

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

“PERULANGAN FOR”

disusun oleh:

Dennis Shauqi Akbar

2511533020

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

Asisten Pratikum: Rahmad DRO



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan tugas kelompok ini dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, keluarga, sahabat, serta para pengikutnya hingga akhir zaman.

Tugas Individu ini kami susun dengan tujuan untuk memenuhi salah satu tugas mata kuliah Praktikum Algoritma dan Pemrograman, dengan Judul “Perulangan For” yang mencakup PerulanganFor1, PerulanganFor2, PerulanganFor3, PerulanganFor4, NestedFor0, NestedFor1, NestedFor2

Kami menyadari bahwa penyusunan tugas ini masih jauh dari sempurna, baik dari segi materi maupun penyajiannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari semua pihak demi kesempurnaan tugas kami di masa mendatang.

Akhir kata, kami mengucapkan terima kasih kepada dosen pembimbing yang telah memberikan arahan sehingga tugas ini dapat terselesaikan tepat waktu. Semoga tugas ini dapat bermanfaat bagi kami khususnya, dan bagi pembaca pada umumnya.

Padang, 01 November 2025

Dennis Shauqi Akbar

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan.....	1
1.3 Manfaat Praktikum.....	2
BAB II PEMBAHASAN	2
2.1 Program PerulanganFor1	2
2.2 Program PerulanganFor2	4
2.3 Program PerulanganFor3	6
2.4 Program PerulanganFor4	7
2.5 Program NestedFor0	9
2.6 Program NestedFor1	11
2.6 Program NestedFor2	12
BAB III KESIMPULAN.....	14
3.1 Kesimpulan	14
3.2 Saran Pengembangan	14
DAFTAR PUSTAKA	15

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, perulangan (loop) merupakan salah satu struktur penting yang berfungsi untuk mengeksekusi perintah secara berulang selama kondisi tertentu masih terpenuhi. Salah satu bentuk perulangan yang paling umum digunakan adalah for loop, karena memungkinkan pelaksanaan pengulangan dengan jumlah langkah yang telah ditentukan sebelumnya.

Melalui kegiatan praktikum ini, mahasiswa mempelajari berbagai variasi penggunaan perulangan for, meliputi PerulanganFor1 sampai PerulanganFor4, serta nested for (perulangan bersarang) seperti NestedFor0 hingga NestedFor2. Setiap bagian membantu mahasiswa memahami cara kerja logika perulangan, mulai dari konsep dasar hingga penerapan yang lebih kompleks.

Penguasaan konsep ini memiliki peran penting dalam pemrograman, terutama dalam pembuatan pola, tabel, maupun pengolahan data. Dengan memahami dan menerapkan perulangan for dengan baik, mahasiswa dapat menulis program yang lebih efisien serta menyusun algoritma yang logis dan terstruktur untuk menyelesaikan berbagai permasalahan dalam komputasi.

1.2 Tujuan

1. Memahami konsep dasar struktur perulangan, khususnya perulangan for dan nested for, dalam pemrograman.
2. Mampu menerapkan perulangan for untuk menyelesaikan berbagai permasalahan yang memerlukan proses berulang.
3. Melatih kemampuan logika dan berpikir sistematis dalam menyusun algoritma menggunakan struktur perulangan.
4. Mengembangkan keterampilan menulis kode program yang efisien, rapi, dan mudah dipahami.

5. Mengetahui perbedaan fungsi dan penerapan antara perulangan tunggal dan perulangan bersarang (nested loop).

1.3 Manfaat Praktikum

1. Mahasiswa dapat memahami cara kerja dan penerapan for loop dalam pemrograman nyata.
2. Meningkatkan kemampuan analisis dan logika berpikir dalam merancang solusi menggunakan algoritma perulangan.
3. Membantu mahasiswa menulis program yang lebih efektif dengan mengurangi pengulangan kode secara manual.
4. Menjadi dasar bagi pemahaman materi pemrograman yang lebih lanjut, seperti pengolahan data, pola visual, dan struktur data dua dimensi.
5. Membentuk kebiasaan berpikir logis, terstruktur, dan efisien dalam penyelesaian masalah komputasi.

BAB II

PEMBAHASAN

2.1 Program PerulanganFor1

```
1 package pekan5;
2
3 public class PerulanganFor1 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         for (int i = 5; i <=5; i-- ) {
9             System.out.println(i);
10        }
11    }
12
13 }
14
```

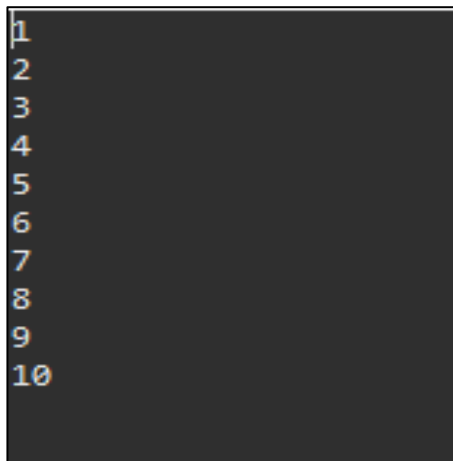
Kode Program 2.1

Program di atas merupakan contoh penerapan struktur perulangan for dalam bahasa pemrograman Java yang ditempatkan di dalam kelas PerulanganFor1. Program ini memiliki tujuan untuk menampilkan nilai variabel *i* secara berulang menggunakan perintah `System.out.println(i);`. Variabel *i* diinisialisasi dengan nilai awal 5, dan pada setiap iterasi nilainya akan dikurangi satu (*i--*). Struktur ini memungkinkan program mengeksekusi blok perintah berulang kali selama kondisi yang ditentukan masih terpenuhi.

Pada kode tersebut, kondisi perulangan yang digunakan adalah *i* <= 5. Karena nilai awal *i* adalah 5 dan setiap iterasi nilainya semakin berkurang, maka kondisi tersebut akan selalu benar. Akibatnya, program akan terus mencetak nilai *i* tanpa henti mulai dari 5, 4, 3, 2, 1, 0, -1, -2, dan seterusnya. Situasi ini disebut infinite loop atau perulangan tak terbatas, karena tidak ada kondisi yang dapat menghentikan proses perulangan secara otomatis.

Agar program berjalan sesuai tujuan, yaitu mencetak angka dari 5 hingga 1 secara menurun, maka kondisi pada perulangan harus disesuaikan menjadi $i \geq 1$. Dengan perubahan tersebut, perulangan akan berhenti ketika nilai i sudah tidak lebih besar atau sama dengan 1. Hasil akhirnya, program akan mencetak angka 5, 4, 3, 2, dan 1 secara berurutan, lalu berhenti karena kondisi perulangan sudah tidak terpenuhi.

Output dari program diatas jika dijalankan adalah



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

Gambar 2.1

2.2 Program PerulanganFor2

```
1 package pekan5;
2
3 public class PerulanganFor2 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for(int i = 5; i >=1; i--) {
8
9             for(int j=2;j<=i;j++) {
10                 System.out.print("_");
11             }
12             for(int j=5;j>=i;j--) {
13                 System.out.print("* ");
14             }
15             System.out.println();
16         }
17     }
18 }
19
20 }
```

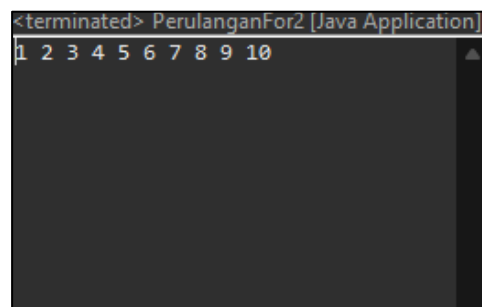
Kode Program 2.2

Program di atas merupakan contoh penerapan nested loop atau perulangan bersarang dalam bahasa pemrograman Java. Program ini menggunakan dua perulangan for di dalam satu perulangan utama. Perulangan luar menggunakan variabel *i* yang bernilai dari 5 hingga 1 (*i--*). Setiap iterasi dari *i* akan menjalankan dua perulangan dalam yang bertanggung jawab untuk mencetak simbol berbeda pada baris yang sama, sebelum akhirnya pindah ke baris berikutnya dengan `System.out.println()`.

Perulangan pertama di dalamnya menggunakan variabel *j* dan berfungsi mencetak tanda garis bawah (`_`) sebanyak nilai *i*. Perulangan ini dimulai dari $j = 2 * i$ (yang tampaknya keliru karena akan menyebabkan hasil tidak terduga) hingga $j \leq i$, dan setiap iterasi menampilkan satu karakter garis bawah tanpa berpindah baris. Setelah itu, perulangan kedua menggunakan variabel *j* juga, namun dengan arah menurun (*j--*), dan mencetak tanda bintang (*). Karena `System.out.print()` digunakan tanpa `ln`, hasilnya akan tersusun dalam satu baris.

Secara keseluruhan, program ini dirancang untuk menampilkan pola gabungan antara karakter garis bawah dan bintang dengan jumlah tertentu pada setiap baris, tergantung nilai variabel *i*. Setelah kedua perulangan dalam selesai, perintah `System.out.println()` dijalankan untuk memindahkan kursor ke baris berikutnya sehingga baris baru bisa dicetak pada iterasi selanjutnya. Namun, karena terdapat kesalahan logika pada inisialisasi $j = 2 * i$, program mungkin tidak menghasilkan output sesuai harapan.

Output dari program diatas jika kita jalankan adalah sebagai berikut:



Gambar 2.2

2.3 Program PerulanganFor3

```

1 package pekan5;
2
3 public class PerulanganFor3 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int jumlah = 0 ;
8         for(int i=1; i <=10; i++) {
9             System.out.print(i);
10            jumlah= jumlah+i;
11            if (i<10) {
12                System.out.print(" + ");
13            }
14        }
15        System.out.println();
16        System.out.println("Jumlah = "+jumlah);
17    }
18 }
19
20
21
22
23

```

Kode Program 2.3

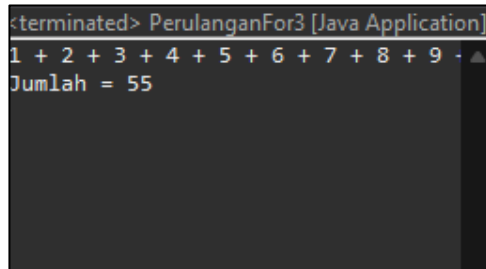
Program di atas merupakan implementasi dari perulangan for dalam Java untuk menampilkan deretan angka dari 1 hingga 10, sekaligus menghitung total penjumlahan dari angka-angka tersebut. Pada awal program, variabel jumlah diinisialisasi dengan nilai 0 sebagai tempat penyimpanan hasil penjumlahan. Kemudian, perulangan for digunakan dengan variabel penghitung i yang dimulai dari 1 dan terus bertambah hingga mencapai 10.

Di dalam blok perulangan, perintah `System.out.print(i);` digunakan untuk menampilkan nilai i tanpa pindah baris. Setelah setiap angka dicetak, nilai i ditambahkan ke variabel jumlah melalui pernyataan `jumlah = jumlah + i;`. Terdapat juga kondisi `if (i < 10)` yang berfungsi untuk menambahkan tanda “+” di antara angka, sehingga tampilan output menjadi lebih rapi seperti `1 + 2 + 3 + ... + 10`. Setelah perulangan selesai, program keluar dari loop dan menampilkan hasil total penjumlahan tersebut.

Hasil akhirnya, program menampilkan deretan angka dari 1 sampai 10 yang dipisahkan oleh tanda tambah, kemudian pada baris berikutnya menampilkan total

penjumlahan dengan format Jumlah = 55. Dengan demikian, program ini tidak hanya memperlihatkan konsep dasar looping, tetapi juga penerapan operasi aritmetika dan penggunaan kondisi sederhana di dalam perulangan. sudah memiliki sim,

Jika Programnya dijalankan maka outputnya adalah sebagai berikut:



Gambar 2.3

2.4 Program PerulanganFor4

```

1 package pekan5;
2
3 import java.util.Scanner;
4
5 public class PerulanganFor4 {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         int jumlah=0;
10        int batas;
11        Scanner input= new Scanner(System.in);
12        System.out.print("Masukkan nilai batas =");
13        batas= input.nextInt();
14        input.close();
15        for (int i=1; i<=batas; i++) {
16            System.out.print(i);
17            jumlah = jumlah+i;
18            if (i<batas) {
19                System.out.print(" + ");
20            } else {
21                System.out.print(" = ");
22            }
23        }
24        System.out.println(jumlah);
25    }
26 }
27 }
28

```

Kode Program 2.4

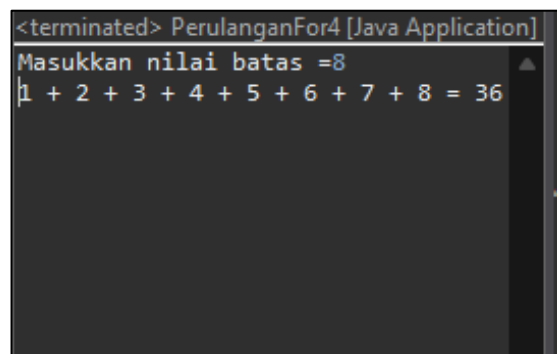
Program di atas merupakan contoh penerapan perulangan for yang dikombinasikan dengan input dari pengguna menggunakan kelas Scanner dalam Java. Pada awal program, dua variabel utama yaitu jumlah dan batas dideklarasikan, di mana jumlah berfungsi untuk menyimpan hasil penjumlahan, sedangkan batas digunakan untuk

menampung nilai yang dimasukkan pengguna sebagai penentu akhir perulangan. Program kemudian meminta pengguna untuk memasukkan nilai batas, yang kemudian dibaca oleh objek Scanner dan disimpan ke dalam variabel batas.

Setelah nilai batas diperoleh, program menjalankan perulangan for dari angka 1 hingga nilai yang dimasukkan pengguna ($i \leq \text{batas}$). Dalam setiap iterasi, nilai i dicetak ke layar dan ditambahkan ke variabel jumlah. Kemudian terdapat kondisi if ($i < \text{batas}$) yang berfungsi menampilkan tanda “+” di antara angka-angka yang dicetak, sedangkan jika iterasi telah mencapai nilai batas, program hanya mencetak spasi agar tampilan hasil tetap rapi.

Setelah seluruh proses perulangan selesai, program mencetak hasil akhir penjumlahan dengan perintah `System.out.println(jumlah);`. Dengan demikian, program ini berfungsi untuk menampilkan deretan angka dari 1 sampai nilai batas yang ditentukan pengguna, disertai dengan tanda tambah di antaranya, serta menghitung total jumlah dari semua angka tersebut. Program ini menggambarkan penggunaan perulangan, percabangan, dan input dinamis secara sederhana namun efektif.

Jika Programnya dijalankan dengan input 8 maka outputnya adalah sebagai berikut



```
<terminated> PerulanganFor4 [Java Application]
Masukkan nilai batas =8
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36
```

Gambar 2.4

2.5 Program NestedFor0

```

1 package pekan5;
2
3 public class NestedFor0 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for(int line = 1; line <=5; line++) {
8             for (int j = 1; j <= (-1 * line + 5 ); j++) {
9                 System.out.print(".");
10            }
11            System.out.print(line);
12            System.out.println();
13        }
14    }
15
16 }
17

```

Kode Program 2.5

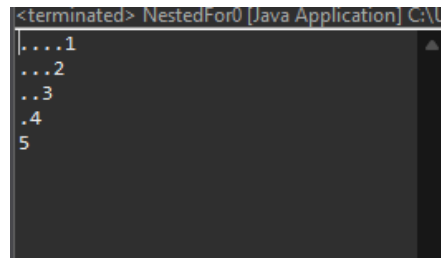
Program di atas merupakan contoh penggunaan nested loop atau perulangan bersarang dalam bahasa Java. Program ini terdiri dari dua perulangan for, di mana perulangan luar dikontrol oleh variabel line dan perulangan dalam dikontrol oleh variabel j. Nilai line dimulai dari 1 hingga 5, dan setiap kali perulangan luar berjalan, perulangan dalam dijalankan untuk mencetak sejumlah karakter titik (.) sesuai dengan ekspresi pada kondisinya. Setelah perulangan dalam selesai, program mencetak nilai line di akhir baris menggunakan `System.out.print(line);`, lalu berpindah ke baris baru dengan `System.out.println();`.

Perulangan bagian dalam memiliki kondisi $j \leq (-1 * \text{line} + 5)$ yang menentukan berapa banyak titik yang akan dicetak pada setiap baris. Kondisi ini menghasilkan pola di mana jumlah titik berkurang setiap kali baris meningkat. Misalnya, pada baris pertama $\text{line} = 1$, maka dicetak empat titik, pada baris kedua tiga titik, dan seterusnya hingga pada baris kelima tidak ada titik yang dicetak. Nilai line kemudian ditampilkan di akhir setiap baris sebagai angka urut yang menunjukkan posisi baris tersebut.

Secara keseluruhan, program ini berfungsi untuk menampilkan pola menurun dari karakter titik (.) diikuti oleh angka barisnya. Hasilnya adalah tampilan yang membentuk pola simetris dari atas ke bawah, dengan jumlah titik berkurang setiap barisnya. Melalui program ini, dapat dipahami bagaimana nested for loop

digunakan untuk membuat pola visual dengan logika matematis pada kondisi perulangannya.

Jika Programnya dijalankan maka outputnya adalah sebagai berikut



```
<terminated> NestedFor0 [Java Application] C:\U
....1
...2
..3
.4
5
```

Gambar 2.5

2.6 Program NestedFor1

```

1 package pekan5;
2
3 public class NestedFor1 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int i = 1; i <=5; i++) {
8             for (int j = 1; j <=5; j++) {
9                 System.out.print("*");
10
11             }
12             System.out.println();
13             // to end the line
14         }
15     }
16 }
17 }
18

```

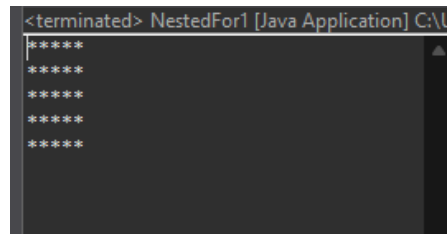
Kode Program 2.6

Kode di atas memperlihatkan penerapan loop bersarang pada Java untuk membuat pola sederhana menggunakan simbol bintang (*). Perulangan luar dikontrol oleh variabel *i* yang menentukan jumlah baris, sedangkan perulangan dalam dikontrol oleh variabel *j* yang menentukan jumlah bintang yang dicetak di setiap baris. Karena kedua perulangan memiliki batas hingga angka 5, hasil akhirnya berupa lima baris dengan lima bintang di setiap baris.

Ketika perulangan luar berjalan satu kali, perulangan dalam dijalankan penuh untuk mencetak lima bintang secara berurutan tanpa pindah baris. Setelah itu, perintah `System.out.println();` dijalankan untuk membuat baris baru, sehingga proses pencetakan bisa dimulai lagi di baris berikutnya. Dengan cara ini, bintang-bintang akan tersusun membentuk pola kotak yang rapi.

Program ini menggambarkan cara kerja dua perulangan yang saling bersarang untuk menghasilkan keluaran berbentuk pola visual. Dengan memodifikasi kondisi pada perulangan bagian dalam atau luar, pola yang dihasilkan bisa diubah menjadi bentuk lain seperti segitiga, tangga, atau piramida. Konsep dasar ini sering digunakan dalam latihan logika pemrograman untuk melatih pemahaman struktur kontrol dan pola output di Java.

Jika Programnya dijalankan maka outputnya adalah sebagai berikut



Gambar 2.6

2.6 Program NestedFor2

```

1 package pekan5;
2
3 public class NestedFor2 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         for (int i = 0; i <= 5; i++) {
8             for (int j = 0; j <= 5; j++) {
9                 System.out.print(i+j+ " ");
10
11             }
12             System.out.println();
13             // to end the line
14         }
15     }
16 }
17
18

```

Kode Program 2.7

Program NestedFor2 di atas menggunakan konsep perulangan bersarang (nested loop) untuk menampilkan deretan angka dengan pola tertentu. Perulangan luar dikontrol oleh variabel *i*, yang bernilai mulai dari 0 hingga 5, sedangkan perulangan dalam dikontrol oleh variabel *j*, yang juga berjalan dari 0 hingga kurang dari 5. Pada setiap iterasi dari perulangan dalam, program mencetak hasil penjumlahan dari variabel *i* dan *j*, kemudian diikuti dengan spasi agar hasil output terlihat rapi.

Setiap kali perulangan dalam selesai, program menjalankan perintah `System.out.println()`; untuk memindahkan posisi cetak ke baris berikutnya. Hal ini membuat hasil keluaran membentuk beberapa baris angka, di mana setiap baris memiliki lima nilai hasil dari operasi $i + j$. Nilai pada baris pertama dimulai dari 0 (karena $i=0$), dan akan meningkat di setiap baris berikutnya karena nilai i bertambah satu di setiap putaran perulangan luar.

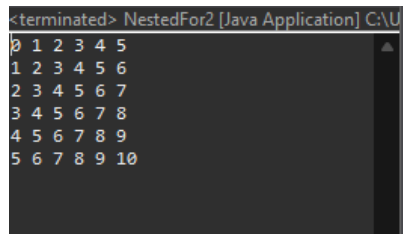
Secara keseluruhan, program ini menampilkan pola tabel angka sederhana, di mana setiap baris berisi hasil penjumlahan antara i dan j . Program ini dapat digunakan untuk memahami bagaimana dua perulangan saling berinteraksi, serta bagaimana variabel dari perulangan luar mempengaruhi hasil dari perulangan dalam. Pola seperti ini sering digunakan untuk latihan logika dasar dalam pemrograman, terutama untuk mempelajari hubungan antara indeks baris dan kolom.

Kode di atas memperlihatkan penerapan loop bersarang pada Java untuk membuat pola sederhana menggunakan simbol bintang (*). Perulangan luar dikontrol oleh variabel i yang menentukan jumlah baris, sedangkan perulangan dalam dikontrol oleh variabel j yang menentukan jumlah bintang yang dicetak di setiap baris. Karena kedua perulangan memiliki batas hingga angka 5, hasil akhirnya berupa lima baris dengan lima bintang di setiap baris.

Ketika perulangan luar berjalan satu kali, perulangan dalam dijalankan penuh untuk mencetak lima bintang secara berurutan tanpa pindah baris. Setelah itu, perintah `System.out.println()`; dijalankan untuk membuat baris baru, sehingga proses pencetakan bisa dimulai lagi di baris berikutnya. Dengan cara ini, bintang-bintang akan tersusun membentuk pola kotak yang rapi.

Program ini menggambarkan cara kerja dua perulangan yang saling bersarang untuk menghasilkan keluaran berbentuk pola visual. Dengan memodifikasi kondisi pada perulangan bagian dalam atau luar, pola yang dihasilkan bisa diubah menjadi bentuk lain seperti segitiga, tangga, atau piramida. Konsep dasar ini sering digunakan dalam latihan logika pemrograman untuk melatih pemahaman struktur kontrol dan pola output di Java.

Jika Programnya dijalankan maka outputnya adalah sebagai berikut



A screenshot of a Java application window titled "<terminated> NestedFor2 [Java Application] C:\U". The window displays a 5x5 grid of numbers. The first row contains numbers 0 through 5. Each subsequent row contains numbers starting from the previous row's first number, incrementing by 1. The grid is as follows:

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 2.7

BAB III

KESIMPULAN

3.1 Kesimpulan

Dari seluruh percobaan program yang telah dilakukan, dapat disimpulkan bahwa konsep perulangan (looping), khususnya perulangan for dan perulangan bersarang (nested for), memiliki peranan penting dalam pemrograman karena memudahkan proses pengulangan instruksi secara otomatis tanpa harus menulis kode berulang. Melalui praktikum ini, diperoleh pemahaman mengenai cara kerja variabel penghitung, kondisi batas perulangan, serta bagaimana pengaruh perubahan nilai variabel terhadap hasil output. Selain itu, implementasi nested for juga menunjukkan bagaimana dua atau lebih perulangan dapat saling berinteraksi untuk membentuk pola atau struktur data tertentu yang lebih kompleks.

Praktikum ini membuktikan bahwa logika pengulangan merupakan dasar utama dalam pengembangan algoritma. Dengan memahami mekanisme perulangan, seorang programmer dapat membuat program yang lebih efisien, terstruktur, dan mudah dimodifikasi. Setiap percobaan (PerulanganFor1–4 dan NestedFor0–2) memberikan gambaran nyata mengenai cara menulis dan menganalisis alur eksekusi program dengan hasil keluaran yang berbeda tergantung pada logika yang diterapkan.

3.2 Saran Pengembangan

Untuk pengembangan selanjutnya, disarankan agar praktikum tidak hanya berfokus pada perulangan sederhana, tetapi juga dikombinasikan dengan struktur kendali lainnya, seperti percabangan (if-else), array, atau fungsi (method) agar logika program menjadi lebih dinamis. Selain itu, hasil perulangan dapat dikembangkan untuk menghasilkan bentuk visual seperti pola bintang, tabel, atau matriks agar lebih interaktif dan mudah dipahami.

Penggunaan input dari pengguna (seperti pada program PerulanganFor4) juga dapat diperluas dengan validasi data dan antarmuka sederhana menggunakan GUI atau aplikasi berbasis web. Dengan begitu, mahasiswa tidak hanya memahami logika dasar perulangan, tetapi juga mampu mengaplikasikannya dalam konteks

pemrograman yang lebih nyata dan bermanfaat di dunia pengembangan perangkat lunak.

DAFTAR PUSTAKA

- [1] A. Sibero, *Belajar Pemrograman Java untuk Pemula*, Yogyakarta: Mediakom, 2013.
- [2] A. Nugroho, *Algoritma dan Pemrograman dengan Java*, Bandung: Informatika Bandung, 2019.
- [3] B. Raharjo, *Pemrograman Java: Dari Dasar Hingga Mahir*, Bandung: Informatika, 2018.
- [4] Wahana Komputer, *Panduan Praktis Pemrograman Java*, Yogyakarta: Andi Publisher, 2016.
- [5] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed., New York: McGraw-Hill, 2010.
- [6] I. Suarga, *Logika dan Algoritma Komputer*, Jakarta: Deepublish, 2021.
- [7] Oracle, "The Java™ Tutorials — For Statements," [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>. [Accessed: Nov. 1, 2025].
- [8] W3Schools, "Java For Loop," [Online]. Available: https://www.w3schools.com/java/java_for_loop.asp. [Accessed: Nov. 1, 2025].
- [9] GeeksforGeeks, "Nested Loops in Java," [Online]. Available: <https://www.geeksforgeeks.org/nested-loops-in-java/>. [Accessed: Nov. 1, 2025].
- [10] D. S. Malik, *Java Programming: From Problem Analysis to Program Design*, 6th ed., Boston: Cengage Learning, 2017.