

**LAPORAN PRATIKUM
PEMROGRAMAN ALGORITMA PEMROGRAMAN**

**LAPORAN PRAKTIKUM PEMROGRAMAN DASAR JAVA: PENERAPAN
PERULANGAN FOR DAN NESTED FOR**

disusun oleh:

Khairunnisa M.

NIM 2511532005

Dosen Pengampu:

DR. Wahyudi, S. T., M. T.

Asisten Pratikum:

Aufan Taufiqurrahman



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis penjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum ini dengan baik. Laporan ini disusun untuk memenuhi salah satu tugas pada mata kuliah Pemrograman Dasar Java dengan topik pembahasan Perulangan For dan Nested For. Melalui praktikum ini, penulis belajar memahami bagaimana struktur perulangan bekerja dalam Bahasa pemrograman Java, baik perulangan tunggal maupun perulangan bersarang. Selain itu, praktikum ini juga membantu penulis dalam melatih logika berpikir secara terstruktur, efisien, dan sistematis dalam menyelesaikan permasalahan pemrograman.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asistensi praktikum, serta rekan-rekan yang telah membantu dalam proses pelaksanaan praktikum ini. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi untuk memahami konsep perulangan dalam Bahasa Java.

Padang, 29 September 2025

Khairunnisa M.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	2
BAB II.....	3
PEMBAHASAN	3
2.1. Perulangan For 1	3
2.2. Perulangan For 2	4
2.3. Perulangan For 3	5
2.4. Perulangan For 4	7
2.5. Nested For 0	8
2.6. Nested For 1	9
2.7. Nested For 2	10
BAB III	11
KESIMPULAN	11
DAFTAR PUSTAKA	12

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, salah satu konsep dasar yang sangat penting untuk dikuasai adalah struktur perulangan (*looping*). Perulangan memungkinkan program untuk menjalankan suatu blok kode secara berulang tanpa harus menuliskannya berkali-kali. Hal ini membuat kode menjadi lebih efisien, mudah dibaca, dan menghemat waktu dalam proses pengembangan program.

Pada Bahasa pemrograman Java, terdapat beberapa jenis perulangan, seperti *for*, *while*, dan *do-while*. Di antara ketiganya, perulangan *for* merupakan struktur yang paling sering digunakan karena memberikan kontrol penuh terhadap inisialisasi, kondisi, dan perubahan nilai variabel dalam satu baris sintaks, selain itu, terdapat pula konsep perulangan bersarang (*nested for*) yang memungkinkan satu perulangan berada di dalam perulangan lainnya untuk menghasilkan pola atau proses berulang yang lebih kompleks.

Melalui praktikum ini, mahasiswa diharapkan dapat memahami dan mengimplementasikan konsep perulangan *for* serta *nested for* dalam berbagai contoh program. Pemahaman terhadap konsep ini menjadi dasar penting untuk membangun logika pemrograman yang kuat dan efisien, terutama dalam menyelesaikan permasalahan yang melibatkan proses berulang seperti perhitungan matematis, pengolahan data, maupun pembuatan pola visual di konsol.

1.2 Tujuan Praktikum

1. Memahami konsep dasar perulangan (*looping*) dalam Bahasa Java, khususnya struktur *for*.
2. Mampu mengimplementasi perulangan *for* untuk menampilkan data secara berurutan dan melakukan operasi perhitungan berulang.
3. Mengenal dan menerapkan konsep perulangan bersarang (*nested for*) untuk membentuk pola atau menampilkan hasil dengan struktur yang lebih kompleks.

4. Melatih kemampuan logika dan berpikir algoritmik, terutama dalam merancang solusi yang efisien menggunakan struktur control perulangan.
5. Meningkatkan keterampilan pemrograman dasar sebagai fondasi untuk memahami konsep pemrograman lanjutan di masa depan.

1.3 Manfaat Praktikum

1. Memperdalam pemahaman terhadap struktur control perulangan yang menjadi dasar dalam setiap bahasa pemrograman.
2. Meningkatkan kemampuan logika pemrograman, khususnya dalam menyusun algoritma yang melibatkan proses berulang dan bersarang.
3. Melatih ketelitian dan pemahaman alur eksekusi program, sehingga mahasiswa dapat menelusuri dan memperbaiki kesalahan logika dengan lebih mudah.
4. Meningkatkan keterampilan problem solving, yaitu kemampuan Menyusun solusi efisien melalui pendekatan algoritmik.
5. Menjadi bekal dalam mempelajari konsep pemrograman lanjutan

BAB II

PEMBAHASAN

2.1. Perulangan For 1

Percobaan pertama dilakukan untuk memahami cara kerja dasar struktur perulangan *for* pada bahasa pemrograman Java. Pada tahap ini, program dibuat untuk menampilkan deretan angka dari 1 sampai 10 ke layar. Tujuan dari percobaan ini adalah agar mahasiswa dapat memahami bagaimana mekanisme inisialisasi, kondisi, dan increment bekerja di dalam satu struktur 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 }
10
11
```

Gambar 2.1. Program Java dengan Perulangan For 1.

Program ini menggunakan satu perulangan *for* untuk mencetak angka dari 1 hingga 10 secara berurutan. Struktur perulangan *for* terdiri dari tiga bagian utama:

1. Inisialisasi (`int i = 1`) → nilai awal variabel penghitung `i`.
2. Kondisi (`i <= 10`) → menentukan kapan perulangan berhenti.
3. Increment (`i++`) → menambah nilai `i` setiap kali perulangan dijalankan.

Langkah kerja:

1. Membuat package baru dengan nama `Pekan5`.
2. Membuat file program Java baru bernama `PerulanganFor1.java`
3. Menuliskan kode program untuk menampilkan angka 1 sampai 10 menggunakan perulangan *for*.

4. Menjalakan program dan mengamati hasil output di layar.
5. Menyimpan hasil percobaan sebagai bahan pembahasan laporan.

Program ini mencetak angka dari 1 sampai 10, masing-masing pada baris baru. Nilai *i* terus bertambah satu setiap kali perulangan dijalankan hingga mencapai nilai maksimum, yaitu 10. Percobaan ini menunjukkan dasar dari konsep perulangan dengan jumlah langkah yang sudah diketahui, atau yang disebut *count-controlled loop*. Struktur *for* seperti ini sangat berguna untuk tugas-tugas yang melibatkan pengulangan tetap, seperti mencetak deret angka, menghitung total, atau mengakses elemen dalam array.

2.2. Perulangan For 2

Percobaan kedua dilakukan untuk memahami perbedaan antara penggunaan *System.out.println* dan *System.out.print* dalam struktur perulangan *for*. Jika pada percobaan sebelumnya setiap angka dicetak di baris baru, maka pada percobaan ini hasil keluaran akan ditampilkan dalam satu baris dengan spasi antar angka.

Struktur perulangan yang digunakan sama seperti pada program PerulanganFor1, yaitu menggunakan perulangan *for*. Namun, terdapat perbedaan pada perintah keluaran. Program ini menggunakan *System.out.print*, sehingga hasil cetakan tidak berpindah ke baris baru setiap kali perulangan berjalan. Selain itu, ditambahkan tanda " " (spasi) setelah setiap angka agar hasil tampilan lebih rapi.

```

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
12 }
13

```

Gambar 2.2. Program Java dengan Perulangan For 2.

Langkah kerja:

1. Membuat file baru bernama PerulanganFor2.java
2. Menuliskan kode program untuk menampilkan deret angka dalam satu baris menggunakan system.out.print.
3. Menjalankan program dan mengamati hasil keluaran di layar.
4. Menyimpan hasil untuk dibahas pada laporan.

Dari hasil percobaan dapat dilihat bahwa semua angka dari 1 hingga 10 ditampilkan dalam satu baris yang sama, dipisahkan oleh spasi. Hal ini membuktikan bahwa penggunaan System.out.print menampilkan data secara berurutan tanpa membuat barisan baru.lanjut

2.3. Perulangan For 3

Percobaan ketiga ini bertujuan untuk memahami bagaimana perulangan for dapat digunakan untuk melakukan operasi penjumlahan berulang. Tidak hanya mencetak angka, tetapi juga menghitung total dari semua angka yang ditampilkan.


```

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 }
17 }
18

```

Gambar 2.3. Program Java dengan Perulangan For 3.

Langkah kerja:

1. Membuat file baru dengan nama PerulanganFor3.java
2. Mendeklarasikan variabel jumlah dengan nilai awal 0 untuk menampung hasil penjumlahan.
3. Menuliskan perulangan for dari angka 1 sampai 10, dan pada setiap iterasi menambahkan nilai I ke variabel jumlah.
4. Menampilkan hasil total penjumlahan setelah perulangan selesai.
5. Menjalankan program dan mencatat hasil keluaran di layar.

Hasil keluaran menunjukkan bahwa program mencetak angka 1 hingga 10 ditampilkan ke layar, lalu dijumlahkan hingga menghasilkan total 55. Program ini menunjukkan bahwa perulangan for bisa digunakan untuk menghitung nilai secara berulang. Dengan menambahkan variabel jumlah, setiap nilai I ditambahkan satu per satu sampai perulangan perulangan selesai.

2.4. Perulangan For 4

Percobaan keempat dilakukan untuk memahami cara menggunakan input dari pengguna dalam perulangan for. Program ini meminta pengguna memasukkan batas angka tertentu, lalu menampilkan angka dari 1 hingga batas tersebut dan menghitung jumlah totalnya. Dengan percobaan ini, mahasiswa dapat memahami cara memadukan perulangan dengan kelas *Scanner* untuk membaca input.

```

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
25 }
26

```

Gambar 2.4. Program Java dengan Perulangan For 4.

Langkah kerja:

1. Membuat class baru dengan nama PerulanganFor4.java.
2. Mengimpor kelas *Scanner* untuk membaca input dari pengguna.
3. Meminta pengguna memasukkan angka batas perulangan.
4. Menjalankan perulangan for dari 1 hingga angka yang dimasukkan

5. Menjumlahkan setiap nilai 1 dan menampilkan hasil akhirnya di layar.

Program berhasil menampilkan angka dari 1 sampai nilai yang dimasukkan pengguna dan menghitung totalnya. Perulangan berjalan sesuai input, menunjukkan cara penggunaan *Scanner* dan for secara bersamaan.

2.5. Nested For 0

Percobaan ini bertujuan untuk mengenalkan konsep perulangan bersarang, yaitu perulangan for di dalam for lainnya. Program ini digunakan untuk memahami bagaimana dua perulangan dapat bekerja secara bertahap, di mana perulangan luar mengatur baris, dan perulangan dalam mengatur kolom

```

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
14 }
15

```

Gambar 2.5. Program Java dengan Nested For 0.

Langkah kerja:

1. Membuat class baru dengan nama NestedFor0.java
2. Menuliskan dua perulangan for, di mana perulangan luar mengatur jumlah baris dan perulangan dalam mengatur jumlah kolom.
3. Menggunakan `System.out.print("* ")` untuk mencetak bintang dalam satu baris dan `System.out.println()` untuk pindah ke baris berikutnya.
4. Menjalankan program dan mengamati pola yang terbentuk.

Program menghasilkan pola bintang berbentuk persegi 5x5. Perulangan luar menentukan jumlah baris, sedangkan perulangan dalam menentukan jumlah bintang di tiap baris. Ini menunjukkan cara kerja dasar *nested loop* dalam membentuk pola berulang dua dimensi.

2.6. Nested For 1

Percobaan pemrograman ini bertujuan untuk memahami bagaimana *nested for* digunakan untuk menampilkan pola angka secara berurutan. Program akan mencetak angka dari 1 hingga 5 dalam bentuk baris dan kolom.

```

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            // to end the line
11        }
12    }
13
14 }
15

```

Gambar 2.6. Program Java dengan Nested For 1.

Langkah kerja:

1. Membuat file *NestedFor1.java* .
2. Membuat dua perulangan for:
3. Perulangan luar untuk baris (i).
4. Perulangan dalam untuk kolom (j).
5. Mencetak nilai j di setiap baris menggunakan *System.out.print* .
6. Menjalankan program dan melihat hasil pola angka.

Program menampilkan pola angka 1 sampai 5 dalam lima baris. Perulangan luar mengatur baris, dan perulangan dalam mencetak angka di setiap baris.

2.7. Nested For 2

Percobaan ini bertujuan untuk memahami bagaimana *nested for* dapat digunakan untuk menampilkan pola angka yang berubah sesuai baris. Setiap baris menampilkan angka yang sama dengan nilai baris tersebut.

```

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            // to end the line
11        }
12    }
13
14 }
15

```

Gambar 2.7. Program Java dengan Nested For 2.

Langkah kerja :

1. Membuat *class* baru *NestedFor2.java* .
2. Membuat dua perulangan for :
3. Perulangan luar (i) menentukan angka yang akan ditampilkan.
4. Perulangan dalam (j) mencetak angka tersebut beberapa kali dalam satu baris.
5. Menjalankan program dan mengamati hasil pola yang terbentuk.

Program mencetak angka sesuai nilai barisnya. Perulangan luar menentukan angka, sedangkan perulangan dalam mengulanginya lima kali di setiap baris.

BAB III

KESIMPULAN

Berdasarkan hasil praktikum mengenai perulangan `for` dan `nested for` pada bahasa pemrograman Java, dapat disimpulkan bahwa penggunaan struktur perulangan sangat membantu dalam mempersingkat proses yang dilakukan secara berulang. Melalui percobaan yang dilakukan, mahasiswa dapat memahami konsep dasar perulangan, mulai dari mencetak nilai berurutan, melakukan penjumlahan otomatis, hingga menampilkan pola dengan perulangan *nested*. Pemahaman ini menjadi dasar penting dalam membangun logika pemrograman yang efisien dan terstruktur.

Sebagai saran, mahasiswa disarankan untuk lebih sering berlatih membuat variasi program perulangan dengan kondisi dan pola yang berbeda agar pemahaman terhadap konsep ini semakin kuat. Selain itu, penguasaan logika dasar perulangan juga akan sangat berguna dalam mempelajari topik lanjutan seperti pengolahan data, pembuatan algoritma, serta pengembangan aplikasi yang lebih kompleks di masa mendatang.

DAFTAR PUSTAKA

Bibliography

- [1] andre, "Tutorial Belajar C: Cara Membuat Perulangan Bersarang (Nested Loop)," Duniaikom, 11 april 2019. [Online]. Available: <https://www.duniaikom.com/tutorial-belajar-c-cara-membuat-perulangan-bersarang-nested-loop/>.
- [2] A. Muhardian, "Belajar Java: Memahami 2 Jenis Perulangan dalam Java," 03 Januari 2016. [Online]. Available: https://www.petanikode.com/java-perulangan/#google_vignette.
- [3] J. Squirrels, "Loop Bersarang Java," CodeGym, 21 Juli 2023. [Online]. Available: <https://codegym.cc/id/groups/posts/id.665.loop-bersarang-java>.