru ditampilkan luarannya.

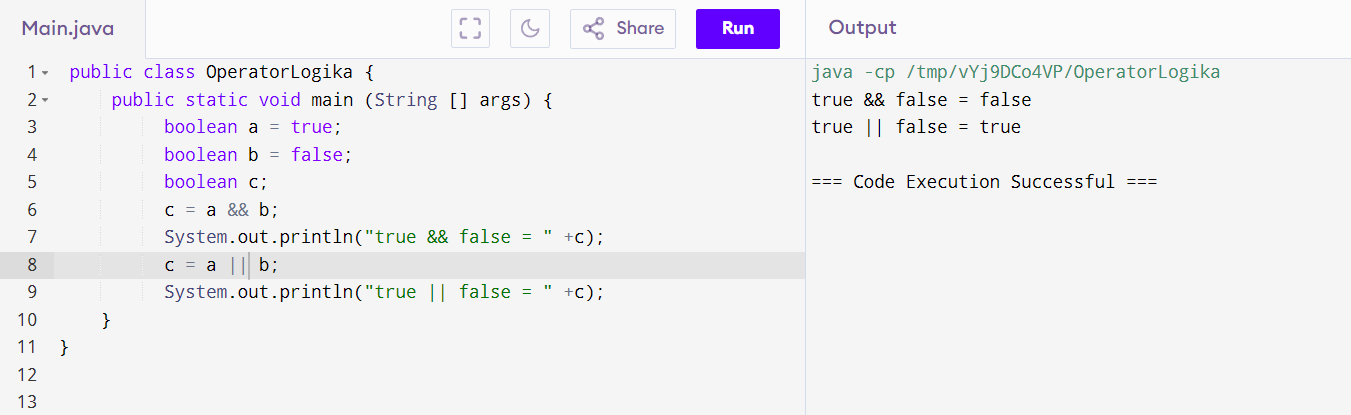
**[No. 5] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

**Contoh 5**: Salin dan tempel kode berikut:  
 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.  
5.2.  Ubahlah nilai a = false dan b = false. Analisa perubahan dan perbedaan boolean yang terjadi!  
5.2.  Apabila diketahui pernyataan a || b && a || !b. Uraikan urutan logika yang akan dikerjakan! Analisa luaran true atau false dari pernyataan tersebut!

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

****

gambar kode program 1

****

gambar kode program 2

Pada kode program 1 diminta operasi AND (a && b) dan OR (a || b) . Operasi AND pada kode program tersebut bernilai *false* karena dalam operasi AND hanya akan bernilai *true* apabila kedua operand nya bernilai *true*, apabila salah satu operand bernilai *false* maka hasilnya adalah *false*. Seperti pada kode program 1 operand a bernilai *true* dan operand b bernilai *false* sehingga luaran yang dihasilkan untuk operator AND adalah *false*. Lalu, untuk operator OR pada kode program 1, bernilai *true* karena jika salah satu operand bernilai *true* dan apabila kedua operand bernilai *false* maka hasil luaran yang keluar adalah *false*. Sedangkan, pada kode program 2 juga diminta operasi AND dan OR. Akan tetapi, nilai variabel a diubah menjadi *false*. Maka dari itu, operasi AND pada kode program 2 akan bernilai *false* karena kedua operandnya bernilai *false*. Kemudian, pada operasi OR akan bernilai *false* karena kedua operandnya bernilai *false*.

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

Untuk urutan logika yang akan dikerjakan itu sesuai dengan prioritas.

1. Operasi AND (&&) akan dikerjakan lebih dulu karena prioritasnya lebih tinggi daripada OR (| |) . Oleh karena itu, maka b && a akan dikerjakan terlebih dahulu.
2. Kemudian, hasil dari b && a akan digabungkan dengan a | |.
3. Selanjutnya, hasil dari a | | (b && a) akan digabungkan dengan | | !b.

Dimisalkan nilai a = *true* dan b = *false*, maka dapat dioperasikan sebagai berikut;

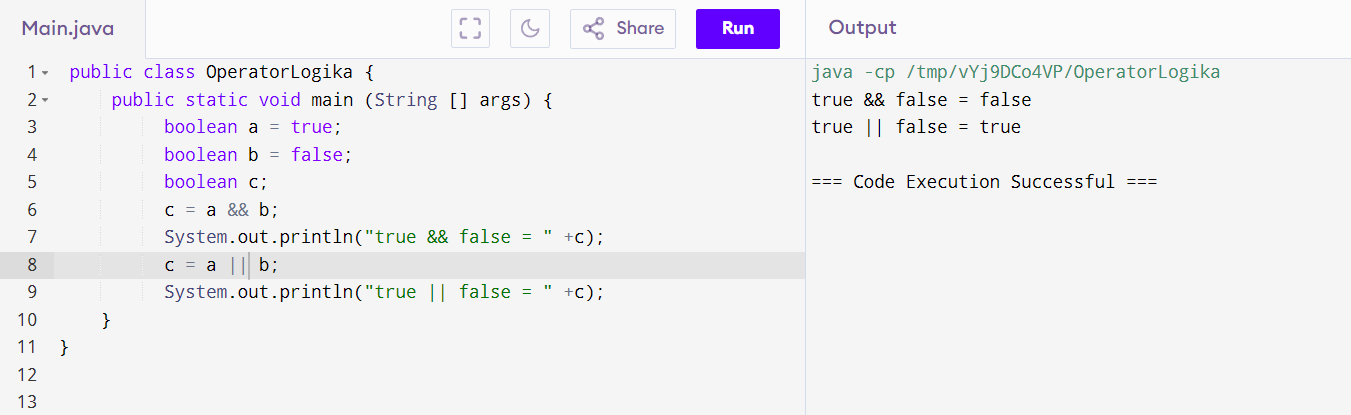
1. b && a, dimana b bernilai *false* dan a bernilai *true,* maka b && a akan bernilai *false* karena kedua operand tidak bernilai *true.*
2. Selanjutnya, kerjakan a | | (b && a) dimana a bernilai *true* dan (b && a) bernilai *false*. Maka, luaran yang dihasilkan adalah *true.*
3. Kemudian, hasil dari a | | (b && a) yang bernilai *true* digabungkan dengan | | !b yang bernilai *true*. Maka, operasi a || b && a || !b akan menampilkan luaran *true* ke layar.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai
2. Deklarasi dan inisialisasi nilai dari variabel a dan b
3. Jalankan operator logika
4. Menampilkan output ke layar
5. Selesai
6. Kode program dan luaran
7. Screenshot/ Capture potongan kode dan hasil luaran

****

1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Hasil operator logika yang ditampilkan telah sesuai dengan kebutuhan dan permintaan.

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

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

Jawaban Analisa:

1. Operator logika adalah operator yang biasanya digunakan untuk melakukan operasi pada nilai Boolean (true atau false). Operator logika dibagi menjadi dua yaitu operator AND dan OR. **Operator AND (**&&**)** memeriksa apakah kedua kondisi bernilai benar, apabila salah satu kondisi salah, maka hasilnya adalah false. Sedangkan, operator OR (| |) memeriksa apakah salah satu dari kondisi bernilai benar, apabila salah satu kondisi bernilai benar, maka hasilnya adalah true.
2. Pada kasus ini saya mengambil dasar dari materi *e-learning* mengenai *system* operasi yaitu operator logika. Dimana, Operator Logika digunakan untuk membandingkan dua nilai variabel atau lebih, hasilnya boolean true atau false.

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

1. Uraikan permasalahan dan variabel

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

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

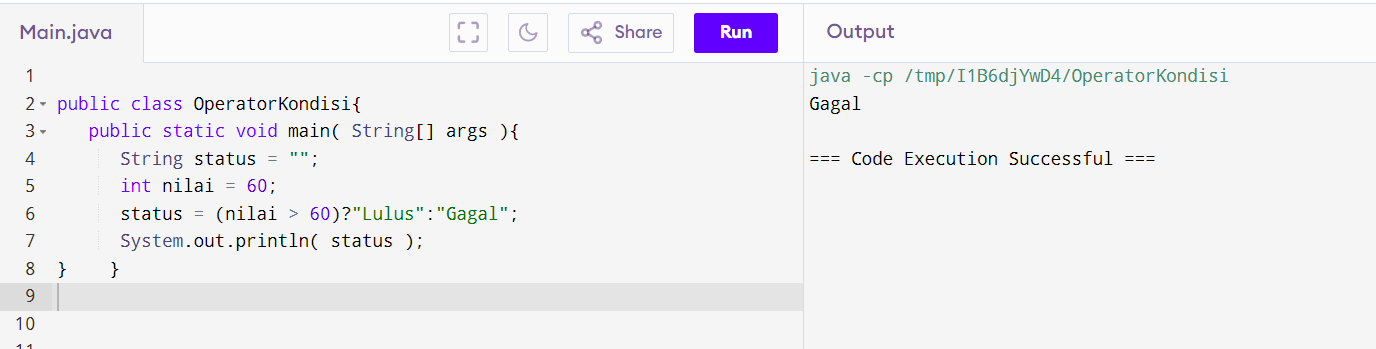
Pada saat nilai diubah menjadi 60, luaran yang dihasilkan adalah false. Hal ini karena nilai 60 tidak lebih dari 60 sehingga system akan mendeteksi bahwa luaran tersebut bernilai false maka status akan mencetak “gagal”.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai
2. Deklarasi nilai dari variabel
3. Evaluasi kondisi
4. Mencetak status ke layar
5. Selesai
6. Kode program dan luaran
7. Screenshot/ Capture potongan kode dan hasil luaran



1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Hasil operator kondisional yang ditampilkan telah sesuai dengan kebutuhan dan permintaan.

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

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

Jawaban Analisa:

1. Pada program diminta untuk mengubah nilai dari variabel menjadi 60. Kemudian, program akan memeriksa apakah nilai dari variabel lebih besar daripada 60. Lalu, karena 60 tidak lebih besar daripada 60. Maka, kondisi bernilai false dan status akan di setting menjadi “gagal”.
2. Pada kasus ini saya mengambil dasar dari materi *e-learning* mengenai *system* operasi yaitu operator kondisional. Dimana 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);

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

1. Uraikan permasalahan dan variabel

**Contoh 7:** Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle  
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 : 2  
Hasil dari a | b : 15  
Hasil dari a ^ b : 13  
Hasil dari ~a : -11  
Hasil dari a >> 1 : 5  
Hasil dari b << 2 : 28

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

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

Saya memilih 3 perhitungan berupa;

1. Bitwise AND ( a & b)

Pada operasi AND, system akan membandingkan setiap bit dari dua angka dan hanya akan menghasilkan 1 apabila kedua bit bernilai 1.

Nilai variabel a = 00001010

Nilai variabel b = 00000111

Operasi AND = 00000010, yang apabila dikonversi ke desimal maka akan bernilai 2.

1. Bitwise OR (a | b)

Pada operasi OR, sytem akan membandingkan setiap bit dari dua angka dan akan menghasilkan 1 apabila salah satu dari bit bernilai 1.

Nilai variabel a = 00001010

Nilai variabel b = 00000111

Operasi OR = 00001111, yang apabila dikonversi ke desimal maka akan bernilai 15.

1. Bitwise XOR (^)

Pada operasi XOR, system akan membandingkan setiap bit dari dua angka dan hanya akan menghasilkan 1 jika satu dari bit yang dibandingkan bernilai 1, tapi tidak dengan keduanya.

Nilai variabel a = 00001010

Nilai variabel b = 00000111

Operasi XOR = 00001101, yang apabila dikonversi ke desimal maka akan bernilai 13.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

a) Mulai

b) Deklarasi nilai dari variabel a dan b

c) Jalankan operasi perhitungan

d) Tampilkan *output* atau luaran ke layar

e) Selesai

2) Kode program dan luaran

1. Screenshot/ Capture potongan kode dan hasil luaran



1. Analisa luaran yang dihasilkan

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

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

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

Jawaban Analisa:

1. Operator yang digunakan untuk melakukan operasi pada tingkat bit dari bilangan bulat. Operator ini menggunakan representasi biner dari angka dengan cara yang sangat spesifik. Operator bitwise ini memiliki beberapa macam operator yaitu;
2. Bitwise AND (&) yang akan menghasilkan bit bernilai 1 hanya apabila kedua bit dari operand adalah 1.
3. Bitwise OR (|) yang akan menghasilkan bit bernilai 1 jika salah satu dari operand memiliki bit 1.
4. Bitwise XOR (^) yang akan menghasilkan bit bernilai 1 hanya apabila bit dari operand berbeda.
5. Bitwise NOT (~) yang akan membalikkan semua bit dari operand (negasi).
6. Bitwise Right Shift (>>) yang akan menggeser bit ke kanan serta membagi nilai dengan 2 untuk setiap posisi geser.
7. Bitwise Left Shift (<<)yang akan menggeser bit ke kiri serta mengalikan nilai dengan 2 untuk setiap posisi geser.
8. Pada kasus ini saya mengambil dasar dari materi *e-learning* mengenai *system* operasi yaitu operator bitwise. Dimana, Operator bitwise merupkan operator yang digunakan untuk operasi bit (biner).  
   Operator ini berlaku untuk tipe data int, long, short, char, dan byte.  
   Operator ini akan menghitung dari bit-ke-bit.