

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
PERULANGAN FOR DAN NASTED FOR

Disusun Oleh:

Khairun Nisa

2511532015

Dosen Pengampu:

DR. WAHYUDI, S.T, M.T

Asisten Pratikum:

Aufan Taufiqurrahman



FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas terselesaikannya laporan praktikum Algoritma dan Pemrograman ini. Laporan ini disusun sebagai dokumentasi hasil kegiatan praktikum yang telah dilaksanakan pada 30 Oktober 2025, dengan pembahasan mengenai perulangan for dan nested for.

Penulis mengucapkan terima kasih kepada Bapak Dosen serta asisten laboratorium yang telah membimbing selama praktikum berlangsung. Semoga laporan ini dapat bermanfaat bagi pembaca dan menjadi referensi untuk perkembangan ilmu pengetahuan di bidang pemrograman.

Padang, 30 September 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Pratikum	1
1.3 Manfaat Pratikum	2
BAB II 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 Nasted For 0.....	8
2.6 Nasted For 1	10
2.7 Nasted For 3	11
BAB III KESIMPULAN	13
3.1 Kesimpulan	13
DAFTAR PUSTAKA	14

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pengembangan perangkat lunak, kemampuan untuk mengotomatisasi tugas berulang merupakan salah satu prinsip dasar pemrograman. Bahasa java, sebagai salah satu Bahasa pemrograman berorientasi objek yang paling populer, menyediakan berbagai struktur kendali alur (*control flow*) untuk mendukung hal tersebut, salah satunya adalah dengan penggunaan perulangan for.

Perulangan merupakan suatu proses dimana kita dapat mengulang satu atau beberapa perintah atau eksekusi blok kode tanpa hentu, selama kondisi yang dijadikan acuan terpenuhi. Biasanya disiapkan variabel untuk iterasi atau variabel penanda kapan perulangan akan diberhentikan [1]. Perulangan for memungkinkan programmer mengeksekusi blok kode berulang kali tanpa harus menulis instruksi yang sama secara manual. Hal ini tidak hanya meningkatkan efisiensi penulisan kode, tetapi juga meminimalkan risiko kesalahan dan mempermudah pemeliharaan program.

Pada perulangan for terdapat 3 komponen utama Perulangan for:

- Start
Merupakan kondisi pada awal perulangan. Biasanya kondisi awal ini berisi perintah untuk memberikan nilai kepada variabel counter. Variabel counter sendiri adalah sebuah variabel yang menentukan berapa banyak perulangan dilakukan. Kebanyakan programmer menggunakan variabel *i* sebagai counter, namun bisa saja menggunakan variabel yang lain.
- Condition
Merupakan ekspresi Boolean yang dievaluasi sebelum setiap iterasi. Selama kondisi bernilai *true*, blok perulangan akan terus dijalankan. Jika kondisi bernilai *false*, perulangan dihentikan dan program melanjutkan ke baris berikutnya di luar loop.
- Increment

Merupakan perintah yang dieksekusi setelah setiap iterasi selesai. Fungsinya untuk mengubah nilai variabel pencacah (misalnya menaikkan dengan `i++` atau menurunkan dengan `i--`) agar suatu saat kondisi menjadi false dan perulangan berhenti.

Dalam konteks pembelajaran pemrograman dasar, penguasaan `for` menjadi fondasi penting untuk memahami konsep seperti pembuatan pola, perhitungan matematis berulang, hingga algoritma sederhana.

1.2 Tujuan Pratikum

1. Memahami konsep dan mekanisme kerja berulang `for` dalam Bahasa pemrograman java.
2. Mampu menerapkan sintaks `for` untuk menyelesaikan berbagai kasus pemrograman sederhana, seperti menampilkan deret angka, menghitung nilai, dan membuat pola karakter.
3. Melatih kemampuan berpikir logis dan algoritmik melalui penggunaan variabel pencacah, kondisi iterasi, dan pengendalian alur program.

1.3 Manfaat Pratikum

1. Memberikan pengalaman langsung dalam menulis, menjalankan, dan memperbaiki kode program yang menggunakan perulangan `for`, sehingga memperkuat pemahaman teoritis.
2. Membantu mahasiswa mengembangkan keterampilan dasar dalam mengotomatisasi tugas berulang, yang merupakan salah satu prinsip utama dalam pemrograman efisien.
3. Menjadi fondasi penting untuk mempelajari konsep lanjutan seperti nested loop, algoritma iterative, dan sebagainya.
4. Meningkatkan kepercayaan diri dalam menulis kode yang terstruktur, rapi, dan mudah dilakukan, dimana keterampilan yang sangat dibutuhkan dalam pengembangan perangkat lunak di dunia nyata.

BAB II

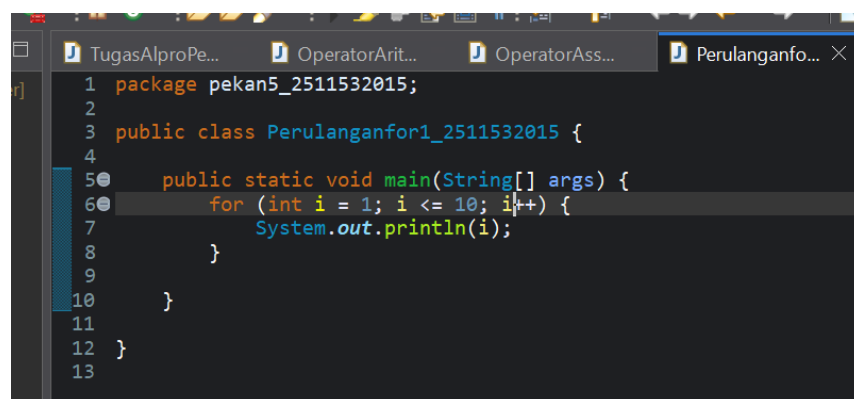
PEMBAHASAN

2.1 Perulangan For 1

Perulangan for adalah salah satu struktur kendali dalam pemrograman yang digunakan untuk menjalankan suatu blok kode secara berulang dalam jumlah yang sudah diketahui sebelumnya. Struktur ini sangat efisien karena menggabungkan tiga elemen penting dalam satu baris, yaitu inisialisasi, kondisi, dan update.

Pada praktikum pekan 5 Algoritma dan Pemrograman, telah dibuat program sederhana menggunakan perulangan for dalam Bahasa Java untuk menampilkan angka dari 1 sampai 10 secara berurutan, masing-masing angka dicetak pada baris baru. Perulangan for dipilih karena jumlah iterasi sudah diketahui dengan pasti, yaitu sebanyak 10 kali.

1. Sebelum mengetikkan kode program, terlebih dahulu membuat package baru dengan cara yang sama seperti praktikum minggu-minggu sebelumnya dan diberi nama 'pekan5'.
2. Di dalam package pekan 5, buat class baru dengan nama 'perulanganFor1'
3. Kemudian, ketikkan kode seperti berikut:



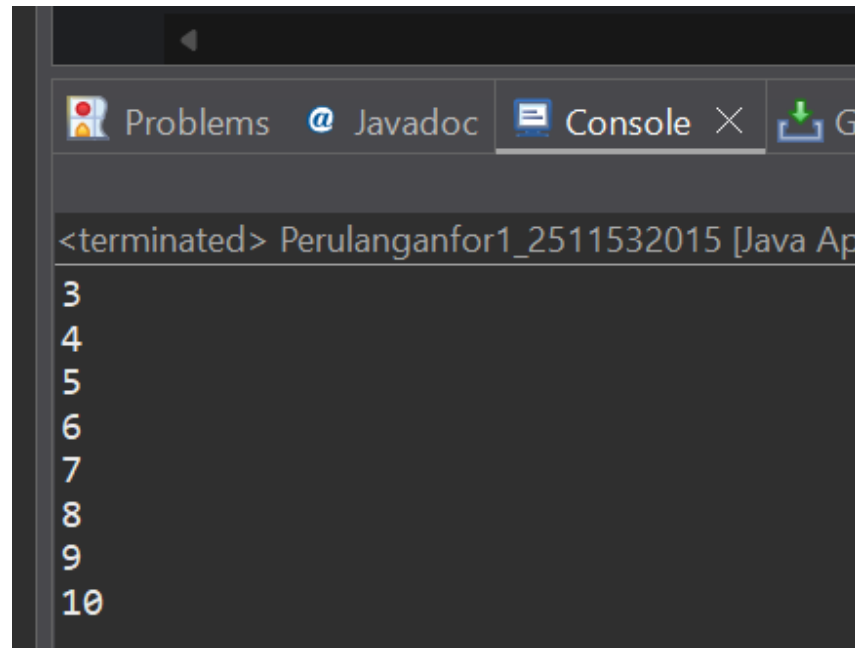
```
1 package pekan5_2511532015;
2
3 public class Perulanganfor1_2511532015 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 10; i++) {
7             System.out.println(i);
8         }
9     }
10 }
11
12 }
13 }
```

Gambar 2.1: Kode program perulanganFor1

4. Struktur for yang digunakan:
 - Inisialisasi: variable pencacah i diberi nilai awal 1 (int i=1)

- Kondisi: perulangan berlangsung selama i kurang dari atau sama dengan 10 ($i \leq 10$)
- Update: setiap akhir iterasi, nilai i dinaikkan 1 ($i++$)

Dengan struktur ini, program dapat mencetak 1, 2, 3, 4, ..., 10, masing-masing pada baris terpisah, tanpa perlu menulis perintah `System.out.println()` sebanyak 10 kali secara manual.



```
<terminated> Perulanganfor1_2511532015 [Java Ap
3
4
5
6
7
8
9
10
```

Gambar 2.2: Output program perulanganFor1

5. Setelah program dijalankan dengan me-run kan kode program, output yang dihasilkan sesuai, yaitu angka 1 sampai 10 muncul secara berurutan, masing-masing pada baris baru. Hal ini menunjukkan perulangan for berhasil mengeksekusi blok perintah `System.out.println(i)` sebanyak 10 kali, dengan nilai i yang berubah dari 1 sampai 10 pada setiap iterasi.

2.2 Perulangan For 2

Program ini menggunakan perulangan for untuk mencetak angka dari 1 sampai 10 secara berurutan dalam satu baris.

1. Sebelum mengetikkan kode program, terlebih dahulu membuat class baru di package pekan 5, dan diberi nama 'perulanganFor2

2. Kemudian ketikkan kode program seperti berikut:

```
1 package pekan5_2511532015;
2
3 public class Perulanganfor2_2511532015 {
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 2.3: Kode program perulanganFor2

3. Setelah dijalankan akan menghasilkan output:

```
<terminated> Perulanganfor2_2511532015 [Java Application]
1 2 3 4 5 6 7 8 9 10
```

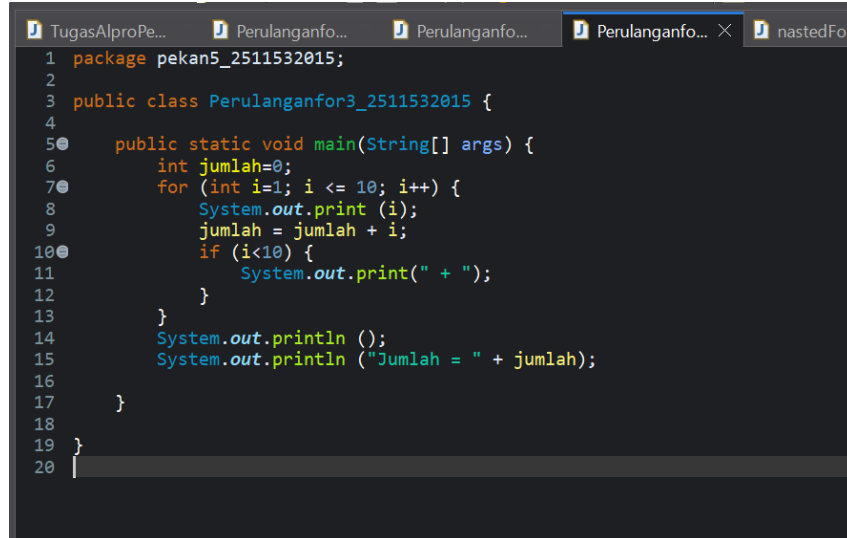
Gambar 2.4: Output program perulanganFor2

Setiap kali angka dicetak, program menambahkan satu spasi di belakangnya dengan perintah `System.out.print(i+ " ")`. Karena menggunakan `print ()`, semua angka tidak dibuat di baris baru, melainkan langsung disambung ke kanan.

2.3 Perulangan For 3

Program ini menggunakan `perulanganfor` untuk menampilkan deret penjumlahan angka dari 1 sampai 10 sekaligus menghitung total hasil penjumlahannya.

1. Sebelum mengetikkan program terlebih dahulu membuat class baru di package pekan 5, dan diberi nama 'perulanganFor3'
2. Ketikkan kode program seperti berikut:



```

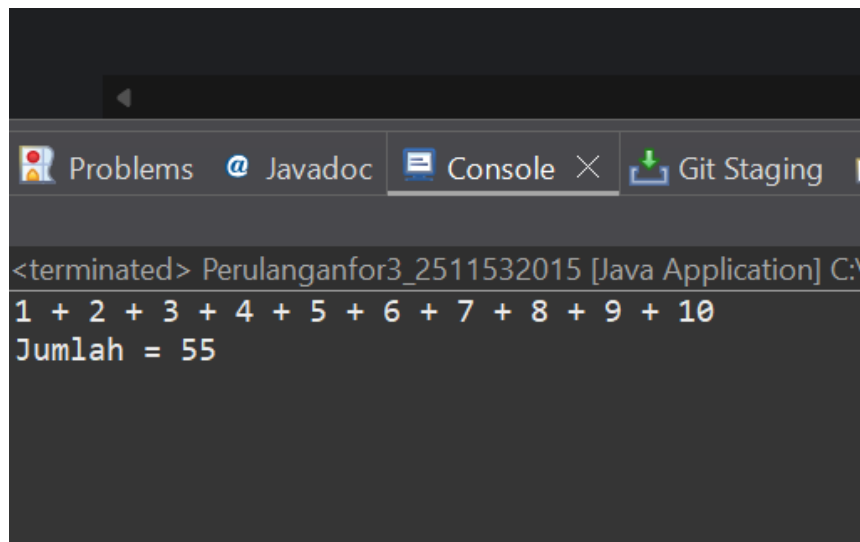
1 package pekan5_2511532015;
2
3 public class Perulanganfor3_2511532015 {
4
5     public static void main(String[] args) {
6         int jumlah=0;
7         for (int i=1; i <= 10; i++) {
8             System.out.print(i);
9             jumlah = jumlah + i;
10            if (i<10) {
11                System.out.print(" + ");
12            }
13        }
14        System.out.println ();
15        System.out.println ("Jumlah = " + jumlah);
16    }
17 }
18
19 }
20

```

Gambar 2.5: Kode program perulangFor3

Di awal, variabel jumlah diberi nilai 0 sebagai wadah akumulasi. Perulangan dimulai dari $i=1$ dan berjalan selama i kurang atau sama dengan 10, dengan i bertambah satu setiap iterasi ($i++$). Pada setiap putaran, angka i dicetak menggunakan `System.out.print(i)`, lalu jika i masih kurang dari 10, program menambahkan tanda “+” agar tampilan output membentuk ekspresi matematika yang utuh. Setelah perulangan selesai, program mencetak baris baru dan menampilkan hasil total penjumlahan dari 1-10.

3. Saat dijalankan program akan menampilkan output:



```

<terminated> Perulanganfor3_2511532015 [Java Application] C:\
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 55

```

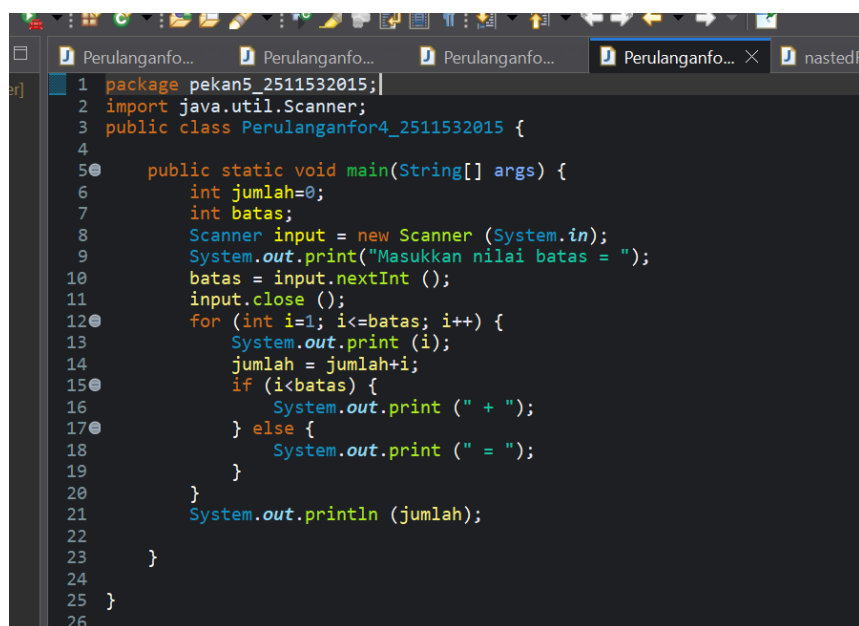
Gambar 2.6: Output program perulanganFor3

Program ini tidak hanya mencetak angka, tapi juga menghitung totalnya sambil menampilkan proses penjumlahan secara visual.

2.4 Perulangan For 4

Program ini bisa menampilkan deret penjumlahan angka dari 1 hingga suatu nilai batas, di mana nilai batas tersebut tidak ditentukan secara tetap dalam kode, melainkan diinputkan langsung oleh pengguna saat program dijalankan. Untuk memungkinkan interaksi ini, program menggunakan kelas Scanner, yang berfungsi membaca input dari keyboard. Dengan Scanner, pengguna bebas memasukkan angka berapapun sebagai batas akhir perulangan. Perulangan for kemudian akan menyesuaikan jumlah iterasinya sesuai input tersebut, sekaligus menghitung dan menampilkan proses penjumlahan secara visual.

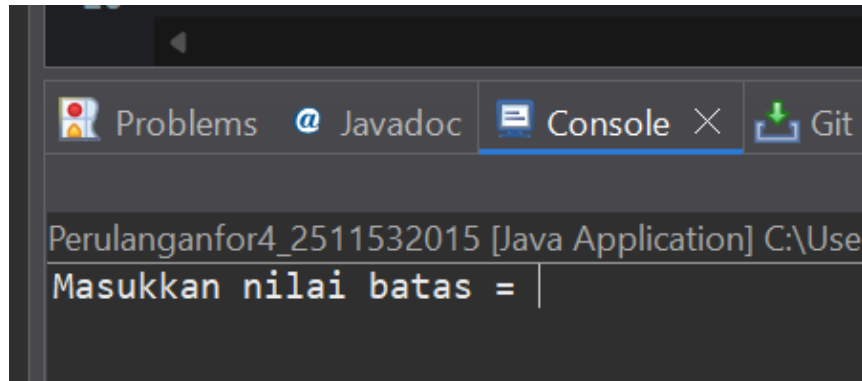
1. Sebelum mengetikkan kode program, terlebih dahulu membuat class baru di package pekan 5, dan diberi nama 'perulanganFor4'.
2. Kemudian, ketikka kode program seperti berikut:

A screenshot of a Java IDE window titled 'Perulanganfor4_2511532015'. The code is as follows:

```
1 package pekan5_2511532015;
2 import java.util.Scanner;
3 public class Perulanganfor4_2511532015 {
4
5     public static void main(String[] args) {
6         int jumlah=0;
7         int batas;
8         Scanner input = new Scanner (System.in);
9         System.out.print("Masukkan nilai batas = ");
10        batas = input.nextInt ();
11        input.close ();
12        for (int i=1; i<=batas; i++) {
13            System.out.print (i);
14            jumlah = jumlah+i;
15            if (i<batas) {
16                System.out.print (" + ");
17            } else {
18                System.out.print (" = ");
19            }
20        }
21        System.out.println (jumlah);
22    }
23 }
24
25 }
26
```

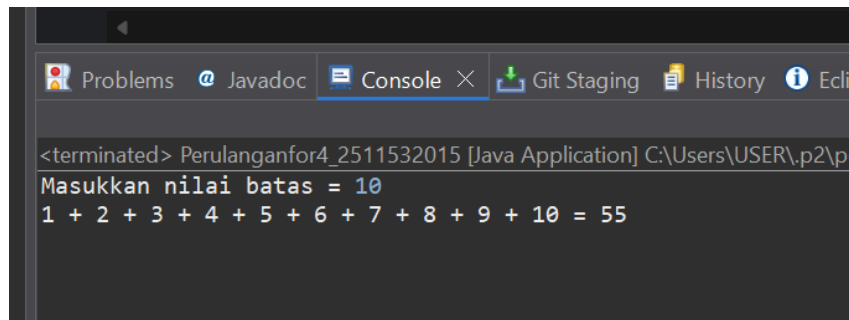
Gambar 2.7: Kode program perulanganFor4

3. Setelah program dijalankan, pengguna diminta memasukkan nilai batas.



Gambar 2.8: Output menentukan batas

4. Setelah pengguna memasukkan nilai batas, misal 10, maka perulangan berjalan tepat sebanyak nilai batas, yaitu 10 dan menghitung dan menampilkan proses penjumlahan.

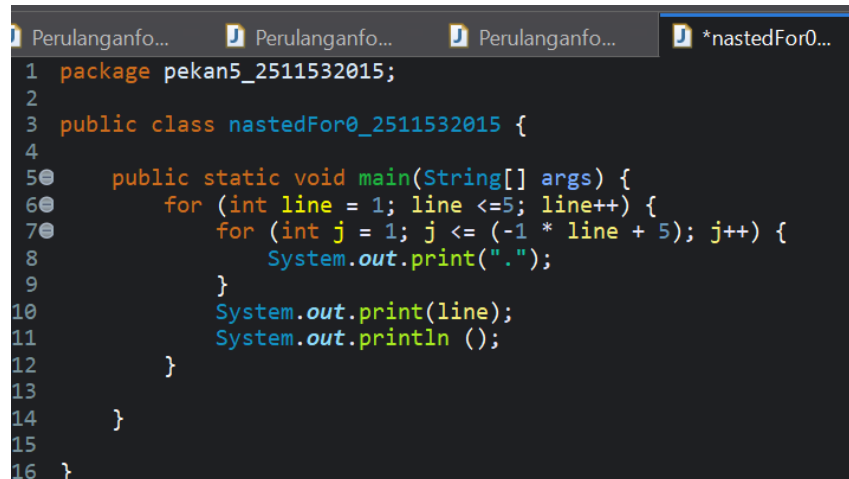


Gambar 2.9: Output program perulanganFor4

2.5 Nasted for 0

Nasted for atau perulangan for bersarang adalah variasi dari perulangan for, di mana terdapat satu perulangan for di dalam perulangan for lainnya. Perulangan ini disebut perulangan luar dan perulangan dalam, tergantung pada posisinya [2]. Pada program ini menggunakan nasted for loop untuk mencetak karakter titik (.) yang jumlahnya berkurang seiring bertambahnya baris, diikuti oleh angka yang menunjukkan nomor baris.

1. Sebelum mengetikkan kode program, terlebih dahulu membuat class baru di package pekan 5 dengan nama 'nastedFor0'.
2. Kemudian ketikkan kode programnya



```

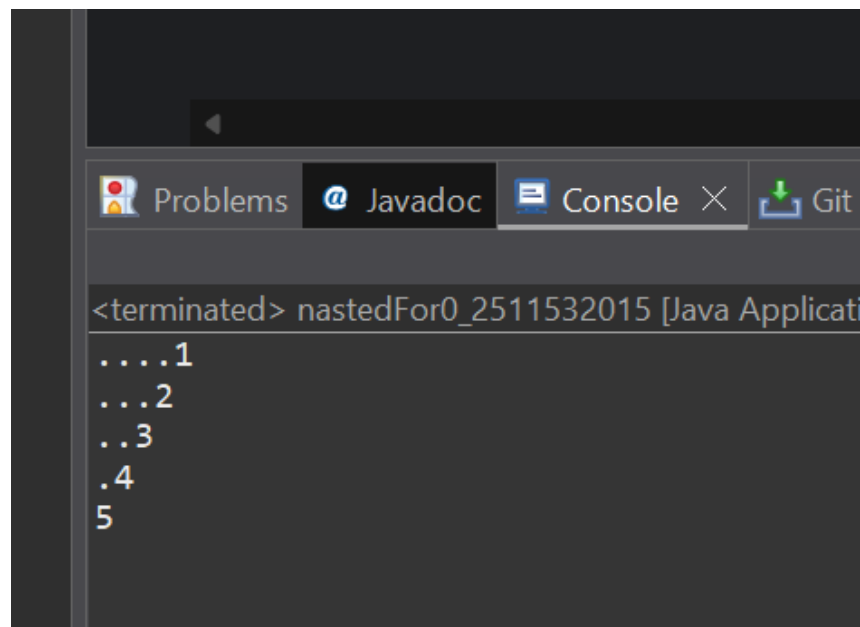
1 package pekan5_2511532015;
2
3 public class nastedFor0_2511532015 {
4
5     public static void main(String[] args) {
6         for (int line = 1; line <=5; line++) {
7             for (int j = 1; j <= (-1 * line + 5); j++) {
8                 System.out.print(".");
9             }
10            System.out.print(line);
11            System.out.println ();
12        }
13    }
14 }
15
16 }

```

Gambar 2.10: Kode program nastedFor0

Struktur utamanya terdiri dari dua perulangan, for pertama untuk mengontrol jumlah baris (variabel line), dan for kedua untuk mengontrol jumlah baris (variabel j). perulangan luar dimulai dari line=1 hingga line<=5, artinya program akan mencetak 5 baris. Di dalamnya, perulangan dalam menggunakan rumus $(-1 * \text{line} + 5)$ sebagai batas atas, ini membuat jumlah titik pada setiap baris semakin kebawah semakin berkurang.

3. Hasil output program setelah dijalankan



```

<terminated> nastedFor0_2511532015 [Java Applicat
....1
...2
..3
.4
5

```

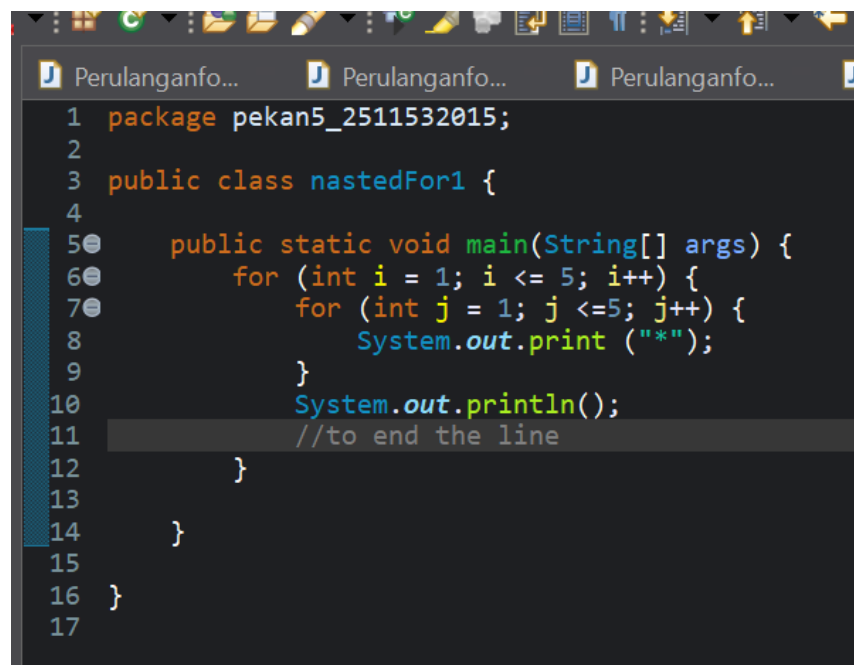
Gambar 2.11: Output program nastedFor0

Pola ini menunjukkan bagaimana perulangan bersarang dapat digunakan untuk membuat desain visual sederhana dengan logika matematis yang tepat.

2.6 Nasted for 1

Program ini menggunakan perulangan bersarang for untuk mencetak pola Bintang (*) dalam bentuk persegi dan kotak.

1. Sebelum mengetikkan kode program, terlebih dahulu membuat class baru di package pekan 5 dengan nama 'nastedFor1'.
2. Kemudian ketikkan kode program

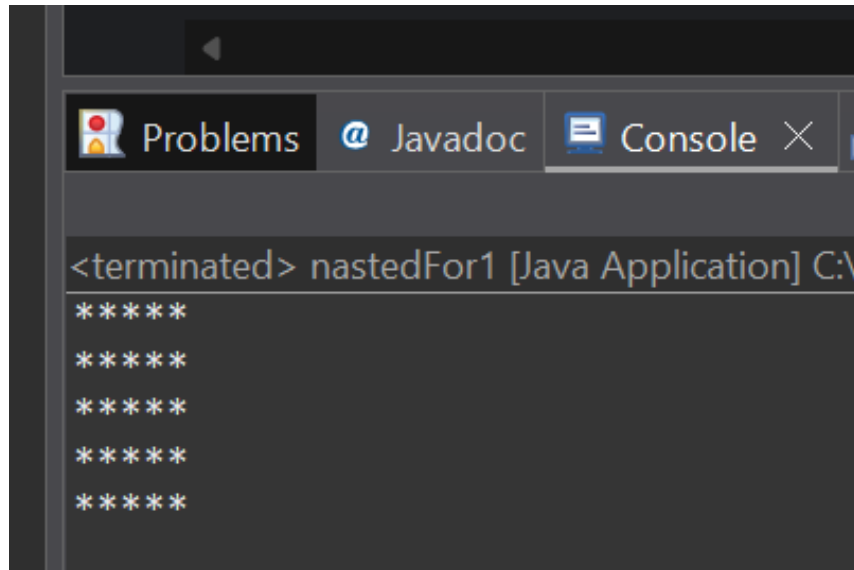


```
1 package pekan5_2511532015;
2
3 public class nastedFor1 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 5; i++) {
7             for (int j = 1; j <= 5; j++) {
8                 System.out.print ("*");
9             }
10            System.out.println();
11            //to end the line
12        }
13    }
14 }
15
16 }
17
```

Gambar 2.12: Kode program nastedFor1

Dengan dua perulangan for, program mengontrol jumlah baris dan jumlah Bintang per baris secara terpisah. Perulangan luar (i) menentukan berapa banyak baris yang akan dicetak, sedangkan perulangan (j) menentukan berapa banyak Bintang yang akan dicetak di setiap baris. Setelah selesai mencetak semua Bintang di satu baris, System.out.println() berfungsi untuk pindah ke baris baru.

3. Output yang dihasilkan Ketika program dijalankan



Gambar 2.13: Output program nastedFor1

2.7 Nasted For 2

Program ini menggunakan perulangan bersarang untuk mencetak pola angka dalam bentuk table atau matriks.

1. Sebelum mengetikkan kode program terlebih dahulu membuat class baru di package pekan 5 dengan nama 'nastedFor2'.
2. Kemudian ketikkan kode program

```

1 package pekan5_2511532015;
2
3 public class nastedFor2 {
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

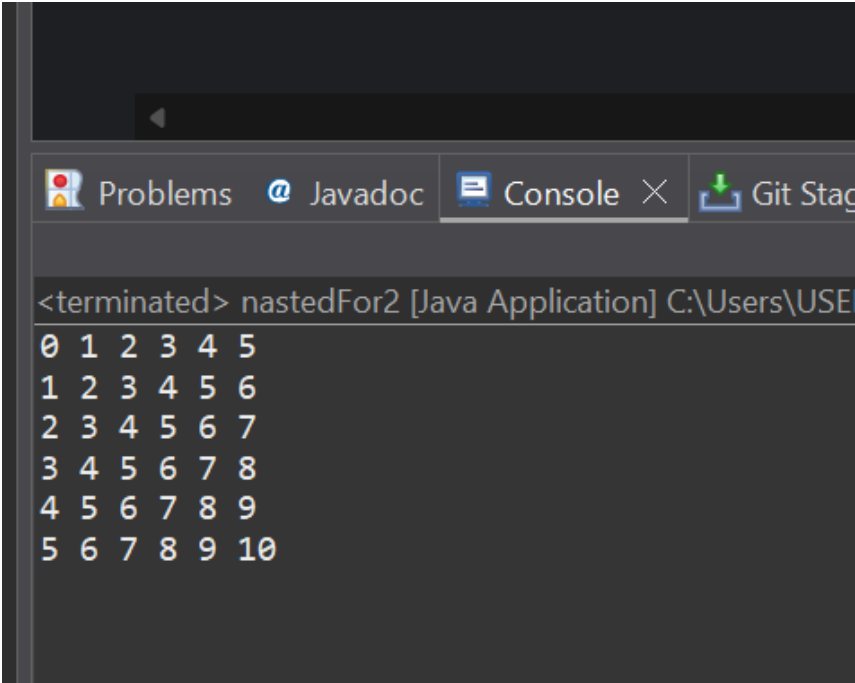
```

Gambar 2.14: Kode program nastedFor2

Program dimulai dengan perulangan luar yang mengontrol jumlah baris, dimana variabel i berjalan dari 0 hingga 5, artinya program akan mencetak 6 baris. Di dalamnya, perulangan for dalam (j) mengontrol

jumlah kolom. Setiap sel dalam tabel akan berisi hasil penjumlahan indeks baris dan kolom.

3. Output yang dihasilkan jika dijalankan



```
<terminated> nestedFor2 [Java Application] C:\Users\USE
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 2.15: Output program nestedFor2

Pola ini menunjukkan bahwa program berhasil mengimplementasikan konsep tabel penjumlahan dua dimensi dengan perulangan bersarang.

BAB III

KESIMPULAN

3.1 Kesimpulan

Perulangan for, baik Tunggal maupun berulang merupakan fondasi penting dalam pemrograman untuk mengotomatisasi proses penting. Dari sekedar mencetak angka 1 sampai 10, menghitung total deret, hingga membuat pola visual dan tabel penjumlahan, struktur for menunjukkan fleksibilitasnya dalam menggabungkan kontrol alur, dan operasi matematika. Pemahaman terhadap inisialisasi, kondisi, dan update dalam for, serta kemampuan mengatur hubungan antara loop pada nested for, menjadi kunci untuk menyelesaikan berbagai masalah pemrograman secara efisien dan terstruktur.

DAFTAR PUSTAKA

- [1] T. Suryana, "Perulangan - Looping," *Repository Unikom*, 2019.
- [2] "Perulangan For Bersarang dan Perulangan For yang Disempurnakan di Java," GroTechMinds, 24 Juni 2024. [Online]. Available: <https://grotechminds.com/what-is-nested-amp-enhanced-for-loops-in-java/>. [Accessed 31 October 2025].