

LAPORAN PRATIKUM PEKAN 5
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“PERULANGAN FOR”



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

Asisten Lab:
Muhammad Zaki Al Hafiz

Disusun Oleh:
Hamdi Sidqi Alifi
2511531009

Fakultas Teknologi Informasi
Departemen Informatika
Universitas Andalas

2025

KATA PENGANTAR

Puji dan syukur senantiasa penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan ini guna memenuhi laporan praktikum mata kuliah Algoritma Pemrograman, dengan judul: “Perulangan For”. Laporan ini berisikan hasil analisis dari praktikum pemrograman java penulis di Praktikum Kelas D yang dilaksanakan pada hari Rabu tanggal 29 Oktober 2025.

Pada kesempatan ini, Penulis ingin mengucapkan terima kasih kepada Bapak DR. Wahyudi. S.T.M.T, dosen pengampu mata kuliah Algoritma Pemrograman, yang telah memberikan tugas ini. Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah mendukung serta membantu penulis dalam penyelesaian laporan praktikum Algoritma Pemrograman.

Penulis sadar bahwa laporan ini masih memiliki beberapa kekurangan dikarenakan terbatasnya pengalaman dan pengetahuan yang penulis miliki. Oleh karena itu, penulis harap adanya bentuk saran dan kritik yang membangun dari berbagai pihak.

Akhir kata, penulis berharap laporan ini dapat bermanfaat bagi perkembangan dunia pendidikan.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Manfaat Pratikum.....	1
BAB II PEMBAHASAN	2
2.1 Teori	2
2.2 Kode Pemrograman	5
BAB III KESIMPULAN	10
3.1 Kesimpulan	10
3.2 Saran	10
DAFTAR PUSTAKA	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam membuat sebuah program, seringkali kita diharuskan untuk membuat kode yang berulang-ulang dan kode tersebut memiliki pola yang sama. Namun, sebenarnya kita bisa menggunakan salah satu utilitas yang berada pada package java. Dengan menggunakan utilitas perulangan tersebut kita dapat membuat algoritma kita seolah-olah “dipersingkat” dan ini dapat meningkatkan efisiensi kegiatan produktif kita.

Konsep perulangan selalu ada dalam setiap bahasa pemrograman walaupun terkadang di tiap-tiap bahasa pemrograman tipe perulangannya berbeda-beda. Pada bahasa pemrograman Java konsep perulangan ini ada yaitu: for-loop, while-loop, dan do-while-loop. Namun, karena yang dipraktikan pada pertemuan kelima hanya konsep perulangan for-loop, maka penulis hanya akan membahas konsep perulangan tersebut pada laporan ini.

1.2 Tujuan

1. Menjelaskan konsep perulangan for-loop dalam bahasa pemrograman Java.
2. Menjelaskan penerapan konsep perulangan for-loop dalam bahasa pemrograman Java.

1.3 Manfaat Pratikum

1. Memahami konsep perulangan for-loop dalam bahasa pemrograman Java.
2. Mampu memahami dan menerapkan konsep perulangan for-loop dalam bahasa Pemrograman Java.

BAB II PEMBAHASAN

2.1 Teori

Dalam kebahasaan *loop* memiliki arti lingkaran, dalam bahasa pemrograman Java *loop* ini berfungsi untuk mengeksekusi blok kode berdasarkan suatu kondisi yang dinilai benar (memenuhi suatu kondisi yang telah ditetapkan). Sehingga runtutan alur program terlihat seperti melingkar. *Loop* sangat berguna karena dapat menghemat waktu dan membuat pembuatan program menjadi lebih mudah.

For-loop digunakan saat kita mengetahui berapa kali baris kode tersebut di jalankan dalam menggunakan for loop kita memerlukan 3 statement, yang mana tiap statement nya memiliki peran masing masing :

```
for (initialization ; conditional ; iteration ) {  
    // Ketik program yang akan kita eksekusi  
}
```

Initialization : memberikan suatu nilai awal terhadap variabel yang telah kita deklarasikan.

Conditional : dapat berfungsi sebagai *range* (jangkauan) terlaksananya program di dalam *bracket for-loop*.

Iteration : pemberian nilai/*value* baru terhadap variabel yang telah dideklarasikan pada statement *initialization*.

Dalam menggunakan for-loop, jika *value*/nilai pada statement *initialization* tidak memenuhi statement *conditional* maka program di dalam loop ini tidak akan pernah berjalan.

2.2 Kode Pemrograman

Disini penulis akan menjelaskan beberapa program yang telah penulis kerjakan pada pertemuan kelima praktikum Algoritma Pemograman dibawah ini :

a. Program PerulanganFor1

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

Penjelasan :

1. Pada line (5) kita mengetikan *function for*.
2. Di dalam bracket pada statement *initialization*, penulis mendeklarasikan sebuah bilangan bulat/*integer* *i* dan diinisialisasikan dengan nilai 1.
3. Pada statement *conditional* kita menerapkan kondisi *integer* harus kecil sama dari 10.
4. Pada statement *iteration* kita menerapkan algoritma untuk setiap i_{baru} memiliki nilai i_{lama} yang dijumlahkan dengan angka 1.
5. Kemudian, pada line (6) kita mengetikkan *function* cetak nilai *I* untuk setiap perulangan.

Output :

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

b. Program PerulanganFor2

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

Penjelasan :

1. Sama seperti program sebelumnya ... (a), kita mengimport *function for* dengan statement seperti program (a).
2. Hanya saja, pada line (6) alih-alih mencetak dengan menggunakan *function*

Output :

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

c. Program PerulanganFor3

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

Penjelasan:

1. Pada line (4) kita inisialisasi var jumlah = 0
2. Kemudian, pada line (5), inisali *for-loop* dengan i=1 dengan *conditional statement* $i \leq 10$, dan iterasi $i++$.
3. Cetak (i)
4. Jumlah += i.
5. Jika $i < 10$ print(" + ")
6. Println() // Untuk memisahkan baris aritmatika dengan alamat hasil
7. Println("jumlah =" +)

Output:

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

d. Program PerulanganFor4

```
1 package pekan5;
2 import java.util.Scanner;
3 public class PerulanganFor4 {
4     public static void main(String[] args) {
5         int jumlah = 0;
6         int batas;
7         Scanner input = new Scanner(System.in);
8         System.out.print("Masukkan nilai batas = ");
9         batas = input.nextInt();
10        input.close();
11        for (int i = 1; i <= batas; 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 }
```

Penjelasan :

1. Inisialisasi batas dan jumlah = 0
2. Inputkan nilai batas dengan bantuan utilitas *Scanner*
3. Tambahkan For i = 1, I <= batas, i++
4. Cetak(i)
5. Jumlah += i
6. Jika i < batas print(" + ")
7. Selain itu print(" = ")
8. Println(jumlah)

Output :

```
Masukkan nilai batas = 10
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55
```


e. Program nestedFor0

```
1 package pekan5;
2 public class nestedFor0 {
3     public static void main(String[] args) {
4         int line = 1;
5         for ( ; 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 }
```

Penjelasan:

1. Loop dengan kondisi `line = 1`, `line <= 5`, `line++`
2. Namun perhatikan pada line (5), disini penulis tidak menginisialisasikan “`int line`”, tetapi terpisah yang berlokasi di line (4). Hal ini dikarenakan terkadang *Java Virtual Machine*, yaitu “mesin” yang kita gunakan untuk memprogram dengan bahasa Java ini dapat terkadang mengalami *syntax bug* dimana program tidak bisa membaca sebuah inisialisasi di dalam *for-loop function* jika variabelnya berbentuk kata, sehingga kita perlu menginisialisasikan variabel tersebut di luar *for-loop function* yang akan kita gunakan.
3. Kemudian di dalam *for-loop* line kita beri *for-loop* dengan kondisi `j = 1`, `j <= (-1 * line + 5)`, `j++`
4. `Print(“.”)`
5. `Println(line)`
6. `Println()`

Output:

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

f. Program nestedFor1

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

Penjelasan:

1. *Inner-loop* (sebagai nilai kolom) dengan kondisi $i = 1, i \leq 5, i++$
2. *Outer-loop* (sebagai nilai baris) yang berada di dalam loop 1 dengan kondisi $j = 1, j \leq 5, j++$ dan `print("*")`
3. `Println()` pada *line* (8) berguna untuk menciptakan baris baru sebelum program mengeksekusi kodingan *inner loop* dengan nilai *integer* selanjutnya.

Output:

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

g. Program nestedFor2

```
1 package pekan5;
2 public class nestedFor2 {
3     public static void main(String[] args) {
4         for (int i = 0; i <= 5; i++) {
5             for (int j=0; j <= 5; j++) {
6                 System.out.print(i+j+" ");
7             }
8             System.out.println();
9             // to end the line
10        }
11    }
12 }
```

Penjelasan :

1. *Inner-loop* (sebagai nilai kolom) dengan kondisi $i = 0, i \leq 5, i++$
2. *Outer-loop* (sebagai nilai baris) yang berada di dalam loop 1 dengan kondisi $j = , j \leq 5, j++$ dan `print (i + j + " ")`
3. `Println()` pada *line* (8) untuk pemisah antar baris.

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
```

BAB III

KESIMPULAN

3.1 Kesimpulan

Penggunaan fungsi perulangan *for* memiliki peran yang sangat penting dalam membuat suatu kode program dalam menggunakan bahasa pemrograman *Java*, karena fungsi *loop* dapat menghemat waktu kita dalam mengeksekusi blok kode yang harus dilakukan secara berulang-ulang.

3.2 Saran

Berdasarkan praktikum yang telah dilakukan mengenai fungsi *loop* dalam bahasa pemrograman *Java*. Penulis harap kita sebagai mahasiswa informatika harus dapat memahami dan mampu menerapkan konsep *loop* dalam membuat sebuah kode program.

DAFTAR PUSTAKA

Oracle. (2023). “*The Java*” *Tutorials: Learning The Java Language - Oracle Documentation*,. docs.oracle.com. Diakses pada tanggal 31 Oktober 2025, dari <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>.

W3schools. (2025). *Java For Loop*. w3schools.com. Diakses pada 31 Oktober 2025, dari https://www.w3schools.com/java/java_for_loop.asp.

