

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“PERULANGAN FOR”



disusun Oleh:

NAIRA RAMADHANI HALIL

2511533027

Dosen Pengampu:

Dr. WAHYUDI, S.T, M.T.

Asisten Praktikum:

JOVANTRI IMMANUEL GULO

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan judul “PERULANGAN FOR” dapat diselesaikan tepat waktu.

Laporan ini disusun untuk memenuhi salah satu tugas mata kuliah Praktikum Algoritma dan Pemrograman, sekaligus sebagai sarana pembelajaran dalam memahami konsep dasar pemrograman khususnya perulangan for pada Bahasa Java.

Pada kesempatan ini, penulis menyampaikan terimakasih kepada:

1. Bapak Dr. Wahyudi, S.T., M.T. selaku dosen pengampu mata kuliah Algoritma dan pemrograman yang telah memberikan bimbingan dan arahan.
2. Bang Jovantri Immanuel Gulo selaku asisten praktikum kelas A yang telah membantu pelaksanaan praktikum.
3. Teman-teman mahasiswa yang telah membantu dan memberi dukungan serta berdiskusi bersama dalam menyelesaikan laporan ini.

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

Akhir kata, semoga laporan ini dapat memberikan manfaat bagi pembaca dan menambah wawasan mengenai pemrograman Java.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum.....	2
BAB II PEMBAHASAN.....	3
2.1 Langkah Kerja Praktikum	3
2.2 Analisis Hasil dan Pembahasan	8
BAB III KESIMPULAN.....	11
3.1 Kesimpulan	11
DAFTAR PUSTAKA

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pemrograman merupakan salah satu keahlian yang harus dalam bidang informatika. Salah satu hal penting yang harus dikuasai adalah perulangan (looping). Perulangan adalah proses untuk mengeksekusi suatu perintah secara berulang sesuai jumlah tertentu atau sampai kondisi tertentu terpenuhi. Perulangan di java itu ada tiga, yaitu for, while, dan do-while. Perulangan for ini digunakan ketika jumlah yang akan kita lakukan pada perulangan diketahui. Contoh penulisannya seperti ini:

```
for (inisial; syarat; perubahan_nilai){  
    perintah_yang_akan_diulangi;  
}
```

Kalau perulangan while digunakan untuk mengulang suatu proses perulangan yang belum diketahui jumlahnya. Pada perulangan while pengecekan kondisi akan dilakukan terlebih dahulu, Jika kondisi masih bernilai benar, maka perulangan akan terus berlanjut. dan sebaliknya jika bernilai salah maka perulangan akan dihentikan. Contoh penulisannya seperti ini: while (syarat){

```
    perintah_yang_akan_diulang;  
}
```

Terakhir, perulangan do-while. Perulangan ini hampir sama dengan perulangan while, namun pernyataan dilakukan terlebih dahulu dan kemudian baru melakukan pengecekan. Contoh penulisannya seperti ini: do{

```
    perintah_yang_akan_diulang;  
    }while (perulangan);  
}
```

Dengan memahami konsep perulangan, mahasiswa mampu membuat program yang lebih dinamis, terstruktur, dan mudah dikembangkan. Oleh karena itu, praktikum ini sangat membantu mahasiswa untuk mendapatkan pemahaman lebih dalam mengenai cara

kerja dan penerapan perulangan for dalam Java melalui beberapa contoh kasus sederhana sehingga memudahkan mahasiswa untuk membuat program yang lebih kompleks.

1.2 Tujuan Praktikum

1. Memahami konsep dasar perulangan (looping) dalam pemrograman.
2. Mengetahui struktur penulisan (*syntax*) perulangan **for** pada bahasa Java.
3. Mampu menerapkan perulangan for untuk berbagai kasus pemrograman.
4. Melatih kemampuan logika dan analisis dalam menyelesaikan permasalahan menggunakan perulangan.
5. Mengembangkan program yang efisien dan mudah dibaca melalui penggunaan struktur perulangan yang tepat.

1.3 Manfaat Praktikum

1. Menambah wawasan mengenai struktur kontrol dalam Java khususnya perulangan for.
2. Melatih keterampilan mahasiswa dalam menuliskan kode yang efektif dan efisien.
3. Membantu memahami masalah komputasi yang membutuhkan proses berulang.
4. Menjadi dasar untuk mempelajari konsep pemrograman yang lebih lanjut, seperti *nested loop*, *array*, dan struktur data lainnya.
5. Meningkatkan kemampuan mengembangkan aplikasi berbasis logika iteratif dalam dunia pemrograman.

BAB II

PEMBAHASAN

2.1 Langkah Kerja Praktikum

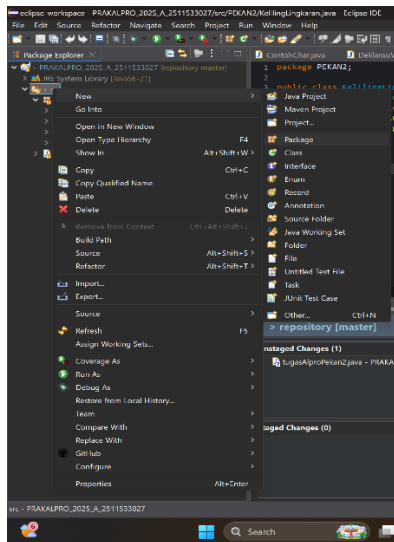
1. Persiapan Awal

- Buka aplikasi Eclipse IDE di laptop atau di komputer.

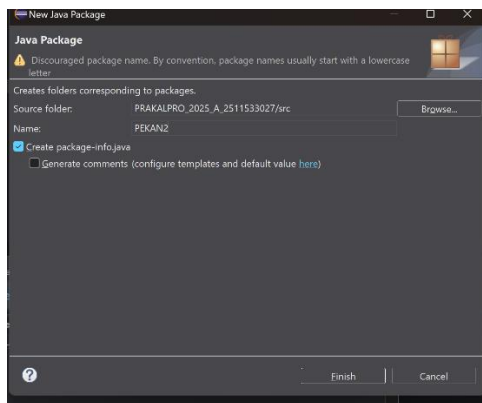


2. Membuat Package

- Klik kanan pada folder **src**, lalu pilih **New**, terakhir klik **Package**.

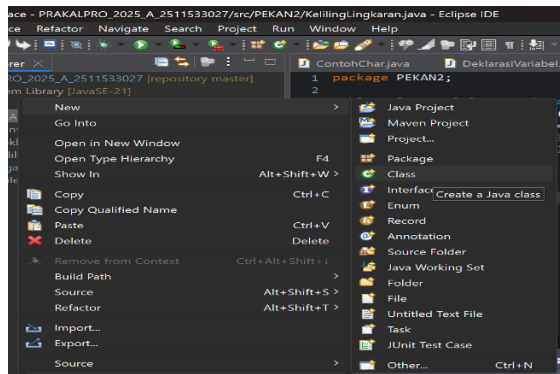


- Beri nama package **PEKAN5**



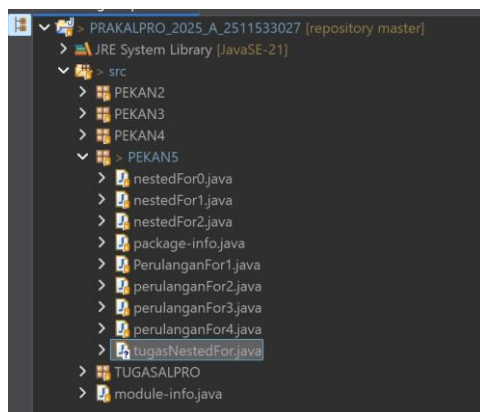
3. Membuat Class Baru untuk Setiap Percobaan

a) Klik kanan pada package **PEKAN5**, lalu pilih **New**, terakhir klik **Class**.



b) Buat beberapa class sesuai percobaan.

- perulanganFor1
- perulanganFor2
- perulanganFor3
- perulanganFor4
- nestedFor0
- nestedFor1
- nestedFor2



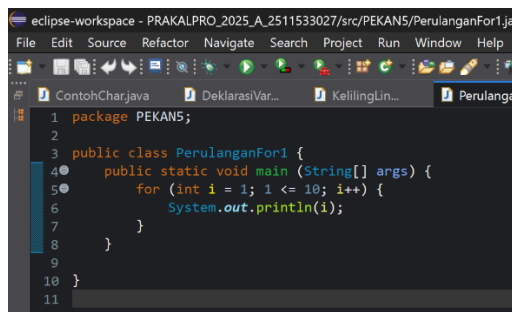
c) Saat membuat class, centang opsi **public static void main(String[] args)** agar otomatis ada fungsi **main**.

4. Menulis Kode Program

a) perulanganFor1.java

- Menuliskan program menggunakan perulangan for untuk menampilkan angka dari 1-10.
- Menggunakan variabel control **i** yang bernilai awal 1 dan meningkat hingga 10.
- Menampilkan hasil ke layar menggunakan **System.out.println()**.

- Sehingga menampilkan:



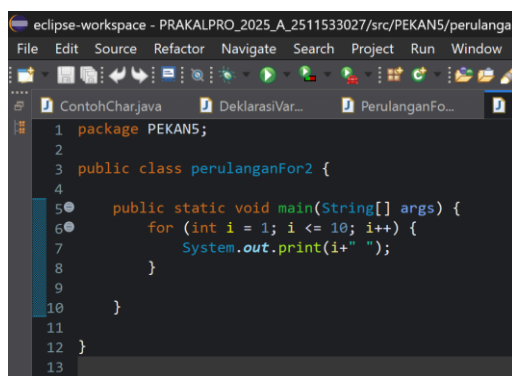
```

1 package PEKANS;
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

```

b) perulanganFor2.java

- Menuliskan program dengan perulangan for untuk menampilkan angka 1-10 pada satu baris.
- Menggunakan **System.out.print()** agar hasil tidak berpindah baris.
- Menambah spasi setelah setiap angka untuk memperjelas tampilan output.
- Sehingga menampilkan:



```

1 package PEKANS;
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

```

c) perulanganFor3.java

- Menambahkan variabel **jumlah** untuk menyimpan hasil penjumlahan bilangan.
- Menggunakan perulangan for untuk menampilkan deret angka dengan tanda “+”.
- Menggunakan pernyataan if agar tanda + tidak muncul setelah angka terakhir.
- Menampilkan hasil jumlah setelah deret bilangan.
- Sehingga menampilkan:


```

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             if (i<10) {
10                 System.out.print(" ");
11             }
12             System.out.println();
13             System.out.println("Jumlah = "+jumlah);
14         }
15     }
16 }
17
18

```

d) perulanganFor4.java

- Menambahkan input dari pengguna menggunakan kelas **Scanner**.
- Membaca nilai batas (limit) untuk menentukan jumlah pengulangan.
- Menggunakan perulangan for untuk menampilkan deret penjumlahan hingga batas input user.
- Menjumlahkan semua bilangan menggunakan **jumlah = jumlah + i**.
- Menampilkan hasil akhir di layar.
- Sehingga menghasilkan:

```

1 package PEKAN5;
2
3 import java.util.Scanner;
4
5 public class perulanganFor4 {
6
7     public static void main(String[] args) {
8         int jumlah=0;
9         int batas;
10         Scanner input = new Scanner (System.in);
11         System.out.print("Masukkan nilai batas = ");
12         batas = input.nextInt();
13         input.close();
14         for (int i=1; i<=batas; i++) {
15             System.out.print(i);
16             jumlah = jumlah+i;
17             if (i<batas) {
18                 System.out.print(" ");
19             } else {
20                 System.out.print(" ");
21             }
22         }
23         System.out.println(jumlah);
24     }
25 }
26
27

```

e) nestedFor0.java

- Membuat dua perulangan for bersarang, yaitu perulangan luar untuk baris dan perulangan dalam untuk menampilkan titik “.”.
- Pada setiap baris, jumlah titik akan berkurang satu di setiap baris.
- Setelah titik ditampilkan, menampilkan angka baris menggunakan **System.out.print(line)**.
- Menggunakan **System.out.println()** di akhir agar kursor pindah ke baris berikutnya.

- Sehingga menampilkan:

```

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            System.out.print(line);
11            System.out.println();
12        }
13    }
14 }
15
16 }
17

```

f) nestedFor1.java

- Membuat dua perulangan for, dimana perulangan for luar (i) mengatur jumlah baris dan membuat perulangan dalam (i) mengatur jumlah kolom.
- Setiap iterasi dari perulangan dalam akan menampilkan tanda Bintang “*” tanpa pindah baris.
- Setelah perulangan dalam selesai, digunakan **System.out.println()** untuk pindah ke baris berikutnya.
- Perulangan dijalankan sebanyak 5 kali, sehingga membentuk persegi Bintang 5x5.
- Sehingga menampilkan:

```

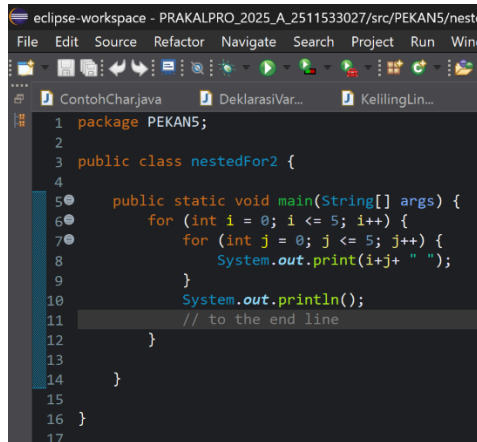
1 package PEKAN5;
2
3 public class nestedFor1 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 5; i++) {
7             for (int j = 1; j <= 5; j++) {
8                 System.out.print("*");
9             }
10            System.out.println();
11            // to end the line
12        }
13    }
14 }
15
16 }
17

```

g) nestedFor2.java

- Membuat dua perulangan for untuk menampilkan hasil penjumlahan antara dua variabel **i** dan **j**.
- Perulangan luar (i) mengatur jumlah baris dari 0-5.
- Perulangan dalam (j) mengatur kolom dari 0-5.
- Pada setiap iterasi, program menampilkan hasil penjumlahan **i+j** diikuti dengan spasi.

- Menggunakan **System.out.println()** di akhir setiap baris untuk berpindah ke baris berikutnya.
- Sehingga menampilkan:



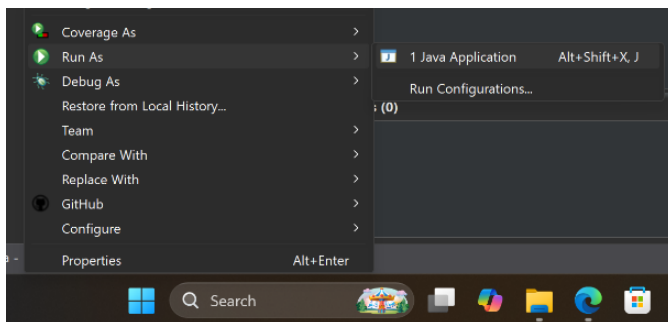
```

1 package PEKANS;
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            System.out.println();
11            // to the end line
12        }
13    }
14 }
15
16 }
17

```

5. Kompilasi dan Menjalankan Program

- Klik kanan pada file program, pilih **Run As**, terakhir klik **Java Application**.
- Lihat hasil output pada tab Console.



2.2 Analisis Hasil dan Pembahasan

1. Program perulanganFor1.java

- Hasil: Program menampilkan angka 1-10 secara berurutan ke bawah menggunakan perintah **System.out.println()**.
- Analisis: Program menggunakan perulangan for dengan variabel **i** yang dimulai dari 1 dan terus bertambah satu hingga mencapai 10.
- Teori Pendukung: Menurut dokumentasi Java SE (Oracle, 2023), perulangan for digunakan Ketika jumlah iterasi sudah diketahui. Stsruktur umumnya adalah **for (inisialisasi; kondisi; perubahan)**.

2. Program perulanganFor2.java

- Hasil: Program menampilkan angka 1-10 dalam satu baris, dengan spasi di antara angka-angkanya.

- b) Analisis: Program ini memanfaatkan perintah **System.out.print()** agar setiap nilai ditampilkan dalam satu baris. Perulangan berhenti setelah **i** mencapai 10.
 - c) Teori Pendukung: Dalam Java, perulangan for sangat efisien untuk menampilkan deret data yang terstruktur karena seluruh bagian control loop berada di satu baris kode.
3. Program perulanganFor3.java
- a) Hasil: Program menampilkan deret angka 1-10 dengan tanda “+” di antara setiap angka. Namun, nilai jumlah masih 0 karena perintah **jumlah += i;** belum ditambahkan.
 - b) Analisis: Program menunjukkan kombinasi penggunaan perulangan for dan pernyataan if. Kondisi **if (i < 10)** digunakan untuk mencegah tanda “+” muncul setelah angka terakhir.
 - c) Teori Pendukung: Menurut *Java: The Complete Reference* (Schildt, 2021), penggabungan loop dan conditional statement memungkinkan kontrol logika yang fleksibel dalam alur program.
4. Program perulanganFor4.java
- a) Hasil: Program menampilkan deret penjumlahan dari 1-sampai batas angka yang dimasukkan user, kemudian menghitung total hasil penjumlahan.
 - b) Analisis: Program menggabungkan kelas **Scanner** untuk menerima input user dan perulangan for untuk melakukan proses iteratif berdasarkan nilai batas yang dimasukkan. Nilai total dihitung dengan menambahkan **jumlah = jumlah + i;** di setiap iterasi.
 - c) Teori Pendukung: Menurut *Java Platform SE 21 Documentation* (Oracle, 2023), kelas **Scanner** digunakan untuk membaca input dari user, dan kombinasi dengan for loop sering dipakai dalam proses pengulangan berbasis input dinamis.
5. Program nestedFor0.java
- a) Hasil: Program menampilkan pola titik (.) yang berkurang jumlahnya setiap baris dan diakhiri dengan angka di sisi kanan.
 - b) Analisis: Perulangan menggunakan for bersarang. Perulangan luar mengatur jumlah baris, sedangkan perulangan dalam mengatur jumlah titik pada setiap baris. Nilai batas pada perulangan dalam dikurangi secara bertahap berdasarkan nilai variabel line, sehingga jumlah titik berkurang di tiap baris. Setelah titik selesai dicetak, angka variabel line ditampilkan di akhir setiap baris.

- c) Teori Pendukung: Menurut *Java Language Specification (JLS)*, perulangan bersarang memungkinkan satu perulangan dijalankan di dalam perulangan lain. Struktur ini umum digunakan untuk menampilkan dua pola dimensi seperti tabel atau bentuk geometris.
6. Program `nestedFor1.java`
- a) Hasil: Program menampilkan pola bintang (*) membentuk persegi berukuran 5x5.
 - b) Analisis: Perulangan luar (i) mengatur jumlah baris, sedangkan perulangan dalam (j) mengatur jumlah kolom. Setiap kali perulangan dalam selesai menampilkan lima bintang, **System.out.println()** digunakan untuk memindahkan kursor ke baris baru. Hasilnya berupa persegi penuh dari karakter bintang, yang menggambarkan fungsi perulangan bersarang dalam membentuk pola dua dimensi.
 - c) Teori Pendukung: Menurut *Java: The Complete Reference* (Herbert Schildt, 2021), perulangan `for` bersarang sering digunakan untuk menampilkan data dalam bentuk tabel, matriks, atau pola tertentu. Teknik ini juga menjadi dasar dalam pemrosesan array dua dimensi.
7. Program `nestedFor2.java`
- a) Hasil: Program menampilkan hasil penjumlahan dari dua variabel i dan j dalam bentuk tabel angka.
 - b) Analisis: Program menggunakan dua perulangan `for` bersarang untuk menghasilkan pola numerik. Variabel i berfungsi sebagai indeks baris, dan j sebagai indeks kolom. Pada setiap iterasi, program mencetak hasil dari i+j diikuti spasi. Setelah kolom selesai, perintah **System.out.println()** menutup satu baris dan melanjutkan ke baris berikutnya. Program ini menunjukkan bagaimana *nested loop* dapat dimanfaatkan untuk membuat tabel atau pola angka.
 - c) Teori Pendukung: Menurut *Java SE 21 Documentation* (Oracle, 2023), perulangan `for` digunakan untuk menjalankan blok kode dengan jumlah iterasi tertentu. Kombinasi dua perulangan memungkinkan pengolahan data dua dimensi seperti matriks atau tabel perhitungan.

BAB III PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilakukan pada materi **Perulangan For**, dapat disimpulkan bahwa perulangan merupakan salah satu struktur control penting dalam pemrograman yang berfungsi untuk mengeksekusi perintah secara berulang dengan jumlah iterasi tertentu.

Struktur for loop digunakan ketika jumlah perulangan sudah diketahui sebelumnya. Dengan memanfaatkan bagian inisialisasi, kondisi, dan increment/decrement, programmer dapat mengontrol jalannya proses iterasi secara efisien.

Dari beberapa percobaan yang telah dilakukan (perulanganFor1.java, perulanganFor2.java, perulanganFor3.java, perulanganFor4.java, nestedFor0.java, nestedFor1.java, nestedFor2.java), diperoleh pemahaman bahwa:

1. Struktur perulangan for digunakan ketika jumlah perulangan sudah pasti, karena seluruh komponen kontrol loop ditulis dalam satu baris perintah.
2. Penggabungan perulangan for dengan pernyataan kondisi if memungkinkan pengaturan tampilan output yang lebih terstruktur, seperti menambahkan tanda “+” di antara angka.
3. Nested for dapat digunakan untuk membuat pola angka atau karakter dalam dua dimensi.
4. Setiap variabel iterasi memiliki peran penting dalam mengatur baris, kolom, atau proses perhitungan.
5. Penggunaan kelas **Scanner** memungkinkan pengguna menentukan batas perulangan secara dinamis.
6. Perulangan for juga dapat dikombinasikan dengan operasi aritmetika untuk menghasilkan nilai penjumlahan atau deret angka tertentu.

Melalui praktikum ini, saya dan teman teman mahasiswa lainnya diharapkan mampu memahami konsep dasar dan penerapan perulangan for, serta dapat menggunakannya untuk menyelesaikan berbagai permasalahan logika dan pola dalam pemrograman yang lebih kompleks di tahap selanjutnya.

DAFTAR PUSTAKA

- [1] Wahyudi, *Perulangan for di Java*, presentasi kuliah, Program Studi Informatika, Universitas Andalas, 2025.
- [2] Oracle Corporation. *Java Platform, Standard Edition 21 Documentation*. 2023.
- [Online]. Tersedia di: <https://docs.oracle.com/javase/21/docs/api/>
- [3] H. Schildt, *Java: The Complete Reference*, 12th ed., New York, NY, USA: McGraw-Hill, 2021.
- [4] GeeksforGeeks. "Loops in Java," 2024.
- [Online] Tersedia di: <https://www.geeksforgeeks.org/java/loops-in-java/>
- [5] J. Gosling, B. Joy, G. Steele, dan G. Bracha, *The Java Language Specification*, 3rd ed., Boston, MA, USA: Addison-Wesley, 2005.
- [6] W3Schools. "Java For Loop," 2024.
- [Online]. Tersedia di: https://www.w3schools.com/java/java_for_loop.asp
- [7] MalasNgoding, "*Perulangan pada Java*"
- [Online]. Tersedia di: <https://www.malasngoding.com/perulangan-pada-java/>
- [8] PetaniKode, "*Belajar Java: Memahami 2 Jenis Perulangan dala Java*"
- [Online]. Tersedia di: <http://petanikode.com/java-perulangan/>