

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
“PERULANGAN FOR PADA BAHASA PEMROGRAMAN JAVA”

disusun Oleh:
RAZIF AL FARISI
NIM 2511532028
Dosen Pengampu: Dr.Wahyudi S.T, M.T

Asisten Laboratorium: Rahmad Dwirizki Olders



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2025

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga laporan praktikum dengan judul **“Perulangan FOR pada Bahasa Pemrograman Java”** dapat diselesaikan dengan baik dan tepat waktu.

Laporan ini disusun sebagai bagian dari kegiatan praktikum pemrograman Java yang bertujuan untuk memahami konsep dan penerapan struktur perulangan (*looping*), khususnya perulangan *FOR*. Melalui kegiatan praktikum ini, peserta praktikum diharapkan mampu memahami logika dasar perulangan, menyusun algoritma yang efisien, serta mengimplementasikannya dalam bentuk program menggunakan bahasa Java secara tepat dan sistematis.

Ucapan terima kasih disampaikan kepada dosen pengampu mata kuliah, asisten laboratorium, dan seluruh pihak yang telah memberikan bimbingan serta dukungan selama pelaksanaan praktikum ini.

Harapan besar agar laporan ini dapat memberikan manfaat, menambah wawasan, serta menjadi referensi dalam mempelajari konsep dasar perulangan dalam bahasa pemrograman Java.

Padang, 31 Oktober 2025

Penulis

Daftar Isi

KATA PENGANTAR	i
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Manfaat	1
BAB II	3
PEMBAHASAN.....	3
2.1 Struktur Kontrol Perulangan	3
2.1.1 Perulangan Tunggal – Perulangan <i>For</i>	3
2.1.2 Perulangan Bersarang – Nested For	3
2.2 Langkah Pengerjaan	4
2.2.1 PerulanganFor1	4
2.2.2 PerulanganFor2	4
2.2.3 NestedFor1	6
2.2.4 NestedFor2	7
BAB III.....	8
PENUTUP	8
3.1 Kesimpulan	8
3.2 Saran	8
DAFTAR PUSTAKA	9

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi di era digital menuntut adanya pemahaman mendalam terhadap bahasa pemrograman sebagai sarana komunikasi antara manusia dan komputer. Bahasa pemrograman memungkinkan seseorang memberikan instruksi logis yang dapat dieksekusi secara otomatis oleh komputer.

Salah satu aspek penting dalam pemrograman adalah struktur kontrol, yaitu mekanisme yang mengatur alur eksekusi suatu program. Struktur kontrol perulangan dibedakan menjadi tiga jenis, yaitu:

1. Struktur urutan (*sequence*), yaitu perintah yang dieksekusi secara berurutan dari atas ke bawah.
2. Struktur percabangan (*selection* atau *branching*), yaitu perintah yang dijalankan berdasarkan kondisi tertentu.
3. Struktur perulangan (*looping*), yaitu perintah yang dijalankan secara berulang selama memenuhi kondisi tertentu.

Fokus laporan ini adalah struktur perulangan (*looping*), khususnya perulangan *FOR*. Struktur ini banyak digunakan karena memberikan kemudahan dalam menulis kode yang melakukan pengulangan dengan jumlah iterasi yang telah diketahui.

1.2 Tujuan

Tujuan dari penyusunan laporan praktikum ini adalah sebagai berikut:

1. Menjelaskan konsep dasar perulangan *FOR* pada bahasa pemrograman Java.
2. Menjelaskan jenis-jenis perulangan *FOR* dan penggunaannya dalam program.
3. Menyajikan contoh penerapan *FOR* dan *nested FOR* dalam pemrograman Java.
4. Menganalisis hasil eksekusi program berdasarkan logika iterasi yang diterapkan.

1.3 Manfaat

1. Sebagai sarana pembelajaran untuk memahami konsep dan implementasi perulangan *FOR* pada Java.

2. Sebagai referensi tambahan bagi dosen maupun asisten laboratorium dalam kegiatan pengajaran struktur kontrol pemrograman.
3. Sebagai acuan bagi pembaca umum dalam mempelajari penggunaan perulangan *FOR* pada bahasa Java.

BAB II

PEMBAHASAN

2.1 Struktur Kontrol Perulangan

Struktur kontrol perulangan atau *looping* merupakan mekanisme yang digunakan untuk mengeksekusi serangkaian perintah berulang kali selama kondisi tertentu terpenuhi. Dalam bahasa Java, terdapat tiga jenis perulangan utama, yaitu *for*, *while*, dan *do-while*. Jenis perulangan *for* digunakan ketika jumlah iterasi sudah diketahui sebelumnya. Struktur *for* terdiri atas tiga bagian utama: inisialisasi, kondisi, dan perubahan nilai variabel iterasi. Dalam praktikum ini, terdapat dua bentuk utama dari perulangan *for*:

1. **Perulangan tunggal (*PerulanganFor*)** – hanya menggunakan satu *for* untuk menjalankan perintah berulang.
2. **Perulangan bersarang (*NestedFor*)** – menggunakan *for* di dalam *for* untuk membentuk pola tertentu atau operasi bertingkat.

2.1.1 Perulangan Tunggal – Perulangan *For*

- Perulangan tunggal digunakan untuk menjalankan perintah berulang secara linear berdasarkan kondisi tertentu.
- Sintaks umum:

```
for (inisialisasi; kondisi; perubahan) {
    // perintah berulang
}
```
- Variabel penghitung umumnya bertipe *int*.
- Kondisi menggunakan operator perbandingan seperti *<*, *<=*, atau *>*.
- Perubahan variabel bisa berupa penambahan (*i++*) atau pengurangan (*i--*).

2.1.2 Perulangan Bersarang – Nested For

- *Nested for* adalah perulangan *for* di dalam *for* lain. Biasanya digunakan untuk membuat bentuk pola, tabel, atau perhitungan dua dimensi.
- Sintaks Umum:

```
for (inisialisasi1; kondisi1; perubahan1) {
    for (inisialisasi2; kondisi2; perubahan2) {
```

- ```

 perintah;
 }
}

 - Perulangan luar menentukan baris, dan perulangan dalam menentukan kolom.
 - Jumlah iterasi total merupakan hasil perkalian dari kedua perulangan.
```

## 2.2 Langkah Pengerjaan

### 2.2.1 PerulanganFor1

1. Deklarasi variabel: membuat variabel *i* untuk digunakan sebagai penghitung iterasi.
2. Menentukan batas perulangan: kondisi *i*  $\leq 10$  agar perulangan berjalan sebanyak sepuluh kali.
3. Menentukan operasi perubahan: *i++* digunakan untuk menambah nilai *i* setiap iterasi.
4. Menuliskan perintah di dalam blok perulangan: misalnya untuk menampilkan bilangan genap dari 1–20.
5. **Kode Program :**

```

public class PerulanganFor1 {
 public static void main(String[] args) {
 for (int i = 1; i <= 10; i++) {
 System.out.println(i);
 }
 }
}

```

6. **Hasil Program :**

```

1
2
3
4
5
6
7
8
9
10

```

### 2.2.2 PerulanganFor2

1. Menggunakan *Scanner* untuk menerima input batas.
2. Melakukan perulangan *for* dari 1 hingga batas yang dimasukkan pengguna.

3. Setiap nilai ditambahkan ke variabel *jumlah*.
4. Program menampilkan proses penjumlahan dan hasil akhirnya.

5. **Kode Program :**

```
import java.util.Scanner;

public class PerulanganFor4 {
 public static void main(String[] args) {
 int jumlah = 0;
 int batas;
 Scanner input = new Scanner(System.in);
 System.out.print("Masukkan nilai batas = ");
 batas = input.nextInt();
 input.close();
 for (int i = 1; i <= batas; i++) {
 System.out.print(i);
 jumlah = jumlah + i;
 if (i < batas) {
 System.out.print(" + ");
 } else {
 System.out.print(" = ");
 }
 }
 System.out.println(jumlah);
 }
}
```

6. **Hasil Program :**

```
Masukkan nilai batas = 4
1 + 2 + 3 + 4 = 10
```

### 2.2.3 NestedFor1

1. Program dimulai dari fungsi main().
2. Program menggunakan dua perulangan *for* — satu di luar (untuk baris), dan satu di dalam (untuk titik).
3. Perulangan luar (*line*) berjalan dari 1 sampai 5.
4. Di setiap perulangan luar, perulangan dalam (*j*) mencetak titik (".") sebanyak (5 - *line*) kali.
  - Saat *line* = 1, program mencetak 4 titik.
  - Saat *line* = 2, mencetak 3 titik.
  - Saat *line* = 3, mencetak 2 titik.
  - Saat *line* = 4, mencetak 1 titik.
  - Saat *line* = 5, tidak mencetak titik.
5. Setelah titik selesai dicetak, program mencetak angka sesuai nilai *line*.
6. Program kemudian pindah ke baris baru dan mengulang langkah tersebut sampai *line* = 5.
7. Setelah selesai, program berhenti.
8. **Kode Program :**

```
public class nestedFor0 {

 public static void main(String[] args) {
 for (int line = 1; line <= 5; line++) {
 for (int j = 1; j <= (-1 * line + 5); j++)
 {
 System.out.print(".");
 }
 System.out.print(line);
 System.out.println();
 }
 }
}
```

**9. Hasil Program :**

```
....1
...2
..3
.4
5
```

**2.2.4 NestedFor2**

1. Inisialisasi  $i = 0$
2. Bandingkan  $i$  dengan 5
3. Untuk  $i \leq 5$  bernilai benar, (untuk baris)
  - Inisialisasi  $j = 0$
  - Untuk  $j \leq 5$  bernilai benar, print  $(i+j+ " ")$  lalu  $i=i+1$  (untuk kolom)
  - Print  $" "($ untuk akhiran tiap baris dan tersusun menjadi persegi)
  - Jika bernilai salah, lanjutkan ke no.3
4. Jika bernilai salah, perulangan berhenti
5. Print hasil program

**6. Kode Program :**

```
public class nestedFor2 {

 public static void main(String[] args) {
 for (int i = 0; i <= 5; i++) {
 for (int j = 0; j <= 5; j++) {
 System.out.print(i+j+ " ");
 }
 System.out.println();
 }
 }
}
```

**7. Hasil Program :**

```
0 1 2 3 4 5
1 2 3 4 5 6
2 3 4 5 6 7
3 4 5 6 7 8
4 5 6 7 8 9
5 6 7 8 9 10
```

## BAB III

# PENUTUP

### 3.1 Kesimpulan

Dari hasil praktikum, dapat disimpulkan bahwa perulangan *for* dan *nested for* merupakan struktur dasar yang sangat penting dalam pemrograman Java.

- *Perulangan for* digunakan untuk pengulangan dengan jumlah iterasi tertentu.
- *Nested for* digunakan ketika dibutuhkan dua atau lebih pengulangan bersarang, seperti pembuatan pola atau perhitungan dua dimensi.

Dengan memahami cara kerja dan sintaks kedua jenis perulangan tersebut, pemrogram dapat membuat program yang lebih efisien dan terstruktur.

### 3.2 Saran

1. Sebelum pelaksanaan praktikum, perlu diberikan pemahaman teoritis mengenai logika iterasi agar mahasiswa lebih siap dalam mengimplementasikan program.
2. Perlu disertakan latihan dengan tingkat kompleksitas berbeda agar mahasiswa dapat memahami variasi penggunaan perulangan *FOR*.

## **DAFTAR PUSTAKA**

Naughton, Patrick, *Java Handbook: Konsep Dasar Pemrograman Java* , Andi,Yogyakarta, 1996

Gary Cornell dan Cay S.Horstmann, Core Java edisi Indonesia , Andi Yo-gyakarta, 1997

ANuff, Penuntun Pemrograman Java, Andi Yogyakarta, 1997

Kadir, Abdul. Dasar Pemrograman Java 2. Andi. Yogyakarta, 2008