**Template Lembar Kerja Individu dan Kelompok**

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Karina Hodiyah Ramadona**  **G1F024040** | **Operator** | **15 September 2024** |
| **[No.1] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable   class Main{           public static boolean F(int a,int b,int c){          return !(a>c) && b > c;     }     public static void main(String[] args) {             System.out.println(F(1,2,3));         System.out.println(F(3,2,1));         System.out.println(F(3,3,3));     }      } **Luaran:** false false false  **Ayo berlatih!** 1. Uraikan urutan penyelesaian masalah (!(a>c) && b > c) berdasarkan prioritas operator! 2. Susun kembali nilai F agar menghasilkan nilai TRUE!   1. Rincikan sumber informasi yang relevan (buku / webpage)   <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | |
| **[No.1] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Saya mengusulkan untuk menambahkan kode public di awal kode program. Kemudian saya juga mengusulkan denganmengganti angka yang telah saya tentukan agar bisa mendapatkan luaran yang bernilai TRUE.   1. Analisis solusi, kaitkan dengan permasalahan.   Alasan saya mengusulkan solusi yang pertama yaitu karena tanpa kode public semua kode yang lain tidak akan bisa di run atau error. Kemudian alasan saya mengusulkan solusi yang kedua adalah jika kita lihat lagi di soal, maka kita akan melihat perintah untuk mrngubah nilai f yang semula false menjadi true. Sedangkan nilai f sendiri sudah mempunyai nilai operator yang harus ditentukan dulu prioritasnya. Oleh karena itu, saya menentukan angka tersebut berdasarkan operator yang ada. Misalnya untuk nilai F yang pertama, melihat dari operatornya yang seperti (!(a>c)) maka kita harus mencari nilai a yang tidak lebih dari c. Kenapa? Hal ini disebabkan oleh adanya operator NOT(!) yang berada sebelum operator (a>c), yang mana operator NOT(!) itu sendiri menandakan luaran yang berkebalikan dengan input. selain itu karena operator NOT(!) memang prioritas operator dibandingkan operator lain yang ada di penyataan kali ini.  Kemudian untuk angka kedua dan ketiga di dapatkan dari pernyataan (&& b>c) yang mana operator ini menyatakan bahwa nilai luaran akan bernilai true apabila kedua operator true. Maka dari itu langsung saja mencari nilai b dan yang, yang mana nilai b harus lebih besar dari c. | | |
| **[No.1] Penyusunan Algoritma dan Kode Program** | | |
| 1. Algoritma 2. Lihat permasalahan pada kode program pada saat sudah selesai membuka e-learning 3. Apabila ada kode program yang kurang, langsung saja tambahkan. Karena apabila ada satu saja kode yang kurang maka, kode tersebut tidak akan bisa berjalan atau error. 4. Jika diberikan perintah untuk menentukan nilai tetapi ada beberapa jenis operatornya, maka harus mengetahui prioritas dari operator-operator tersebut. 5. Saat sudah selesai semua, cobalah untuk run agar memastikan semuanya sudah benar. 6. Selesai. 7. Tuliskan kode program dan luaran     **Luaran** | | |
| **[No.1] Kesimpulan** | | |
| 1. Kreasi 2. Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?   Ya, ada. Saya jadi lebih mengatahui dan lebih memahami tentang prioritas dari macam macam operator. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Karina Hodiyah Ramadona**  **G1F024040** | **Operator** | **15 September 2024** |
| **[No.2] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable   public class operator {     public static void main(String[] args) {            System.out.println(3+4\*4>1+5\*(4-2)||5<7);     } }  **Luaran:** True  **Ayo berlatih!** 1.    Uraikan ekspresi tersebut hingga mendapatkan kesimpulan TRUE! 2.    Susun ekspresi yang berbeda untuk menghasilkan kesimpulan FALSE!  2) Rincikan sumber informasi yang relevan (buku / webpage)  <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | |
| **[No.2] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Saya mengusulkan untuk menghitung semua jenis operator dimulai dari operator yang paling di prioritaskan. Kemudian saya juga mengusulkan untuk mengubah salah satu tanda operator agar sesuai dengan perintah yang diberikan.   1. Analisis solusi, kaitkan dengan permasalahan.   Alasan saya mengusulkan Solusi tersebut ialah karena tanpa diketahui jumlah atau hasil dari operator tersebut maka kita tidak akan tau cara mengubah nilainya agar menjadi false. Cara nya yaitu kita dahulukan operator yang di prioritaskan. Seperti pada contoh (3+4\*4>1+5\*(4-2)||5<7), maka perhitungannya yaitu:  (3+4\*4>1+5\*(2)||5<7) dahulukan pengurangan didalam kurung  (3+16>1+10||5<7) kalikan angkanyang bisa di kalikan  (19>11||5<7) jumlahkan  Apabila sudah seperti ini maka kita bisa melihat luaran yang akan dihasilkan. Karena disini menggunakan operator OR(||) yang mana operator tersebut akan menghasilkan true apabila kedua dan salah satu bernilai true. Untuk di contoh sekarang keduanya bernilai true, maka luarannya adalah true. Untuk menghasilkan nilai false maka kita harus mengubah salah satu nilai dari lebih dari atau kurang dari, dan saya juga mengubah angka yang awalnya \*4 menjadi \*0 pada kode (3+4\*4) | | |
| **[No.1] Penyusunan Algoritma dan Kode Program** | | |
| 1. Algoritma 2. Lihat permasalahan pada kode program pada saat sudah selesai membuka e-learning 3. Apabila ada kode program yang kurang, langsung saja tambahkan. Karena apabila ada satu saja kode yang kurang maka, kode tersebut tidak akan bisa berjalan atau error. 4. Jika diberikan perintah untuk menentukan nilai tetapi ada beberapa jenis operatornya, maka harus mengetahui prioritas dari operator-operator tersebut. 5. Saat sudah selesai semua, cobalah untuk run agar memastikan semuanya sudah benar. 6. Selesai. 7. Tuliskan kode program dan luaran     **Luaran** | | |
| **[No.2] Kesimpulan** | | |
| 1. Kreasi 2. Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?   Ya, ada. Saya jadi lebih mengatahui dan lebih memahami tentang prioritas dari macam macam operator. | | |
|  | | |

