**Komputer Dan Pemrograman**

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **NAMA:MUHAMMAD NABIL**  **NPM:G1F024068** | **Operator** | **14 September 2024** |
| **Unit 1 Operator** | | |
| 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,. Contoh lain : -c simbol “ – “ (minus) merupakan unary, karena hanya memiliki satu operand (yaitu c).  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. | | |
| **Unit 2 Jenis Operator** | | |
| **1. Operator Aritmatika: +, -, \*, /** Operator aritmatika digunakan untuk melakukan perhitungan matematika. Jika operator memiliki prioritas yang sama, operator sebelah kiri akan diutamakan untuk dikerjakan terlebih dahulu. Tanda kurung biasa digunakan untuk mengubah urutan pengerjaan.  Misalnya:    x = (2 + 3) \* 2 ;     Menghasilkan x = 10, sebab 2 + 3 dahulu dan hasilnya dikalikan dengan 2.  **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));  **Latihan 1.** 1.1.  Tambahkan baris System.out.println("a + b = "  + (a + b));  Ubahlah operator ( + ) dengan tanda ( -, \*, /, %) 1.2.  Analisa perhitungan matematika yang terjadi!  **Jawab:**  1.1      Kode:  public class operator {  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));  }  }  1.2  Nilai a: 20  Nilai b: 3  Penjumlahan:  Operasi: a + b  Hasil: 20 + 3 = 23  Output: "a + b = 23"  Pengurangan:  Operasi: a - b  Hasil: 20 - 3 = 17  Output: "a - b = 17"  Perkalian:  Operasi: a \* b  Hasil: 20 \* 3 = 60  Output: "a \* b = 60"  Pembagian:  Operasi: a / b  Hasil: 20 / 3 ≈ 6  Output: "a / b = 6"  Sisa Bagi:  Operasi: a % b  Hasil: 20 % 3 = 2 (sisa dari pembagian)  Output: "a % b = 2"  **2. Operator Penugasan  ( = )** Operator Penugasan adalah operator yang digunakan untuk memberikan nilai ke dalam variabel tertentu.  **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=0         System.out.println("Sisa Bagi: " + b);     } }  **Luaran:** Penambahan : 23 Pengurangan : 3 Perkalian : 60 Pembagian : 3 Sisa Bagi: 3  **Latihan 2.** 2.1.  Bandingkan hasil Contoh 1 dengan Contoh 2!  **Jawab :**  2.1  Kedua kode memiliki hasil yang berbeda karena operasi yang dilakukan berbeda.  -Contoh Pertama:  Hanya melakukan operasi penjumlahan dan mencetak hasilnya.  Hasilnya adalah: "a + b = 23"  -Contoh Kedua:  Melakukan operasi penugasan berurutan dan mencetak hasilnya setelah setiap operasi.  Hasilnya adalah:  Penambahan : 23  Pengurangan : 3  Perkalian : 60  Pembagian : 3  Sisa Bagi: 3  **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 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  **Jawab :**  3.1  Nilai A dan B: Kedua nilai A dan B sekarang sama, yaitu 4.  -Lebih Besar dari (>): hasil = nilaiA > nilaiB => hasil = 4 > 4 => hasil = false  -Lebih Kecil dari (<): hasil = nilaiA < nilaiB => hasil = 4 < 4 => hasil = false  -Lebih Besar Sama dengan (>=): hasil = nilaiA >= nilaiB => hasil = 4 >= 4 => hasil = true  -Lebih Kecil Sama dengan (<=): hasil = nilaiA <= nilaiB => hasil = 4 <= 4 => hasil = true  -Sama dengan (==): hasil = nilaiA == nilaiB => hasil = 4 == 4 => hasil = true  -Tidak Sama dengan (!=): hasil = nilaiA != nilaiB => hasil = 4 != 4 => hasil = false  Output    3.2  Perubahan nilai A dan B menjadi 4 dan 4 akan menghasilkan output yang berbeda dari kode asli. Karena kedua nilai sama, maka semua perbandingan logika akan menghasilkan true kecuali untuk perbandingan > dan < yang akan menghasilkan false.  **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 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!  **Jawaban :**  4.1  Post Increment vs Pre Increment:   * Post Increment: Nilai a digunakan terlebih dahulu, lalu ditingkatkan. * Pre Increment: Nilai a ditingkatkan terlebih dahulu, lalu digunakan.   Post Decrement vs Pre Decrement:   * Post Decrement: Nilai c digunakan terlebih dahulu, lalu dikurangi. * Pre Decrement: Nilai c dikurangi terlebih dahulu, lalu digunakan.   **5. Operator Logika:  &&, ||, !** Operator Logika untuk membandingkan dua nilai variabel atau lebih, hasilnya boolean true atau false. Asumsikan variabel a bernilai true, b bernilai false dan c bernilai true. Perbedaan dasar antara operator && dan & adalah && mensupports evaluasi per bagian, sementara operator & tidak. Misal pernyataan: exp1 AND exp2 Dengan operator && akan mengevaluasi pernyataan exp1, dan segera mengembalikan nilai false dan menyatakan bahwa exp1 bernilai false. Jika exp1 bernilai false, operator tidak akan pernah mengevaluasi exp2 karena hasil operasi operator akan menjadi false tanpa memperhatikan nilai dari exp2. Dengan operator & selalu mengevaluasi kedua nilai dari exp1 danexp2 sebelum mengembalikan suatu nilai jawaban. Demikian juga antara operator || dan |. Tanda && dan & serta || dan | memiliki arti berbeda. & dan | akan tetap mengevaluasi kondisi apa pun yang terjadi. && akan menghentikan evaluasi kondisi jika salah satu ekspresi bernilai salah. Jika salah satu kondisi ekspresi salah, dipastikan hasil operasi logika bernilai salah. Sedangkan || akan menghentikan evaluasi kondisi jika salah satu ekspresi bernilai benar. Jika salah satu kondisi ekspresi benar, dipastikan hasil operasi logika bernilai benar.  **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! 5.3.  Apabila diketahui pernyataan a || b && a || !b. Uraikan urutan logika yang akan dikerjakan! Analisa luaran true atau false dari pernyataan tersebut!  **Jawab :**  5.1  5.2    Perubahan nilai a dan b menjadi false akan menghasilkan output yang berbeda dari kode asli. Pada operasi AND, hasilnya akan selalu false karena salah satu operasi atau kedua operasi harus benar untuk membuat hasilnya benar. Pada operasi OR, hasilnya akan benar jika salah satu operasu benar, tetapi jika kedua operasi salah, maka hasilnya akan salah.  Perbedaan antara kode asli dan kode dengan nilai diubah adalah:   * Operasi AND (&&): Kedua kode akan menghasilkan false karena tidak ada salah satu operasi yang benar. * Operasi OR (||): Kode asli akan menghasilkan true karena salah satu operasi (a) adalah true, sedangkan kode dengan nilai diubah akan menghasilkan false karena tidak ada salah satu operasi yang benar.   Output    **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** Berdasarkan Contoh 6, ubahlah nilai = 60. Analisis hasil dan proses yang terjadi!  **Jawab :**    hasil dari kode tersebut adalah bahwa nilai status akan menjadi "Gagal" karena nilai nilai (80) tidak lebih besar dari 90.  Proses yang Terjadi:   * Pengujian Kondisi: Kode menguji apakah nilai nilai (80) lebih besar dari 90. * Penetapan Status: Jika kondisi tidak terpenuhi (80 > 90 adalah false), maka nilai status akan menjadi "Gagal". * Pencetakan Hasil: Nilai status "Gagal" akan dicetak ke konsol.   Output    **7.    Operator Bitwise: &, |, ^, ~, <<, >>, >>>** 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.  **Operator    Nama             Contoh    Biner                 Hasil (biner)    Hasil (decimal)** &                 AND               10 & 12    1010 & 1100             1000             8 |                  OR                  10 | 12     1010 | 1100               1110             14 ^                 XOR                10 ^ 12    1010 ^ 1100              0110             6 ~                 NOT                    ~10    ~1010                         0101            -11 <<               Geser kiri        10 << 1    1010 << 1              10100             20 >>               Geser kanan   10 >> 1    1010 >> 1                  101             5  **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!  **Jawab :**  **Perhitungan 1**: hasil = a & b;  Operasi: Bitwise AND (&)  Operasi bitwise AND (&) menghasilkan 1 jika kedua bit yang diperiksa memiliki nilai 1. Jika salah satu atau kedua bit memiliki nilai 0, maka hasilnya akan 0.  Contoh  a = 10 (1010 biner)  b = 7 (0111 biner)  Hasil bitwise AND:  1010 (a)  0111 (b)  ------ 0010 (hasil)  Jadi, hasilnya adalah: 2 (0010 biner)  Kesimpulan: Hasil dari a & b : 2  **Perhitungan 2**: hasil = a | b;  Operasi: Bitwise OR (|)  Operasi bitwise OR (|) menghasilkan 1 jika salah satu atau kedua bit yang diperiksa memiliki nilai 1. Jika kedua bit memiliki nilai 0, maka hasilnya akan 0.  Contoh  a = 10 (1010 biner)  b = 7 (0111 biner)  Hasil bitwise OR:  1010 (a)  0111 (b)  ------ 1111 (hasil)  Jadi, hasilnya adalah: 15 (1111 biner)  Kesimpulan : Hasil dari a | b : 15  **Perhitungan 3**: hasil = a ^ b;  Operasi: Bitwise XOR (^)  Operasi bitwise XOR (^) menghasilkan 1 jika salah satu bit yang diperiksa memiliki nilai 1 dan bit lainnya memiliki nilai 0. Jika kedua bit memiliki nilai yang sama (baik 0 atau 1), maka hasilnya akan 0.  Contoh  a = 10 (1010 biner)  b = 7 (0111 biner)  Hasil bitwise XOR:  1010 (a)  0111 (b)  ------ 1101 (hasil)  Jadi, hasilnya adalah: 13 (1101 biner)  Kesimpulan : Hasil dari a ^ b : 13 | | |
|  | | |
|  | | |

Contoh Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Putri  G1A000001** | **Tipe Data** | **26 Agustus 2022** |

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

1. Uraikan permasalahan dan variabel

Contoh:

Tuliskan kembali soal:

Pada soal masih ada pesan kesalahan \_\_\_\_\_\_\_\_\_\_\_\_

Atau  
Diketahui dari soal : variabel \_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara \_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Alasan solusi ini karena \_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Perbaikan kode program dengan cara \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Misalkan algoritma memasak mi instan:

1. Masak air
2. Buka bungkus
3. Masukkan mie
4. Masukkan bumbu
5. Hasilnya mie matang, taruh di piring
6. Mie siap disantap.
7. Kode program dan luaran



1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan   
   Contoh:

Luaran sudah sesuai dengan program yang disusun.   
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.  
  
(Tuliskan penjelasan dari program yang dibuat, apakah kode dan luaran sudah benar?)

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

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

Contoh jawaban Analisa:

Pada program itu saya menggunakan bentuk kelas public karena \_\_\_\_\_\_\_\_\_\_  
 Perbaikan program dengan menambahkan \_\_\_\_\_\_\_\_\_\_\_\_ karena struktur java   
 mengharuskan \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
 (penjelasan analisa mengulangi kembali materi yang sudah diberikan)  
 (penjelasan mengkaitkan dengan materi yang ada)

1. **Evaluasi**
2. Apa konsekuensi/dampak dari kode program yang dibuat?
3. Evaluasi input program, proses perhitungan, dan luaran yang dihasilkan! (jika ada)

Contoh jawaban Evaluasi:

Pada program itu saya mengkonversi bentuk kelas public karena \_\_\_\_\_\_\_\_\_\_  
 Setelah dikonversi, saya mengevaluasi bahwa tipe data \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ lebih baik   
 digunakan untuk bentuk data seperti \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(penjelasan evaluasi mengulangi kembali materi yang sudah diberikan dan   
mengetahui kekurangan dari materi hasil eksperimen   
(misal tipe data \_\_\_ ternyata tidak dapat dipakai untuk \_\_\_\_\_\_ karena \_\_\_\_\_ )

1. **Kreasi**
2. Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?
3. Susunlah hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

Contoh jawaban Kreasi:

Pada program itu saya mengkonversi bentuk kelas public karena \_\_\_\_\_\_\_\_\_\_  
 Setelah dikonversi, saya mengevaluasi bahwa tipe data \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ lebih baik   
 digunakan untuk bentuk data seperti \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Saya telah mencoba mengubah menjadi kelas private dan protected, ternyata menghasilkan \_\_\_\_\_\_\_\_\_\_

Berarti kelas private dan protected mempengaruhi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
(sampaikan temuan Anda yang baru diketahui, eksperimen baru diluar materi yang diberikan)

(penjelasan kreasi mengulangi kembali materi yang sudah diberikan dan

menambahkan pengetahuan baru dari pengalaman dari hasil eksperimen )

Lanjutkan ke soal nomor 2 – 3 – … – dan seterusnya

**Refleksi**

(Tuliskan singkat tentang pengalaman belajar, pemaknaan pengetahuan yang baru, tantangan yang dihadapi pada minggu tersebut. Ringkasan singkat dari semua soal, bukan per soal)