

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
PERULANGAN**



Disusun oleh:

Ghinada Fathanawafa Algma
2511533008

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

Asisten Praktikum:
Rahmad Dwirizki Olders

**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

DAFTAR ISI

DAFTAR ISI	i
KATA PENGANTAR.....	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan.....	1
1.3 Manfaat	2
BAB II PEMBAHASAN	3
2.1 Dasar Teori	3
2.2 Langkah Kerja.....	4
2.2.1 Membuat package dan class.....	4
2.2.2 Program 1 : Perulangan For 1	6
2.2.3 Program 2 : Perulangan For 2	7
2.2.4 Program 3 : Perulangan For 3	8
2.2.5 Program 4 : Perulangan For 4	8
2.2.6 Program 5 : Nested For 0	9
2.2.7 Program 6 : Nested For 1	10
2.2.8 Program 7 : Nested For 2	11
BAB III KESIMPULAN.....	13
3.1 Kesimpulan	13
DAFTAR PUSTAKA	14

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT karena atas rahmat dan karunia-Nya, laporan praktikum pemrograman Java dengan topik pengulangan menggunakan struktur for ini dapat diselesaikan dengan baik. Praktikum ini merupakan bagian dari upaya pembelajaran dalam memahami konsep dasar pengulangan (*looping*) dalam pemrograman, khususnya penggunaan perulangan for dalam bahasa Java.

Laporan ini disusun untuk memenuhi salah satu tugas mata kuliah Algoritma Pemrograman, serta sebagai dokumentasi atas kegiatan praktikum yang telah dilaksanakan. Dalam laporan ini, penulis membahas teori dasar pengulangan, implementasi kode program menggunakan for, serta analisis hasil praktikum yang diperoleh.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi perbaikan di masa mendatang. Ucapan terima kasih penulis sampaikan kepada dosen pembimbing, asisten praktikum, serta rekan-rekan yang telah memberikan dukungan dan bantuan selama proses praktikum berlangsung.

Padang, 27 Oktober 2025

Penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, struktur pengulangan merupakan salah satu konsep dasar yang sangat penting untuk dipahami. Pengulangan memungkinkan program untuk menjalankan suatu blok kode secara berulang berdasarkan kondisi tertentu, sehingga dapat meningkatkan efisiensi dan mengurangi redundansi dalam penulisan kode.

Bahasa pemrograman Java menyediakan beberapa jenis struktur pengulangan, salah satunya adalah for. Struktur for digunakan ketika jumlah iterasi telah diketahui sebelumnya, sehingga sangat cocok untuk berbagai kasus seperti pencetakan data berulang, perhitungan matematis, dan pengolahan array.

Praktikum ini bertujuan untuk memperkenalkan dan melatih kemampuan mahasiswa dalam menggunakan struktur for secara tepat dan efektif. Melalui praktikum ini, mahasiswa diharapkan mampu memahami sintaks dasar, logika pengulangan, serta penerapan for dalam berbagai kasus pemrograman. Pemahaman terhadap konsep ini menjadi fondasi penting dalam pengembangan aplikasi yang lebih kompleks di masa mendatang.

1.2 Tujuan

1. Memahami konsep dasar perulangan for pada Java.
2. Memahami cara kerja perulangan for pada Java.
3. Mengimplementasikan perulangan for dalam kasus sederhana.
4. Menumbuhkan keterampilan berpikir logis dan sistematis dalam menyusun algoritma berbasis pengulangan.

1.3 Manfaat

1. Menambah pemahaman mengenai konsep perulangan for pada Java.
2. Meningkatkan kemampuan dalam menulis, menguji, dan memperbaiki kode program yang digunakan.
3. Mahasiswa belajar cara menganalisis alur masalah dan menyelesaikan masalah menggunakan penggunaan struktur for.
4. Dengan memahami pengulangan, mahasiswa dapat menulis kode yang lebih efisien dan terstruktur, menghindari pengulangan manual yang tidak perlu.

BAB II

PEMBAHASAN

2.1 Dasar Teori

Pengulangan (*looping*) merupakan salah satu konsep fundamental dalam pemrograman yang memungkinkan eksekusi suatu blok kode secara berulang berdasarkan kondisi tertentu. Struktur pengulangan sangat berguna untuk menghemat penulisan kode, meningkatkan efisiensi, dan menyederhanakan logika program.

Dalam bahasa pemrograman Java, terdapat beberapa jenis struktur pengulangan, yaitu *for*, *while*, dan *do-while*. Struktur *for* digunakan ketika jumlah iterasi telah diketahui sebelumnya. Sintaks dasar dari perulangan *for* adalah sebagai berikut:

```
for (inisialisasi; kondisi; increment/decrement) {  
    // blok kode yang akan diulang  
}
```

Penjelasan komponen:

- Inisialisasi: Menentukan nilai awal variabel kontrol.
- Kondisi: Mengevaluasi apakah pengulangan akan dilanjutkan.
- Increment/Decrement: Mengubah nilai variabel kontrol setiap iterasi.

Struktur *for* sering digunakan dalam berbagai kasus seperti:

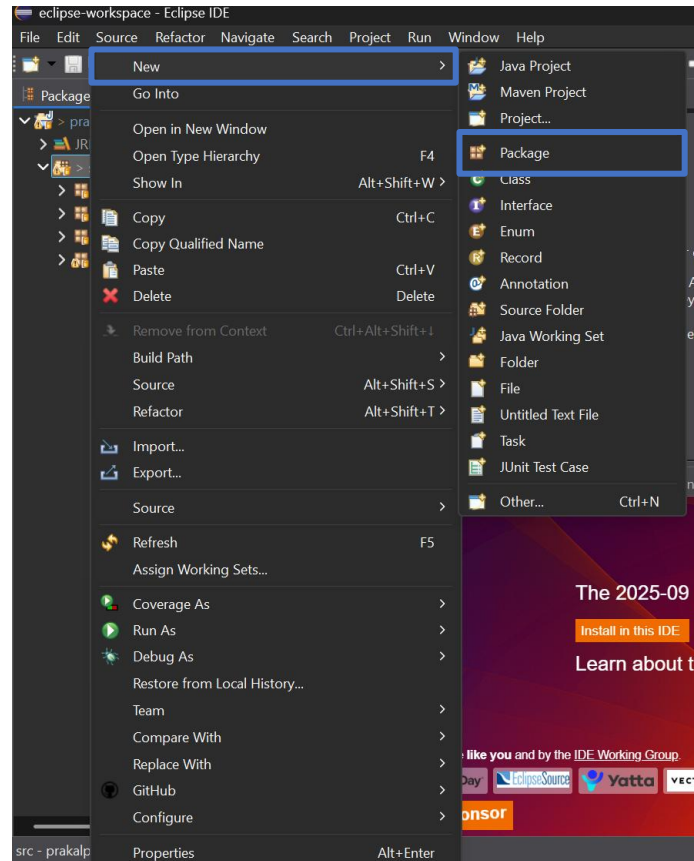
- Menampilkan data dalam jumlah tertentu
- Mengakses elemen array
- Melakukan perhitungan berulang seperti faktorial, deret, atau pola

Pemahaman terhadap struktur *for* sangat penting karena menjadi dasar dalam membangun algoritma yang lebih kompleks, seperti pengolahan data, pencarian, dan pengurutan (*sorting*).

2.2 Langkah Kerja

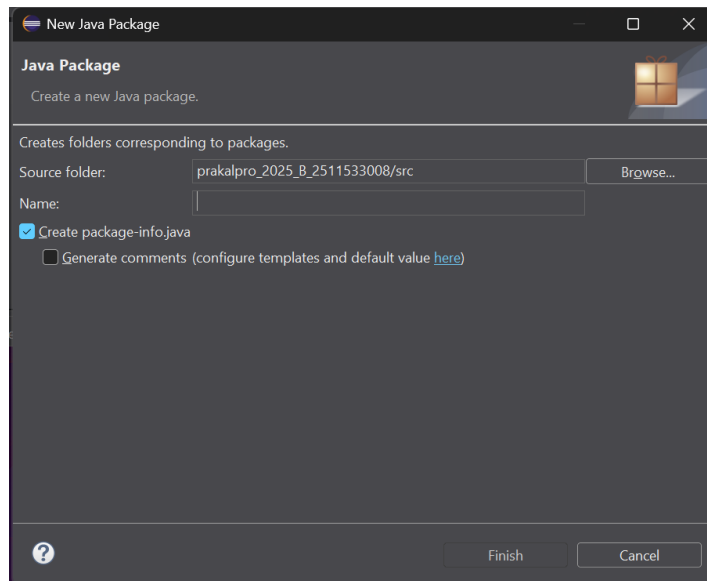
2.2.1 Membuat package dan class

1. Sebelum membuat atau menuliskan kode program pertama kita harus membuat *package*. Dengan cara klik kanan pada *src*, lalu pilih *New* dan pilih *package*.



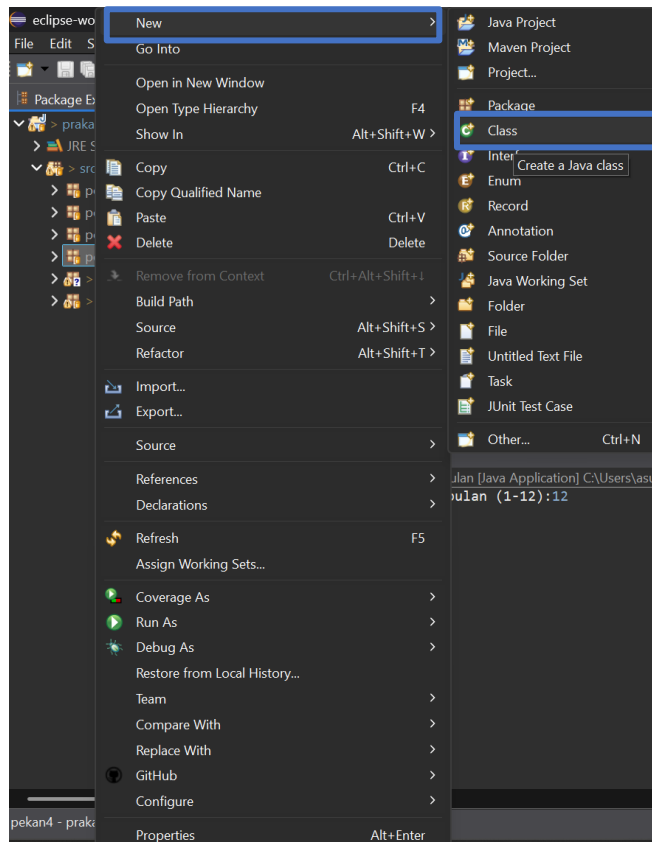
Gambar 2.1 membuat package

2. Setelah mengklik *package* kita akan diminta untuk mengisi nama *package* pada menu “Java Package”.



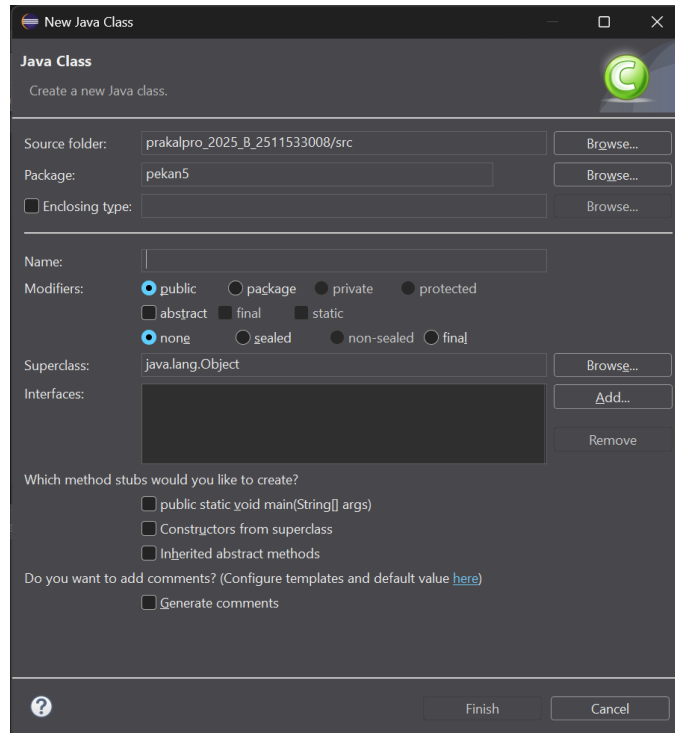
Gambar 2.2 membuat nama package

- Setelah selesai membuat *package* klik kanan pada *package* yang telah dibuat lalu pilih *New* dan pilih *class*.



Gambar 2.3 membuat class

- Setelah itu akan muncul menu “Java Class” di sana kita diminta untuk mengisi nama *class*.



Gambar 2.4 membuat nama class

2.2.2 Program 1 : Perulangan For 1

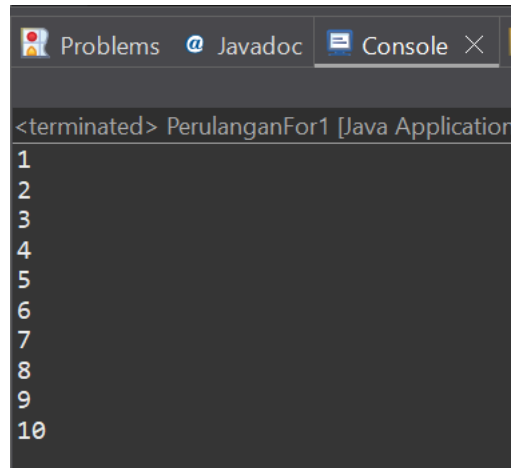
- Pada *class* yang telah dibuat masukkan kode program berikut.

```
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 }
13
```

Kode program 2.1 Kode program perulangan for 1

- Setelah selesai membuat kode program maka tekan tombol run untuk menampilkan output dari kode program yang dibuat.

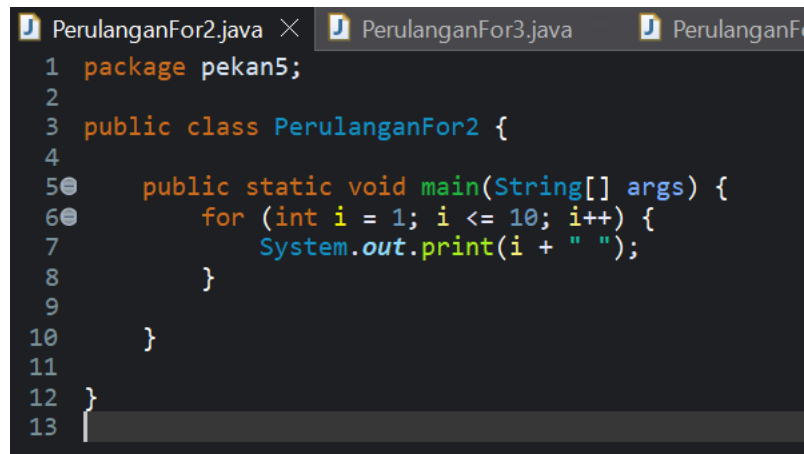
3. Tampilan dari output kode program di atas akan tampil pada menu “Console” seperti berikut.



Gambar 2.5 output perulangan for 1

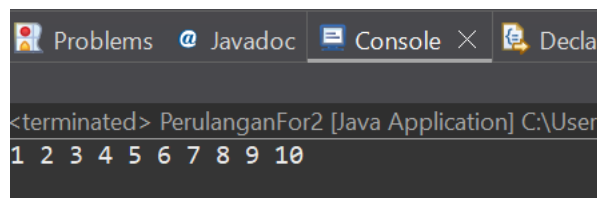
2.2.3 Program 2 : Perulangan For 2

1. Buat *class* baru pada package yang sama lalu pada *class* baru tersebut tulis kode program berikut.



Kode program 2.2 Kode program Perulangan for 2

2. Untuk menjalankan kode program tekan tombol run dan akan muncul output seperti berikut pada menu “console”



Gambar 2.6 output kode program perulangan for 2

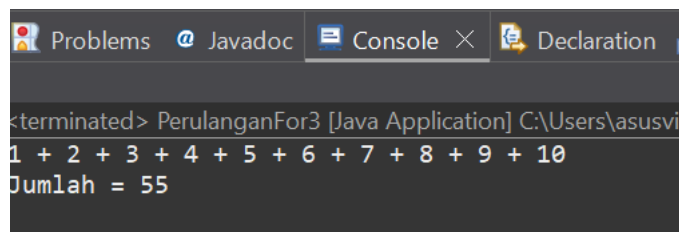
2.2.4 Program 3 : Perulangan For 3

1. Pada *class* baru buat kode program seperti berikut ini.

```
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 Kode program perulangan for 3

2. Setelah membuat kode program klik tombol run untuk menjalankan program dan pada menu “console” akan tampil output seperti ini.



```
<terminated> PerulanganFor3 [Java Application] C:\Users\asusv...
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 55
```

Gambar 2.7 output dari perulangan for 3

2.2.5 Program 4 : Perulangan For 4

1. Pada *class* baru buat kode program seperti berikut ini.

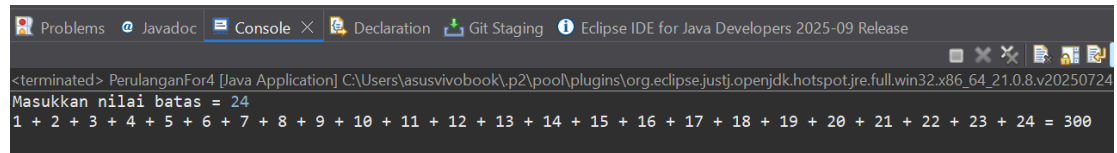
```

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 }
28

```

Kode program 2.4 kode program perulangan for 4

2. Untuk menjalankan kode program klik tombol run dan akan muncul output seperti berikut di menu “console”.



```

<terminated> PerulanganFor4 [Java Application] C:\Users\asusvivobook\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724
Masukkan nilai batas = 24
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 + 19 + 20 + 21 + 22 + 23 + 24 = 300

```

Gambar 2.8 output perulangan for 4

2.2.6 Program 5 : Nested For 0

1. Pada *class* baru buat kode program seperti berikut ini.

```

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

```

Kode program 2.5 kode program nested for 0

2. Untuk menjalankan program klik tombol run dan akan muncul output pada menu “console” seperti berikut.

```

<terminated> nestedFor0 [Java Application] C:\...
....1
...2
..3
.4
5

```

Gambar 2.9 output nested for 0

2.2.7 Program 6 : Nested For 1

1. Pada *class* baru buat kode program berikut ini.

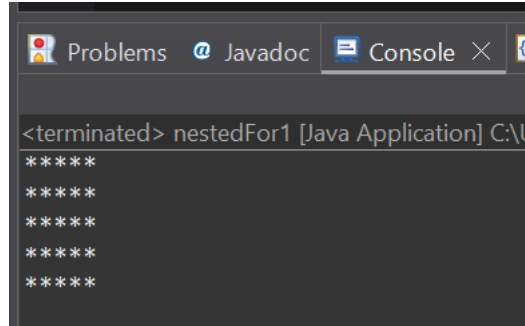
```

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

```

Kode program 2.6 kode program neted for 1

2. Untuk menjalankan program klik tombol run dan akan muncul output pada menu “console” seperti berikut ini.



Gambar 2.10 output nested for 1

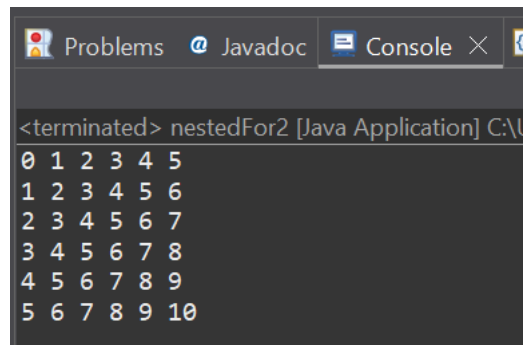
2.2.8 Program 7 : Nested For 2

1. Pada *class* baru buat kode program seperti berikut.

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

Kode program 2.7 kode program nested for 2

2. Untuk menjalankan program klik tombol run dan akan muncul output pada menu “console” seperti berikut.



The screenshot shows an IDE window with three tabs: 'Problems', 'Javadoc', and 'Console'. The 'Console' tab is active, displaying the output of a Java application named 'nestedFor2'. The output consists of six lines of numbers, each line representing the result of an inner loop iteration for a specific value of the outer loop. The numbers are arranged in a grid-like pattern, with each line containing a sequence of numbers from 0 to 10, incrementing by 1 for each subsequent line.

```
<terminated> nestedFor2 [Java Application] C:\U
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
```

Gambar 2.11 output nested for 2

BAB III

KESIMPULAN

3.1 Kesimpulan

Berdasarkan hasil praktikum yang telah dilaksanakan, dapat disimpulkan bahwa:

Struktur pengulangan `for` merupakan salah satu komponen penting dalam pemrograman Java yang memungkinkan eksekusi kode secara berulang dengan jumlah iterasi yang telah ditentukan.

Penggunaan `for` sangat efektif dalam menyelesaikan berbagai permasalahan logika, seperti pencetakan pola, perhitungan matematis, dan pengolahan data secara berulang.

Praktikum ini membantu mahasiswa memahami sintaks dasar, logika kontrol, serta penerapan struktur `for` dalam konteks nyata pemrograman.

Melalui latihan dan analisis program, mahasiswa dapat meningkatkan kemampuan berpikir logis, efisiensi penulisan kode, dan keterampilan debugging.

Dengan pemahaman yang baik terhadap struktur `for`, mahasiswa memiliki bekal awal yang kuat untuk mempelajari konsep pemrograman yang lebih kompleks di masa mendatang.

DAFTAR PUSTAKA

- [1] H. Schildt, *Java: The Complete Reference*, 11th ed. New York: McGraw-Hill Education, 2019.
- [2] J. Bloch, *Effective Java*, 3rd ed. Boston: Addison-Wesley, 2018.
- [3] Oracle, “The for Statement (The Java™ Tutorials > Learning the Java Language > Language Basics),” Oracle, 2025. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>
- [4] D. Liang, *Introduction to Java Programming and Data Structures*, 12th ed. Pearson, 2021.
- [5] R. L. Sebesta, *Concepts of Programming Languages*, 12th ed. Pearson, 2020.