

LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGAMAN

PERULANGAN FOR

Disusun Oleh:

IHSANUL ZAKY EL-MUHAMMADY

NIM:2511533001

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

ASISTEN LABORATORIUM: AUFAN TAUFIQUR RAHMAN



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Dengan penuh rasa syukur, saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa, yang telah memberikan kemudahan dan kesempatan sehingga laporan praktikum ini dapat selesai dengan baik.

Saya juga ingin menyampaikan penghargaan yang setinggi-tingginya kepada Dr. Wahyudi, S.T., M.T. selaku dosen pengampu Mata Kuliah Algoritma dan Pemrograman, yang telah memberikan bimbingan dan motivasi selama perkuliahan. Ucapan terima kasih juga saya sampaikan kepada Uda Aufan Tafiqurrahman, asisten laboratorium, yang dengan sabar dan penuh perhatian telah membimbing kami dalam praktikum ini.

Laporan ini disusun sebagai salah satu tugas dalam Mata Kuliah Pemrograman Dasar dengan tema “Struktur Control Percabangan”.

Saya menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, segala masukan dan saran yang membangun sangat saya harapkan untuk perbaikan di masa mendatang.

Padang, 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....

DAFTAR ISI

BAB I PENDAHULUAN

- 1.1 Latar Belakang
- 1.2 Tujuan.....
- 1.3 Manfaat.....

BAB II PEMBAHASAN

- 2.1 Program Perulangan for 1
- 2.2 Program Perulangan for 2
- 2.3 Program Perulangan for 3
- 2.4 Program Perulangan for 4
- 2.5 Program Nested for 0
- 2.6 Program Nested for 1

BAB III KESIMPULAN.....

- 3.1 Kesimpulan
- 3.2 Saran.....

DAFTAR PUSTAKA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perulangan merupakan proses menjalankan suatu perintah secara berulang-ulang sesuai dengan jumlah atau batas yang telah ditentukan. Dalam bahasa pemrograman **Java**, terdapat beberapa jenis pernyataan (statement) perulangan yang dapat digunakan, salah satunya adalah perulangan “for”

Perulangan dengan teknik *for* dikendalikan oleh tiga komponen utama yang terletak di dalam tanda kurung dan dipisahkan oleh tanda titik koma. Komponen pertama adalah **inisialisasi**, yaitu tahap mendeklarasikan variabel sebagai titik awal perulangan. Biasanya variabel ini memiliki tipe data *integer* atau *float*. Komponen kedua adalah **kondisi**, yang berfungsi untuk memeriksa apakah perulangan masih memenuhi syarat untuk dijalankan. Jika kondisi ini terpenuhi, maka perintah di dalam blok perulangan akan dieksekusi. Komponen ketiga adalah increment atau decrement, yaitu bagian yang menentukan perubahan nilai variabel setiap kali perulangan dijalankan.

Jika kondisi pada komponen kedua sudah tidak terpenuhi, maka program tidak lagi mengeksekusi bagian ketiga, dan proses perulangan akan berhenti secara otomatis.

1.2 Tujuan

- 1 Menjelaskan konsep dasar dan cara kerja perulangan for di Java.
- 2 Menunjukkan contoh penerapan perulangan for dalam program.
- 3 Melatih kemampuan logika pemrograman dengan menggunakan struktur perulangan for

1.3 Manfaat

1. Menambah pemahaman tentang struktur perulangan for dalam bahasa Java.
2. Membantu dalam penerapan logika pemrograman untuk *problem solving*.
3. Menjadi dasar pengetahuan dalam pengembangan program yang lebih kompleks di masa mendatang, terutama yang melibatkan pengolahan data dan algoritma berulang.

BAB II

PEMBAHASAN

2.1 Program Perulangan for 1

```
1 package Pekan5;
2
3 public class PerulanganFor1_2511533001 {
4     public static void main(String[] args) {
5         for (int i=1; i <= 10; i++) {
6             System.out.println(i);
7         }
8     }
9 }
```

- a) Inisialisasi for: int i = 1;
Variabel lokal **i** dibuat dan diberi nilai awal 1.
- b) Pemeriksaan kondisi i <= 10
Jika benar, masuk ke badan loop.
Jika salah, keluar dari loop dan program selesai.
- c) Badan loop: System.out.println(i);
Mencetak nilai **i** ke konsol dan menambahkan baris baru.
- d) Update/Increment i++
Setelah badan loop selesai, nilai **i** ditambah 1.
- e) Ulangi langkah 2–4 sampai kondisi i <= 10 menjadi false.
Karena dimulai dari i = 1 dan ditambah 1 setiap iterasi, loop berjalan untuk i = 1, 2, ..., 10 (total 10 kali).
Setelah i menjadi 11, kondisi i <= 10 salah dan perulangan berhenti.

2.2 Program Perulangan for 2

```
1 package Pekan5;
2
3 public class PerulanganFor2_2511533001 {
4     public static void main(String[] args) {
5         for (int i=1; i <= 10; i++) {
6             System.out.print(i+" ");
7         }
8     }
9 }
```

- a. for (int i = 1; i <= 10; i++)
int i = 1 → variabel penghitung (*counter*) dimulai dari angka 1.
i <= 10 → loop akan terus berjalan selama nilai **i** kurang dari atau sama dengan 10.
i++ → setiap kali loop selesai dijalankan, nilai **i** bertambah 1.

- b. Badan loop: `System.out.print(i + " ")`
 Menampilkan nilai **i** ke layar.
 Menggunakan **print**, bukan **println**, sehingga tidak membuat baris baru setelah setiap angka.
 Tanda " " memberikan spasi di antara angka-angka agar output terlihat lebih rapi.
- c. Loop berulang dari **i = 1** sampai **i = 10**, lalu berhenti setelah **i** bertambah menjadi 11 (karena kondisi **i <= 10** tidak lagi terpenuhi).

2.3 Program Perulangan for 3

```

1 package Pekan5;
2
3 public class PerulanganFor3_2511533001 {
4     public static void main(String[] args) {
5         int jumlah=0;
6         for (int i=1; i<=10; i++) {
7             System.out.print(i);
8             jumlah+= jumlah+i;
9             if (i<10) {
10                 System.out.print(" + ");
11             }
12         }
13     }
14 }
```

<terminated> PerulanganFor3_2511533001 [Java Application] C:\Users\FA506NCI

- a) `for (int i = 1; i <= 10; i++)`

Variabel **i** dimulai dari 1.

Perulangan berjalan selama **i ≤ 10**.

Setelah setiap iterasi, nilai **i** bertambah 1 (**i++**).

- b) Di dalam loop:

- `System.out.print(i);` → menampilkan nilai **i** di layar.
- `jumlah = jumlah + i;` → menambahkan nilai **i** ke variabel **jumlah**.
- `if (i < 10)` → menampilkan tanda “ + ” di antara angka, kecuali setelah angka terakhir (10), agar hasil tampilan lebih rapi.

- c) Setelah loop selesai:

- `System.out.println();` → membuat baris baru.
- `System.out.println("Jumlah = " + jumlah);` → menampilkan total hasil penjumlahan semua angka.

2.4 Program Perulangan for 4

```

1 package Pekan5;
2
3 import java.util.Scanner;
4
5 public class PerulanganFor4_2511533001 {
6
7     public static void main(String[] args) {
8         int jumlah=0;
9         int batas;
10        Scanner input= new Scanner (System.in);
11        System.out.print("Masukkan nilai batas = ");
12        batas=input.nextInt();
13        input.close();
14        for (int i=1; i<=batas;i++) {
```

<terminated> PerulanganFor4_2511533001 [Java Application] C:\Users\FA506NCI\p2\pool\plugins\org.eclipse.jst

1. Deklarasi variabel

- int jumlah = 0; → digunakan untuk menyimpan hasil penjumlahan.
- int batas; → digunakan untuk menyimpan batas angka yang dimasukkan oleh pengguna.

2. Input dari pengguna

- Scanner input = new Scanner(System.in); → membuat objek Scanner untuk membaca input dari keyboard.
- System.out.print("Masukkan nilai batas = "); → menampilkan pesan agar pengguna memasukkan angka.
- batas = input.nextInt(); → membaca angka yang dimasukkan pengguna dan menyimpannya ke variabel batas.
- input.close(); → menutup objek Scanner (praktik yang baik untuk mencegah kebocoran *resource*).

3. Perulangan for (int i = 1; i <= batas; i++)

- Perulangan dimulai dari i = 1 dan berakhir di i = batas.
- Setiap iterasi:
 - Menampilkan nilai i.
 - Menambahkan nilai i ke variabel jumlah.
 - Jika i masih lebih kecil dari batas, tampilkan tanda " + ".
 - Jika i sudah mencapai batas, tampilkan tanda " = ".

4. Menampilkan hasil akhir

Setelah perulangan selesai, program mencetak nilai total dari jumlah di akhir baris.

2.5 Program Nested for 0

The screenshot shows a Java code editor with the following code:

```
1 package Pekan5;
2
3 public class nestedFor0_2511533001 {
4     public static void main(String[] args) {
5         for (int line = 1; line <= 5; line++) {
6             for (int j = 1; j <= (-1 * line + 5); j++) {
7                 System.out.print(".");
8             }
9             System.out.print(line);
10            System.out.println();
11        }
12    }
13 }
```

Below the code editor, there is a terminal window showing the output of the program:

```
<terminated> nestedFor0_2511533001 [Java Application] C:\Users\FA506NCG\p2\pool\p
....1
....2
....3
....4
....5
```

1. Perulangan pertama (luar)

```
for (int line = 1; line <= 5; line++)
```

- Variabel line berfungsi sebagai penghitung baris.
- Nilainya dimulai dari 1 hingga 5.
- Dengan demikian, loop luar akan berjalan sebanyak 5 kali (untuk baris ke-1 sampai ke-5).

2. Perulangan kedua (dalam)

```
for (int j = 1; j <= (-1 * line + 5); j++)
```

- Loop dalam digunakan untuk mencetak tanda titik "..".
- Jumlah titik bergantung pada nilai line.
- Rumusnya: $(-1 * line + 5) \rightarrow$ menghasilkan jumlah titik yang berkurang setiap kali baris bertambah.

3. `System.out.print(line);`

- Setelah mencetak titik, program akan mencetak nilai dari line di akhir setiap baris.

4. `System.out.println();`

- Digunakan untuk pindah ke baris baru setelah satu baris selesai dicetak.

2.6 Program Nested for 1

The screenshot shows the Eclipse IDE interface with a Java code editor and a terminal window. The code in the editor is:

```
1 package Pekan5;
2
3 public class nestedFor_2311533001 {
4     public static void main(String[] args) {
5         for (int line = 1; line <= 5; line++) {
6             for (int j = 1; j <= (-1 * line + 5); j++) {
7                 System.out.print(".");
8                 System.out.print(line);
9                 System.out.println();
10            }
11        }
12    }
13 }
```

The terminal window shows the output of the program:

```
*****  
*****  
*****  
*****  
*****
```

1. Loop luar (i)

```
for (int i = 1; i <= 5; i++)
```

- Mengatur jumlah baris.
- Loop berjalan sebanyak 5 kali, yaitu untuk baris ke-1 sampai ke-5.

2. Loop dalam (j)

```
for (int j = 1; j <= 5; j++)
```

- Mengatur jumlah kolom di setiap baris.
- Setiap kali loop luar berjalan satu kali, loop dalam akan mencetak 5 bintang (*) pada satu baris.

3. System.out.print("*");

- Mencetak tanda bintang tanpa pindah ke baris baru.

4. System.out.println();

- Setelah satu baris selesai dicetak (5 bintang tercetak), perintah ini membuat program pindah ke baris berikutnya.

2.7 Program Nested for 2

The screenshot shows a Java code editor with the following code:

```
1 package Pekan5;
2
3 public class nestedFor2_2511533001 {
4
5     public static void main(String[] args) {
6         for (int i = 0; i <= 5; i++) {
7             for (int j = 0; j <= 5; j++) {
8                 System.out.print(i+j + " ");
9             }
10            System.out.println();
11            // to end the line
12        }
13    }
14 }
```

Below the code, the Java console tab shows the output of the program:

```
<terminated> nestedFor2_2511533001 [Java Application] C:\Users\FA506N\
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
```

1. Perulangan luar (i)

for (int i = 0; i <= 5; i++)

- Loop ini mengatur jumlah baris.
- Nilai i dimulai dari 0 hingga 5, sehingga menghasilkan total 6 baris.

2. Perulangan dalam (j)

for (int j = 0; j <= 5; j++)

- Loop ini mengatur jumlah kolom pada setiap baris.
- Nilai j juga berjalan dari 0 sampai 5, sehingga setiap baris berisi 6 angka.

3. System.out.print(i + j + " ");

- Mencetak hasil penjumlahan i + j diikuti dengan spasi " ".
- Karena menggunakan print (bukan println), hasilnya akan dicetak sejajar dalam satu baris.

4. System.out.println();

- Setelah loop dalam selesai (satu baris penuh tercetak), perintah ini membuat baris baru agar hasil berikutnya ditulis di bawahnya.

BAB III

KESIMPULAN

3.1 Kesimpulan

Dari hasil praktikum dapat disimpulkan bahwa perulangan for merupakan struktur dasar dalam pemrograman Java yang digunakan untuk mengeksekusi perintah secara berulang sesuai jumlah iterasi tertentu. Melalui percobaan, diperoleh pemahaman bahwa perulangan for dapat dimanfaatkan untuk mencetak deret angka, menghitung nilai, serta membentuk pola sederhana maupun kompleks dengan nested for. Secara keseluruhan, penggunaan perulangan for membantu menyederhanakan kode, meningkatkan efisiensi, dan melatih kemampuan logika pemrograman.

3.2 Saran

Sebaiknya dosen menyelenggarakan sesi pra-praktikum di kelas sebagai pengantar materi, agar mahasiswa memperoleh pemahaman awal yang lebih baik. Dengan demikian, potensi kepanikan atau kesalahan selama praktikum dapat diminimalkan. Disarankan agar materi praktikum dibagikan terlebih dahulu melalui platform iLearn, sehingga mahasiswa memiliki waktu yang cukup untuk mempelajari dan mempersiapkan diri sebelum pelaksanaan praktikum.

DAFTAR PUSTAKA

[1] “The for Statement (The Java™ Tutorials > Learning the Java Language)”, Oracle, 2024.
Available:

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>