

# **LAPORAN PRATIKUM ALGORITMA PEMROGRAMAN**

## **PERULANGAN FOR**

Disusun Oleh:

Endy Pardilian 2511531017

Dosen Pengampu:

Wahyudi. Dr., S.T,M.T

Asisten Pratikum:

Aufan Taufiqurrahman



**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**PADANG**

**2025**

## **KATA PENGANTAR**

Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum algoritma pemrograman yang membahas mengenai Perulangan For dalam bahasa pemrograman java. Melalui laporan ini, penulis dapat memahami materi pratikum secara mendalam. Penulisan laporan ini juga dapat melatih ketelitian, keteraturan, serta kemampuan menulis sesuai kaidah akademik. Dengan demikian, laporan pratikum ini dapat berfungsi sebagai sarana belajar, dokumentasi kegiatan, dan referensi untuk praktikum atau pembelajaran jenjang berikutnya.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan, baik dari isi maupun penyajiannya. Oleh sebab itu, saran dan kritik sangat penulis harapkan untuk laporan berikutnya.

Padang, 2025

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>i</b>
<b>DAFTAR ISI .....</b>	<b>ii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan.....	1
1.3 Manfaat.....	1
<b>BAB II PEMBAHASAN .....</b>	<b>2</b>
2.1 Program Perulangan <i>for</i> 1 .....	2
2.2 Program Perulangan <i>for</i> 2 .....	3
2.3 Program Perulangan <i>for</i> 3 .....	4
2.4 Program Perulangan <i>for</i> 4 .....	6
2.5 Program Nested <i>for</i> 0 .....	7
2.6 Program Nested <i>for</i> 1 .....	9
2.7 Program Nested <i>for</i> 2 .....	10
<b>BAB III KESIMPULAN.....</b>	<b>12</b>
3.1 Kesimpulan.....	12
3.2 Saran.....	12
<b>DAFTAR PUSTAKA.....</b>	<b>13</b>

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perulangan adalah melakukan perintah yang ada di dalam blok perulangan tersebut secara berulang-ulang sesuai dengan nilai yang ditentukan atau sampai mencapai batas yang diinginkan. Dalam bahasa pemrograman JAVA terdapat beberapa statement perulangan yang dapat digunakan salah satunya yaitu “*for*”.

Perulangan dengan teknik ini dikontrol oleh tiga bagian yang ada dalam tanda kurung dan masing-masing bagian ini dipisahkan oleh titik-koma. Pada bagian pertama (inisialisasi ekspresi), sebuah variabel akan di deklarasikan sebagai sebuah titik awal dari perulangan, biasanya variable ini mempunyai tipe data *integer* atau *float*. Sementara pada bagian kedua perulangan diperiksa apakah masih memenuhi syarat atau tidak, jika memenuhi syarat maka statement dibawahnya akan di eksekusi. Sedangkan bagian ketiga adalah bagian dimana jika bagian kedua memenuhi syarat maka nilai variabel ditambahkan sesuai syarat yang dituliskan. Bagian ketiga, ini otomatis akan tidak dibaca oleh program jika kondisi pada bagian ke-dua sudah tidak lagi memenuhi syarat, dan perulangan pun menjadi terhenti.

### 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 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 {
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
13 }
14
```

Code 2.1

Nama file: PerulanganFor1

1. Inisialisasi for: `int i = 1;`

Variabel lokal `i` dibuat dan diberi nilai awal 1.

2. Pemeriksaan kondisi `i <= 10`

Jika benar, masuk ke badan loop.

Jika salah, keluar dari loop dan program selesai.

3. Badan loop: `System.out.println(i);`

Mencetak nilai `i` ke konsol dan menambahkan baris baru.

4. Update/Increment `i++`

Setelah badan loop selesai, `i` ditambah 1.

5. Ulangi langkah 5–7 sampai kondisi `i <= 10` menjadi false.

Karena dimulai `i = 1` dan diincrement 1 per iterasi, loop akan berjalan untuk `i = 1, 2, ..., 10` (total 10 iterasi). Setelah `i` menjadi 11, kondisi `i <= 10` salah → loop berhenti.

Output:

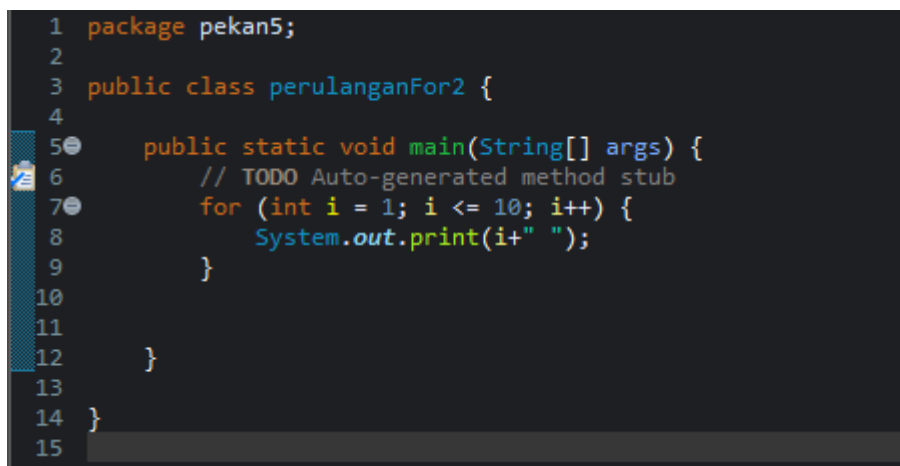


```
1
2
3
4
5
6
7
8
9
10
```

Gambar 2.1

Setiap baris pada output adalah hasil panggilan `System.out.println(i)`; pada setiap iterasi. Karena `println` menulis nilai lalu pindah baris, setiap angka muncul pada baris baru. Loop berjalan dari  $i = 1$  sampai  $i = 10$  (termasuk), sehingga yang dicetak adalah angka 1 sampai 10, satu per baris. Setelah  $i$  bertambah menjadi 11, kondisi  $i \leq 10$  gagal dan loop berhenti, tidak ada angka di atas 10 yang dicetak.

## 2.2 Program Perulangan *for* 2



```
1 package pekan5;
2
3 public class perulanganFor2 {
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     }
13
14 }
15
```

Code 2.2

Nama file: perulanagnFor2

### 1. `for (int i = 1; i <= 10; i++)`

`int i = 1` , variabel penghitung (counter) dimulai dari angka 1.

`i <= 10` , loop akan terus berjalan selama  $i$  kurang dari atau sama dengan 10.

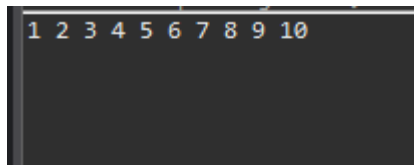
`i++` , setiap kali loop selesai dijalankan, nilai  $i$  bertambah 1.

2. Badan loop: `System.out.print(i + " ");`

Menampilkan nilai `i` ke layar. Menggunakan `print`, bukan `println`, sehingga tidak membuat baris baru setelah setiap angka. " " memberikan spasi di antara angka-angka agar output lebih rapi.

3. Loop berulang dari `i = 1` sampai `i = 10`, lalu berhenti setelah `i` bertambah menjadi 11 (karena kondisi `i <= 10` tidak terpenuhi).

Output:

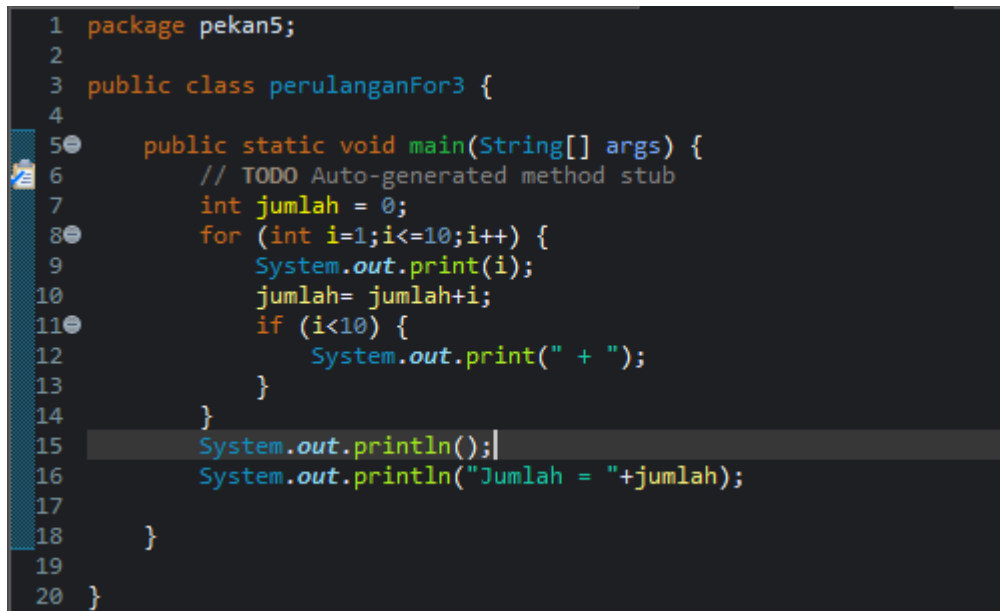


```
1 2 3 4 5 6 7 8 9 10
```

Gambar 2.2

Program menampilkan deret angka dari 1 sampai 10 dalam satu baris. Karena menggunakan `System.out.print()` (tanpa `ln`), semua angka dicetak dalam satu baris yang sama, dipisahkan oleh spasi. Setelah loop selesai, program berakhir tanpa baris baru di akhir.

### 2.3 Program Perulangan *for* 3



```
1 package pekan5;
2
3 public class perulanganFor3 {
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
20 }
```

Code 2.3

Nama file: perulanganFor3

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

`i` dimulai dari 1. Perulangan terus berjalan selama  $i \leq 10$ . Setelah setiap iterasi, `i` bertambah 1 (`i++`).

2. Di dalam loop:

`System.out.print(i);` (Menampilkan nilai `i` di layar.)

`jumlah = jumlah + i;` ( Menambahkan nilai `i` ke variabel `jumlah`.)

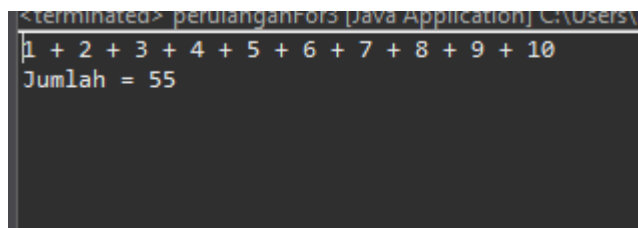
`if (i < 10),` (Menampilkan tanda " + " di antara angka, kecuali setelah angka terakhir (10) agar hasil rapi.)

3. Setelah loop selesai:

`System.out.println();` (membuat baris baru.)

`System.out.println("Jumlah = " + jumlah);` , (menampilkan total penjumlahan semua angka.)

Output:



```
<terminated> perulanganFor3 [Java Application] C:\Users\...
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 55
```

Gambar 2.3

Program mencetak semua angka dari 1 sampai 10 dalam satu baris, dipisahkan oleh tanda "+". Setelah selesai, program menghitung total semua angka tersebut:

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

Nilai total tersebut disimpan di variabel `jumlah`, lalu ditampilkan dengan perintah `"System.out.println("Jumlah = " + jumlah);"`. Hasil akhirnya, program menampilkan dua baris, yaitu inputan angka dan hasil penjumlahannya.



## 2.4 Program Perulangan *for* 4

```
1 package pekan5;
2
3 import java.util.Scanner;
4
5 public class PerulanganFor4 {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         int jumlah=0;
10        int batas;
11        Scanner input= new Scanner(System.in);
12        System.out.print("Masukan nilai batas = ");
13        batas= input.nextInt();
14        input.close();
15        for (int i=1;i<=batas;i++) {
16            System.out.print(i);
17            jumlah= jumlah+i;
18            if (i<batas) {
19                System.out.print(" + ");
20            } else {
21                System.out.print(" = ");
22            }
23        }
24        System.out.println(jumlah);
25    }
26
27 }
```

Code 2.4

Nama file: PerulanganFor4

### 1. Deklarasi variabel

int jumlah = 0; untuk menyimpan hasil penjumlahan. int batas; untuk menyimpan batas angka yang dimasukkan pengguna.

### 2. Input dari pengguna

Scanner input = new Scanner(System.in);. Membuat objek Scanner untuk membaca input dari keyboard. "System.out.print("Masukan nilai batas = ");", menampilkan pesan agar pengguna memasukkan angka. "batas = input.nextInt();" , membaca angka yang dimasukkan pengguna dan menyimpannya di variabel batas. "input.close();" , menutup objek Scanner (praktik yang baik untuk menghindari kebocoran resource).

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

Perulangan dimulai dari  $i = 1$  dan berakhir di  $i = \text{batas}$ . Setiap iterasi menampilkan nilai  $i$ , menambahkan  $i$  ke variabel jumlah.

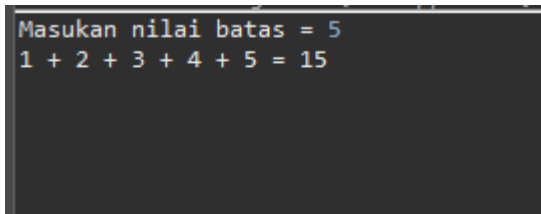
Jika  $i$  masih lebih kecil dari batas, tampilkan " + ".

Jika sudah mencapai batas, tampilkan " = ".

4. Menampilkan hasil akhir

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

Output:



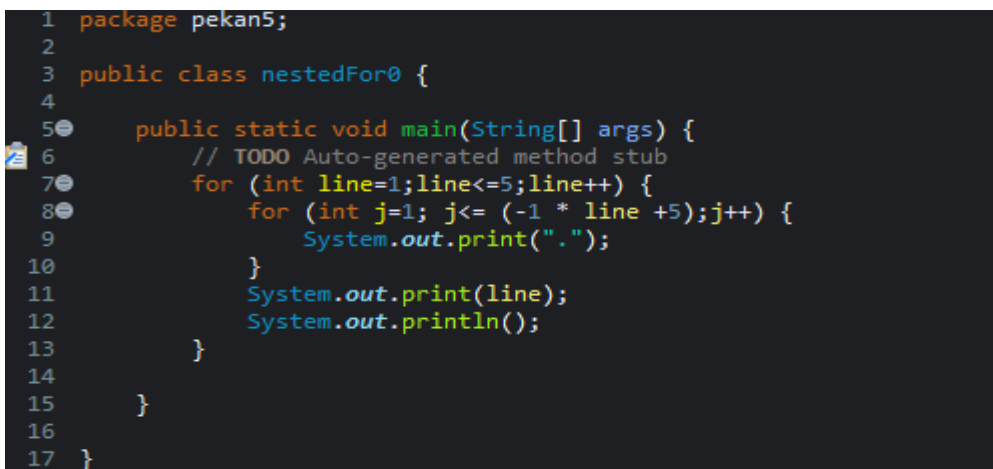
```
Masukan nilai batas = 5
1 + 2 + 3 + 4 + 5 = 15
```

Gambar 2.4

Program menampilkan deret angka dari 1 sampai dengan batas yang dimasukkan pengguna. Setiap angka dipisahkan oleh tanda " + ", kecuali angka terakhir yang diikuti tanda " = ". Nilai hasil penjumlahan (jumlah) dihitung dengan menambah semua angka dari 1 sampai batas. Misalnya jika batas = 5:

$1 + 2 + 3 + 4 + 5 = 15$

2.5 Program Nested *for* 0



```
1 package pekan5;
2
3 public class nestedFor0 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int line=1; line<=5; line++) {
8             for (int j=1; j<= (-1 * line +5); j++) {
9                 System.out.print(".");
10            }
11            System.out.print(line);
12            System.out.println();
13        }
14    }
15 }
16
17 }
```

Code 2.5

Nama file: nestedFor0

1. Perulangan pertama (luar):

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

Variabel line berfungsi sebagai penghitung baris. Nilainya mulai dari 1 hingga 5. Jadi loop luar akan berjalan 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 titik ".". Jumlah titik tergantung dari nilai line. Rumusnya:  $(-1 * \text{line} + 5)$  → ini menghasilkan jumlah titik yang berkurang setiap baris.

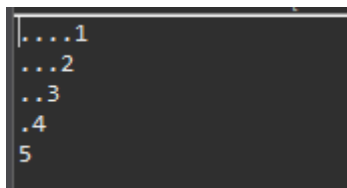
3. `System.out.print(line);`

Setelah mencetak titik, program mencetak angka baris (line) di akhir setiap baris.

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

Pindah ke baris baru setelah satu baris selesai dicetak.

Output:



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

Gambar 2.5

Program mencetak 5 baris karena line berjalan dari 1 sampai 5. Setiap baris terdiri dari beberapa titik "." yang berkurang setiap kali baris naik. Angka dari 1 sampai 5 yang muncul di posisi semakin ke kiri setiap baris berikutnya. Hasil akhirnya membentuk pola angka menurun ke kiri seperti segitiga miring.

## 2.6 Program Nested *for* 1

```
1 package pekan5;
2
3 public class nestedFor1 {
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        }
13    }
14 }
15
16 }
17 }
```

Code 2.6

Nama file: nestedFor1

### 1. Loop luar (i)

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

Mengatur jumlah baris. Berjalan sebanyak 5 kali (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, loop dalam mencetak 5 bintang (\*) pada satu baris.

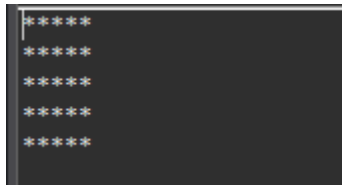
### 3. `System.out.print("*");`

Mencetak bintang tanpa pindah ke baris baru.

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

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

Output:



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

Gambar 2.6

## 2.7 Program Nested *for* 2



```
1 package pekan5;
2
3 public class nestedFor2 {
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        }
13    }
14
15 }
16
17 }
```

Code 2.7

Nama file: nestedFor2

### 1. Perulangan luar (i)

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

Loop ini mengatur baris. Nilai i dimulai dari 0 hingga 5, total 6 baris.

### 2. Perulangan dalam (j)

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

Loop ini mengatur kolom di setiap baris. Nilai j juga dari 0 sampai 5, setiap baris berisi 6 angka.

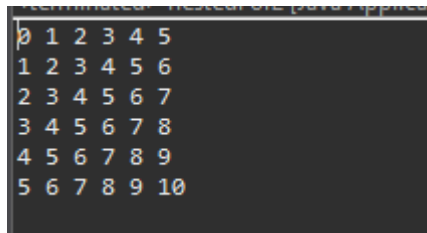
### 3. `System.out.print(i + j + " ");`

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

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

Setelah loop dalam selesai (satu baris penuh tercetak), baris baru dibuat agar hasil berikutnya ditulis di bawahnya.

Output:



```
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
```

Gambar 2.7

## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Dari hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa perulangan `for` merupakan salah satu struktur dasar dalam bahasa pemrograman Java yang berfungsi untuk mengeksekusi perintah secara berulang berdasarkan jumlah iterasi yang sudah ditentukan. Melalui percobaan yang dilakukan, diperoleh pemahaman bahwa perulangan `for` dapat digunakan untuk mencetak deret angka, menghitung jumlah nilai, serta membentuk berbagai pola seperti kotak dan tabel angka. Selain itu, dengan menggunakan `nested for` atau perulangan bersarang, program dapat menghasilkan pola yang lebih kompleks dan teratur. Secara keseluruhan, praktikum ini memberikan pemahaman bahwa penggunaan perulangan `for` dapat membantu menyederhanakan kode program, meningkatkan efisiensi, serta melatih kemampuan berpikir logis dalam pemrograman Java.

#### **3.2 Saran**

Sebagai saran, sebaiknya penjelasan materi saat praktikum bisa lebih detail dan perlahan, supaya mahasiswa yang belum terlalu paham coding bisa mengikuti dengan baik. Selain itu, dosen atau asisten praktikum sebaiknya memberikan kriteria pelaksanaan tugas yang lebih jelas, agar mahasiswa mengetahui apa saja yang harus dikerjakan dan bagaimana penilaiannya.

## 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>.