|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Iqbal Tri Agustian**  **G1F024043** | **Operator Relasional dan Operator Logiika** | **08 September 2024** |
| **[Materi Pengayaan 1] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variabel      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> * Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | |
| **[Materi Pengayaan 1] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan.  * 1. Ekspresi !(a > c) && b > c melibatkan operator NOT !, perbandingan >, dan AND &&. * Prioritas operator di Java:   1. > (lebih besar dari)   2. ! (NOT logis)   3. && (AND logis) * Jadi, ekspresi akan mengevaluasi a > c dan b > c terlebih dahulu, kemudian menerapkan operator !, dan terakhir mengevaluasi operator &&. * 2. Saya menyusun kembali nilai F agar menghasilkan nilai TRUE! yaitu dengan mengganti nilai dari variabel int a, int b, int c yang awalnya * System.out.println(F(1,2,3)); System.out.println(F(3,2,1)); System.out.println(F(3,3,3));   Menjadi   * System.out.println(F(1,4,3));  System.out.println(F(0,2,1)); System.out.println(F(3,4,3));  1. Analisis solusi, kaitkan dengan permasalahan.  * Fungsi F akan mengembalikan true hanya jika:   + !(a > c) bernilai true, artinya a harus lebih kecil atau sama dengan c.   + b > c bernilai true, artinya b harus lebih besar dari c. | | |
| **[Materi Pengayaan 1] Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma  * Membuat nama kelas sama dengan nama file * Membuat method main * Mendeklarasikan variabel * Memasukan operasi relasional dan operasi logika * Membuat perintah luaran dan memasukkan nilai variabel yang akan dijadikan luaran dengan tipe boolean  1. Tuliskan kode program dan luaran  * Codingan Materi Pengayaan 1      * Luaran codinagan Materi Pengayaan 1      * Luaran yang ditampilkan sudah sesuai dengan perintah dan tidak terjadi eror atau kendala saat ingin menampilkan luaran | | |
| **[Materi Pengayaan 1] Kesimpulan** | | |
| 1. Analisa 2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!  * Kode program menggunakan metode boolean, metode boolean adalah metode yang digunakan untuk menentukan kebenaran dari suatu nilai atau “TRUE OR FALSE”, dengan mendeklarasi variabel yang sudah diberi operasi relasional, dan operasi logika, maka boolean akan menampilkan luaran sesuai dengan operasi. Jika operasi benar maka akan ditampilkan luaran “TRUE” dan jika nilai salah, maka akan ditampilkan luaran”FALSE”  1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?  * Alasannya untuk mempermudah dalam menentukan apakah nilai tersebut bersifat “TRUE” atau “FALSE” dengan menggunakan metode boolean | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Iqbal Tri Agustian**  **G1F024043** | **Operator Relasional dan Operator Logika** | **09 September 2024** |
| **[Materi Pengayaan 2] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variabel      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> * Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | |
| **[Materi Pengayaan 2] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan.  * 1. Ekspresi 3+4\*4>1+5\*(4-2) || 5<7Melibatkan perbandingan > dan, perbandingan < dan operator OR || * **Ekspresi Aritmatika**:   + 3 + 4 \* 4:     - Operasi perkalian terlebih dahulu: 4 \* 4 = 16     - Kemudian penjumlahan: 3 + 16 = 19   + 1 + 5 \* (4 - 2):     - Kurung terlebih dahulu: 4 - 2 = 2     - Kemudian perkalian: 5 \* 2 = 10     - Kemudian penjumlahan: 1 + 10 = 11 * **Ekspresi Relasional**   + (19 > 11) adalah True   + 5 < 7 adalah True * **Operator Logika &&**:   + - Ekspresi logika menjadi True && True, yang menghasilkan “TRUE” * 2. Saya menyusun kembari eksperasi agar menghasilkan luaran “FALSE” dengan mengganti ekspresi yang awalnya System.out.println(3+4\*4>1+5\*(4-2)||5<7); Menjadi System.out.println(3+4\*4<1+5\*(4-2)||5>7);   + (19 < 11) adalah False   + 5 > 7 adalah False * Ekspresi logika menjadi False || False , yang menjadi “FALSE”  1. Analisis solusi, kaitkan dengan permasalahan.  * Ekspresi logika OR || menjadi “FALSE” hanya jika kedua nilai bersifat “FALSE” | | |
| **[Materi Pengayaan 2] Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma  * Mengubah ekspresi agar menghasilkan luaran “FALSE”  1. Tuliskan kode program dan luaran  * Codingan      * Luaran codingan      * Luaran sudah keluar sesuai perintah dan tidak terjadi eror | | |
| **[Materi Pengayaan 2] Kesimpulan** | | |
| 1. Analisa 2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!  * Kode program menggunakan metode boolean, metode boolean adalah metode yang digunakan untuk menentukan kebenaran dari suatu nilai atau “TRUE OR FALSE”, dengan mendeklarasi variabel yang sudah diberi operasi relasional, dan operasi logika, maka boolean akan menampilkan luaran sesuai dengan operasi. Jika operasi benar maka akan ditampilkan luaran “TRUE” dan jika nilai salah, maka akan ditampilkan luaran”FALSE”  1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?  * Alasannya untuk mempermudah dalam menentukan apakah nilai tersebut bersifat “TRUE” atau “FALSE” dengan menggunakan metode boolean | | |