

LAPORAN PRAKTIKUM

PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

“PENGUNAAN PERULANGAN *FOR* DAN *NESTED FOR* DALAM
BAHASA PEMROGRAMAN JAVA”

disusun oleh

Hafizh Habibullah

2511531002

Dosen Pengampu: Dr. Wahyudi, S.T., M.T.

Asisten Praktikum: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

TAHUN 2025

KATA PENGANTAR

Puji syukur kita panjatkan ke hadirat Allah SWT. karena atas rahmat dan karunia-Nya laporan praktikum Algoritma dan Pemrograman ini dapat diselesaikan dengan baik.

Laporan ini disusun untuk memenuhi tugas praktikum pada mata kuliah Algoritma dan Pemrograman dengan topik *Penggunaan Perulangan **For** dan **NestedFor** dalam bahasa pemrograman Java*. Melalui laporan ini, penulis berusaha menjelaskan hasil praktikum yang meliputi kode program, penjelasan tentang cara kerja program yang dibuat, serta hasil output dari program.

Penulis sangat mengharapkan kritik dan saran yang membangun agar dapat menjadi bahan perbaikan di kemudian hari. Semoga laporan ini dapat memberikan manfaat bagi penulis maupun pembaca.

Padang, 31 Oktober 2025

Hafizh Habibullah

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR LAMPIRAN	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Manfaat.....	1
BAB 2 PEMBAHASAN.....	2
2.1 Perulangan For.....	2
2.2 Nested For	6
BAB III KESIMPULAN.....	9
3.1 Kesimpulan.....	9
3.2 Saran Pengembangan.....	9
DAFTAR PUSTAKA	10

DAFTAR LAMPIRAN

Gambar 2. 1 - Kode Program PerulanganFor1.....	2
Gambar 2. 2 - Kode Program PerulanganFor2.....	3
Gambar 2. 3 - Kode Program PerulanganFor3.....	4
Gambar 2. 4 - Kode Program PerulanganFor4.....	5
Gambar 2. 5 - Kode Program NestedFor0	6
Gambar 2. 6 - Kode Program NestedFor1	7
Gambar 2. 7 - Kode Program NestedFor2	8
Tabel 2. 1 - Hasil Output dari PerulanganFor1	3
Tabel 2. 2 - Hasil Output dari PerulanganFor2	4
Tabel 2. 3 - Hasil Output dari PerulanganFor3	4
Tabel 2. 4 - Hasil Output dari PerulanganFor4	5
Tabel 2. 5 - Hasil Output dari NestedFor0	7
Tabel 2. 6 - Hasil Output dari NestedFor1	7
Tabel 2. 7 - Hasil Output dari NestedFor2	8

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman, *looping* adalah salah satu konsep fundamental yang sangat krusial karena memungkinkan suatu proses dilakukan berulang-ulang tanpa perlu menulis kode yang identik berulang-ulang. Salah satu tipe perulangan yang paling umum digunakan adalah *for loop*. Struktur ini memudahkan pengaturan jumlah iterasi secara jelas melalui inisialisasi, kondisi, dan peningkatan nilai variabel secara rutin.

Selain *for loop* tunggal, ada juga konsep *nested for* atau perulangan bersarang, di mana satu perulangan terletak di dalam perulangan yang lain. Konsep ini sering diterapkan untuk menciptakan pola tertentu, memproses data hierarkis, atau menyajikan struktur berulang seperti tabel, matriks, dan bentuk visual sederhana menggunakan karakter.

Dalam praktikum ini, dipelajari cara kerja **for loop** dan **nested for**, serta penerapannya untuk menciptakan program yang terstruktur dan efisien.

1.2 Tujuan

Adapun tujuan dari praktikum ini adalah:

1. Memahami konsep dasar dan struktur sintaks perulangan *for*.
2. Menerapkan *nested for* untuk membentuk pola tertentu.
3. Melatih logika berpikir dalam menyusun program yang menggunakan perulangan bersarang.

1.3 Manfaat

Melalui praktikum ini, diharapkan dapat membantu dalam memahami logika perulangan secara mendalam, mengembangkan kemampuan berpikir sistematis, serta mampu membuat program yang efisien dengan menggunakan *for loop* dan *nested for*. Pemahaman ini juga menjadi dasar penting dalam pengembangan algoritma yang lebih kompleks di tahap selanjutnya. Pembiasaan menulis kode dengan struktur yang rapi dan sesuai dengan aturan bahasa pemrograman.

BAB 2

PEMBAHASAN

Pada praktikum pekan ini, perhatian utamanya adalah pada penerapan struktur perulangan *for* dan *nested for* dalam pemrograman. Melalui pembahasan ini, dipelajari cara penggunaan perulangan untuk mengatasi berbagai masalah logika serta membentuk pola-pola tertentu secara efisien.

Selama praktikum, dibuat beberapa aplikasi sederhana yang menerapkan konsep *looping*, seperti menampilkan deret angka, mencetak pola segitiga dan sebagainya dengan *nested loop*. Setiap program dibuat untuk memperdalam pemahaman mengenai cara *for loop* berfungsi, termasuk perubahan variabel iterasi dan interaksi antara perulangan bersarang dalam menghasilkan output yang diinginkan.

Bagian ini juga menguraikan teori fundamental mengenai *looping*, struktur sintaks *for*, serta cara kerja *nested for* yang menjadi landasan dari program yang dikerjakan.

2.1 Perulangan For

Perulangan *for* adalah salah satu struktur kontrol yang dipakai untuk menjalankan sekumpulan perintah secara berulang dengan jumlah iterasi yang telah ditentukan. Struktur ini terdiri dari tiga komponen utama, yaitu pengaturan variabel, syarat perulangan, dan modifikasi nilai variabel (umumnya penambahan atau pengurangan). Dalam praktikum ini, dibuat empat program sederhana yang memanfaatkan konsep perulangan *for*.

```
1 package pekan5;
2
3 public class PerulanganFor1 {
4     public static void main(String[] args) {
5         for (int i = 1; i <= 10; i++) {
6             System.out.println(i);
7         }
8     }
9 }
```

Gambar 2. 1 - Kode Program PerulanganFor1

Pada Kode Program pertama, dibuat program sederhana yang menggunakan konsep perulangan *for*, yang dimana Kode Program ini dibuat untuk menampilkan output urutan angka 1 sampai 10 secara berurutan. Pada Program ini, Digunakan variabel counter **i** yang dimulai dengan nilai **1**. Kondisi pengulangan ditentukan agar **i** terus meningkat selama nilainya **kurang dari atau sama dengan 10**, dengan setiap pengulangan menambahkan **i** sebanyak **1 unit** melalui ekspresi **i++**. Dalam blok perulangan, terdapat instruksi **System.out.println(i);** yang menampilkan nilai **i** di setiap tahap perulangan.

Hasil Output
1
2
3
4
5
6
7
8
9
10

Tabel 2. 1 - Hasil Output dari PerulanganFor1

Selanjutnya digunakan Kode Program yang sama, namun dengan sedikit perubahan. Pada Kode Program *PerulanganFor2*, berbeda dari output sebelumnya yang menampilkan urutan angka kebawah karena perintah **System.out.println(i);**, output kali ini akan menampilkan urutan angka menyamping dengan menggunakan **System.out.print(i+" ");**.

```

1 package pekan5;
2
3 public class PerulanganFor2 {
4     public static void main(String[] args) {
5         for (int i = 1; i <= 10; i++) {
6             System.out.print(i+" ");
7         }
8     }
9 }

```

Gambar 2. 2 - Kode Program PerulanganFor2

Saat dijalankan, Kode Program diatas akan menampilkan Output sebagai berikut:

Hasil Output										
1	2	3	4	5	6	7	8	9	10	

Tabel 2. 2 - Hasil Output dari PerulanganFor2

Pada *PerulanganFor3*, ditambahkan perintah yang akan menjumlahkan urutan angka 1 sampai 10. Perintah *if* juga digunakan supaya Program menampilkan output “ + “ hanya jika *i* lebih kecil dari **10**. Setelah itu, semua angka dijumlahkan menggunakan perintah **jumlah= jumlah+i**.

```

1 package pekan5;
2
3 public class PerulanganFor3 {
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         System.out.println();
14         System.out.println("Jumlah = "+jumlah);
15     }
16 }

```

Gambar 2. 3 - Kode Program PerulanganFor3

Akhirnya, digunakan perintah **System.out.println("Jumlah = "+jumlah);** untuk menampilkan jumlah dari angka-angka yang ditampilkan sebelumnya. Hasil Output dari Kode Program diatas adalah sebagai berikut:

Hasil Output
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 55

Tabel 2. 3 - Hasil Output dari PerulanganFor3

Pada *PerulanganFor4*, ditambahkan perintah yang meminta pengguna untuk memasukkan nilai batas. Untuk menerima input dari pengguna, digunakan **java.util.Scanner** untuk menerima input dari pengguna yang nantinya digunakan saat menjalankan program yang sama dengan program sebelumnya.

```

1 package pekan5;
2
3 import java.util.Scanner;
4
5 public class PerulanganFor4 {
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        for (int i=1;i<=batas;i++) {
14            System.out.print(i);
15            jumlah= jumlah+i;
16            if (i<batas) {
17                System.out.print(" + ");
18            } else {
19                System.out.print(" = ");
20            }
21        }
22        System.out.println(jumlah);
23    }
24 }

```

Gambar 2. 4 - Kode Program PerulanganFor4

Sebagai contoh, Pengguna akan memasukkan angka 15. Program pun akan menjalankan perintah-perintah yang sama dengan *PerulanganFor3*, bedanya kali ini batas angka tidak ditentukan dari Program-nya, melainkan dari input pengguna.

Hasil Output
Masukkan nilai batas = 15
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 = 120

Tabel 2. 4 - Hasil Output dari PerulanganFor4

2.2 Nested For

Selanjutnya pada praktikum ini dilakukan percobaan yang membahas tentang **perulangan bersarang (nested for)**. Konsep ini merupakan pengembangan dari yang sebelumnya, yaitu perulangan tunggal. Dalam *nested for*, dalam satu perulangan terdapat perulangan lain yang berjalan secara terstruktur.

Tujuan dari percobaan ini adalah untuk memahami bagaimana **loop dalam loop** bekerja, serta bagaimana urutan eksekusi program dapat membentuk pola tertentu, seperti tabel angka, segitiga bintang, atau bentuk visual lainnya di konsol. Pada bagian ini, dibuat 3 Kode Program sebagai percobaan dalam menerapkan konsep *nested loop*.

```
1 package pekan5;
2
3 public class NestedFor0 {
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 }
```

Gambar 2. 5 - Kode Program NestedFor0

Pada program ini, digunakan konsep *nested for*. Program diatas digunakan untuk mencetak pola kombinasi titik dan angka. Perulangan luar (**line**) digunakan untuk mengatur jumlah baris 1 sampai 5, sedangkan Perulangan dalam (**j**) digunakan untuk mencetak titik (.) dengan jumlah yang semakin berkurang tiap barisnya karena adanya perintah (**-1 * line + 5**) yang mengurangi jumlah titik seiring bertambahnya nilai **line**. Hasil outputnya akan menjadi seperti berikut:

Hasil Output
<pre>12 ...3 ..4 .5 5 </pre>

Tabel 2. 5 - Hasil Output dari NestedFor0

Selanjutnya, Program *NestedFor1* dibuat untuk mencetak pola berbentuk kotak bintang (*) berukuran 5x5. *loop* luar (*i*) mengatur jumlah barisnya, sedangkan *loop* dalam (*j*) mengatur jumlah bintang yang dicetak pada tiap baris. Pada perulangan *j*, program mencetak satu karakter bintang (*) tanpa berpindah baris. Setelah satu baris selesai dicetak, perintah **System.out.println();** akan dijalankan untuk berpindah ke baris berikutnya.

```

1 package pekan5;
2
3 public class NestedFor1 {
4     public static void main(String[] args) {
5         for (int i = 1; i <= 5; i++) {
6             for (int j = 1; j <= 5; j++) {
7                 System.out.print("*");
8             }
9             System.out.println();
10        }
11    }
12 }

```

Gambar 2. 6 - Kode Program NestedFor1

Ketika dijalankan, Program akan mencetak Output seperti berikut:

Hasil Output
<pre> ***** ***** ***** ***** ***** </pre>

Tabel 2. 6 - Hasil Output dari NestedFor1

Terakhir, dibuat Program *NestedFor2* yang merupakan Program untuk mencetak tabel angka. Program menjumlahkan dua variabel perulangan, yaitu **i** dan **j**. Perulangan luar (**i**) menentukan baris 0 sampai 5, sedangkan perulangan dalam (**j**) menentukan kolom pada setiap baris. Setiap kali loop dijalankan, program akan mencetak hasil penjumlahan (**i + j**) yang diikuti spasi. Setelah satu baris selesai dicetak, dijalankan perintah **System.out.println();** yang akan memindahkan program ke baris baru.

```

1 package pekan5;
2
3 public class NestedFor2 {
4     public static void main(String[] args) {
5         for (int i = 0; i <= 5; i++) {
6             for (int j = 0; j <= 5; j++) {
7                 System.out.print(i+j+ " ");
8             }
9             System.out.println();
10        }
11    }
12 }

```

Gambar 2. 7 - Kode Program NestedFor2

Hasil Output dari Program diatas akan menampilkan tabel angka yang dimana setiap elemen merepresentasikan nilai penjumlahan dari indeks baris dan kolomnya. Program akan mencetak Output sebagai berikut:

Hasil 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					

Tabel 2. 7 - Hasil Output dari NestedFor2

BAB III

KESIMPULAN

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilaksanakan, dapat disimpulkan bahwa struktur perulangan **for** dan **nested for** memiliki peran yang sangat penting dalam mengontrol proses pengulangan dalam suatu program. Melalui praktikum ini juga dipahami bahwa perulangan bersarang (**nested loop**) bekerja dengan prinsip “loop dalam loop”, dimana setiap pengulangan loop luar akan mengeksekusi seluruh pengulangan loop dalam terlebih dahulu.

3.2 Saran Pengembangan

Untuk pengembangan selanjutnya, Praktikum dapat diperluas dengan menerapkan perulangan pada kasus yang lebih dinamis, seperti mencetak pola sesuai keinginan pengguna, menggabungkan lebih banyak logika percabangan, atau menambah **validasi input** agar program lebih aman digunakan. Selain itu, dapat pula dieksplorasi penerapan **nested loop lebih dari dua tingkat** untuk memahami dampaknya terhadap kompleksitas program.

DAFTAR PUSTAKA

- [1] J. Schildt, *Java: The Complete Reference*, 12th ed. New York: McGraw-Hill, 2021.
- [2] H. M. Deitel and P. J. Deitel, *Java: How to Program*, 11th ed. Boston: Pearson Education, 2019.
- [3] Oracle, “The for Statement (The Java™ Tutorials),” *Oracle Documentation*, 2024. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>
- [4] Oracle, “Nested Loops in Java,” *Oracle Documentation*, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>
- [5] E. Kurniawan, *Belajar Pemrograman Java dari Nol*, Bandung: Informatika, 2020.
- [6] R. A. Nugroho, “Struktur Perulangan dalam Java,” *Jurnal Teknologi Informasi dan Komputer (JTIK)*, vol. 5, no. 2, pp. 45–52, 2022.