|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Zahra Sari Fhadilah**  **G1F024025** | **Operator Aritmatika** | **5 September 2024** |
| **[1] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable     Pada kode program masih ada sedikit kesalahan yaitu pada bagian yang menampilkan penjumlahan.   1. Rincikan sumber informasi yang relevan (buku / webpage)   **Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada laman**[**https://www.youtube.com/watch?v=PzCMZObexZM**](https://www.youtube.com/watch?v=PzCMZObexZM) | | |
| **[1.] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Saya mengusulkan Solusi yang bisa mengatasi eror pada output dengan cara menambah operator (+) sebelum (a + b) untuk menggabungkan string.  System.out.println("a + b = "  (a + b));  Menjadi,  System.out.println("a + b = "  + (a + b));   1. Analisis solusi, kaitkan dengan permasalahan.   Solusi yang saya usulkan bisa menambah pemahaman untuk memahami dasar dari operasi aritmatika pemrograman. | | |
| **[1.] Penyusunan Algoritma dan Kode Program** | | |
| 1. Algoritma  * Mulai * Deklarasi kelas   public class OperatorAritmatika{   * Deklarasi method mainn   public static void main(String[] args)  {   * Deklarasi variable   a = 20  b = 3   * Menampilkan nilai variable * Operasi aritmatika * Menampilkan hasil hitung aritmatika   Menggunakan System.out.println(   * Selesai  1. Tuliskan kode program dan luaran 2. Beri komentar pada kode  * Pada kode program yang pertama saya memperbaiki kode dengan cara menambah operasi (+) sebelum a + b untuk menggabungkan string. * Kemudian paada kode program yang telah saya perbaiki saya menambahkan baris untuk menampilkan perhitungan dengan operator ( -, \*, /, %).  1. Uraikan luaran yang dihasilkan  * Output yang ditampilkan sudah tidak mengalami eror karna perbaikan yang saya lakukan, serta karna saya menambah baris untuk menampilkan perhitungan dengan operator yang berbeda membuat output menampilkan hasil perhitungan dari berbagai operasi perhitungan.  1. Kode dan hasil luaran  * Kode program | | |
| **[1.] Kesimpulan** | | |
| 1. Kode program ini menunjukkan bahwa menggabungkan String sangat berpengaruh untuk menjalankan kode program. 2. Output yang awalnya ditampilkan sebagai eror bisa berjalan dengan baik dengan cara memperbaiki kesalahan pada penggabungan String. | | |

|  |  |  |
| --- | --- | --- |
| Nama & NPM | Topik: | Tanggal: |
| Zahra Sari Fhadilah  G1F024025 | Operator Penugasan | 5 September 2024 |
| [2.] Identifikasi Masalah: | | |
| 1. Uraikan permasalahan dan variable      1. Rincikan sumber informasi yang relevan (buku / webpage)   **Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada laman**[**https://www.youtube.com/watch?v=PzCMZObexZM**](https://www.youtube.com/watch?v=PzCMZObexZM) | | |
| [2.] Analisis dan Argumentasi | | |
| 1. Uraikan rancangan solusi yang diusulkan.   2.2 Argumentasi   * Luaran : Pada Contoh 1, nilai awal dari variabel a dan b serta hasil penjumlahannya ditampilkan, sehingga informasi yang diberikan menjadi lebih komprehensif. Sebaliknya, Contoh 2 hanya menampilkan hasil akhir dari penambahan tanpa mencantumkan nilai awal, sehingga tampilannya menjadi lebih sederhana. * Waktu Eksekusi : kedua contoh memiliki waktu eksekusi yang sangat cepat dan hampir sama, karena mereka hanya melakukan operasi dasar tanpa perhitungan yang rumit. | | |
| [2.] Penyusunan Algoritma dan Kode Program | | |
| 1. Algoritma  * Mulai * Deklarasi kelas   public class OperatorPenugasan {   * Deklarasi method main   public static void main(String[] args) {   * Deklarasi variable   a = 20  b = 3   * Operasi penugasan * Menampilkan hasil hitung   Menggunakan System.out.println(   * Selesai  1. Tuliskan kode program dan luaran 2. Beri komentar pada kode  * Kode program ini sudah dibuat dengan jelas karena dapat mendeklarasikan dan menginisialisasi variable dengan jelas, menggunakan operator penugasan dan mencetak hasil perhitungan dengan label yang jelas. * Kemudian pada kode program itu, saya menambahkan baris untuk menampilkan perhitungan dengan operator  ( -=, \*=,  /=, %=).  1. Uraikan luaran yang dihasilkan  * Output yang ditampilkan sudah sesuai dengan yang diinginkan, serta karna saya menambah baris untuk menampilkan perhitungan dengan operator yang berbeda membuat output menampilkan hasil perhitungan dari berbagai operasi perhitungan.  1. Kode dan hasil luaran  * Kode program      * Output (luaran) | | |
| [2.] Kesimpulan | | |
| 1. Secara keseluruhan, algoritma dan implementasi kode program ini memberikan pemahaman yang baik tentang penggunaan operator penugasan dalam Java, serta menunjukkan pentingnya kejelasan dan efisiensi dalam penulisan kode. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama & NPM | | Topik: | | Tanggal: |
| Zahra Sari Fhadilah  G1F024025 | | Operator Relasional | | 5 September 2024 |
| [3.] Identifikasi Masalah: | | | | |
| 1. Uraikan permasalahan dan variable      1. Rincikan sumber informasi yang relevan (buku / webpage)   **Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada laman**[**https://www.youtube.com/watch?v=PzCMZObexZM**](https://www.youtube.com/watch?v=PzCMZObexZM) | | | | |
| [3.] Analisis dan Argumentasi | | | | |
| 1. Uraikan rancangan solusi yang diusulkan.   2.2 Argumentasi   * Luaran : Pada Contoh 1, nilai awal dari variabel a dan b serta hasil penjumlahannya ditampilkan, sehingga informasi yang diberikan menjadi lebih komprehensif. Sebaliknya, Contoh 2 hanya menampilkan hasil akhir dari penambahan tanpa mencantumkan nilai awal, sehingga tampilannya menjadi lebih sederhana. * Waktu Eksekusi : kedua contoh memiliki waktu eksekusi yang sangat cepat dan hampir sama, karena mereka hanya melakukan operasi dasar tanpa perhitungan yang rumit. | | | | |
| [3.] Penyusunan Algoritma dan Kode Program | | | | |
| 1. Algoritma  * Mulai * Deklarasi kelas   public class OperatorRelasional {   * Deklarasi method main   public static void main(String[] args) {   * Deklarasi variable   a = 12  b = 4   * Menampilkan hasil hitung   Menggunakan System.out.println(   * Operasi perbandingan * Selesai  1. Tuliskan kode program dan luaran 2. Beri komentar pada kode  * Kode program ini sudah dibuat dengan jelas karena dapat mendeklarasikan dan menginisialisasi variable dengan jelas, menggunakan operator perbandingan dan mencetak hasil perbandingan dengan label yang jelas.  1. Uraikan luaran yang dihasilkan  * Output yang ditampilkan sudah sesuai dengan yang diinginkan, output menampilkan hasil perbandingan dari berbagai operasi perbandingan menggunakan tipe data Boolean.  1. Kode dan hasil luaran  * Kode program      * Output (luaran) | | | | |
| [3.] Kesimpulan | | | | |
| 1. Secara keseluruhan, algoritma dan implementasi kode program ini memberikan pemahaman yang baik tentang penggunaan operator perbandingan dalam Java, serta menunjukkan pentingnya kejelasan dan efisiensi dalam penulisan kode. | | | | |
| Nama & NPM | Topik: | | Tanggal: | |
| Zahra Sari Fhadilah  G1F024025 | Operasi Increment Dan Decrement | | 5 September 2024 | |
| [4.] Identifikasi Masalah: | | | | |
| 1. Uraikan permasalahan dan variable      1. Rincikan sumber informasi yang relevan (buku / webpage)   **Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada laman**[**https://www.youtube.com/watch?v=PzCMZObexZM**](https://www.youtube.com/watch?v=PzCMZObexZM) | | | | |
| [4.] Analisis dan Argumentasi | | | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Saya mengusulkan untuk memecahkan masalah dengan cara mengubah perintah yang awalnya menggunakan post increment menjadi pre increment.  System.out.println("b: " + (a++));  Menjadi,  System.out.println("b: " + (a++)); | | | | |
| [4.] Penyusunan Algoritma dan Kode Program | | | | |
| 1. Algoritma  * Mulai * Deklarasi kelas   public class Operator {   * Deklarasi method main   public static void main(String[] args) {   * Deklarasi variable   a = 5   * Membuat perintah menampilkan luaran dengan operator pre increment * Menampilkan hasil   Menggunakan System.out.println(   * Selesai  1. Tuliskan kode program dan luaran 2. Beri komentar pada kode  * Kode program ini sudah dibuat dengan jelas karena dapat mendeklarasikan dan menginisialisasi variable dengan jelas, menggunakan fungsi operator pre increment  1. Uraikan luaran yang dihasilkan  * Output yang ditampilkan sudah sesuai dengan yang diinginkan.  1. Kode dan hasil luaran  * Kode program      * Output (luaran) | | | | |
| [4.] Kesimpulan | | | | |
| 1. Secara keseluruhan, algoritma dan implementasi kode program ini memberikan pemahaman yang baik tentang penggunaan operasi increment dan decrement dalam Java, serta menunjukkan pentingnya kejelasan dan efisiensi dalam penulisan kode. Karena menggunakan pre increment maka variable yang awalnya b=5 akan berubah menjadi b=6 karna nilai a sudah ditambah nilainya. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama & NPM | Topik: | | Tanggal: | |
| Zahra Sari Fhadilah  G1F024025 | Operator Logika | | 5 September 2024 | |
| [5.] Identifikasi Masalah: | | | | |
| 1. Uraikan permasalahan dan variable      1. Rincikan sumber informasi yang relevan (buku / webpage)   **Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman** <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | | | |
| [5.] Analisis dan Argumentasi | | | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Saya mengusulkan untuk memecahkan masalah dengan cara mengubah nilai dari Boolean b menjadi true agar dapat menghasilkan luaran true dengan operator && dan operator ||  Boolean a = true;  Boolean b = false;  Menjadi,  Boolean a = true;  Boolean b = true; | | | | |
| [5.] Penyusunan Algoritma dan Kode Program | | | | |
| 1. Algoritma  * Mulai * Deklarasi kelas   public class Operator {   * Deklarasi method main   public static void main(String[] args) {   * Deklarasi variable dengan nilai true * Membuat perintah menampilkan hasil logika   Menggunakan System.out.println(   * Selesai  1. Tuliskan kode program dan luaran 2. Beri komentar pada kode  * Kode program ini sudah dibuat dengan jelas karena dapat mendeklarasikan dan menginisialisasi variable dengan jelas, menggunakan fungsi operator logika  1. Uraikan luaran yang dihasilkan  * Output yang ditampilkan sudah sesuai dengan yang diinginkan.  1. Kode dan hasil luaran  * Kode program      * Output (luaran) | | | | |
| [5.] Kesimpulan | | | | |
| 1. Secara keseluruhan, algoritma dan implementasi kode program ini memberikan pemahaman yang baik tentang penggunaan operator logika dalam Java, serta menunjukkan pentingnya kejelasan dan efisiensi dalam penulisan kode.    * Untuk operasi AND (a && b), hasilnya adalah true karena kedua variabel a dan b bernilai true.    * Untuk operasi OR (a1 || b1), hasilnya adalah true karena salah satu operand (a1) bernilai true | | | | |
| Nama & NPM | | | Topik: | | Tanggal: |
| Zahra Sari Fhadilah  G1F024025 | | | **Operator Kondisional (Ternary)** | | 5 September 2024 |
| [6.] Identifikasi Masalah: | | | | | |
| 1. Uraikan permasalahan dan variable      1. Rincikan sumber informasi yang relevan (buku / webpage)   **Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman** <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | | | | |
| [6.] Analisis dan Argumentasi | | | | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Saya mengusulkan untuk memecahkan masalah dengan cara mengganti nilai dari variabel status agar milai 60 bisa lulus.  status = (nilai > 60)?"Lulus":"Gagal";  Menjadi,  status = (nilai >= 60)?"Lulus":"Gagal"; | | | | | |
| [6.] Penyusunan Algoritma dan Kode Program | | | | | |
| 1. Algoritma  * Mulai * Deklarasi kelas   public class OperatorKondisi {   * Deklarasi method main   public static void main(String[] args) {   * Deklarasi variable dengan nilai 60 * Membuat deklarasi variable status dengan nilai >=60 dengan operator kondisional Lulus atau Gagal * Menampilkan hasil luaran   Menggunakan System.out.println(   * Selesai  1. Tuliskan kode program dan luaran 2. Beri komentar pada kode  * Kode program ini sudah dibuat dengan jelas karena dapat mendeklarasikan dan menginisialisasi variable dengan jelas, menggunakan fungsi operator kondisional  1. Uraikan luaran yang dihasilkan  * Output yang ditampilkan sudah sesuai dengan yang diinginkan.  1. Kode dan hasil luaran  * Kode program      * Output (luaran) | | | | | |
| [6.] Kesimpulan | | | | | |
| 1. Secara keseluruhan, algoritma dan implementasi kode program ini memberikan pemahaman yang baik tentang penggunaan operasi kondisional dalam Java, serta menunjukkan pentingnya kejelasan dan efisiensi dalam penulisan kode. Karena mengubah tanda operator menjadi >=60 maka nilai 60 dapat dikatakan lulus karena memenuhi syarat. | | | | | |
| Nama & NPM | | | Topik: | | Tanggal: |
| Zahra Sari Fhadilah  G1F024025 | | | **Operator Bitwise** | | 8 September 2024 |
| [7.] Identifikasi Masalah: | | | | | |
| 1. Uraikan permasalahan dan variable      1. Rincikan sumber informasi yang relevan (buku / webpage)   **Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman** <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | | | | |
| [7.] Analisis dan Argumentasi | | | | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Saya mencari tahu nilai biner dari 10 = 1010, kemudian saya mencari tahu apa itu ~(NOT) = Botwise complement (komplemen bit) merupakan operasi yang dapat mengubah setiap bit dari suatu angka biner menjadi nilai yang berlawanan: bit 1 menjadi 0, dan bit 0 menjadi 1. Dalam bahasa pemrograman seperti Java, operasi ini direpresentasikan dengan simbol ~.   1. Analisis solusi, kaitkan dengan permsalahan   Penyebab hasil a~ = -11   * **Cara Kerja Bitwise Complement** * Bitwise complement adalah operasi yang diterapkan pada angka biner. Untuk memahami ini, mari kita lihat bagaimana bitwise complement bekerja pada angka dalam biner: * Contoh: * Jika kita memiliki angka a = 10, dalam biner, 10 direpresentasikan sebagai 0000 1010 (dalam 8-bit). * **Langkah 1: Inversi Setiap Bit** * 0000 1010 menjadi 1111 0101. * Angka hasil inversi adalah representasi dari bitwise complement. * **Hubungan dengan 2's Complement** * Dalam konteks bilangan negatif dalam komputer, **2's complement** digunakan untuk mewakili bilangan negatif. Untuk mendapatkan 2's complement dari suatu bilangan: * Ambil **bitwise complement** dari angka tersebut. * Tambahkan 1 ke hasil inversi. * Misalnya, ~10 (bitwise complement dari 10): * Bilangan 10 dalam biner: 0000 1010. * Bitwise complement dari 10 adalah: 1111 0101. * Untuk menghitung nilai desimal dari 1111 0101 menggunakan 2's complement: * Hitung kembali inversi (bitwise complement) dari 1111 0101, yaitu 0000 1010. * Tambahkan 1: 0000 1010 + 1 = 0000 1011 (yang sama dengan 11). * Karena bit pertama adalah 1, ini menunjukkan angka negatif, jadi hasil akhirnya adalah -11. | | | | | |
| [7.] Penyusunan Algoritma dan Kode Program | | | | | |
| 1. Rancang desain solusi atau algoritma  * Mencari nilai biner dari 10 * Memahami apa itu ~(NOT) * Menghitung nilai biner dari 10 dengan cara kerja dari ~(NOT)  1. Tuliskan kode program dan luaran 2. Beri komentar pada kode   Kode program ini sudah dibuat dengan jelas karena dapat mendeklarasikan dan menginisialisasi variable dengan jelas, menggunakan fungsi operator bitwise   1. Uraikan luaran yang dihasilkan   Output yang ditampilkan sudah sesuai dengan yang diinginkan.   1. Kode dan hasil luaran  * Kode program      * Output (luaran) | | | | | |
| [7.] Kesimpulan | | | | | |
| 1. Bitwise Complement (~) mengubah setiap bit 1 menjadi 0 dan setiap bit 0 menjadi 1. 2. Dalam konteks sistem bilangan negatif, hasil dari bitwise complement adalah dasar untuk mendapatkan representasi 2's complement. Ini digunakan untuk merepresentasikan bilangan negatif di komputer.   **Refleksi**  Setelah menyelesaikan tugas ini, saya memperoleh banyak pengalaman dan pengetahuan baru. Materi ini merupakan hal baru yang saya temui di perkuliahan, sehingga saya mengalami kesulitan pada awalnya. Namun, saya tidak menyerah dan terus belajar agar tidak tertinggal dari teman-teman saya. Teman-teman juga sering membantu saya dalam mengerjakan tugas ini, sehingga saya akhirnya dapat menyelesaikannya dengan baik.  Banyak hal baru yang saya pelajari dari mengerjakan tugas ini, seperti penggunaan tipe data, penempatan berbagai elemen dan bagaimana cara mengoperasikan sebuah nilai dan lain sebagainya, yang sangat berpengaruh terhadap kelancaran kode program. | | | | | |