

# LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN

## Perulangan For Loop dan Nested Loop



Oleh: Afif Naufal Zahran

NIM: 2511533009

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

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

## **Kata Pengantar**

Pedoman ini disusun sebagai rujukan resmi bagi mahasiswa Departemen Informatika dalam penyusunan laporan praktikum pada mata kuliah *Pemrograman Dasar dengan Java*. Dokumen ini tidak hanya memberikan gambaran umum mengenai format penulisan, tetapi juga menguraikan secara rinci sistematika laporan, tata cara penyajian isi, serta contoh penulisan kode program yang dilengkapi dengan referensi ilmiah. Melalui panduan ini, mahasiswa diharapkan mampu menyusun laporan yang tidak sekadar memenuhi aspek administratif, tetapi juga mencerminkan ketelitian, keteraturan, dan penerapan kaidah penulisan akademik pada tingkat dasar. Dengan demikian, laporan praktikum yang dihasilkan dapat berfungsi sebagai media pembelajaran, dokumentasi kegiatan, sekaligus sarana untuk melatih keterampilan menulis ilmiah yang akan bermanfaat dalam jenjang studi selanjutnya.

Padang, 2025

Tim Penyusun

## DAFTAR ISI

BAB I .....	
1.1 Latar Belakang.....	
1.2 Tujuan .....	
1.3 Manfaat .....	
BAB II .....	
2,1 Teori.....	
2,2 Program.....	
BAB III.....	
3,1 Kesimpulan.....	
3,2 Saran .....	

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada bahasa pemograman ketika kita ingin menjalankan suatu program berulang kali 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* dan *While*. Namun pada pertemuan minggu ke-5 kita hanya akan membahas perulangan *For* dan *Nested For*.

### **1.2 Tujuan**

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

### **1.3 Manfaat**

Manfaat dari praktikum tentang operator agar pembaca mendapatkan pengetahuan tentang perulangan *For* dan *Nested For*. Manfaat untuk penulis agar dapat memahami operator pada bahasa pemograman *Java*.

## BAB II

### PEMBAHASAN

#### 2,1 Teori

Perulangan *For* adalah perulangan yang biasa digunakan untuk mengulangi 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*. Berikut contoh sintaks untuk kode *For* di *Java*:

```
for(inisialisasi;kondisi;update){
```

```
    //Kode anda disini
```

```
}
```

Atau kode java nya:

```
for(int i = 0;i<10;i++){
```

```
    //Kode yang akan diulang sebanyak 10 kali
```

```
    //Dimulai dari 0-9
```

```
}
```

#### 2,2 Program

Pada pertemuan pekan ke-5 terdapat 7 program yang telah dibuat yakni: *nestedFor0.java*, *nestedFor1.java*, *nestedFor2.java*, *PerulanganFor1.java*, *PerulanganFor2.java*, *PerulanganFor3.java*, dan *PerulanganFor4.java*.

a. PerulanganFor1.java

```
1 package pekan5_2511533009;
2
3 public class PerulanganFor1_2511533009 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int i = 1; i <= 10; i++) {
8             System.out.println(i);
9         }
10    }
11
12 }
```

gambar 2.1

Pada gambar 2.1 program kali ini kita membuat perulangan sederhana dengan melakukan perulangan sebanyak 10 kali dimulai dari 1 hingga 10 dan menjalankan perintah `System.out.println(i);` dan akan menghasilkan output 1 hingga 10 tiap baris

b. PerulanganFor2.java

```
1 package pekan5_2511533009;
2
3 public class perulanganFor2_2511533009 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int i = 1; i <= 10; i++) {
8             System.out.print(i+" ");
9         }
10    }
11
12 }
```

gambar 2.2

Pada gambar 2.2 program kali ini melakukan perulangan sederhana juga seperti gambar 2.1 tetapi yang membedakan perintah yang dijalankan adalah `System.out.print(i+" ");`; dimana outputnya adalah 1 2 3 4 5 6 7 8 9 10 pada 1 baris karena perintah “print” akan mengeluarkan output pada 1 baris.

c. PerulanganFor3.java

```
1 package pekan5_2511533009;
2
3 public class perulanganFor3_2511533009 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int jumlah=0;
8         for(int i = 1;i<=10;i++) {
9             System.out.print(i);
10            jumlah= jumlah+i;
11            if(i<10) {
12                System.out.print(" + ");
13            }
14        }
15        System.out.println();
16        System.out.println("Jumlah = "+jumlah);
17    }
18
19 }
```

gambar 2.3

Pada gambar 2.3 program kali ini menjalankan perulangan dengan percabangan if didalamnya dan menjumlahkan semua bilangan 1 hingga 10 dimana setelah angka 1 hingga 9 akan ditambahkan *character* '+' dan menghasilkan output "1+2+3+4+5+6+7+8+9+10"  
"Jumlah = 55"

d. PerulanganFor4.java

```
1 package pekan5_2511533009;
2 import java.util.Scanner;
3
4 public class PerulanganFor4_2511533009 {
5
6     public static void main(String[] args) {
7         int jumlah=0;
8         int batas;
9         Scanner input = new Scanner(System.in);
10        System.out.print("Masukkan nilai batas = ");
11        batas = input.nextInt();
12        input.close();
13
14        for(int i = 1;i<=batas;i++) {
15            System.out.print(i);
16            jumlah= jumlah+i;
17            if(i<batas) {
18                System.out.print(" + ");
19            } else {
20                System.out.print(" = ");
21            }
22        }
23
24        System.out.println(jumlah);
25    }
26
27 }
```

gambar 2.4

Pada gambar 2.4 program melakukan input sebagai batas dari perulangan for yang dilakukan sehingga kita dapat membuat program penjumlahan 1 hingga batas yang kita input.

e. NestedFor0.java

```
1 package pekan5_2511533009;
2
3 public class nestedFor0_2511533009 {
4
5     public static void main(String[] args) {
6         for (int line = 1;line <= 5;line++) {
7             for (int j = 1; j <= (-1 * line + 5);j++) {
8                 System.out.print(".");
9             }
10            System.out.print(line);
11            System.out.println();
12        }
13
14    }
15
16
17 }
```

gambar 2.5

Pada gambar 2.5 program melakukan pengulangan, dan didalam perulangan juga menjalankan perulangan dengan kondisi  $j \leq -1 * \text{line} + 5$  sehingga menghasilkan pola segitiga siku-sikut dan tiap ujung baris terdapat angka yang menunjukan baris seberapa.



f. NestedFor1.java

```
1 package pekan5_2511533009;
2
3 public class nestedFor1_2511533009 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int i = 1; i <= 5; i++) {
8             for (int j = 1; j <= 5; j++) {
9                 System.out.print("*");
10            }
11            System.out.println();
12            // to end the line
13        }
14
15    }
16 }
17
18 }
```

gambar 2.6

Pada gambar 2.6 program menjalankan perulangan dan di dalam perulangan juga menjalankan perulangan dengan kondisi yang sama seperti pada perulangan pertama, dan menghasilkan pola persegi dengan *character* '\*'.

g. NestedFor2.java

```
1 package pekan5_2511533009;
2
3 public class nestedFor2_2511533009 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int i = 0; i <= 5; i++) {
8             for (int j = 0; j <= 5; j++) {
9                 System.out.print(i+j + " ");
10            }
11            System.out.println();
12            // to end the line
13        }
14
15    }
16 }
17
18 }
```

gambar 2.7

Pada gambar 2.7 program menjalankan perulangan yang sama seperti program pada gambar 2.6 tetapi mengeluarkan penjumlahan i dan j dan menghasilkan pola persegi.

## **BAB III**

### **PENUTUPAN**

#### 3,1 Kesimpulan

Perulangan *For* pada bahasa pemrograman *Java* sangat penting karena hal yang sangat fundamental pada bahasa pemrograman terutama bahasa pemrograman *Java*.

#### 3,2 Saran

Laporan praktikum ini masih memiliki kekurangan. Karena itu, penulis sangat terbuka terhadap saran dan kritikan agar dapat meningkatkan kualitas laporan ini dan laporan-laporan selanjutnya.