|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Anissa Shanniyah Aprilia**  **G1A022044** | **Operator** | **5 September 2022** |
| **[No.1] Identifikasi Masalah:** | | |
| **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  **Latihan 1.** 1.1.  Tambahkan baris System.out.println("a + b = "  + (a + b));  Ubahlah operator ( + ) dengan tanda ( -, \*, /, %) 1.2.  Analisa perhitungan matematika yang terjadi!  Pada latihan ini dapat kita lihat kita harus menggunakan semua jenis operator yaitu pengurangan, perkalian, pembagian, dan sisa bagi. Lalu menganalisa perhitungan matematika yang terjadi.  Sumber informasi:  <https://pdfslide.tips/documents/laporan-praktikum-3-561d7b57cfed9.html>  <https://kumparan.com/how-to-tekno/operator-aritmatika-pengertian-jenis-jenis-dan-contohnya-1xWSjeRCftV> | | |
| **[No.1] Analisis dan Argumentasi** | | |
| 1.1 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.  Perbaikan kode program dengan cara menambahkan operator penghubung kedua  variable pada soal menjadi :  System.out.println("a + b = " + (a + b));  Lalu merubah operator (+) dengan operator pengurangan, perkalian, pembagian, dan  sisa bagi (-,\*./.%). | | |
| **[No.1] Penyusunan Algoritma dan Kode Program** | | |
| Hasil luarannya sudah sesuai dengan program yang disusun dan analisa saya, dan juga penggunaan operator aritmatika yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.  1.2 Perhitungan matematika yang terjadi pada program ini adalah pertambahan, pengurangan, perkalian, pembagian, dan sisa bagi(modulus) | | |
| **[No.1] Kesimpulan** | | |
| Operator aritmatika digunakan untuk melakukan perhitungan matematika. Jika operator memiliki prioritas yang sama, operator sebelah kiri akan diutamakan untuk dikerjakan terlebih dahulu, serta tanda kurung biasanya digunakan untuk mengubah urutan pengerjaan.  Pada program ini saya menambahkan baris  System.out.println("a + b = "  + (a + b));  Lalu saya mengganti operator (+) dengan tanda (-,\*,/,%). Karena pada soal di perintahkan untuk menambahkan baris diatas, lalu saya mendapatkan hasil:  a+b = 13 a-b = 17  a\*b =60 a/b = 6  a%b = 2 | | |
| **No.2] Identifikasi Masalah:** | | |
| **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!  Pada latihan ini kita hanya perlu mebandingkan hasil contoh 1 dan contoh 2  Sumber Informasi:  <https://pdfslide.tips/documents/laporan-praktikum-3-561d7b57cfed9.html>  <https://www.duniailkom.com/tutorial-belajar-c-plus-plus-jenis-jenis-operator-assignment-bahasa-c-plus-plus/> | | |
| **[No.2] Analisis dan Argumentasi** | | |
| 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. Sedangkan operator penugasan adalah operator yang digunakan untuk memberikan nilai ke dalam variabel tertentu.  Pada contoh 1 jenis operator yang digunakan adalah operator aritmatika sedangkan pada contoh 2 jenis operator yang digunakan adalah operator penugasan.  Pada kedua contoh ini terdapat 2 variabel yang sama dan nilai yang sama, yaitu variabel a dengan nilai 20 dan variabel b dengan nilai 3.  Perbedaan hasil pada contoh 1 dan contoh 2 disebabkan oleh pendeklarasian nilainya, pada contoh 1 menggunakan deklarasi nilai awal untuk dioperasikan pada semua operator, sedangkan pada contoh 2 menggunakan deklarasi awal integer b di awal dan setelah itu dilanjutkan dengan menggunakan nilai b yang telah dioperasikan sebelumnya. | | |
| **[No.2] Penyusunan Algoritma dan Kode Program** | | |
| **Contoh 1**  **Contoh 2** | | |
| **[No2] Kesimpulan** | | |
| Perbedaan hasil pada contoh 1 dan contoh 2 disebabkan oleh pendeklarasian nilainya, pada contoh 1 menggunakan deklarasi nilai awal untuk dioperasikan pada semua operator, sedangkan pada contoh 2 menggunakan deklarasi awal integer b di awal dan setelah itu dilanjutkan dengan menggunakan nilai b yang telah dioperasikan sebelumnya.  Dasar pengambilan keputusan saya adalah menganalisis contoh 1 dan 2 kemudian  saya membandingkannya. | | |

|  |
| --- |
| **[No.3] Identifikasi Masalah:** |
| **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!  Pada latihan ini kita diperintahkan untuk mengubah nilai A=4 dan B=4 lalu analisa perubahan yang terjadi setelah itu bandingkan bagaimana perbedaan nilai A dan dapat mempengaruhi nilai luaran.  Sumber Informasi:  <http://staffnew.uny.ac.id/upload/131781453/pendidikan/MATEMATIKA.pdf>  <https://rpubs.com/fauzil_h/817277> |
| **[No.3] Analisis dan Argumentasi** |
| Pada soal 3.1 kita diperintahkan untuk mengubah A=12 dan B=4 menjadi A=4 dan B=4.Perubahan yang akan terjadi  A>B akan berubah menjadi false karena nilai A dan B sama  A==B akan berubah menjadi true karena yang tadinya nilainya tidak sama menjadi sama  A!=B akan berubah menjadi false Karena yang tadinya nilainya tidak sama menjadi sama  Pada soal 3.2 nilai A dan B menjadi sama, ada 4 status yang berubah yaitu A>B, A<=B,  A==B, dan A!=B. |
| **[No.3] Penyusunan Algoritma dan Kode Program** |
| Sebelum diubah      Setelah diubah      Luaran sudah sesuai dengan program yang disusun, yaitu bernilai true untuk operator relasional yang bernilai “sama dengan” dan bernilai false untuk operator relasional yang bernilai “tidak sama dengan”. |
| **[No.3] Kesimpulan** |
| Pada program ini, terlihat adanya perubahan nilai hasil. Nilai (A > B) & (A != B) yang sebelumnya bernilai true menjadi false, dan nilai (A <= B) & (A == B) yang sebelumnya false menjadi true. Hal ini dapat terjadi karena nilai A yang sebelumnya lebih besar dari B, menjadi sama besar dengan nilai B.  Dasar pengambilan keputusan saya adalah karena adanya perubahan nilai yang terjadi. |

|  |
| --- |
| **[No.4] Identifikasi Masalah:** |
| **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!  Pada latihan ini kita diperintahkan untuk membandingkan hasil Post dan Pre Increment dan hasil Post dan Pre Decrement.  Sumber Informasi:  <https://www.duniailkom.com/tutorial-belajar-c-jenis-jenis-operator-increment-dan-decrement-bahasa-c/> |
| **[No.4] Analisis dan Argumentasi** |
| Operator Increment digunakan untuk menambah variabel sebanyak 1 angka, sedangkan operator decrement digunakan untuk mengurangi variabel sebanyak 1 angka. Penulisannya menggunakan tanda tambah 2 kali (++) untuk increment, dan tanda kurang 2 kali (--) untuk decrement. Penempatan tanda tambah atau kurang ini boleh diawal, atau diakhir variabel, namun keduanya memiliki perbedaan, sehingga increment dan decrement terbagi lagi menjadi 2 yaitu Post dan Pre. |
| **[No.4] Penyusunan Algoritma dan Kode Program** |
| Perbedaannya terdapat pada baris 8 dan 17, yakni ketika kita akses langsung pada saat operasi increment dilakukan. Terlihat bahwa **post-increment** (a++), akan menampilkan hasilnya terlebih dahulu, baru nilai variabel a naik 1 angka, namun dengan **pre-increment** (++b), variabel b akan ditambahkan 1 angka, baru nilainya ditampilkan.  Perbedaannya terdapat pada baris 26 dan 35, yakni ketika kita akses langsung pada saat operasi decrement dilakukan. Terlihat bahwa **post-decrement** (c--), akan menampilkan hasilnya terlebih dahulu, baru nilai variabel c dikurangi 1 angka, sedangkan **pre-decrement** (--d), variabel d langsung dikurangi 1 angka, lalu nilainya ditampilkan. |
| **[No.4] Kesimpulan** |
| Pada latihan ini kita hanya membandingkan hasil Post dan Pre Increment dan hasil Post dan Pre Decrement.  **Post Increment** akan menampilkan hasilnya terlebih dahulu, baru nilai variabelnya ditambahi 1 angka sedangkan **Pre Increment** variabelnya langsung ditambahi 1 .  **Post Decrement** akan menampilkan hasilnya terlebih dahulu, baru nilai variabelnya dikurangi 1 angka sedangkan **Pre Decrement** variabelnya langsung dikurangi 1.  Dasar pengambilan keputusan saya adalah karena adanya perbedaan hasil pada Post dan Pre Increment serta Post dan Pre Decrement. |

|  |
| --- |
| **[No.5] Identifikasi Masalah:** |
| **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!  Sumber Informasi:  <https://www.duniailkom.com/pengertian-dan-jenis-jenis-operator-logika-dalam-php/> |
| **[No.5] Analisis dan Argumentasi** |
| Pada latihan ini operator yang kita gunakan adalah operator logika  untuk membandingkan dua nilai variabel atau lebih, yang hasilnya boolean true atau false. Dengan operator logika **AND (&&)**, **OR (||)**, dan **NOT (!)**. |
| **[No.5] Penyusunan Algoritma dan Kode Program** |
| **5.1 Tambahkan baris kode untuk memeriksa a || b.**    Hasil luaran yang ditampilkan pada gambar 5.1 sesuai dengan program yang disusun dengan menggunakan **&&** (AND) dan **||** (OR).  **5.2 Mengubah nilai a = false dan b = false dan menambahkan logika a || b && a || !b**    Pada gambar 5.2 tidak ada perubahan yang terjadi pada operator logika AND walaupun nilai a sudah dirubah menjadi false, karena pada operator logika AND kedua nilainya harus true agar hasil luarannya true. Sedangkan pada operator logika OR terjadi perubahan pada hasil luarannya dari true menjadi false, karena pada operator logika OR jika salah satu nilainya true maka hasil luarannya akan true.  Pada logika **a || b && a ||!b**, urutan logika yang akan dikerjakan yang pertama **a || b**, yang kedua **b && a**, dan yang terakhir **a||! b**. Sehingga hasil luarannya adalah true, karena **false OR false menghasilkan false, false AND false menghasilkan false, dan false OR NOT false menghasilkan true** |
| **[No.5] Kesimpulan** |
| Pada latihan ini operator yang digunakan adalah operator logika untuk membandingkan dua nilai variabel atau lebih, yang hasilnya boolean true atau false. Operator logika AND untuk mengoperasikan a && b, operator logika OR untuk mengoperasikan a || b, dan menambahkan operator logika NOT untuk mengoperasikan a || b && a ||!b |

|  |
| --- |
| **[No.6] Identifikasi Masalah:** |
| **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!    Pada latihan ini kita diperintahkan untuk mengubah nilainya dari 80 menjadi 60, lalu analisis hasil dan proses yang terjadi  Sumber Informasi:  <https://www.belajarcpp.com/tutorial/cpp/ternary/> |
| **[No.6] Analisis dan Argumentasi** |
| Pada latihan ini setelah kita mengubah nilainya dari 80 menjadi 60 hasilnya akan berubah dari lulus menjadi gagal karena ekspresi pada program ini tanda yang digunakan adalah lebih besar (>), sedangkan angka 60 tidak lebih besar dari 60, kecuali jika ekspresi pada program ini tanda yang digunakan adalah lebih besar sama dengan (>=) maka hasilnya tidak akan berubah walaupun nilainya sudah diubah menjadi 60. |
| **[No.6] Penyusunan Algoritma dan Kode Program** |
| Sebelum diubah    Setelah diubah |
| **[No.6] Kesimpulan** |
| Pada latihan ini hasilnya berubah dari lulus menjadi gagal karena pada ekspresi tanda yang digunakan adalah lebih besar (>), yang mana angka 60 tidak lebih besar dari angka 60 itu sendiri, kecuali jika pada ekspresi tanda yang digunakan adalah lebih besar sama dengan (>=) maka hasilnya tidak akan berubah menjadi gagal walaupun nilainya sudah diubah menjadi 60. |

|  |
| --- |
| **[No.7] Identifikasi Masalah:** |
| **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 : 6 Hasil dari a | b : 7 Hasil dari a ^ b : 1 Hasil dari ~a : -11 Hasil dari a >> 1 : 3 Hasil dari b << 2 : 28  **Latihan 7** Pilihlah 3 perhitungan Contoh 7, kemudian uraikan perhitungan biner! Simpulkan hasilnya!  Pada latihan ini kita diperintahkan untuk memilih 3 perhitungan dari contoh 7 lalu uraikan perhitungan binernya.  Sumber Informasi:  <https://www.duniailkom.com/tutorial-belajar-c-jenis-jenis-operator-bitwise-bahasa-c/> |
| **[No.7] Analisis dan Argumentasi** |
| Karena diperintahkan untuk memilih 3 perhitungan dari 6 perhitungan yang ada pada contoh. Saya memilih untuk menggunakan perhitungan AND, perhitungan OR, dan perhitungan Shift Left. Alasan saya memilih ketiga perhitungan ini karena ketiga perhitungan ini adalah jenis perhitungan Bitwise yang paling saya pahami dan juga termasuk perhitungan yang tidak rumit untuk mencari binernya.  Pada contoh 7, nilai a adalah 10 dan nilai b adalah 7. Jika di konversi ke dalam bentuk biner, keduanya berisi angka berikut:  **a = 10 (desimal) = 1010 (biner)**  **b = 7 (desimal) = 0111 (biner)**  Berikut perhitungan biner Bitwise “AND”:  a     = 1010  b     = 0111          ----  a & b = 0010 = 0+0+2+0= 2 (desimal)  Berikut perhitungan biner bitwise “OR”:  a     = 1010  b     = 0111          ----  a | b = 1111 = 8+4+2+1= 15 (desimal)  Berikut perhitungan biner Bitwise “Shift Left”:  b       =   0111 = 0+4+2+1= 7  b << 1  =  01110 = 0+8+4+2+0= 14 (desimal)  b << 2  = 011100 = 0+16+8+4+0+0= 28 (desimal) |
| **[No.7] Penyusunan Algoritma dan Kode Program** |
| 3 perhitungan yang saya pilih dari 6 perhitungan yang ada pada contoh 7 adalah perhitungan AND, perhitungan OR, dan perhitungan Shift Left. Serta hasil luaran yang dihasilkan sudah sesuai dengan uraian binernya. |
| **[No.7] Kesimpulan** |
| Pada latihan ini diperintahkan untuk memilih 3 perhitungan dari 6 perhitungan yang ada pada contoh. Saya memilih untuk menggunakan perhitungan AND, perhitungan OR, dan perhitungan Shift Left. Alasan saya memilih ketiga operator ini karena ketiga operator ini adalah jenis perhitungan Bitwise yang paling saya pahami dan juga termasuk perhitungan yang tidak rumit untuk mencari binernya. |