

**LAPORAN PRAKTIKUM
AGORITMA PEMROGRAMAN**

“PERULANGAN FOR”

DISUSUN OLEH:

SOFIAN ARBA’I

2511533029

DOSEN PENGAMPU:

Dr. WAHYUDI, S.T, M.T

ASISTEN PRAKTIKUM:

JOVANTRI IMMANUEL GULO



**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga Laporan Praktikum Pemrograman Java Pekan ke-5 ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai bentuk dokumentasi dan refleksi atas kegiatan praktikum yang telah dilaksanakan, khususnya mengenai materi perulangan for dan nested for.

Materi perulangan merupakan salah satu komponen penting dalam pemrograman, karena memungkinkan eksekusi kode secara berulang dengan kontrol yang terstruktur. Praktikum ini memberikan kesempatan kepada mahasiswa untuk memahami dan menerapkan konsep perulangan dalam berbagai kasus nyata, serta mengembangkan logika pemrograman yang sistematis dan efisien.

Penyusunan laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan berbagai pihak. Oleh karena itu, saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Dosen pengampu mata kuliah Praktikum Algoritma dan Pemrograman yang telah memberikan ilmu, arahan, serta bimbingannya.
2. Asisten praktikum yang telah membimbing dan memberikan penjelasan selama kegiatan praktikum berlangsung.

Saya menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang.

Padang, 28 Oktober 2025

Sofian Arba'i

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Manfaat	1
BAB II PEMBAHASAN	3
2.1 Perulangan For	3
2.2 Perulangan Nested For.....	3
2.3 Aplikasi Perulangan For Pada Java	4
2.3.1 Kode Program Perulangan For	5
2.3.2 Kode Program Perulangan For	6
2.3.3 Kode Program Perulangan For	7
2.3.4 Kode Program Perulangan For	8
2.4 Aplikasi Perulangan Nested For Pada Java	9
2.4.1 Kode Program Perulangan Nested For	9
2.4.2 Kode Program Perulangan Nested For	11
2.4.3 Kode Program Perulangan Nested For	12
BAB III PENUTUP	14
3.1 Kesimpulan	14
3.2 Saran	14
DAFTAR PUSTAKA.....	15

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, kemampuan untuk mengulang eksekusi suatu blok kode secara sistematis merupakan fondasi penting dalam membangun aplikasi yang efisien dan dinamis. Salah satu struktur kontrol yang paling umum digunakan untuk tujuan ini adalah perulangan for. Struktur ini memungkinkan programmer untuk menentukan jumlah iterasi secara eksplisit, sehingga sangat cocok digunakan dalam berbagai kasus seperti pengolahan data, pencetakan pola, dan manipulasi array.

Praktikum ini dirancang untuk memberikan pemahaman mendalam mengenai cara kerja perulangan for dan nested for, serta bagaimana mengimplementasikannya dalam bahasa pemrograman Java. Melalui latihan-latihan yang diberikan, mahasiswa diharapkan tidak hanya memahami sintaksis dan logika dasar dari perulangan, tetapi juga mampu mengembangkan solusi kreatif terhadap berbagai permasalahan pemrograman yang menuntut efisiensi dan ketelitian.

1.2 Tujuan

1. Memahami sintaks dan cara kerja perulangan for dalam Java.
2. Menerapkan konsep nested for untuk menyelesaikan masalah yang melibatkan struktur data bertingkat.
3. Melatih kemampuan logika dan algoritma melalui studi kasus yang melibatkan perulangan.
4. Mengembangkan keterampilan debugging dan analisis alur eksekusi program.

1.3 Manfaat

1. Meningkatkan kemampuan menulis kode yang efisien dan terstruktur menggunakan perulangan.

2. Meningkatkan kemampuan dalam menyelesaikan masalah yang membutuhkan iterasi kompleks, seperti pencetakan pola dan pengolahan array dua dimensi.
3. Menjadi bekal dasar untuk memahami struktur kontrol lanjutan dan algoritma yang lebih kompleks.
4. Meningkatkan pemahaman supaya bekal penting dalam pengembangan aplikasi yang membutuhkan pengolahan data secara berulang dan sistematis.

BAB II

PEMBAHASAN

2.1 Perulangan For

Pengulangan atau disebut sebagai looping adalah instruksi khusus dalam bahasa pemrograman dan algoritma yang digunakan untuk mengulang beberapa perintah sesuai dengan jumlah yang telah ditentukan. tujuannya adalah untuk mempermudah pengerjaan program dan untuk mempersingkat instruksi program. Dalam bahasa pemrograman Java, terdapat tiga bentuk pengulangan, yaitu For, While, dan Do-While.

Bentuk pengulangan yang sederhana, yaitu :

1. Pengulangan digunakan untuk mengerjakan suatu atau beberapa perintah secara berulang – ulang sesuai dengan yang diinginkan.
2. Pengulangan sederhana yaitu pengulangan yang hanya membutuhkan 1 kali pengulangan.
3. Di dalam pengulangan sederhana tidak ada pengulangan bersarang.

Pernyataan for digunakan untuk mengerjakan pernyataan atau sekelompok pernyataan secara berulang, dalam hitungan yang sudah pasti. Karakteristik dari pengulangan for yaitu :

1. Digunakan untuk pengulangan yang batasnya sudah diketahui dengan jelas, misalnya 5 kali.
2. Memerlukan dua buah penanda (nilai awal) dan akhir pengulangan (nilai akhir).
3. Nilai penghitung akan secara otomatis bertambah atau berkurang setiap kali sebuah pengulangan dilaksanakan, tergantung jenis pengulangannya.
4. Memerlukan suatu aksi yang dapat diulangi sebanyak pengulangannya.

Bentuk umumnya adalah sebagai berikut.

for (inisialisasi; kondisi; increment/decrement)

2.2 Perulangan Nested For

Selain perulangan `for` diatas, terdapat juga `nested for`, atau perulangan bersarang. *Nested* atau perulangan bersarang adalah sebuah konsep dimana satu perulangan berada di dalam perulangan lainnya.

Hal Ini berarti bahwa perulangan luar akan mengulang beberapa kali, dan di setiap iterasi dari perulangan luar, perulangan dalam juga akan diulang.

Perulangan ini sangat berguna saat kamu berhadapan dengan masalah yang membutuhkan lebih dari satu perulangan atau ketika kamu bekerja dengan struktur data yang lebih kompleks, seperti matriks (*array* dua dimensi) atau pola tertentu.

Nested biasanya digunakan dalam situasi dimana kita memerlukan lebih dari satu lapisan perulangan, Penggunaan `Nested Loop` :

1. Mencetak pola, Seperti mencetak bintang dalam bentuk segitiga atau kotak.
2. Mengakses array dua dimensi, Misalnya saat bekerja dengan matriks atau tabel.
3. Perulangan kompleks, Ketika perlu menyelesaikan masalah yang melibatkan dua atau lebih iterasi berulang.

2.3 Aplikasi Perulangan For Pada Java

Pada pertemuan praktikum pekan 5 mahasiswa diinstruksikan untuk menyalin kode sebagai latihan dan contoh dari perulangan `for` dan `nested for` itu sendiri. Berikut adalah kode program sebagai contoh dari kode program perulangan.

2.3.1 Kode Program Perulangan For

```

1 package pekan5;
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
12 }

```

Kode Program 2.1

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for pada kode program
3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Kode diatas diketik dalam bahasa Java dan berfungsi untuk menampilkan angka dari 1 sampai 10 menggunakan perulangan for. Penjelasan baris demi baris:

1. `package pekan5;` Menunjukkan bahwa program ini berada dalam paket bernama pekan5.
2. `public class PerulanganFor1;` Mendefinisikan kelas dengan nama PerulanganFor1. Semua kode program ditulis di dalam kelas ini.
3. `public static void main(String[] args);` Merupakan method utama (main method) yang akan dijalankan pertama kali saat program dieksekusi.
4. `for (int i = 1; i <= 10; i++);` Ini adalah perulangan **for**.
 - a. `int i = 1` → variabel penghitung (counter) i dimulai dari 1.

- b. $i \leq 10 \rightarrow$ perulangan akan terus berjalan selama i kurang dari atau sama dengan 10.
- c. $i++ \rightarrow$ setiap kali perulangan selesai, nilai i akan bertambah
- 5. `System.out.println(i)`, Menampilkan nilai i ke layar, dan berpindah ke baris berikutnya setelah mencetak.
- 6. Setelah i mencapai 10, perulangan berhenti dan program selesai dijalankan.

2.3.2 Kode Program Perulangan For

Pada perulangan for kedua menggunakan kode program yang hampir mirip dengan kode program perulangan for yang pertama.

```

1 package pekan5;
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
11 }
```

Kode Program 2.2

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Membuat perulangan for serta perintah yang dijalankan
3. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas juga ditulis dalam bahasa Java dan berfungsi untuk menampilkan angka dari 1 sampai 10, namun kali ini dalam satu baris. Program dimulai dengan mendeklarasikan paket `pekan5`, lalu mendefinisikan kelas bernama `PerulanganFor2`. Di dalam metode `main`, digunakan perulangan for dengan inisialisasi `int i = 1`, kondisi `i <= 10`, dan peningkatan `i++` pada setiap iterasi. Setiap kali perulangan berjalan, baris `System.out.print(i + " ");` akan mencetak nilai i diikuti satu spasi.

Perbedaan utama dengan program sebelumnya terletak pada penggunaan `print` alih-alih `println`. Pada program sebelumnya (PerulanganFor1), `println` digunakan sehingga setiap angka dicetak di baris baru. Sedangkan pada program ini, `print` membuat semua angka dicetak dalam satu baris dengan spasi di antaranya.

2.3.3 Kode Program Perulangan For

Penerapan kode program untuk perulangan for ketiga yaitu menampilkan deretan angka dari 1 sampai 10 beserta hasil penjumlahannya.

```

1 package pekan5;
2
3 public class PerulanganFor3 {
4
5     public static void main(String[] args) {
6         int jumlah=0;
7         for (int i=1; i<=10; i++) {
8             System.out.print(i);
9             jumlah= jumlah+i;
10            if (i<10) {
11                System.out.print(" + ");
12            }
13        }
14        System.out.println();
15        System.out.println("jumlah = "+jumlah);
16    }
17 }
18
19 }
```

Kode Program 2.3

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Buat perulangan dan perintah
3. Berikan juga logika atau percabangan `if`
4. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas merupakan program Java yang menampilkan deretan angka dari 1 sampai 10 beserta hasil penjumlahannya. Program

diawali dengan mendefinisikan paket `pekan5` dan kelas bernama `PerulanganFor3`. Di dalam metode `main`, terdapat variabel `jumlah` yang diinisialisasi dengan nilai 0 untuk menyimpan hasil penjumlahan. Kemudian digunakan perulangan `for` yang berjalan dari `i = 1` hingga `i = 10`. Pada setiap iterasi, program mencetak nilai `i` dan menambahkan nilai tersebut ke dalam variabel `jumlah`. Jika nilai `i` belum mencapai 10, maka program akan menampilkan tanda “+” agar hasil cetakan terlihat seperti operasi penjumlahan. Setelah perulangan selesai, program menampilkan hasil akhir penjumlahan dari semua angka, yaitu 55. Dengan demikian, output program ini menampilkan deretan “1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10” diikuti oleh hasil akhirnya “jumlah = 55”. Program ini berbeda dari dua program sebelumnya karena tidak hanya mencetak angka, tetapi juga menghitung dan menampilkan total hasil penjumlahannya.

2.3.4 Kode Program Perulangan For

Pada kode program perulangan `for` terakhir ini yaitu kode program yang menampilkan deretan angka dari 1 sampai 10 serta menghitung total penjumlahannya.

```

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

Kode Program 2.4

Kode Program 2.5

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Buat scanner input supaya bisa data bisa diinputkan sesuai keinginan pengguna
3. Gunakan perulangan for dari $i = 1$ sampai $i = 10$.
4. Tambahkan i ke variabel jumlah dan tampilkan angka.
5. Gunakan if untuk menampilkan tanda “+” atau “=”.

Program PerulanganFor4 berfungsi menampilkan angka dari 1 sampai 10, menambahkan semuanya, dan menampilkan hasilnya. Program ini juga meminta pengguna memasukkan nilai batas, yaitu angka terakhir sebelum tanda “+” berubah menjadi “=”. Program menggunakan Scanner untuk membaca input, lalu menjalankan perulangan for dari 1 hingga 10. Jika nilai i masih kurang dari batas, program mencetak tanda “+”, sedangkan jika sudah mencapai batas, program mencetak tanda “=”. Setelah perulangan selesai, hasil total penjumlahan ditampilkan.

2.4 Aplikasi Perulangan Nested For Pada Java

Aplikasi perulangan nested for pada java dilakukan dengan 3 kode program dengan fungsi dan penulisan yang berbeda. Berikut adalah aplikasi perulangan nested for.

2.4.1 Kode Program Perulangan Nested For

Pada kode program perulangan nested for yang pertama yaitu menampilkan sebuah pola yaitu menurun dimana baris pertama memiliki banyak titik, sedangkan baris terakhir tidak terdapat titik.

```

1 package pekan5;
2
3 public class nestedFor0 {
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            }
11            System.out.print(line);
12            System.out.println();
13        }
14    }
15 }
16
17 }

```

Kode Program 2.5

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Buat perulangan for pertama dengan variabel line dari 1 hingga 5 dan perintah
3. Buat perulangan for kedua dengan variabel j di mana kondisi batasnya ditentukan oleh rumus $(-1 * \text{line} + 5)$ agar jumlah titik berkurang setiap baris.
4. Gunakan `System.out.print(".")` di dalam perulangan dalam untuk mencetak titik tanpa berpindah baris.
5. Setelah perulangan dalam selesai, **cetak nilai line** menggunakan `System.out.print(line)` untuk menampilkan nomor baris.
6. Gunakan `System.out.println()` agar program berpindah ke baris baru sebelum mencetak baris berikutnya.
7. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas merupakan contoh penggunaan nested loop atau perulangan bersarang dalam bahasa Java. Program ini menggunakan dua perulangan for, di mana perulangan luar (line) mengatur jumlah baris yang akan dicetak, sedangkan perulangan dalam (j) mengatur

jumlah titik (.) yang muncul pada setiap baris. Variabel *line* dimulai dari 1 hingga 5, sehingga program akan mencetak sebanyak 5 baris. Pada setiap baris, jumlah titik ditentukan oleh rumus $(-1 * \text{line} + 5)$, sehingga semakin besar nilai *line*, semakin sedikit jumlah titik yang dicetak. Setelah mencetak titik-titik, program mencetak nilai *line* di akhir setiap baris, lalu berpindah ke baris berikutnya menggunakan `System.out.println()`. Hasil akhirnya membentuk pola menurun, di mana baris pertama memiliki lebih banyak titik, dan baris terakhir hanya menampilkan angka tanpa titik di depannya.

2.4.2 Kode Program Perulangan Nested For

Kode program perulangan nested for yaitu sebuah kode program yang berfungsi untuk mencetak pola bintang berbentuk persegi berukuran 5x5. program ini menunjukkan cara menggunakan dua perulangan bersarang untuk membuat pola dua dimensi sederhana. Berikut kode programnya:

```

1 package pekan5;
2
3 public class nestedFor1 {
4
5     public static void main(String[] args) {
6         for (int i=1; i<=5; i++) {
7             for (int j=1; j<=5; j++) {
8                 System.out.print("*");
9             }
10            System.out.println();
11            // to end the line
12        }
13    }
14
15 }
```

Kode Program 2.6

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Buat perulangan for pertama dengan variabel *line* dari 1 hingga 5 untuk menentukan jumlah baris

3. Buat perulangan dalam (for) dengan variabel j dari 1 sampai 5 untuk menentukan jumlah bintang di setiap baris. Gunakan `System.out.print(".")` di dalam perulangan dalam untuk mencetak titik tanpa berpindah baris.
4. Di dalam perulangan dalam, gunakan `System.out.print("*");` untuk mencetak bintang tanpa berpindah baris.
5. Setelah perulangan dalam selesai, gunakan `System.out.println();` agar kursor pindah ke baris berikutnya.
6. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

2.4.3 Kode Program Perulangan Nested For

Kode program perulangan nested for yang terakhir yaitu perulangan yang digunakan untuk menampilkan pola angka hasil penjumlahan dari dua variabel perulangan, yaitu i dan j. Berikut adalah kode program:

```

1 package pekan5;
2
3 public class nestedFor2 {
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            }
11            System.out.println();
12            // to end the line
13        }
14    }
15 }
16
17 }
```

Kode Program 2.7

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Buat perulangan luar (for) dengan variabel i dari 0 sampai 5 untuk mengatur jumlah baris.

3. Buat perulangan dalam (for) dengan variabel j dari 0 sampai 5 untuk mengatur jumlah kolom di setiap baris.
4. Di dalam perulangan dalam, tulis perintah `System.out.print(i + j + " ");` untuk mencetak hasil penjumlahan i + j diikuti spasi.
5. Setelah perulangan dalam selesai, gunakan `System.out.println();` agar kursor pindah ke baris baru.
6. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

Program di atas merupakan contoh nested loop (perulangan bersarang) dalam bahasa Java yang digunakan untuk menampilkan pola angka hasil penjumlahan dari dua variabel perulangan, yaitu i dan j. Program diawali dengan deklarasi paket `pekan5` dan kelas bernama `nestedFor2`. Di dalam metode `main`, terdapat dua perulangan `for`. Perulangan luar dengan variabel i berjalan dari 0 hingga 5, sedangkan perulangan dalam dengan variabel j juga berjalan dari 0 hingga 5. Pada setiap iterasi, program mencetak hasil penjumlahan antara i dan j menggunakan perintah `System.out.print(i + j + " ");`, sehingga setiap baris berisi hasil penjumlahan yang berubah sesuai nilai i. Setelah satu baris selesai, `System.out.println();` digunakan untuk berpindah ke baris berikutnya. Hasil akhirnya adalah pola angka berbentuk tabel di mana setiap baris menunjukkan penjumlahan antara indeks baris (i) dan kolom (j). Contohnya, baris pertama berisi 0 1 2 3 4 5, baris kedua 1 2 3 4 5 6, dan seterusnya hingga baris terakhir 5 6 7 8 9 10.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum pada pekan 5 ini yang membahas tentang *perulangan for* dan *nested for* dalam bahasa Java, dapat disimpulkan bahwa struktur perulangan merupakan komponen penting dalam pemrograman untuk menjalankan serangkaian perintah secara berulang dengan kontrol yang teratur. Melalui latihan yang dilakukan, mahasiswa memahami bagaimana perulangan *for* digunakan untuk mengulang proses dengan batas yang jelas, serta bagaimana *nested for* dapat dimanfaatkan untuk membentuk pola dua dimensi atau memproses data bertingkat seperti matriks. Praktikum ini juga membantu memperkuat pemahaman mengenai logika iterasi, alur eksekusi program, serta efisiensi dalam menulis kode yang terstruktur dan mudah dipahami.

3.2 Saran

Supaya pembelajaran mengenai perulangan semakin efektif, disarankan untuk lebih banyak melakukan eksperimen dengan berbagai variasi pola dan kondisi pada perulangan, seperti penggunaan kombinasi *for* dengan *if*, atau penerapan *nested for* dalam kasus nyata. Selain itu, mahasiswa perlu membiasakan diri menuliskan kode secara sistematis dan memahami setiap bagian dari alur logika yang dibuat agar dapat meminimalisir kesalahan saat debugging. Dosen dan asisten praktikum juga diharapkan memberikan contoh kasus yang lebih kompleks dan kontekstual, sehingga mahasiswa dapat mengasah kemampuan berpikir algoritmik dan siap menghadapi permasalahan pemrograman yang lebih menantang.

DAFTAR PUSTAKA

[1] Evi Pratiwi, Konsep Dasar Algoritma dan Pemrograman dengan Bahasa Java, , 2020.

[2] ITBOX, “Belajar Java Looping: Menguasai Jenis Perulangan dalam Java,”.

[Daring]. Tersedia pada: [Belajar Java Looping: Menguasai Jenis Perulangan dalam Java - ITBOX by Course-Net](#) [Diakses: 27-Okt-2025].