

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
PERULANGAN FOR DAN NESTED FOR



Oleh: Afif Naufal Zahran

NIM: 2511533009

DOSEN PENGAMPU: DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

DAFTAR ISI

BAB I PENDAHULUAN
1.1 Latar Belakang.....
1.2 Tujuan
1.3 Manfaat Praktikum
BAB II PEMBAHASAN.....
2,1 Pengertian Perulangan
2,2 Langkah-Langkah Penggunaan Beserta Flowchart dan PseudoCode
BAB III KESIMPULAN
DAFTAR PUSTAKA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada bahasa pemograman ketika kita ingin menjalankan suatu program berulangkali dengan batas yang ditentukan atau hingga suatu kondisi terpenuhi kita dapat menggunakan perulangan dengan *keyword* yang telah disediakan dari bahasa pemograman yang kita gunakan. Untuk *Java* sendiri kita dapat menggunakan perulangan *For*. Dengan perulangan kita juga dapat membuat suatu pola seperti pola segitiga, segitiga siku-siku, segitiga terbalik, belah ketupat dan sebagainya.

1.2 Tujuan

Tujuan dari laporan tugas praktikum ini agar pembaca dan penulis dapat mendalami dan memahami tentang perulangan dan implementasinya di bahasa pemograman *Java*.

1.3 Manfaat

Manfaat dari tugas praktikum tentang perulangan agar pembaca mendapatkan pengetahuan tentang perulangan. Manfaat untuk penulis agar dapat memahami perulangan pada bahasa pemograman *Java*.

BAB II

PEMBAHASAN

2,1 Pengertian Perulangan

Perulangan *For* adalah perulangan yang biasa digunakan untuk mengulangan sebuah proses yang sudah diketahui jumlahnya. Struktur *For* lebih efisien dari lainnya karena simple. Perulangan *For* dipakai untuk *looping* dengan informasi yang mengenai batas awal, syarat, dan perubahan yang akan terjadi. Jika persyaratan di atas sudah terpenuhi, maka pernyataan *For*.

2,2 Langkah-Langkah Penggunaan Beserta Flowchart dan PseudoCode

Pada tugas kali ini penulis diharuskan membuat program yang menghasilkan output dengan pola belah ketupat seperti berikut:

```
#=====#
|      <><>      |
|      <>....<>    |
|      <>.....<>   |
|<>.....<>       |
|<>.....<>       |
|      <>.....<>   |
|      <>....<>     |
|          <><>     |
#=====#
```

Agar mendapatkan output seperti di atas kita akan menggunakan dua perulangan utama. Perulangan pertama berfungsi untuk membuat segitiga bagian atas, dan perulangan ke-2 bergungsi untuk membuat pola segitiga terbalik bagian bawah. Berikut Program untuk perulangan pertama:

```
5● public static void main(String[] args) {  
6     // TODO Auto-generated method stub  
7     int max = 4;  
8     int maxTitik = 4;  
9     //Untuk Membuat Atap  
10    System.out.printf("#");  
11    //menggunakan rumus max*maxTitik untuk rumus atap agar dapat menyesuaikan dengan pola belah ketupat  
12●   for(int i = 0;i<max*maxTitik;i++){  
13       System.out.print("-");  
14   }  
15   System.out.printf("#");  
16   System.out.println(); //Akhir dari line  
17   String s = "<>"; //Agar Memudahkan Pembacaan Kode  
18   //Pengulangan pertama untuk segitiga atas  
19●   for (int i = 0;i<max;i++){//Loop untuk line  
20       System.out.print("");  
21●       for (int j = i;j<(max*2-i)-2;j++){//Loop untuk spasi  
22           //Menggunakan rumus (max*2-i)-2 untuk mendapatkan pola spasi  
23           System.out.print(" ");  
24       }  
25       //lalu digabungkan dengan segitiga "<>"  
26       System.out.print(s);  
27       //Kemudia membuat pola titik dimana selalu ditambahkan 4  
28●       for (int j = 0;j<i*maxTitik;j++){  
29           //menggunakan rumus j*maxTitik untuk mendapatkan pola titik  
30           System.out.print(".");  
31       }  
32       System.out.print(s);  
33       //Menggunakan rumus yang sama seperti pengulangan di atas untuk membuat pola dinding luar  
34●       for (int j = i;j<(max*2-i)-2;j++){  
35           System.out.print(" ");  
36       }  
37       System.out.println("|");  
38   }
```

Gambar 2.1

Pada gambar 2.1 program yang ditulis adalah program untuk menghasilkan pola segitiga dan atap. Pertama kita menginisiasikan variabel max dengan tipe data int dengan value 4, dan maxTitik dengan value 4 sebagai batas berapa banyak segitiga yang akan kita buat dan berapa buah titik yang harus di letakkan di antara *character* “<>”. Lalu kita melakukan perintah `System.out.print("#")` untuk membuat sudut atap, dan menggunakan perulangan dengan kondisi “`max*maxTitik`” berfungsi untuk mencari panjang maksimal lalu di dalam perulangan kita memanggil perintah `System.out.print(“=”)` dan di luar perulangan kita memanggil perintah yang sama seperti di atas.

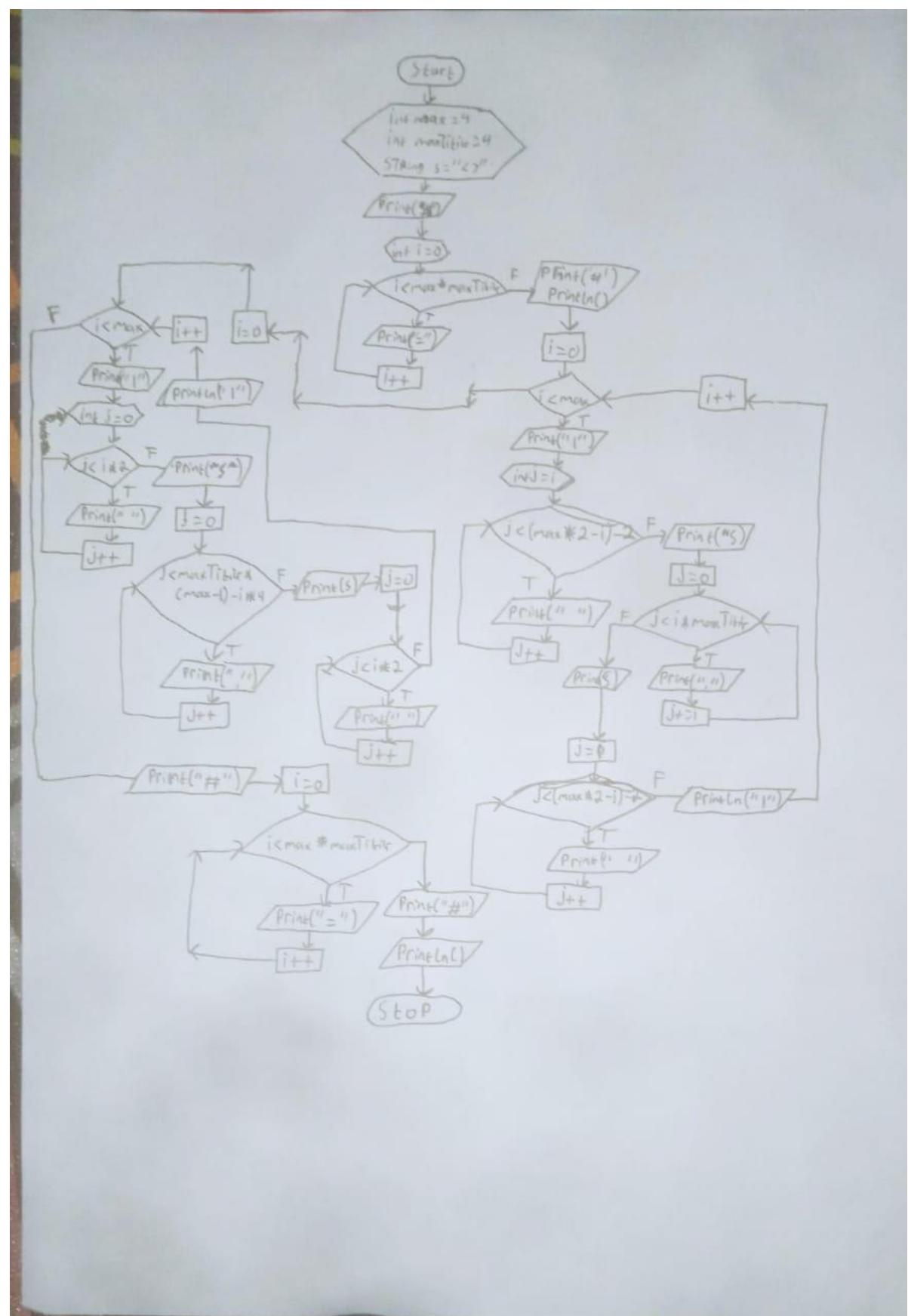
Setelah membuat perulangan untuk atap, kita dapat membuat perulangan untuk membuat pola segitiga atas. Di dalam perulangan kita memanggil perintah `print(|)` sebagai batas samping dan membuat perulangan untuk membuat pola spasi dan segitiga dengan inisiasi `j = i` dan kondisi perulanagn “`j<(max*2-i)-2`” karena kita ingin membuat line yang memiliki spasi dengan pola tiap baris berkurang 2 dengan spasi pada baris pertama 6.

Setelah membuat perulangan untuk pola spasi kita dapat memnggabungkan dengan *character* “<>” dan membuat perulangan untuk membuat pola ‘.’, dimana pada baris pertama tidak ada ‘.’ yang muncul namun pada baris ke-2 terdapat 4 titik dan baris selanjutnya bertambah 4 titik. Kita dapat menggunakan perulangan dengan inisiasi $j = 0$ lalu kondisi “ $j < i * \text{maxTitik}$ ”. Alasan mengapa pada perulangan ini menggunakan kondisi $i * \text{maxTitik}$, misalkan pada baris pertama dimana value $i = 0$; ketika i dikalikan dengan maxTitik akan menghasilkan 0, sehingga para baris pertama tidak ada titik yang dimunculkan, sedangkan ketika $i=1$ atau pada baris kedua, ketika i dikalikan dengan maxTitik akan menghasilkan nilai 4 dan itu akan menjadi batas seberapa banyak titik yang harus di keluarkan.

Setelah membuat pola untuk titik, kita dapat menutupnya menggunakan *character* “<>” dengan memanggil perintah `System.out.print("<>")`, lalu menggunakan perulangan yang sama untuk pola spasi di awal dan menutupnya dengan perintah `System.out.println("|")` untuk membuat batas penutup saming kanan dan melanjutkan pada baris baru.

Setelah itu kita dapat membuat perulangan utama kedua untuk segitiga terbalik dibawah. Di dalam perulangan ini kita juga membuat perulanagn untuk pola spasi namun dengan kondisi “ $j < i * 2$ ” karena sekarang kita menginginkan pada baris pertama tidak ada spasi, lalu baris kedua memiliki 2 spasi dan pada baris setelahnya bertambah 2 spasi lagi. Setelah itu kita memanggil perintah `System.out.print("<>")` untuk memunculkan *character* “<>”. Lalu kita membuat perulanagn untuk membuat pola titik. Pada pola kali ini kita menginginkan pada baris pertama titik yang muncul berjumlah maksimal lalu berkurang 4 dan seterusnya hingga 0. Kita dapat menggunakan kondisi “ $j < \text{maxTitik} * (\text{max}-1) - i * 4$ ” sehingga pada baris pertama kita mendapatkan kondisi dimana maksimal titik yang harus dimunculkan ada 12, lalu pada baris berikutnya berkurang 4, dan seterusnya. Lalu kita menutupnya dengan perintah `System.out.println("|")` agar perulangan setelahnya dilakukan pada baris baru.

a. Flowchart



b. PseudoCode

JUDUL Program Membuat Pola Belah Ketupat {program membuat pola belah ketupat dengan tinggi tiap segitiga atas dan segitiga terbalik bawah 4 baris }
DEKLARASI VAR max,maxTitik,i,j:integer VAR s:string
ALGORITMA max=4 maxTitik=4 s="<>" print("#") for (int i = 0; i<max*maxTitik;i++) print("=".") end for println("#") for (int i = 0; i<max;i++) print(" ") for(int j = i;j<(max*2-i)-2;j++) print(" ") end for print(s) for(int j = 0;j<i*maxTitik;j++) print("."") end for print(s) for(int j = 0;j<(max*2-i)-2;j++) print(" ") end for println(" ") end for

```
for (int i = 0; i<max;i++)
    print("|")
    for(int j =0;j<i*2;j++)
        print(" ")
    end for
    print(s)
    for(int j =0;j<maxTitik*(max-1)-i*4;j++)
        print(".")
    end for
    print(s)
    for(int j = 0;j<i*2;j++)
        print(" ")
    end for
    println("|")
end for
print("#")
for (int i = 0; i<max*maxTitik;i++)
    print(">")
end for
println("#")
```

BAB III

KESIMPULAN

Kesimpulan dari praktikum kali ini adalah perulangan pada bahasa pemograman *Java* begitu penting karena dapat digunakan untuk membuat pola atau melakukan *looping* pada data.

DAFTAR PUSTAKA

- Sumber daring (website) [Diakses: 31-10-2025]
[1] w3schools, “Java Operators”, Tersedia pada:
https://www.w3schools.com/java/java_for_loop.asp. [Diakses: 31-10-2025].