

Laporan Praktikum Algoritma Pemrograman



Disusun Oleh :

Ahda Rindang Al-Amin (2311531003)

Dosen Pengampu : Dr. Wahyudi, S.T, M.T

Asisten Labor : Rahmad Dwirizki Olders

Departemen Informatika
Fakultas Teknologi Informasi
Universitas Andalas
Tahun 2023

Perulangan for

A. Tujuan Praktikum

1. Memahami perulangan *for* pada bahasa pemrograman Java.

B. Pendahuluan

for Loops

Anda dapat mengeksekusi serangkaian kode berulang kali dengan membuat loop. Perulangan digunakan kapan pun Anda perlu melakukan tugas yang berulang karena jauh lebih sederhana dan mudah daripada mencoba menulis urutan pernyataan yang sama berulang kali. Java menyediakan beragam konstruksi loop yang kuat, salah satunya adalah perulangan for. Bentuk perulangan for yang paling sederhana:

for(inisialisasi; kondisi; iterasi) pernyataan;

Dalam bentuknya yang paling umum, bagian inisialisasi loop menetapkan variabel kontrol loop ke nilai awal. Kondisinya adalah ekspresi Boolean yang menguji variabel kontrol loop. Jika hasil pengujian tersebut benar, pernyataan dijalankan dan perulangan for terus melakukan iterasi. Jika salah, perulangan dihentikan. Ekspresi iterasi menentukan bagaimana variabel kontrol loop diubah setiap kali loop diiterasi. Berikut adalah program singkat yang mengilustrasikan perulangan for:

```
/*
  Demonstrate the for loop.

  Call this file ForDemo.java.
*/
class ForDemo {
    public static void main(String args[]) {
        int count;

        for(count = 0; count < 5; count = count+1) ← This loop iterates five times.
            System.out.println("This is count: " + count);

        System.out.println("Done!");
    }
}
```

Output yang dihasilkan oleh program ini:

```
This is count: 0
This is count: 1
This is count: 2
This is count: 3
This is count: 4
Done!
```

Dalam contoh ini, count adalah variabel kontrol loop. Ini diatur ke nol di bagian inisialisasi for. Pada awal setiap iterasi (termasuk iterasi pertama), jumlah pengujian bersyarat < 5 dilakukan. Jika hasil dari pengujian ini benar, pernyataan `println()` dijalankan, dan kemudian bagian iterasi dari perulangan dijalankan, yang menambah jumlah sebanyak 1. Proses ini berlanjut hingga pengujian kondisional bernilai salah, dan pada saat itulah eksekusi dilanjutkan. Menariknya, dalam program Java yang ditulis secara profesional, Anda hampir tidak akan pernah melihat bagian iterasi dari loop ditulis seperti yang ditunjukkan pada program sebelumnya. Artinya, Anda jarang akan melihat pernyataan seperti ini:

```
count = count + 1;
```

Alasannya adalah Java menyertakan operator kenaikan khusus yang melakukan operasi ini dengan lebih efisien. Operator kenaikannya adalah `++` (yaitu, dua tanda tambah berturut-turut). Operator kenaikan menambah operannya sebanyak satu. Dengan menggunakan operator kenaikan, pernyataan sebelumnya dapat ditulis seperti ini:

```
count++;
```

Jadi, for pada program sebelumnya biasanya akan ditulis seperti ini:

```
for(count = 0; count < 5; count++)
```

Java juga menyediakan operator pengurangan, yang ditetapkan sebagai `--`. Operator ini mengurangi operannya sebanyak satu.

C. Metode Praktikum

1. Perulangan for 1

```
1 package pekan4;
2
3 public class PerulanganFor1 {
4
5     public static void main(String[] args) {
6         for (int i=1;i<=10;i++) {
7             System.out.println(i);
8         }
9     }
10
11 }
```

Ini adalah contoh program perulangan for yang sederhana, dimana output yang dihasilkan adalah sebagai berikut:

```
1
2
3
4
5
6
7
8
9
10
```

2. Perulangan for 2

```
1 package pekan4;
2
3 public class PerulanganFor2 {
4
5     public static void main(String[] args) {
6         for (int i=1;i<=10;i++) {
7             System.out.print(i+" ");
8         }
9
10        System.out.println();
11
12        for (int j=10;j>=1;j--) {
13            System.out.print(j+" ");
14        }
15    }
16
17 }
18
```

Pada program ini, iterasi untuk perulangan menggunakan operator unik ++ dan --, dimana i++ berarti i=i+1, sedangkan j-- berarti j=j-1. Output dari program tersebut adalah sebagai berikut:

```
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
```

3. Perulangan for 3

Pada program ini dibutuhkan output seperti berikut :

```
10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1  
jumlah = 55
```

Untuk menghasilkan output tersebut, dapat menggunakan perulangan for. Program yang menggunakan perulangan for dan menghasilkan output diatas dapat berupa seperti berikut :

```
1 package pekan4;  
2  
3 public class PerulanganFor3 {  
4     public static void main(String[] args) {  
5         int j=0;  
6         for (int i=10;i>=1;i--) {  
7             if (i>1) {  
8                 System.out.print(i+" + ");  
9             } else {  
10                System.out.println(i);  
11            }  
12            j=j+i;  
13        }  
14        System.out.println("jumlah = " + j );  
15    }  
16 }  
17
```

4. Perulangan for 4

Program berikut menghasilkan output yang hampir sama dengan program 'Perulangan for 3'. Perbedaannya, pada program 'Perulangan for 3' nilai tertinggi di awal adalah 10, sedangkan pada program ini nilai tertinggi di awal diinput setelah program dijalankan.

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class PerulanganFor4 {
6     public static void main(String[] args) {
7
8         //input batas angkanya
9         int batas;
10        Scanner input = new Scanner(System.in);
11        System.out.print("Batas = ");
12        batas=input.nextInt();
13        input.close();
14
15        int j=0;
16        //penjumlahannya
17        for (int i=batas;i>=1;i--) {
18            if (i>1) {
19                System.out.print(i+" + ");
20            } else {
21                System.out.println(i);
22            }
23            j=j+i;
24        }
25        //jumlahnya
26        System.out.println("jumlah = " + j );
27    }
28 }
29

```

Batas adalah nilai tertinggi yang ada di awal. Nilai batas diinput setelah program dijalankan. Berikut output dari program tersebut dengan contoh input nilai batas adalah 23:

```

Batas = 23
23 + 22 + 21 + 20 + 19 + 18 + 17 + 16 + 15 + 14 + 13 + 12 + 11 + 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1
jumlah = 276

```

5. Perulangan for 5

Program ini membutuhkan output seperti berikut:

```

*
**
***
****

```

Untuk menghasilkan output seperti diatas, dapat dilakukan dengan menggunakan nested for loop. Dimana loop for yang di luar untuk perulangan baris output, sedangkan loop for yang di dalam berfungsi untuk perulangan bintang dalam satu baris. Program yang menggunakan perulangan for dan menghasilkan output diatas dapat berupa seperti berikut :

```
1 package pekan4;
2
3 public class PerulanganFor5 {
4
5     public static void main(String[] args) {
6         int i,j;
7         for (i=1; i<=4; i++) {
8             for (j=1;j<=i;j++) {
9                 System.out.print("*");
10            }
11            System.out.println();
12        }
13
14
15    }
16
17 }
```

D. Kesimpulan Praktikum

Pada bahasa pemrograman Java, anda dapat mengeksekusi serangkaian kode berulang kali dengan membuat loop. Perulangan digunakan kapan pun Anda perlu melakukan tugas yang berulang karena jauh lebih sederhana dan mudah daripada mencoba menulis urutan pernyataan yang sama berulang kali. Java menyediakan beragam konstruksi loop yang kuat, salah satunya adalah perulangan for.