

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
IMPLEMENTASI PERULANGAN FOR DAN NESTED FOR**



Disusun Oleh:

Haliya Isma Husna Putri Ahmadi

2511532002

Dosen Pengampu:

Wahyudi. Dr. S.T.M.T

Asisten Praktikum:

Muhammad Zaky Al Hafiz

**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Puji dan Syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas Rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan Praktikum Algoritma dan Pemrograman pekan 5 dengan judul “Implementasi Perulangan For dan Nested For” pada bahasa pemrograman *Java* tepat pada waktunya.

Penyusunan laporan ini bertujuan untuk memenuhi salah satu tugas dalam mata kuliah Praktikum Algoritma dan Pemrograman sekaligus sebagai bentuk pembelajaran bagi penulis untuk memahami konsep dan penerapan struktur pengulangan, khususnya macam-macam implementasi perulangan *for* dan *nested for* dalam bahasa pemrograman *Java*. Melalui praktikum ini, diharapkan mahasiswa mampu mengimplementasikan logika perulangan untuk menyelesaikan berbagai permasalahan pemrograman secara efisien.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang membangun demi perbaikan di masa yang akan datang.

Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada dosen pengampu, asisten praktikum, serta semua pihak yang terlibat dalam pelaksanaan praktikum dan penyusunan laporan ini. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi penulis maupun pembaca.

Padang, 27 September 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I.....	4
1.1 Latar Belakang.....	4
1.2 Tujuan	4
1.3 Manfaat	4
BAB II	5
2.1 Dasar Teori.....	5
2.1.1 Perulangan <i>For</i>	5
2.1.2 Perulangan <i>Nested For</i>	6
2.2 Langkah Kerja	7
2.2.1 Pembuatan <i>Package</i> dan <i>Class</i>	7
2.2.2 Program Pertama: <i>PerulanganFor1.java</i>	11
2.2.3 Program Kedua: <i>PerulanganFor2.java</i>	12
2.2.4 Program Ketiga: <i>PerulanganFor3.java</i>	13
2.2.5 Program Keempat: <i>PerulanganFor4.java</i>	14
2.2.6 Program Kelima: <i>NestedFor0.java</i>	15
2.2.7 Program Keenam: <i>NestedFor1.java</i>	16
2.2.8 Program Ketujuh: <i>NestedFor2.java</i>	17
2.2.9 Upload <i>Package</i> ke GitHub.....	19
BAB III.....	21
3.1 Kesimpulan	21
3.2 Saran	21
DAFTAR PUSTAKA.....	22

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam mempelajari bahasa pemrograman, pemahaman terhadap perulangan (*looping*) merupakan salah satu konsep dasar yang sangat penting. Dengan perulangan, memungkinkan program untuk menjalankan satu atau lebih perintah secara berulang hingga kondisi tertentu terpenuhi. Adanya struktur perulangan membuat penulisan kode menjadi lebih efisien dan mudah dibaca.

Salah satu jenis perulangan yang umum digunakan adalah perulangan *for*, yang berfungsi untuk mengulang perintah dengan jumlah yang ditentukan. Selain itu, terdapat juga *nested for*, perulangan *for* yang berada di dalam perulangan *for* lainnya. Struktur ini sering digunakan untuk membuat pola, table, atau proses berulang lainnya.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

- 1.2.1 Memahami konsep dasar dan struktur *syntax* perulangan *for* dalam bahasa pemrogram *Java*.
- 1.2.2 Mampu mengimplementasikan berbagai jenis perulangan, sesuai dengan kasus yg dibutuhkan.
- 1.2.3 Membangun logika pemrograman dalam menyusun programan dan algoritma perulangan.

1.3 Manfaat

Manfaat yang diharapkan dari pelaksanaan praktikum ini antara lain:

- 1.3.1 Memberikan pemahaman mengenai konsep dasar perulangan, khususnya perulangan *for* dan *nested for* dalam bahasa *Java*.
- 1.3.2 Memberikan pengalaman langsung bagi mahasiswa dalam mengimplementasikan perulangan untuk suatu kasus *real*.

BAB II

PEMBAHASAN

2.1 Dasar Teori

Perulangan (*looping*) adalah proses menjalankan satu atau beberapa perintah secara berulang hingga mencapai kondisi tertentu terpenuhi. Dalam *Java* terdapat tiga perulangan utama yaitu *for*, *while*, dan *do-while*. Pada pekan ini akan lebih difokuskan dengan perulangan *for* dan *nested for*. Perulangan digunakan untuk mempersingkat kode saat harus mengulang proses yang sama berkali-kali.

Perulangan *for* digunakan ketika telah mengetahui kondisi atau jumlah perulangan yang diinginkan. Perulangan *for* sangat efisien untuk mengiterasi melalui koleksi data seperti *array*, *list*, *string*, atau *range*.

2.1.1 Perulangan *For*

Perulangan *for* digunakan ketika jumlah pengulangan sudah diketahui. Perulangan *for* memiliki struktur umum, yaitu :

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

Keterangan :

1. *Inisialisasi*

Inisialisasi menentukan nilai awal variabel.

2. *Kondisi*

Kondisi adalah ketentuan untuk variabel, yang akan dicek sebelum iterasi.

3. *Increment/Decrement*

Increment/decrement mengatur perubahan nilai pada variabel.

a) *Increment* (Penambahan)

Operasi kenaikan/ penjumlahan, untuk menaikkan nilai suatu variabel sebesar 1 (++). Sedangkan untuk menambah selain 1 menggunakan (+=).

b) *Decrement* (Pengurangan)

Operator penurunan, menurunkan nilai variabel sebesar 1 (--). Untuk mengurangi nilai variabel selain 1 dapat menggunakan (-=).

2.1.2 Perulangan *Nested For*

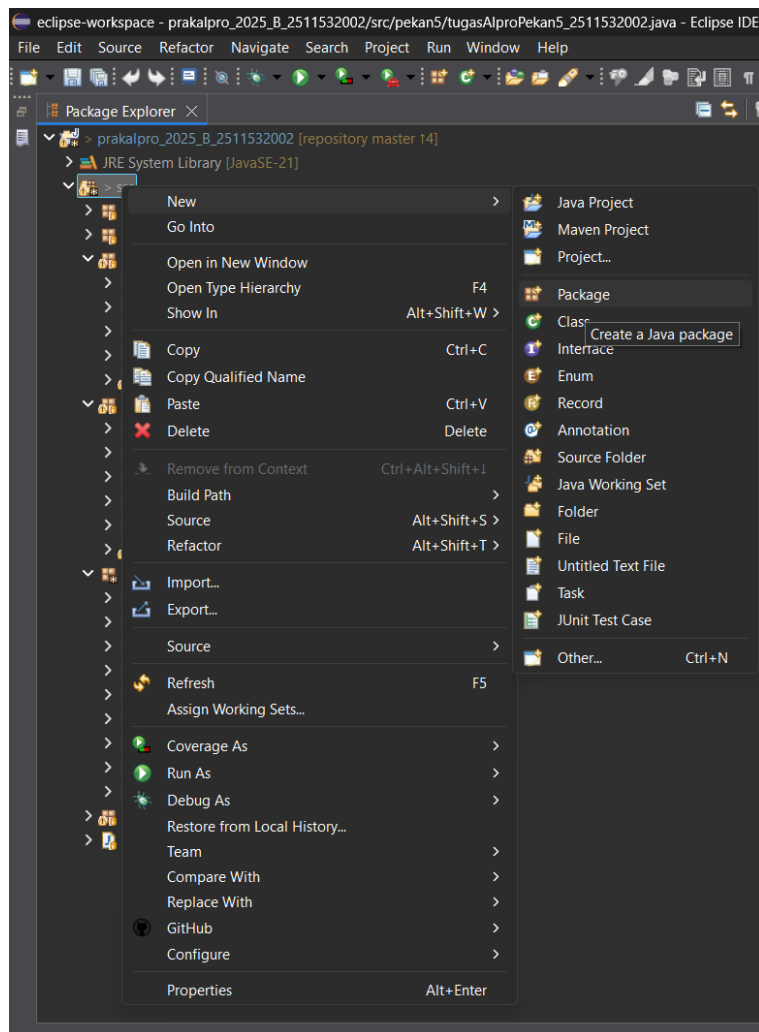
Nested for adalah perulangan *for* didalam *for* lainnya, berfungsi untuk menampilkan pola atau memproses data. Umumnya digunakan saat mengulang elemen-elemen dalam struktur multidimensi, seperti 2 dimensi atau saat melakukan tindakan untuk kombinasi elemen dari dua set atau lebih.

Keunggulan dari perulangan *for* adalah untuk efisiensi pengulangan dengan jumlah kondisi yang sudah didekatahui, namun jika salah *syntax* salah dalam penulisan dapat menyebabkan *infinity loops* jika kondisi salah.

2.2 Langkah Kerja

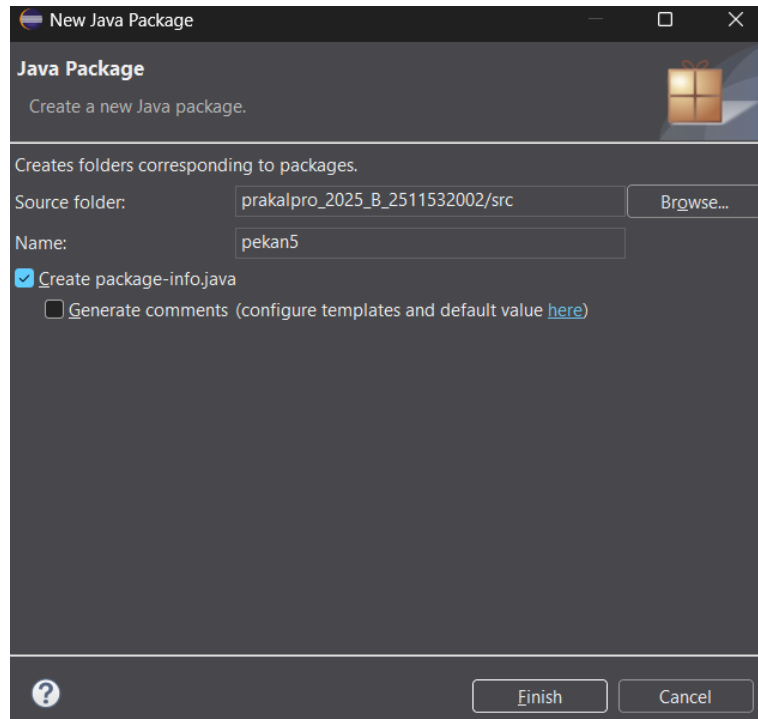
2.2.1 Pembuatan *Package* dan *Class*

1. Sebelum membuat suatu program *user* perlu untuk membuat *package* terlebih dahulu. Caranya dengan klik kanan pada *src* lalu pilih “New” dan klik “Package”.



2. 1 Pembuatan *Package*

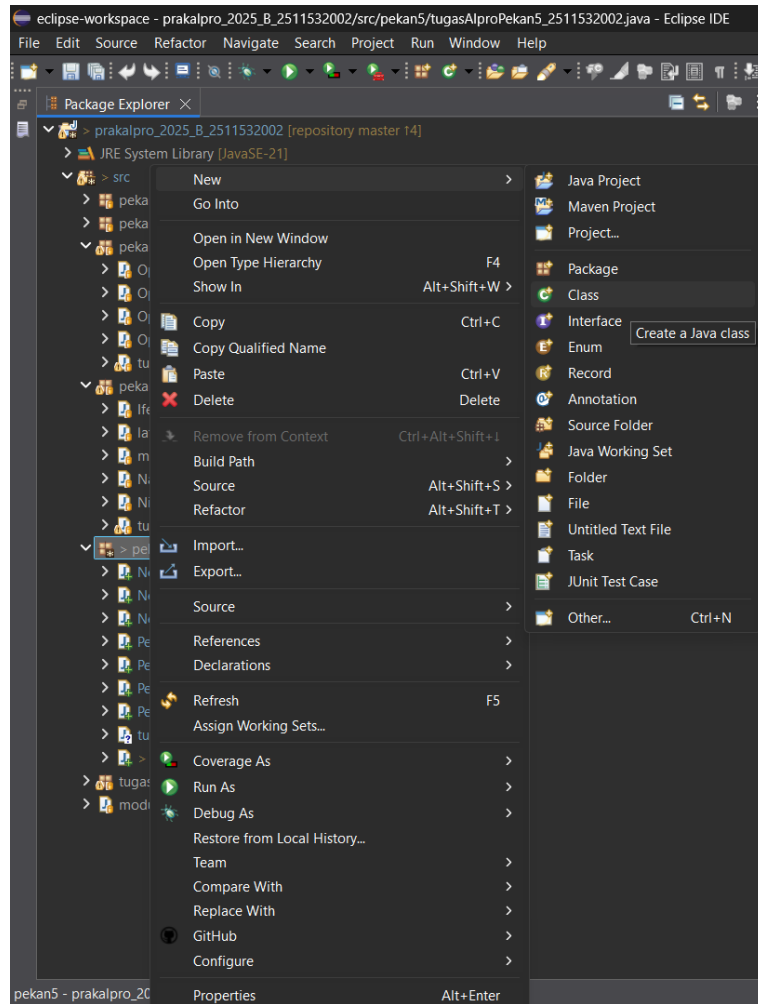
- Setelah itu akan muncul menu “*Java Package*”, buat nama *package* dengan ketentuan tanpa *capslock*, *space*, dan karakter khusus lainnya. Lalu “*Finnish*”.



2. 2 Pembuatan Package

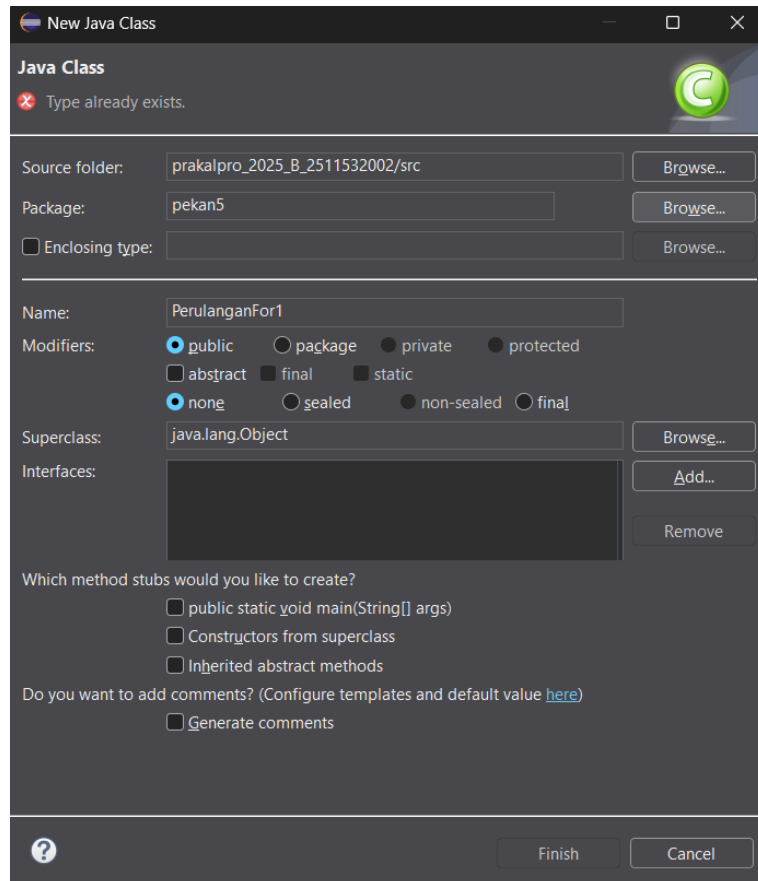
Maka *user* telah berhasil membuat *package*.

3. Selajutnya adalah membuat Membuat *class*, pada *package* yang telah dibuat tapi klik kanan lalu pilih “New” dan pilih “Class”.



2. 3 Pembuatan Class

4. Setelah itu pada menu “Java Class” buat nama *class*, dengan ketentuan nama harus *capslock* diawal kalimat dan tanpa spasi. Lalu pilih “*Public static void main (String[] grs)*” agar tidak perlu membuat *methode main()* secara manual. Selanjutnya “*finish*”.



2. 4 Pembuatan Class

2.2.2 Program Pertama: *PerulanganFor1.java*

1. Pada *class* yang sudah dibuat, masukan *syntaxis* seperti berikut.

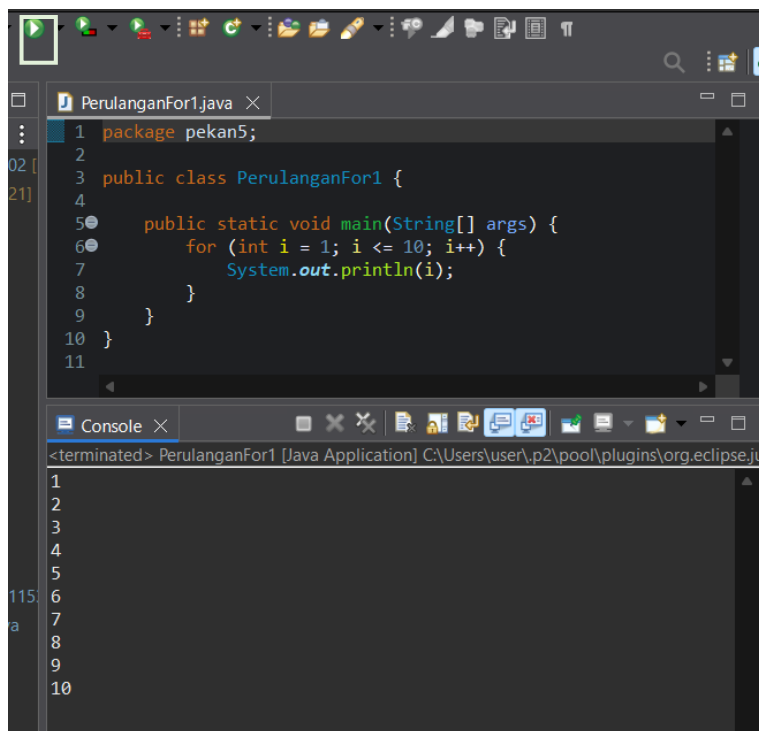
```

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

```

2. 5 Syntax *PerulanganFor1*

2. Setelah itu jalan kan program dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka nantinya akan menampilkan hasil program seperti berikut



2. 6 Output Program *PerulanganFor1*

Maka pada menu *console* akan muncul hasil program seperti berikut, disini dapat dilihat bahwa dari perulangan berikut menghasilkan *output* angka dari 1 sampai 10 yang ditampilkan berurutan kebawah.

2.2.3 Program Kedua: *PerulanganFor2.java*

1. Buat kembali *class* baru pada *package* *pekan5* dengan nama kelas “*PerulanganFor2*”.
2. Selanjutnya masukan *syntax* seperti berikut.

```

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

```

2. 7 Syntax *PerulanganFor2*

3. Lalu jalankan program dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka nantinya akan menampilkan hasil program seperti berikut.

```

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

```

<terminated> PerulanganFor2 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.j...

1 2 3 4 5 6 7 8 9 10

2. 8 Output Program *PerulanganFor2*

Pada menu *console* terlihat hasil *output* berupa angka “1 2 3 4 5 6 7 8 9 10” secara berurutan dalam satu baris. Hal ini disebabkan perintah yang digunakan adalah “*System.out.print(i + " ");*” bukan “*System.out.println(i + " ");*”. Sehingga hasil akan ditampilkan pada *line* yang sama.

2.2.4 Program Ketiga: *PerulanganFor3.java*

1. Buat kembali *class* baru dengan nama kelas “PerulanganFor3”.
2. Selanjutnya masukan *syntax* seperti berikut.

```

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 }

```

2. 9 Syntax Program *PerulanganFor3*

3. Lalu jalankan program dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka nantinya akan menampilkan hasil program seperti berikut.

```

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 }

```

Console

```

<terminated> PerulanganFor3 [Java Application] C:\Users\user\.p2\pool\plugins\org.eclipse.j
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 55

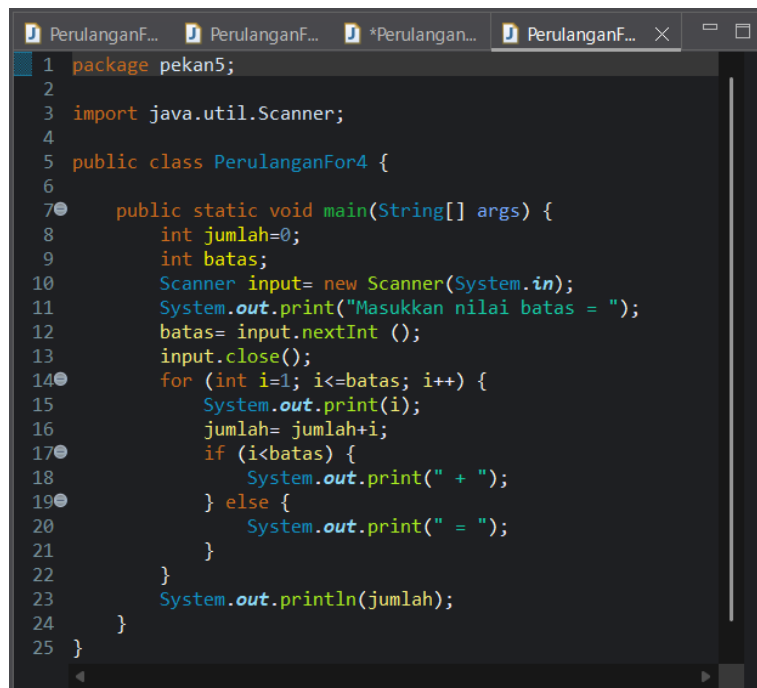
```

2. 10 Output Program *PerulanganFor3*

Pada menu *console* terlihat hasil *output* berupa hasil perulangan penjumlahan “1+2+3+4+5+6+7+8+9+10” dengan hasil “Jumlah = 55” telah sesuai.

2.2.5 Program Keempat: *PerulanganFor4.java*

1. Buat kembali *class* baru dengan nama kelas “PerulanganFor4”.
2. Selanjutnya masukan *syntax* seperti berikut.



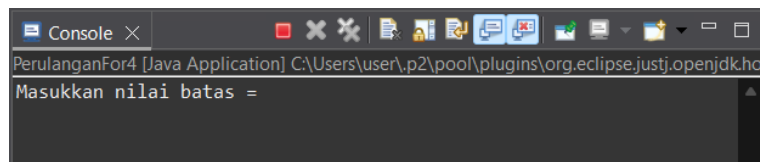
```

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 }

```

2. 11 *Syntax Program PerulanganFor4*

3. Lalu jalankan program dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka nantinya akan menampilkan hasil program seperti berikut.



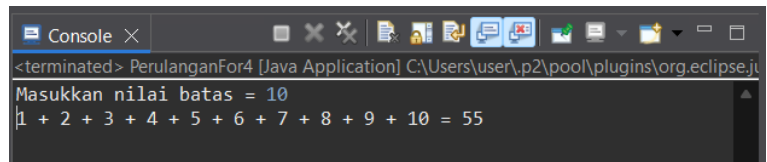
```

Console
PerulanganFor4 [Java Application] C:\Users\user\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Masukkan nilai batas =

```

Pada menu *console* kita dapat memasukkan batasan nilai perulangan yang nantinya akan dijumlahkan. Disini kita akan mencoba dengan perulangan 10x dan 7x.

a) Percobaan ke-1



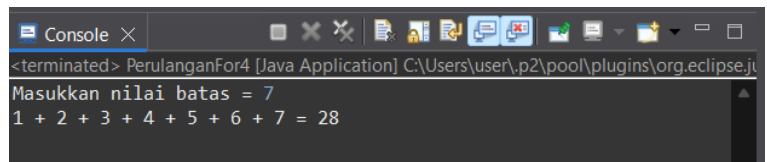
```

Console
<terminated> PerulanganFor4 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.j
Masukkan nilai batas = 10
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55

```

2. 12 Hasil Percobaan Program PerulanganFor4

b) Percobaan ke-2



```

Console
<terminated> PerulanganFor4 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.j
Masukkan nilai batas = 7
1 + 2 + 3 + 4 + 5 + 6 + 7 = 28

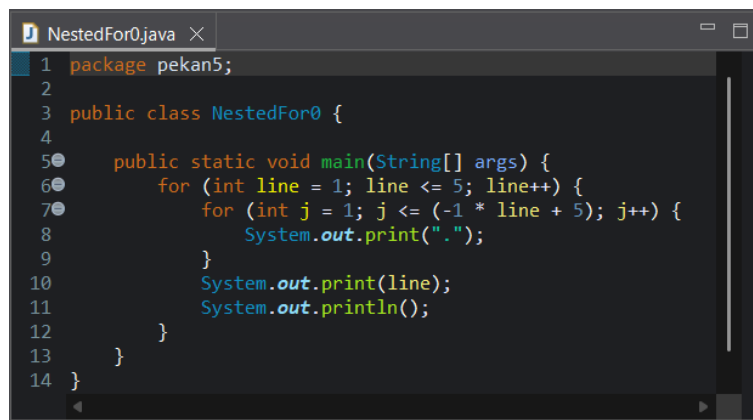
```

2. 13 Hasil Percobaan Program PerulanganFor4

Dapat dilihat dari dua percobaan penjumlahan bilangan perulangan dengan batasan perulangan yang di *input* kan menghasilkan hasil yang sesuai.

2.2.6 Program Kelima: *NestedFor0.java*

1. Buat kembali class baru dengan nama kelas “NestedFor0”.
2. Setelah itu masukan *syntax* seperti berikut.



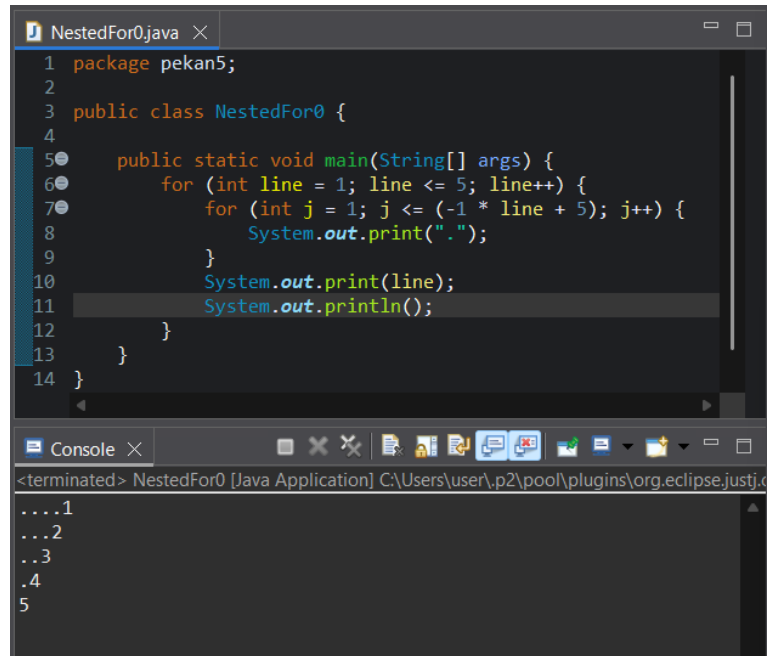
```

NestedFor0.java
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 }

```

2. 14 Syntax Program NestedFor0

- Setelah itu, jalankan program dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka nantinya akan menampilkan hasil program pola segitiga terbalik dari titik (.), kemudian di akhir baris menampilkan nomor dengan perulangan dari 1 hingga 5 seperti berikut.



```

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 }

```

```

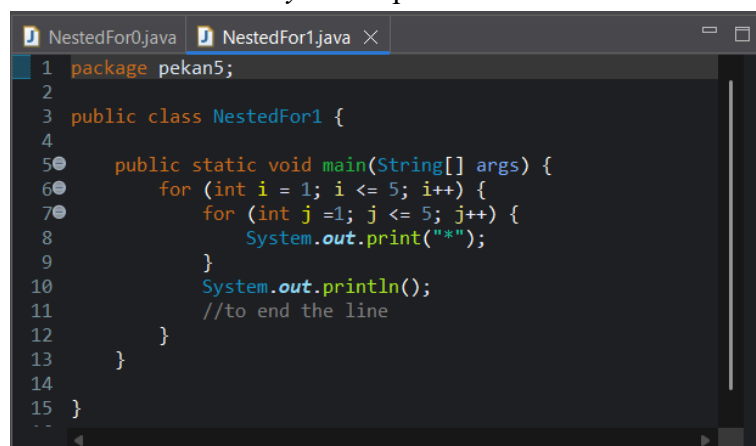
<terminated> NestedFor0 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.justj...
....1
...2
..3
.4
5

```

2. 15 Output Program NestedFor0

2.2.7 Program Keenam: *NestedFor1.java*

- Buat kembali class baru dengan nama kelas “NestedFor1”.
- Setelah itu masukan *syntax* seperti berikut.



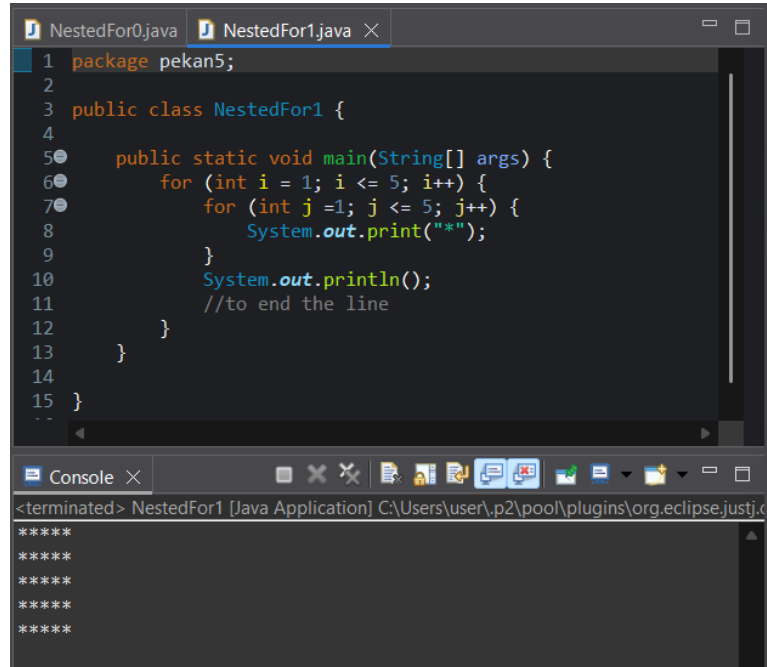
```

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 }

```

2. 16 Syntax Program NestedFor1

- Setelah itu, jalankan program dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka nantinya akan menampilkan hasil program pola dua dimensi (kolom dan baris, 5x5) yang diisi dengan bintang “*”.



```

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 }

```

Console

```

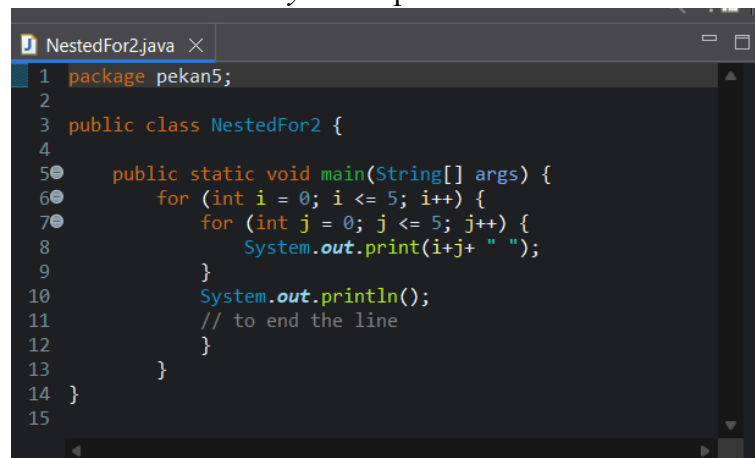
<terminated> NestedFor1 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.justj.c
*****
*****
*****
*****
*****

```

2. 17 Output Program NestedFor1

2.2.8 Program Ketujuh: *NestedFor2.java*

- Buat kembali class baru dengan nama kelas “NestedFor2”.
- Setelah itu masukan *syntax* seperti berikut.



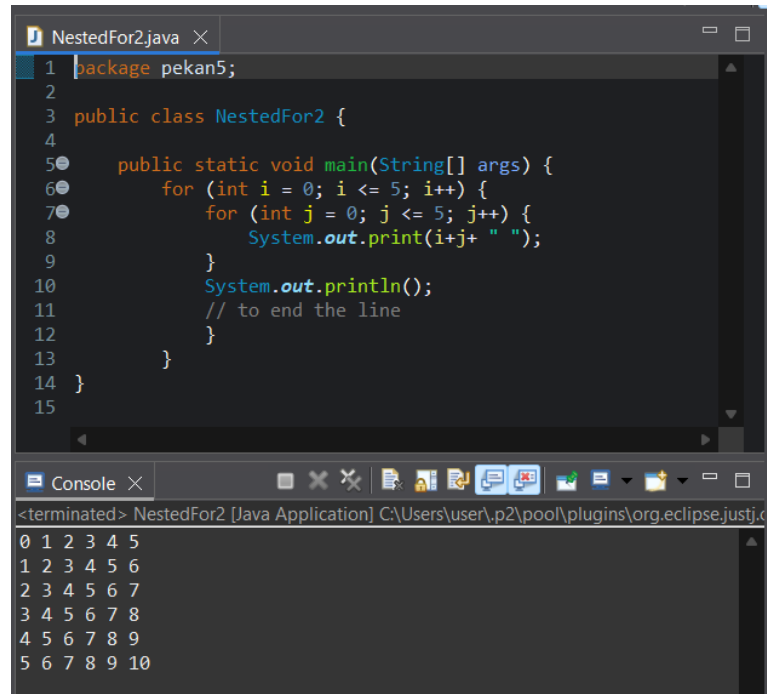
```

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 }

```

2. 18 Syntax Program NestedFor2

3. Setelah itu, jalankan program dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka nantinya akan menampilkan hasil kombinasi hasil penjumlahan dari 2 variabel *i* (baris) dan *j* (kolom).



The screenshot shows the Eclipse IDE with a Java file named `NestedFor2.java` open. The code is as follows:

```
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
```

Below the code editor, the `Console` view shows the output of the program:

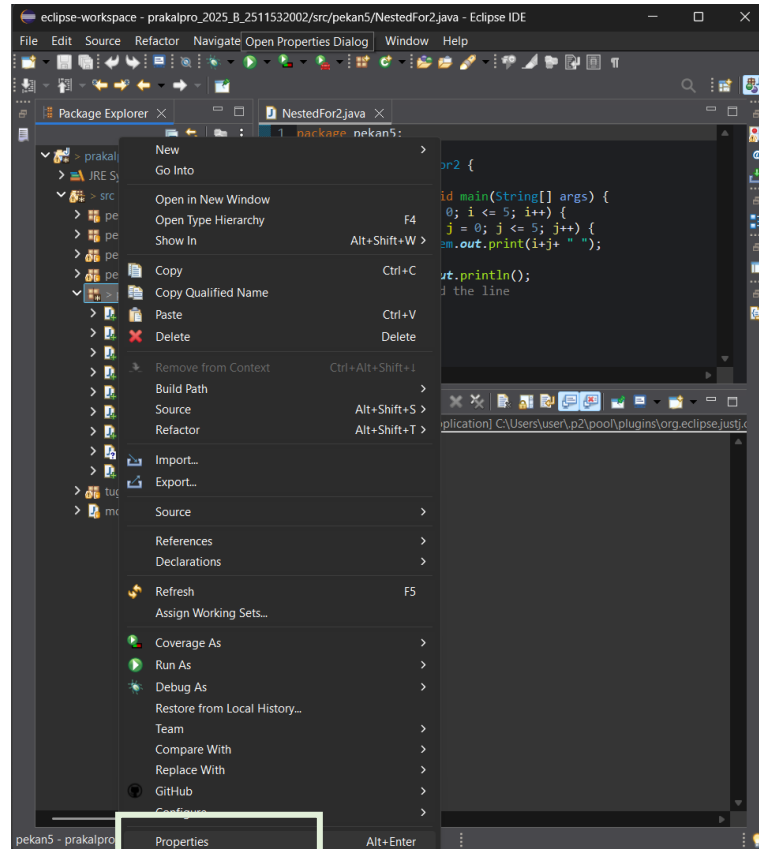
```
<terminated> NestedFor2 [Java Application] C:\Users\user\.p2\pool\plugins\org.eclipse.justj.c
0 1 2 3 4 5
1 2 3 4 5 6
2 3 4 5 6 7
3 4 5 6 7 8
4 5 6 7 8 9
5 6 7 8 9 10
```

2. 19 Output Program *NestedFor2*

2.2.9 Upload Package ke GitHub

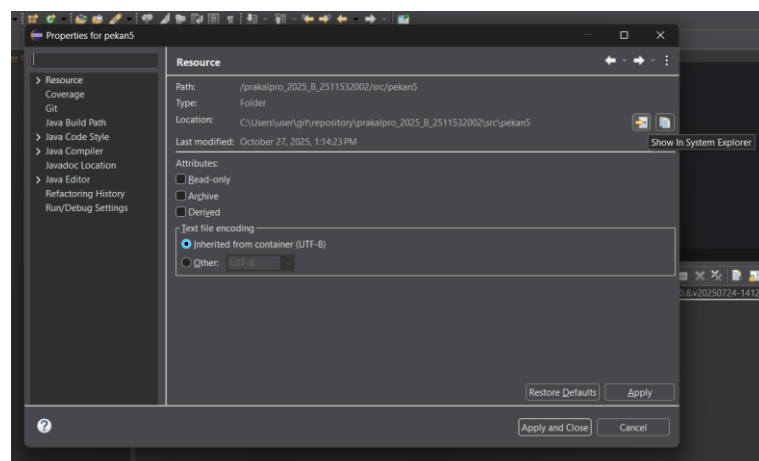
1. Setelah semua program selesai dibuat, selanjutnya adalah *user* perlu untuk memasukkan program yang dibuat di Eclipse ke *repository* GitHub. Selain dengan cara “*Team*” atau “*Commit and Push*”.

Bisa juga dengan cara upload manual. Caranya klik kanan pada *package*, lalu pilih *properties*.



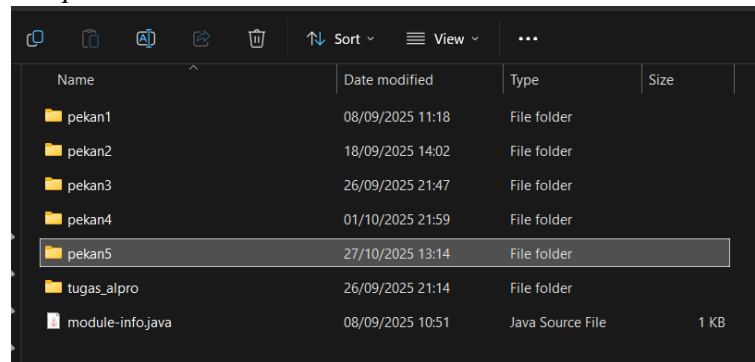
2. 20 Upload Package ke GitHub

2. Maka akan muncul *pop up* menu “*Properteis for Pekan5*”. Lalu klik icon “*Show in System Folder*”.



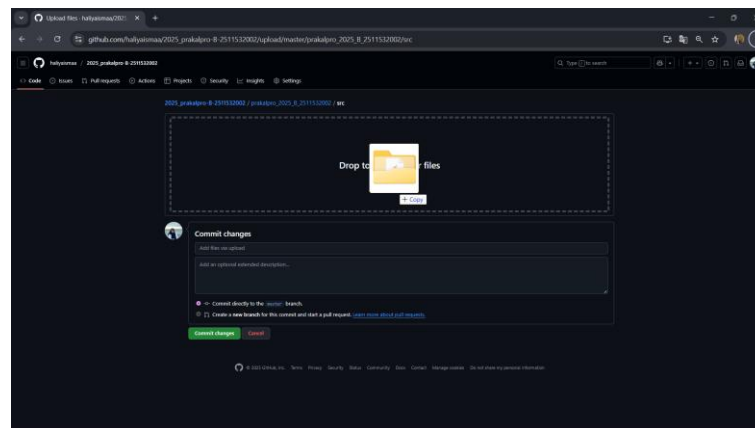
2. 21 Upload Package ke GitHub

3. Maka akan menunjukan lokasi file “pekan5” pada *File Explorer computer*.



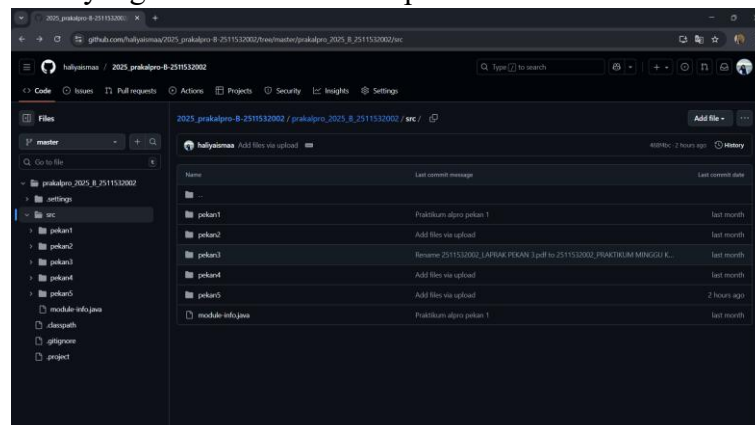
2. 22 Upload Package ke GitHub

4. Setelah itu *drag and drop* file “pekan5” ke GitHub. Lalu tambahkan keterangan jika perlu. Setelah itu klik *upload*.



2. 23 Upload Package ke GitHub

Maka dapat dilihat pada *repository* GitHub bahwa file “pekan5” telah terunggah *via Upload*. Maka akan *muncul file package* dan *class* yang telah di buat di Eclipse.



2. 24 Upload Package ke GitHub

BAB III

KESIMPULAN

3.1 Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan dengan “Implementasi Perulangan *For* dan *Nested For*” pada bahasa pemrograman *Java*. Dapat disimpulkan bahwa perulangan merupakan salah satu konsep dasar penting dalam pemrograman yang berfungsi untuk mengeksekusi perintah secara berulang sesuai kondisi tertentu. Selain itu dapat diketahui bahwa:

1. Pengulangan *for* digunakan ketika jumlah pengulangan sudah diketahui, dimana strukturnya terdiri dari inisialisasi, kondisi, dan *increment/decrement*.
2. Dengan praktikum perulangan *for* dan *nested for*, mahasiswa dapat memahami dan meningkatkan kemampuan logika perulangan serta pentingnya efisiensi dalam penulisan kode program untuk menyelesaikan suatu permasalahan.

3.2 Saran

Saran untuk pelaksanaan praktikum kedepannya adalah diharapkan agar mahasiswa lebih teliti dalam memahami dan menulis *syntax* agar meminimalisir kesalahan penulisan kode dan eror, terutama dalam perulangan *for* yang berisiko terjadi *infinite loop* jika tidak menset kondisi dan *increment/decrement* dengan tepat.

DAFTAR PUSTAKA

- [1] W3Schools, “Java For Loop,” dokumen daring. Tersedia pada: https://www.w3schools.com/java/java_for_loop.asp [Diakses: 28 Oktober 2025].
- [2] Petanikode, “Perulangan di Java: For, While, dan Do While,” dokumen daring. Tersedia pada: <https://www.petanikode.com/java-perulangan/> [Diakses: 28 Oktober 2025].
- [3] GeeksforGeeks, “Loops in Java,” dokumen daring. Tersedia pada: https://www.geeksforgeeks.org/java/loops-in-java/?utm_source=chatgpt.com [Diakses: 28 Oktober 2025].