**Lembar Kerja Individu**

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Ariiq Ashar Sofyan**  **G1F024052** | **For dan While** | **10 Oktober 2024** |
| **[Nomor Soal] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variabel 2. Rincikan sumber informasi yang relevan (buku / webpage) 3. Uraikan rancangan solusi yang diusulkan (jika ada). 4. Analisis susunan solusi, parameter solusi (jika ada). | | |
| **[Nomor Soal] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan. 2. Analisis solusi, kaitkan dengan permasalahan. | | |
| **[Nomor Soal] Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma 2. Tuliskan kode program dan luaran 3. Beri komentar pada kode 4. Uraikan luaran yang dihasilkan 5. Screenshot/ Capture potongan kode dan hasil luaran | | |
| **[Nomor Soal] Kesimpulan** | | |
| 1. Analisa 2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! 3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? 4. Evaluasi 5. Apa konsekuensi dari skenario pemprograman ini? 6. Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada) 7. Kreasi 8. Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi? 9. Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada) | | |
|  | | |

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

1. Uraikan permasalahan dan variabel

**Latihan 1**  
1.1.     Evaluasi penyebab kesalahan dan perbaiki kode pada Contoh 1!  
           Rekomendasikan kata kunci yang tepat diletakkan pada baris kode yang kosong 1 dan 2 untuk dapat menghasilkan luaran berikut:  
Luaran contoh 1:  
0   
2   
4   
6

1.2.    Cermati contoh kode 2 pada kode  //baris kode kosong.  
           Rekomendasikan kode yang tepat menggunakan break atau continue terhadap pertama atau kedua agar menghasilkan luaran berikut:  
Luaran Contoh 2:  
i = 1; j = 1  
i = 1; j = 2  
i = 2; j = 1  
i = 2; j = 2  
   
1.3.    Cermati kode contoh 3. Apabila ingin menghasilkan luaran berikut:  
Luaran berbentuk piramida  
Masukan Input: 7  
       \*  
      \*\*\*  
     \*\*\*\*\*  
    \*\*\*\*\*\*\*  
   \*\*\*\*\*\*\*\*\*  
  \*\*\*\*\*\*\*\*\*\*\*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*  
        Rekomendasikan kode untuk menghasilkan luaran tersebut!

1.4.  Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

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

1. Saya mengusulkan permasalahan kode pada contoh 1.1 dapat diatasi dengan cara :
   1. Format yang benar untuk loop for adalah : for (inisialisasi; kondisi; increment)
   2. for (double y <= 15; y=0; y++) diperbaiki menjadi for (double y = 0 ; y <= 15; y++)\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Alasan solusi ini karena dengan perbaikan program diatas, program akan menghasilkan output yang diinginkan.
3. Perbaikan kode program dengan cara :

mengubah kesalahan sintaks pada contoh 1,berikut kode program perbaikan pada contoh 1 :

public class ContohFor {

public static void main(String[] args) {

for (double y = 0; y <= 15; y++) {

if (y % 2 == 1) { //kondisi 1

continue; // baris kode kosong 1

} else if (y == 8) { //kondisi 2

break; // baris kode kosong 2

} else

System.out.println(y + " ");

}

}}

1. Untuk mendapatkan luaran yang diinginkan sesuai dengan soal 1.2 maka sebaiknya penggunaan kode yang tepat menggunakan kode break pertama karena kita perlu menghentikan seluruh proses loop setelah i mencapai nilai 2. Oleh itu, pada kondisi if (i == 2), kita menggunakan break pertama. Jika kita menggunakan kode lain seperti :
   1. Break; (tanpa nama) – hanya akan menghentikan loop dalam
   2. Break kedua; - juga hanya menghentikan loop dalam
   3. Continue pertama; - akan melanjutkan ke literasi berikutnya dari loop luar, yang tidak kita inginkan.

**[No.1 ] Penyusunan Algoritma, Kode Program, dan Diagram Flowchart**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma membuat program yang menghasilkan luaran segitiga:

* Mulai
* Menyalin kode program contoh 1
* Mengubah tipe data double menjadi int
* Membalikkan inisialisasi y
* Dan output kode program
* Lanjut ke soal kedua meminta output 0,2,4,6
* Sebelum membuat kode program yang menghasilkan 0,2,4,6 kita memperbaiki kode program contoh 2 terlebih dahulu
* Mengubah inisialisasi dari kode program 2 yaitu mengubah y =0 : y <=6: y++
* Mengeluarkan output
* Melampirkannya di laporan
* Lanjut kesoal ketiga yaitu kode program yang menghasilkan output yang telah tertera pada soal
* Dari contoh 2 kita perlu mengubah i=1: i<3: i++ begitu pula dengan j.
* Menambahkan continue atau break pada kode program tersebut pada kalimat //kode yang hilang
* Melampirkannya pada laporan
* Lanjut ke soal ketiga
* Membuat kode program yang menghasilkan piramida
* Menggunakan 3 for
* Runkode program yang telah dibuat dan
* Menganalisa kode tersebut
* Melampirkannya pada laporan
* Membuat flowchart dan melampirkannya pada laporan



1. Kode program dan luaran



1. Kode Program

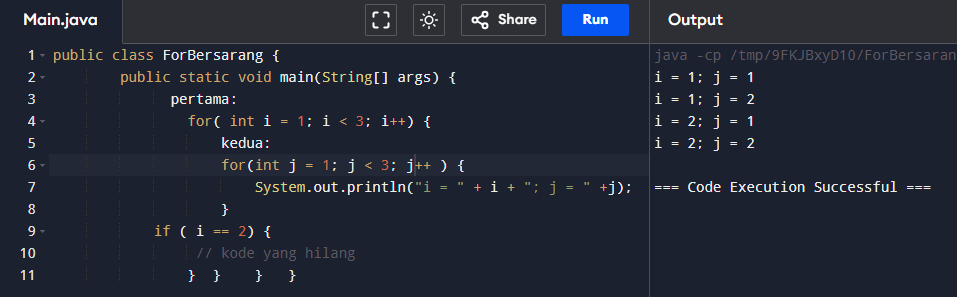
• Kesalahan kode program yaitu pada tipe data double yang seharusnya int

• Kemudian peletakan saat inisialisasi y terbalik, yang benar adalah ( y= 0: y <= 15 : y++)

• Kemudian dari soal memerintahkan untuk menghasilkan luaran 0, 2, 4, 6 saja, kode programnya yaitu



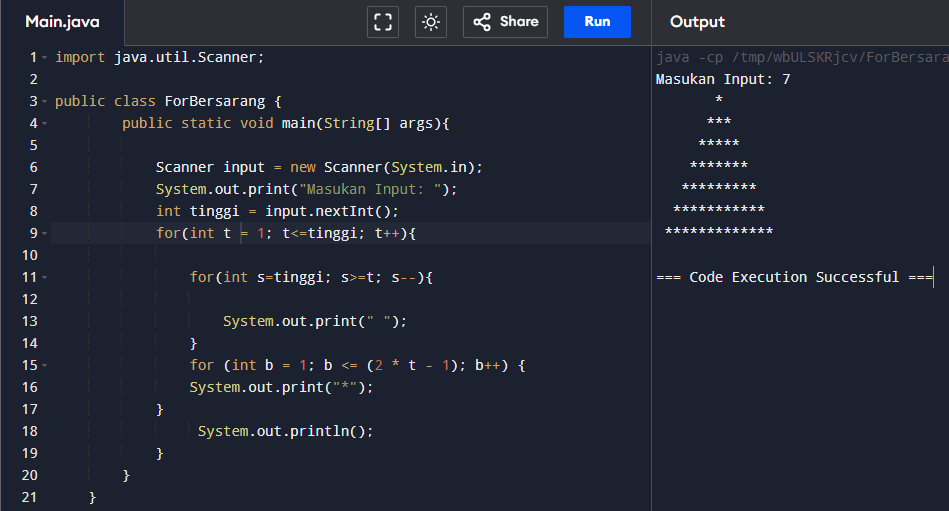
* Dari kode program diatas yang diubah adalah pada saat inisialisasi yaitu membuat y<=6
* Kode program selanjutnya yaitu :



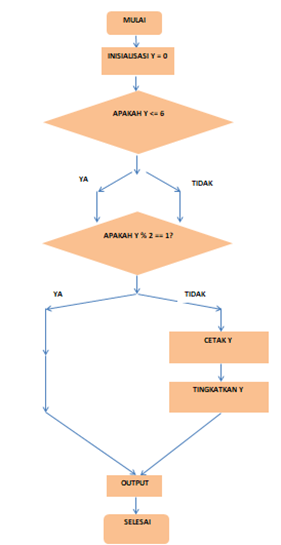
* Kode program yang menampilkan luaran yang telah diminta, perbaikan yang dilakukan yaitu pada line 4 dan line 6 saat inisialisasi.
* Kode program selanjutnya kode program yang telah menambahkan continue pada program diatas



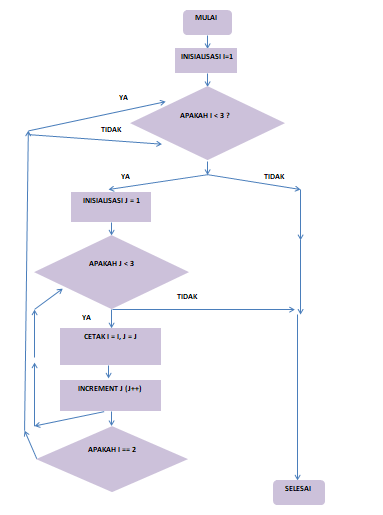
* Peletakan continue pada tulisan kode yang hilang tersebut.
* Kode program yang menghasilkan piramida yaitu



* Pada kode diatas akan menghasilkan piramida dengan tinggi yang sesuai dengan input yang dimasukkan oleh pengguna
* Seperti input yang saya masukkan yaitu 7, maka akan mencetak piramida dengan tinggi 7 tingkat.
* Kemudian flowchart untuk latihan 1.1. dan latihan 1.2 yaitu



* Gambar diatas adalah flowchart dari latihan 1.1, kemudia untuk flowchart 1.2 adalah



**[No.1] Kesimpulan**

1. **Analisa**
2. Program untuk membuat segitiga bintang menggunakan loop for bersarang adalah cara yang baik untuk memahami pengulangan dalam pemrograman. Program ini menggunakan dua loop for: satu loop luar untuk mengatur jumlah baris segitiga dan satu loop dalam untuk mencetak bintang di setiap baris. Setiap kali loop luar berjalan, jumlah bintang yang dicetak ditentukan oleh nomor baris. Misalnya, di baris pertama dicetak satu bintang, di baris kedua dua bintang, dan seterusnya hingga tinggi segitiga yang diinginkan. Loop dalam bertugas mencetak bintang dan biasanya diakhiri dengan karakter newline untuk pindah ke baris berikutnya.
3. Pengambilan keputusan dalam program ini melibatkan beberapa pertimbangan. Pertama, kita perlu memahami tujuan program, yaitu mencetak pola segitiga bintang yang simetris. Keputusan untuk menggunakan loop bersarang diambil karena metode ini memberikan kontrol lebih baik atas jumlah elemen yang dicetak di setiap baris. Selain itu, struktur kontrol ini juga dipilih karena mudah dipahami dan diimplementasikan, terutama untuk pemula dalam pemrograman.

**[No. 2] Identifikasi Masalah:**

1. Uraikan permasalahan dan variabel

**Contoh 4:** Salin dan tempel kode program berikut ke Eclipse.

public class ContohWhile{   
public static void main(String[] args) {  
    int i=1;  
        while(i<=6){  
            System.out.println(i);  
            i++;  
                 if(i==4){               
                break;        //ubah1   
            }}}

**Luaran:**  
1  
2  
3

**Contoh 5:**Salin dan tempel kode program berikut ke Eclipse.

import java.util.Scanner;

public class ForBersarang {   
        public static void main(String[] args) {  
            Scanner dataKata = new Scanner(System.in);  
            System.out.print("Masukkan Kata yang ingin diulang : ");  
            String kata = dataKata.nextLine();  
              
            Scanner dataJumlah = new Scanner(System.in);  
            System.out.print("Masukkan Jumlah ingin diulang : ");  
            int jumlah = dataJumlah.nextInt();  
              
            int i = 0; //Inisialisasi batas dasar  
            while(i < jumlah){  
                System.out.println(kata);  
                i++;//Faktor pengulang Increment  
            }  
        }  
}

**Luaran Contoh 5:**  
Masukkan Kata yang ingin diulang : Fakultas Teknik  
Masukkan Jumlah ingin diulang : 5  
Fakultas Teknik  
Fakultas Teknik  
Fakultas Teknik  
Fakultas Teknik  
Fakultas Teknik

**Latihan 2**

2.1.  Ubahlah baris kode pada Contoh 4  
        //Ubah1 menjadi  if(i % 3 == 0){   ◊ running, periksa hasilnya  
        //Ubah2 menjadi continue;  ◊ running, periksa hasilnya  
        Evaluasi perbandingan luaran sebelum dan setelah diubah! Simpulkan maksud dari perubahan tersebut!

2.2. Cermati Contoh 5. Periksa luaran, bila ketika di eksekusi, jumlah yang diulang = 0!   
       Evaluasi luaran, bila kode diubah menjadi do … while dengan masukan sama jumlah yang diulang = 0.  
       Simpulkan perbedaan while dan do … while!

2.3. Bila diketahui pernyataan pseudocode berikut:  
       [1]    inisiasi idPelajaran  
       [2]    inisiasi nilai pelajaran  
       [3]    inisiasi nilai rata-rata  
       [4]    Minta pengguna untuk menuliskan jumlah pelajaran  
       [5]    Ketika idPelajaran lebih kecil dari jumlah pelajaran  
       [6]    Minta pengguna untuk menuliskan nilai pelajaran  
       [7]    Hitung nilai rata-rata = (nilai pelajaran + nilai rata-rata) / 2       
       [8]    Tambah satu ke idPelajaran  
       [9]    Tampilkan nilai rata-rata  
       Rekomendasikan kode untuk menyelesaikan Pseudocode tersebut!

2.4.     Rancang diagram flowchart dari Latihan 2.1, Latihan 2.2, dan Latihan 2.3!

**[No.2] Analisis dan Argumentasi**

1. Evaluasi perubahan kode program pada contoh 4 :
   1. Pada perubahan pertama, saya mengganti kode if(i==) dengan if(i % 3 == 0). Dengan perubahan ini, program akan memeriksa apakah nilai i adalah kelipatan 3. Jika iya, pernyataan break akan dieksekusi dan menghentikan loop.
   2. Pada perubahan kedua, saya mengganti break dengan continue. Dengan continue, program tidak akan menghentikan loop, tetapi akan melewati sisa iterasi saat kondisi if(i % 3 == 0) terpenuhi. Ini berarti ketika i adalah 3, program akan mencetak angka tersebut dan langsung melanjutkan ke iterasi berikutnya tanpa mencetak angka 4. Jadi, pada perubahan pertama, program mencetak angka 1, 2, dan 3, tetapi setelah perubahan kedua, program melanjutkan ke iterasi berikutnya tanpa mencetak 4, lalu mencetak 5 dan 6.
2. Alasan solusi ini adalah penggunaan struktur kontrol yang tepat sangat penting dalam pemrograman untuk mencapai hasil yang diinginkan. Dalam contoh program dengan loop while, kondisi dievaluasi sebelum blok kode dieksekusi. Ini berarti jika jumlah pengulangan yang dimasukkan adalah 0, tidak ada output yang dihasilkan karena kondisi awal tidak terpenuhi.
3. Perbaikan kode program dengan cara :

Perbaikan kode program dengan cara:

a. Menggunakan Satu Scanner: Daripada menggunakan dua objek Scanner untuk input, kita bisa menggunakan satu objek Scanner untuk kedua input, sehingga kode lebih efisien.

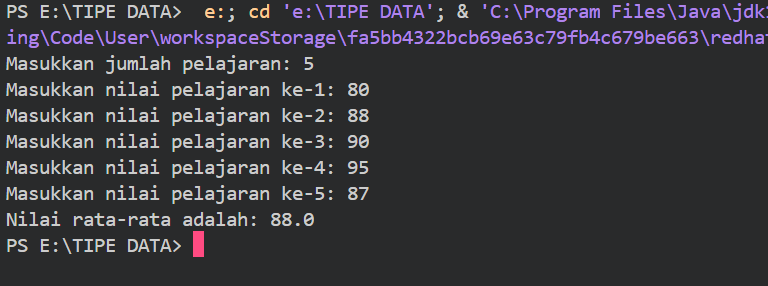
b. Menangani Input Negatif: Kita bisa menambahkan logika untuk menangani kasus saat pengguna memasukkan jumlah pengulangan negatif. Misalnya, jika jumlah yang dimasukkan kurang dari 0, kita bisa memberikan pesan kesalahan dan meminta input ulang.

c. Menggunakan Do While untuk Eksekusi Pasti: Jika kita ingin memastikan bahwa kata selalu dicetak setidaknya sekali, kita bisa menggunakan struktur do while untuk mencetak kata, meskipun jumlah yang dimasukkan 0.

**[No.2 ] Penyusunan Algoritma ,Kode Program, dan diagram Flowchart**

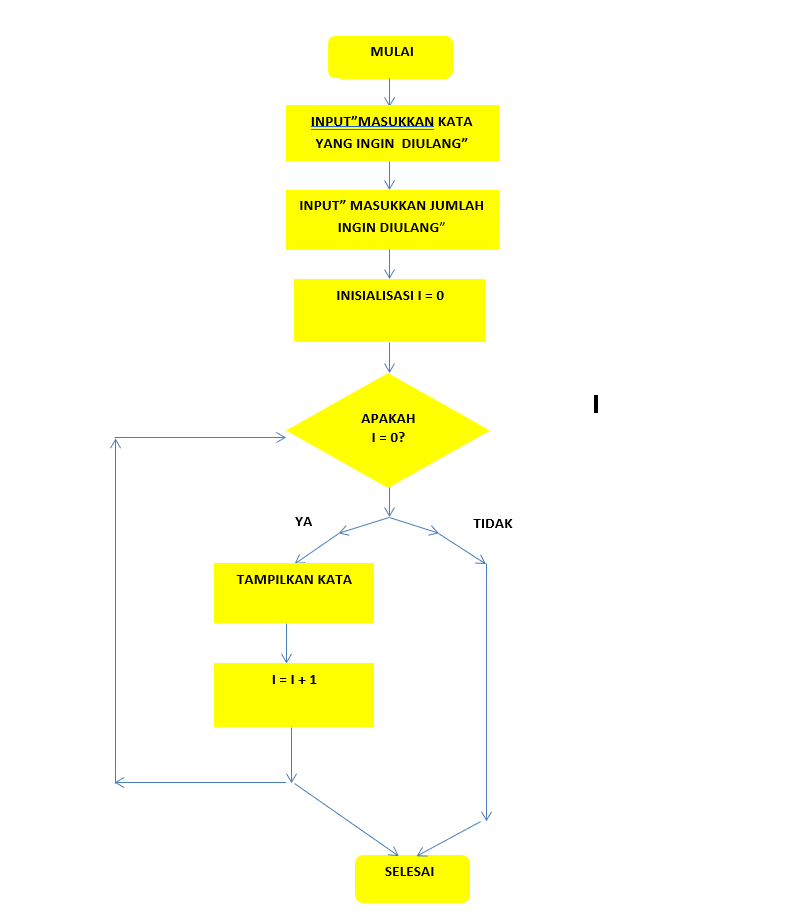
1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma program menghitung Rata Rata Nilai:

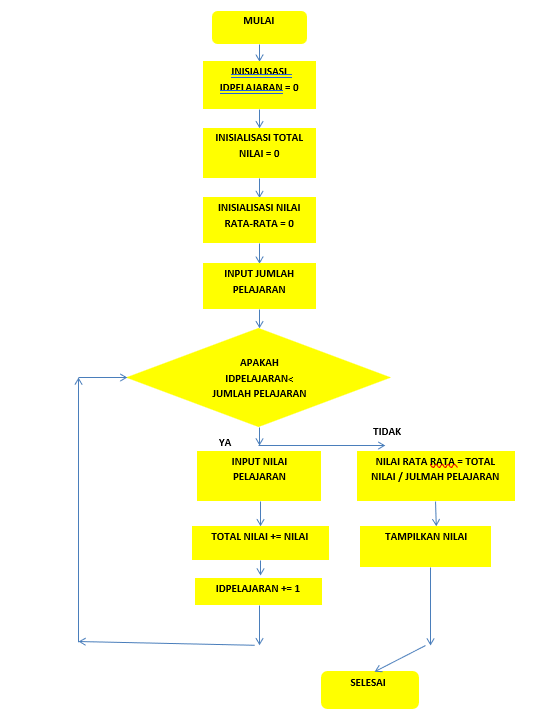
1. Mulai.
2. inisiasi idPelajaran
3. inisiasi nilai pelajaran
4. inisiasi nilai rata-rata
5. Minta pengguna untuk menuliskan jumlah pelajaran
6. Ketika idPelajaran lebih kecil dari jumlah pelajaran
7. Minta pengguna untuk menuliskan nilai pelajaran
8. Hitung nilai rata-rata = (nilai pelajaran + nilai rata-rata) / 2
9. Tambah satu ke idPelajaran
10. Tampilkan nilai rata-rata
11. Selesai.
12. Kode program dan luaran
13. Kode Program
14. Luaran
15. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Secara Keseluruhan program ini efektif dalam menghitung dan menampilkan nilai rata rata, tetapi perlu penanganan kesalahan untuk meningkatkan kualitasnya dan cukup dapat diandalkan untuk perhitungan.

1. Diagram Flowchart
   1. Diagram Flowchart dai Latihan 2.1 dan 2.2



* 1. Diagram Flowchart dari Latihan 2.3



**[No.1] Kesimpulan**

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

Jawab :

Dari permasalahan di atas, saya menyimpulkan perbedaan antara while dan do while: Perbedaan utama terletak pada cara mereka memeriksa kondisi untuk melanjutkan iterasi. Pada loop while, kondisi diperiksa sebelum menjalankan blok kode. Jadi, jika kondisi tidak terpenuhi dari awal, blok kode tidak akan dieksekusi sama sekali. Sebaliknya, pada loop do while, blok kode dijalankan terlebih dahulu sebelum memeriksa kondisi, sehingga setidaknya satu iterasi akan selalu terjadi, bahkan jika kondisi awal tidak terpenuhi.

Dengan demikian, do while lebih cocok digunakan ketika kita ingin memastikan bahwa suatu aksi dilakukan setidaknya sekali, sementara while lebih tepat digunakan ketika kita ingin menghindari aksi jika kondisi tertentu tidak terpenuhi dari awal.

Saya memilih topik perbedaan antara while dan do while karena sesuai dengan yang diminta oleh soal.

**Refleksi**

Saya menyelesaikan tugas ini dalam satu malam karena tugas ini termasuk tingkat kesulitan menengah. Saya berusaha sebaik mungkin dan belajar banyak hal baru. Saya juga bisa meningkatkan kemampuan saya dengan mengganti variabel atau ciri-ciri yang diinstruksikan. Untuk materi selanjutnya, saya berharap mendapatkan materi yang lebih sulit, karena dengan begitu saya bisa mendapatkan lebih banyak ilmu dari tugas tersebut.