|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cut Fitriani G1F024079** | **Operator aritmatika** | **10 September 2024** |

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

1. Uraikan permasalahan dan variabel

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));  //menampilkan hasil penjumlahan  
}   }

**Luaran:**  
        Exception in thread "main" java.lang.Error: Unresolved compilation problems:   
    Syntax error on token ""a + b = "", AssignmentOperator expected after this token  
    The left-hand side of an assignment must be a variable

**Latihan 1**  
1.1. Rekomendasikan perbaikan kode agar program Contoh 1 dapat berjalan!  
1.2. Tambahkan baris untuk menampilkan perhitungan dengan operator ( -, \*, /, %) pada Contoh 1!

-Pada soal masih ada kesalahan sintaks pada penggabungan string dan ekspresi. System.out.println("a + b = "  (a + b));  //menampilkan hasil penjumlahan untuk mencetak hasil penjumlahan a + b dengan format string yang benar, yaitu harus menyertakan hasil perhitungan dalam string yang diprint dengan menggunakan operator + untuk menggabungkan string dengan nilai-nilai lain. Perbaikan ini menjelaskan bahwa hasil penjumlahan (a + b) harus digabungkan dengan string "a + b = " menggunakan operator +.

-Diketahui dari soal : variable yang digunakan ada 2, yaitu **int a**: variabel ini dideklarasikan dengan tipe data int dan diinisialisasi dengan nilai 20. **int b**: Variabel ini dideklarasikan dengan tipe data int dan diinisialisasi dengan nilai 3.

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara memperhatikan masalah utamanya, sintaks di baris System.out.println("a + b = " (a + b)); tidak valid. Maka memerlukan operator + untuk menggabungkan string dan nilai numerik.
2. Alasan solusi ini karena agar output yang diinginkan akan mencetak nilai variabel dan hasil penjumlahan dengan benar.
3. Perbaikan kode program dengan cara mengubah baris System.out.println("a + b = "  (a + b));  //menampilkan hasil penjumlahan tersebut sehingga menggunakan operator + untuk menggabungkan string dengan hasil perhitungan. Berikut adalah perbaikan yang benar: **System.out.println("a + b = " + (a + b)); // menampilkan hasil penjumlahan } }**

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma Operator Aritmatika :

1. Mulai Program
2. Mendeklarasikan dua variabel int bernama a dan b, lalu memberikan nilai awal pada masing-masing variabel
3. Melalukan operasi aritmatika seperti penjumlahan, pengurangan, perkalian, pemabgian dan modulus (sisa bagi) antara variabel a dan b
4. Menampilkan hasil dengan menggunakan sistem perintah System.out.println()
5. Selesai
6. A screenshot of a computer

   Description automatically generatedKode program dan luaran
7. Screenshot/ Capture potongan kode dan hasil luaran

// program ini merupakan komentar satu baris, digunakan untuk memberikan penjelasan singkat mengenai baris kode tertentu.

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**

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

jawaban Analisa:

1. Kode program ini merupakan contoh sederhana dari program Java yang digunakan untuk melakukan perhitungan aritmatika dasar. Program ini menunjukkan bagaimana cara mendeklarasikan variabel, melakukan operasi matematika, dan menampilkan output di layar. Berdasarkan kode yang diberikan, masalah yang ingin diselesaikan adalah melakukan operasi aritmatika dasar (penjumlahan, pengurangan, perkalian, pembagian, dan modulus) terhadap dua bilangan bulat yang diwakili oleh variabel a dan b. Hasil dari setiap operasi kemudian akan ditampilkan ke layar.

Algoritma:

1. Deklarasikan dua variabel bertipe integer, a dan b, untuk menyimpan nilai bilangan yang akan dioperasikan. Berikan nilai awal pada variabel a dan b.
2. Lakukan operasi penjumlahan, pengurangan, perkalian, pembagian, dan modulus antara variabel a dan b.
3. Tampilkan hasil dari setiap operasi ke layar menggunakan perintah System.out.println().

Kode program yang diberikan telah mengimplementasikan algoritma di atas dengan cukup baik. Kode tersebut secara berurutan melakukan: mendeklarasikan variabel a dan b serta memberikan nilai awal, melakukan berbagai operasi aritmatika, mencetak hasil setiap operasi ke layar.

1. Pengenalan Konsep Variabel: kode ini memperkenalkan konsep variabel dengan tipe data integer (bilangan bulat) untuk menyimpan nilai.

Pemahaman Operator Aritmatika: Program ini menjelaskan cara menggunakan berbagai operator aritmatika seperti +, -, \*, /, dan %.

1. Output: kode ini menunjukkan cara mencetak hasil perhitungan ke layar menggunakan System.out.println().

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cut Fitriani G1F024079** | **Operator Penugasan** | **10 September 2024** |

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

1. Uraikan permasalahan dan variable

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

**Luaran:**  
Penambahan : 23

**Latihan 2.**  
2.1.  Tambahkan baris Contoh 2 untuk menampilkan perhitungan dengan operator ( -=, \*=,  /=, %=)!  
2.2.  Berikan argumentasi tentang perbedaan luaran dan waktu eksekusi Contoh 1 dan Contoh 2!

-Pada soal tidak ada kesalahan teknis atau logika yang jelas. Kode tersebut menggunakan operator penugasan dengan benar, dan output yang dihasilkan sesuai dengan operasi yang dilakukan. Jika melihat output "Penambahan : 23" di layar itu adalah hasil yang benar berdasarkan operasi penambahan yang dilakukan oleh b += a.

-Diketahui dari soal : variable yang digunakan ada 2, yaitu int a = 20; , int b = 3;. Variabel a diinisialisasi dengan nilai 20 dan variabel b diinisialisasi dengan nilai 3.

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan memahami bahwa b += a sama dengan b = b + a. Ini akan menambahkan nilai a ke b dan menyimpan hasilnya kembali di b.
2. Alasan solusi ini karena pemahaman yang benar**, j**ika memahami bahwa b += a berarti b ditambahkan dengan nilai a, maka hasilnya akan sesuai dengan ekspektasi. Di sini, b yang awalnya 3 akan menjadi 23 setelah b += a.
3. Perbaikan kode program dengan cara mendeklarasikan nilai dengan dipisahkan dalam baris berbeda untuk keterbacaan, komentar diperjelas agar lebih sesuai dengan operasi yang dilakukan dan gaya penulisan, menjaga konsistensi dan membuat kode lebih mudah dibaca.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma Operator Penugasan :

1. Mulai Program
2. Mendeklarasikan variable, dua variabel bilangan bulat a dan b dideklarasikan dan diinisialisasi dengan nilai 20 dan 3 masing-masing. Variabel-variabel ini akan digunakan dalam perhitungan.
3. Operasi Penjumlahan:

Nilai a ditambahkan ke b dan hasilnya ditampilkan ke layar menggunakan perintah System.out.println().

1. Operasi Pengurangan:

Nilai b dikurangkan dari a dan hasilnya ditampilkan ke layar.

1. Operasi Perkalian:

Nilai a dikalikan dengan b dan hasilnya ditampilkan ke layar.

1. Operasi Pembagian:

Nilai a dibagi dengan b dan hasilnya ditampilkan ke layar.

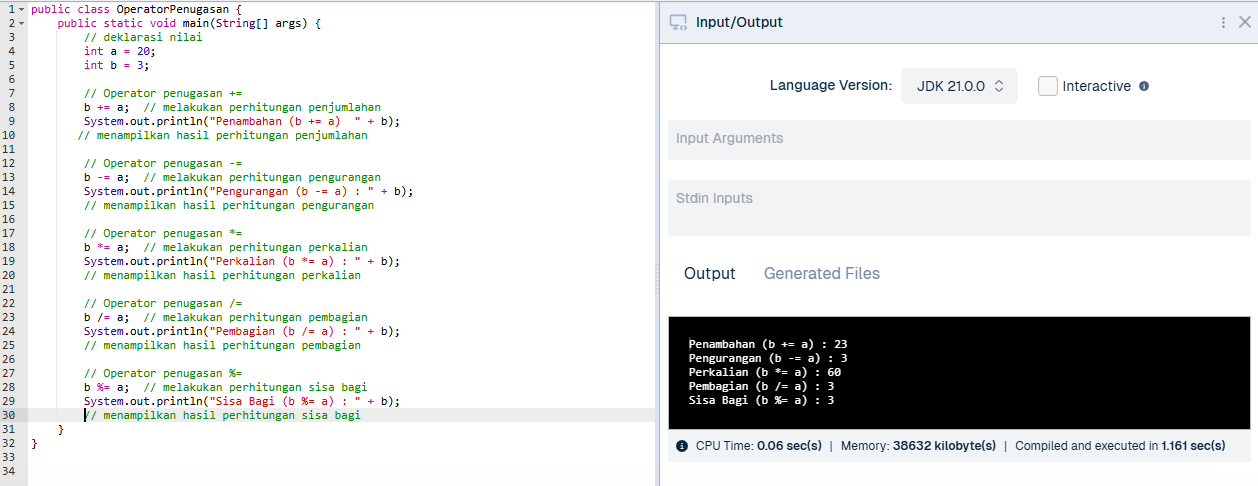
1. Operasi Modulus:

Sisa hasil pembagian a dengan b (modulus) dihitung dan ditampilkan ke layar.

1. Tampilkan Hasil:

System.out.println("Penambahan : " + b); - Ini mencetak hasil akhir ke layar. Output yang dihasilkan adalah "Penambahan : 23".

1. Selesai
2. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran
2. // program ini merupakan komentar satu baris, digunakan untuk memberikan penjelasan singkat mengenai baris kode tertentu.
3. int a = 20; - Komentar menjelaskan bahwa variabel a dideklarasikan dan diinisialisasi dengan nilai 20.
4. int b = 3; - Komentar menjelaskan bahwa variabel b dideklarasikan dan diinisialisasi dengan nilai 3.
5. b += a; - Komentar menjelaskan bahwa operasi ini menambahkan nilai dari a ke b dan menyimpan hasilnya kembali ke dalam b. Juga disertakan penjelasan bahwa ini setara dengan b = b + a.
6. System.out.println("Penambahan : " + b); - Komentar menjelaskan bahwa baris ini mencetak hasil akhir penambahan ke layar, menunjukkan hasil akhir dari variabel b setelah operasi penugasan
7. 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**

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

jawaban Analisa:

1. Program ini melakukan operasi aritmatika dasar (penjumlahan, pengurangan, perkalian, pembagian, dan modulus) pada dua bilangan bulat yang telah ditentukan sebelumnya. Kode program ini merupakan contoh sederhana dari penggunaan operator aritmetika dalam bahasa pemrograman Java. Program ini menunjukkan bagaimana kita dapat melakukan perhitungan sederhana dan menampilkan hasilnya. Meskipun sederhana, konsep yang diimplementasikan dalam program ini sangat penting dalam pemrograman, terutama dalam aplikasi yang melibatkan perhitungan numerik

Algoritma:

1. Inisialisasi: Program memulai dengan mendeklarasikan dua variabel bilangan bulat, a dan b, dan memberikan nilai awal.
2. Perhitungan: Secara berurutan, program melakukan operasi penjumlahan, pengurangan, perkalian, pembagian, dan modulus pada variabel a dan b.
3. Output: Hasil dari setiap operasi kemudian dicetak ke layar menggunakan perintah System.out.println().

Kode program ini mengimplementasikan algoritma di atas dengan jelas. Setiap baris kode memiliki tujuan yang spesifik:

1) Deklarasi Variabel: Menentukan tipe data dan nama variabel yang akan digunakan.

2) Operasi Aritmatika: Melakukan perhitungan menggunakan operator aritmetika seperti +, -, \*, /, dan %.

3) Output: Mencetak hasil perhitungan ke layar menggunakan metode println().

1. Untuk pembelajaran dasar pemrograman: Mengenal Konsep Variabel dan Tipe Data: kode ini memperkenalkan konsep variabel bilangan bulat (integer) dan bagaimana mendeklarasikan serta menginisialisasinya.

Memahami Operator Aritmatika: kode ini menjelaskan penggunaan berbagai operator aritmatika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Belajar Output: kode ini menunjukkan cara mencetak hasil perhitungan ke layar menggunakan System.out.println().

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cut Fitriani G1F024079** | **Operator Relasional** | **10 September 2024** |

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

1. Uraikan permasalahan dan variable

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**  
3.1.  Ubahlah nilai A = 4 dan B = 4 pada Contoh 3. Simpulkan perubahan yang terjadi!

-Pada soal masih ada kesalahan pada output yang dihasilkan dari program belum tepat, dan harus memasukkan variabel int nilaiA = 4

-Diketahui dari soal : variable yang digunakan ada nilaiA dan nilaiB, digunakan dalam operasi perbandingan untuk menentukan hubungan antara dua nilai integer. Variabel ini diinisialisasi dengan nilai 12 dan 4, masing-masing. Hasil: digunakan untuk menyimpan dan menampilkan hasil dari perbandingan antara nilaiA dan nilaiB. Hasil ini kemudian dicetak ke layar untuk menunjukkan hasil dari berbagai operator relasional seperti >, <, >=, <=, ==, dan !=.

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

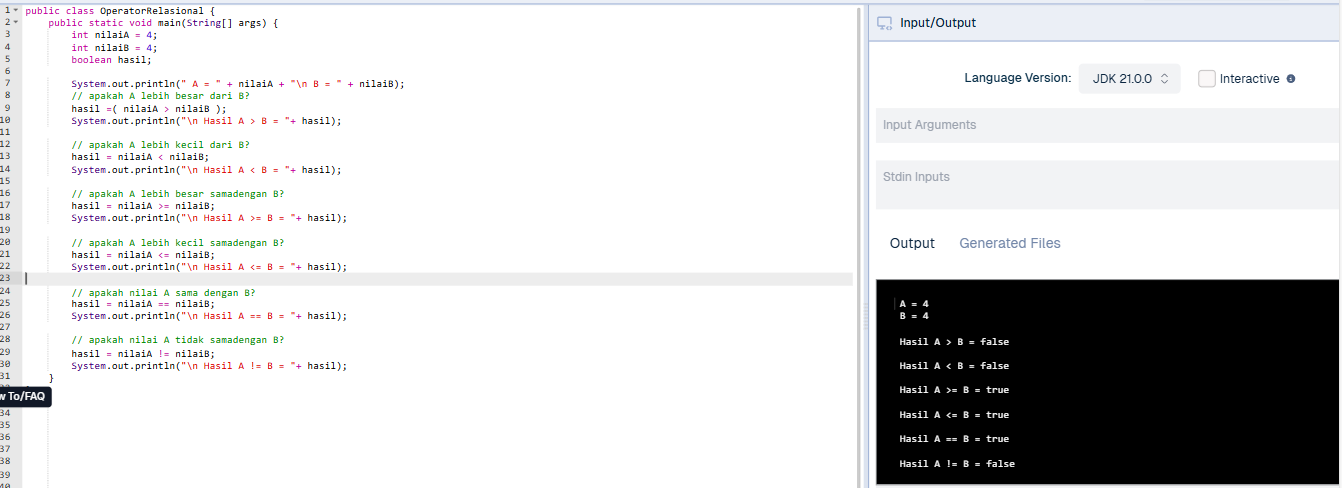
1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara, verifikasi bahwa keluaran dari setiap operasi perbandingan adalah sesuai dengan yang diharapkan. Misalnya, nilaiA > nilaiB harus menghasilkan true jika nilaiA lebih besar dari nilaiB, dan seterusnya. Simpulkan hasil tunjukkan bagaimana hasil dari setiap perbandingan sesuai dengan logika yang diterapkan. Sebagai contoh, jika nilaiA adalah 4 dan nilaiB adalah 4, maka perbandingan seperti nilaiA < nilaiB harus menghasilkan false.
2. Alasan solusi ini karena kode ini sudah memenuhi tujuannya untuk mendemonstrasikan penggunaan operator relasional dengan nilai tetap, memberikan gambaran yang jelas tentang bagaimana setiap operator bekerja. Namun, untuk meningkatkan fleksibilitas dan efisiensi, bisa mempertimbangkan penanganan input dinamis dan optimasi format output seperti yang telah diajarkan.
3. Perbaikan kode program dengan cara memasukkan variabel int nilaiA = 4 dan kode siap untuk di jalankan.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma Operator Penugasan:

1. Mulai Program
2. Memasukkan variable nilai intA menjadi 4
3. Kode program siap dijalankan.
4. Selesai
5. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran

// program ini merupakan komentar satu baris, digunakan untuk memberikan penjelasan singkat mengenai baris kode tertentu.

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**

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

jawaban Analisa:

1. Kode program di atas bertujuan untuk membandingkan dua bilangan bulat (nilaiA dan nilaiB) menggunakan operator relasional dalam bahasa pemrograman Java. Operator-operator relasional yang digunakan adalah:

>: Lebih besar dari

<: Lebih kecil dari

>=: Lebih besar sama dengan

<=: Lebih kecil sama dengan

==: Sama dengan

!=: Tidak sama dengan

Kode program ini merupakan contoh sederhana dari penggunaan operator relasional dalam bahasa Java. Program ini dapat digunakan sebagai dasar untuk mempelajari konsep perbandingan yang lebih kompleks dalam pemrograman. Tidak ada permasalahan yang signifikan dalam kode program ini. Kode berjalan dengan baik dan menghasilkan output yang sesuai dengan perbandingan yang dilakukan.

Algoritma :

1. Deklarasi variabel: Mendefinisikan dua variabel bilangan bulat (nilaiA dan nilaiB) dan satu variabel boolean (hasil) untuk menyimpan hasil perbandingan.
2. Inisialisasi variabel: Memberikan nilai awal pada variabel nilaiA dan nilaiB.
3. Perbandingan: Melakukan perbandingan antara nilaiA dan nilaiB menggunakan berbagai operator relasional.
4. Penugasan: Menyimpan hasil perbandingan ke dalam variabel hasil.
5. Output: Mencetak hasil perbandingan ke layar.

Tujuan utama dari program ini adalah untuk memahami konsep perbandingan dalam pemrograman dan cara menggunakan operator relasional untuk membandingkan nilai.

1. Kode ini menggunakan berbagai operator relasional seperti >, <, >=, <=, ==, dan != untuk membandingkan nilaiA dan nilaiB. Setiap operator relasional menghasilkan nilai boolean (true atau false) berdasarkan hasil perbandingan. Nilai boolean inilah yang menjadi dasar untuk pengambilan keputusan.

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cut Fitriani G1F024079** | **Operator Increment dan Decrement** | **10 September 2024** |

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

1. Uraikan permasalahan dan variable

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. Berikan saran operasi apa yang diperlukan (pre/post increment, pre/post decrement) agar Contoh 4 menghasilkan nilai a = 5 dan b = 6?   
4.2. Simpulkan hasil eksperimen Anda!

**Jawaban:**

4.1. Saran operasi yang saya gunakan yaitu post increment (a++) menaikkan nilai a setelah nilai a digunakan dalam pernyataan. Dan memastikan bahwa a tetap 5 setelah mencetak b, dan b harus mencetak nilai a yang sudah meningkat setelah a++ dieksekusi.

4.2. Post-Increment (a++), yaitu menaikkan nilai variabel setelah nilai tersebut digunakan dalam ekspresi. Dalam kode yang diubah, kita tidak mengubah nilai a secara langsung menggunakan a++. Sebagai gantinya, kita menambahkan 1 dalam ekspresi untuk mencapai nilai b yang diinginkan tanpa mengubah nilai asli dari a. Dan dengan menambahkan 1 ke a dalam ekspresi System.out.println("b: " + (a + 1)), kita mendapatkan nilai b sebagai 6 sementara a tetap 5.

-Pada soal masih ada kesalahan, yaitu **a**: Setelah pernyataan System.out.println("b: " + (a++)); dieksekusi, nilai a menjadi 6.

**b**: Nilai yang dicetak untuk b adalah nilai a sebelum increment, yaitu 5.

Jadi, hasil akhir dari output adalah:

a: 5

b: 5

-Diketahui dari soal : variable yang digunakan bertipe data int. Variabel ini dideklarasikan dan diinisialisasi dengan nilai 5. Dalam pernyataan System.out.println("b: " + (a++));, nilai a dicetak sebagai bagian dari ekspresi post-increment (a++), yang menghasilkan nilai 5 sebelum a ditingkatkan menjadi 6.

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara untuk mendapatkan hasil a = 5 dan b = 6, harus memodifikasi kode sedemikian rupa sehingga b mencetak nilai 6 tetapi a tetap 5. Karena post-increment (a++) secara otomatis meningkatkan nilai a setelah digunakan.
2. Alasan solusi ini int b = a + 1 : Menyimpan nilai a yang telah ditingkatkan (menambahkan 1) ke dalam variabel b. Nilai ini adalah 6. System.out.println("a: " + a) : Mencetak nilai asli dari a, yang tetap 5. System.out.println("b: " + b) : Mencetak nilai b yang baru dihitung sebagai 6.
3. Perbaikan kode program dengan cara perhitungan terpisah untuk b dan menghindari perubahan langsung pada nilai a dengan menggunakan variabel sementara atau operasi tambahan. **Variabel b**: Menghitung nilai b sebagai a + 1 sehingga b menjadi 6 tanpa mempengaruhi nilai a.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma Operator Post Increment:

1. Mulai Program
2. Mendeklarasikan variabel a dengan tipe data int dan memberi nilai 5.
3. Mencetak nilai awa; dari variabel ke layer
4. Melakukan operasi post increment pada variavel a (menambahkan 1)
5. \Menyimpan nilai a sebelum increment ke dalam variabel b
6. Mencetak nilai akhir dari variabel b ke layar
7. Selesai
8. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran

Mendeklarasikan variabel sebuah variabel bernama a dengan tipe data int dan memberikan nilai awal 5.

Mencetak nilai, baris System.out.println("a: " + a); digunakan untuk mencetak nilai dari variabel a ke layar.

// program ini merupakan komentar satu baris, digunakan untuk memberikan penjelasan singkat mengenai baris kode tertentu.

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**

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

jawaban Analisa:

1. Kesimpulannya yaitu pre-increment(++a) nilai variabel diubah terlebih dahulu dan digunakan operasi selanjutnya dan post-increment nilai variabel digunakan terlebih dahulu kemudian diubah setelah operasi .

Algoritma :

1. Mendeklarasikan dua variabel integer, a dan b.
2. Memberikan nilai awal pada a.
3. Mencetak nilai a ke layar.
4. Mengincrement nilai a dan menyimpannya ke b.
5. Mencetak nilai b ke layar.

Kode ini mungkin dibuat untuk menjelaskan cara kerja operator increment (++) dalam bahasa Java. Operator ini digunakan untuk menambah nilai suatu variabel dengan 1. Kode ini bisa jadi merupakan latihan sederhana untuk pemula dalam belajar pemrograman Java dan menunjukkan cara mendeklarasikan dan menggunakan variabel dalam sebuah program.

1. Dasar pengambilan keputusan yaitu pemilihan antara pre dan post-increment dan decrement bergantung pada kapan kita ingin variabel diubah dalam urutan operasi . dalam contoh ini untuk mendapatkan b = 6 sementara a=5 di output awal maka post-increment (a++) digunakan sehingga nilai a digunakan terlebih dahulu dan ditambah setelah digunakan untuk nilai b.

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cut Fitriani G1F024079** | **Operator Logika** | **10 September 2024** |

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

1. Uraikan permasalahan dan variabel

public class OperatorLogika {  
    public static void main(String[] args) {  
        // deklarasi nilai  
        boolean a = true;  
        boolean b = false;  
          
        System.out.println("Hasil logika (a && b) : " + (a && b));  //menampilkan hasil logika AND  
}    }

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

**Latihan 5**  
5.1.  Rekomendasikan berapa nilai a dan b apabila ingin menghasilkan luaran *true*dengan operator && dan  operator | | ?    
5.2.  Berikan kesimpulan dari latihan 5.1.

**Jawab:**

5.1 Untuk menghasilkan luaran true dengan operator && (AND), Kedua nilai a dan b harus true .contoh : a = true dan b = true. Operator || (OR) yaitu setidaknya salah satu dari a atau b harus true .contoh : a = true dan b false  ,a = false dan b = true dan a = true b = true

5.2 Kesimpulan dari Latihan 5.1 adalah operator && (AND) hasilnya akan true hanya jika kedua operan ( a dan b) bernilai true, jika salah satu operan bernilai false maka hasilnya akan false. Operator OR|| Untuk menghasilkan true ,cukup salah satu operan yang true . singkatnya AND besifat lebih kaku karena semua kondisi harus terpenuhi. OR bersifat lebih fleksibel ,cukup satu kondisi yang terpenuhi.

-Pada soal masih ada kesalahan, jika salah satu dari a atau b bernilai false, maka hasil a && b akan selalu false. Jadi, Anda harus memastikan bahwa kedua variabel adalah true untuk mendapatkan hasil true.

-Diketahui dari soal : variable yang digunakan yaitu, **a**: Bertipe data boolean dan diinisialisasi dengan nilai true. **b**: Bertipe data boolean dan diinisialisasi dengan nilai false.

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara untuk menghasilkan hasil true dengan operator logika && (AND) dan || (OR), perlu memahami bagaimana kedua operator ini bekerja.
2. Alasan solusi ini karena agar output yang diinginkan akan mencetak dan hasil dengan benar.
3. Perbaikan kode program dengan cara adalah hasilnya akan true hanya jika kedua operan ( a dan b) bernilai true, jika salah satu operan bernilai false maka hasilnya akan false.

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

1) Algoritma adalah langkah-langkah penyelesaian masalah.  
 Algoritma Operator Logika :

a) Mulai Program

b) Deklarasikan dua variabel Boolean a dan b

c) inisialisasi variabel yang akan dimasukkan kedalam kode

d) Gunakan operator AND (&&) atau OR (||) untuk menggabungkan nilai a dan b

e) Tampilkan hasil operasi logika AND ( a && b ) kelayar

f) selesai

1. Kode program dan luaran

A screenshot of a computer

Description automatically generated

1. Screenshot/ Capture potongan kode dan hasil luaran

// program ini merupakan komentar satu baris, digunakan untuk memberikan penjelasan singkat mengenai baris kode tertentu.

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**

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

jawaban Analisa:

1. Kode program ini bertujuan untuk mendemonstrasikan penggunaan operator logika AND (&&) dan OR (||) dalam bahasa pemrograman Java. Operator-operator ini digunakan untuk menggabungkan beberapa kondisi boolean (benar atau salah) menjadi satu kondisi yang lebih kompleks.

Algoritma:

1. Mendeklarasikan variabel, variabel boolean a dengan nilai false dan variabel boolean b dengan nilai true. Variabel boolean hanya dapat memiliki dua nilai, yaitu true (benar) atau false (salah).
2. Operasi Logika, dilakukan operasi logika AND (&&) antara variabel a dan b. Hasil operasi ini akan menjadi false karena salah satu kondisi (a) bernilai false. Dilakukan operasi logika OR (||) antara variabel a dan b. Hasil operasi ini akan menjadi true karena salah satu kondisi (b) bernilai true.
3. Mencetak hasil

Hasil dari kedua operasi logika tersebut dicetak ke layar menggunakan perintah System.out.println().

Kode Program:

public class OperatorLogika {

public static void main(String[] args) {

boolean a = false;

boolean b = true;

System.out.println("Hasil logika (a && b): " + (a && b)); // Menampilkan hasil logika AND

System.out.println("Hasil logika (a || b): " + (a || b)); // Menampilkan hasil logika OR

}

1. Kode ini sering digunakan sebagai contoh dasar untuk menjelaskan konsep operasi logika dalam pemrograman.

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cut Fitriani G1F024079** | **Operator Kondisional** | **10 September 2024** |

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

1. Uraikan permasalahan dan variabel

 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**  
Rekomendasikan apa bentuk tanda operator agar nilai = 60 memenuhi untuk Lulus !

-Pada soal masih ada kesalahan pada operator > hanya memeriksa apakah nilai lebih besar dari 60. Artinya, nilai 60 tidak termasuk dalam kategori "Lulus", karena 60 tidak lebih besar dari 60; ia hanya sama dengan 60.

-Diketahui dari soal : variable yang digunakan yaitu, **a**: Bertipe data boolean dan diinisialisasi dengan nilai true. **b**: Bertipe data boolean dan diinisialisasi dengan nilai false.

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara operator >= memeriksa apakah nilai lebih besar dari atau sama dengan 60. Dengan menggunakan operator ini, nilai 60 akan memenuhi kriteria dan menghasilkan "Lulus", karena 60 sama dengan 60.
2. Alasan solusi ini karena agar output yang diinginkan akan mencetak dan hasil dengan benar.
3. Perbaikan kode program dengan cara menggunakan operator >= dalam ekspresi ternary. Ini memastikan bahwa nilai 60 termasuk dalam kategori "Lulus", bukan hanya nilai yang lebih besar dari 60.

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

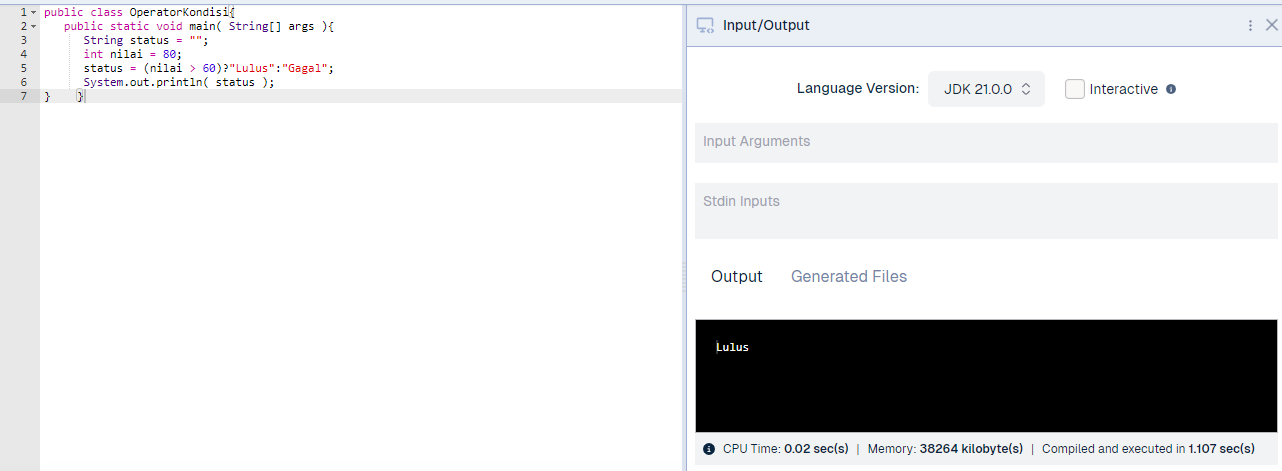
1) Algoritma adalah langkah-langkah penyelesaian masalah.  
 Algoritma Operator Kondisional:

a) Mulai Program

b) Ubah operator > lebih besar dari menjadi < lebih kecil dari agar nilainya lulus.

c) selesai

1. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran

// program ini merupakan komentar satu baris, digunakan untuk memberikan penjelasan singkat mengenai baris kode tertentu.

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**

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

jawaban Analisa:

1. Program ini dirancang untuk menentukan status kelulusan seseorang berdasarkan nilai yang diperoleh. Jika nilai lebih besar dari atau sama dengan 60, maka statusnya "Lulus", jika tidak, maka statusnya "Gagal". Kode program ini merupakan implementasi sederhana dari konsep percabangan (kondisional) dalam bahasa pemrograman Java. Program ini menggunakan operator ternary untuk menentukan status kelulusan berdasarkan nilai yang diberikan. Operator ternary adalah cara yang ringkas untuk menulis pernyataan if-else dalam satu baris.

Algoritma:

1. Deklarasi variabel
2. Inisialisasi nilai
3. Perbandingan dan penentuan status: Menggunakan operator perbandingan >= untuk memeriksa apakah nilai lebih besar dari atau sama dengan 60.

\* Jika benar, maka nilai variabel status diset menjadi "Lulus".

\* Jika salah, maka nilai variabel status diset menjadi "Gagal". Ini dilakukan dalam satu baris menggunakan operator ternary.

4) Mencetak hasil

5) Selesai

Kode Program:

public class OperatorKondisi {

public static void main(String[] args) {

int nilai = 80;

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

System.out.println(status);

}

1. Kode program yang di berikan merupakan contoh sederhana dari pengambilan keputusan dalam pemrograman, khususnya dalam bahasa Java. Dalam kode ini, keputusan diambil berdasarkan suatu kondisi atau syarat tertentu.

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Cut Fitriani G1F024079** | **Operator Bitwise** | **10 September 2024** |

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 );    
            
        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**  
Evaluasi penyebab hasil ~a = -11 ? Buktikan jawaban Anda dalam perhitungan biner!

Jawab :

1. Representasi Biner dari a

Pertama, kita perlu mengetahui nilai biner dari a. Dalam kode Anda, a adalah 10.

Representasi Biner dari 10:

* Decimal 10 dalam biner adalah 0000 1010 (dalam format 8-bit).

2. Operator ~ (Bitwise NOT)

Operator bitwise ~ membalikkan setiap bit dari operandnya. Ini berarti semua bit 0 akan menjadi 1, dan semua bit 1 akan menjadi 0.

3. Konversi dari Biner ke Desimal (2's Complement)

Untuk mengonversi hasil biner 1111 0101 kembali ke desimal, kita perlu mengubahnya dari format 2's complement menjadi desimal. Berikut langkah-langkahnya:

1. Hitung nilai positif dari 1111 0101 dalam 2's complement:
2. Karena kita mulai dengan representasi negatif, hasil akhirnya adalah -11.

**Kesimpulan**

Hasil dari operasi ~a adalah -11 karena operator bitwise ~ membalikkan setiap bit dari a, dan nilai biner yang dihasilkan (1111 0101) merepresentasikan angka negatif -11 dalam format 2's complement.

Ringkasan Perhitungan:

* Nilai 10 dalam biner: 0000 1010
* Hasil ~10 dalam biner: 1111 0101
* Nilai desimal dari 1111 0101: -11

**Refleksi**

Dari tugas yang diberikan disini kami dapat mempelajari materi operator java dan mengetahui berbagai jenis operator dan kode programnya. Operator merupakan simbol dalam program untuk melakukan suatu operasi atau memproses data hingga memberikan hasil.  
Contoh : a + b      
simbol “ + “ merupakan operand untuk melakukan penjumlahan dari a dan b. Operator penjumlahan tergolong sebagai operator binary karena melibatkan dua operand,. **Ekspresi**adalah pernyataan yang melibatkan variabel, data, dan konstanta. Ekspresi sangat penting dalam pemrograman karena perhitungan banyak menggunakan ekspresi. Penyusunan ekspresi memakai lambang-lambang tertentu yang disebut operator.