**LAPORAN PRAKTIKUM**

**AGORITMA PEMROGRAMAN**

**“PERULANGAN FOR”**

**DISUSUN OLEH:**

**ABDUL JABBAR**

**2511533021**

**DOSEN PENGAMPU:**

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

**ASISTEN PRAKTIKUM:**

**JOVANTRI IMMANUEL GULO**

A logo of a tree with a candle

AI-generated content may be incorrect.

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**2025**

**KATA PENGANTAR**

Assalamu’alaikum Warahmatullahi Wabarakatuh

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan *Laporan Praktikum Pemrograman Java Pekan ke-5* ini dengan baik dan tepat waktu. Laporan ini disusun sebagai bentuk tanggung jawab serta dokumentasi atas kegiatan praktikum yang telah dilaksanakan, khususnya mengenai pembahasan materi perulanganfor dan nestedfor.

Konsep perulangan merupakan bagian mendasar dalam dunia pemrograman karena memungkinkan suatu proses dijalankan berulang kali secara otomatis dan terstruktur. Melalui praktikum ini, mahasiswa diharapkan mampu memahami serta mengimplementasikan konsep tersebut dalam berbagai permasalahan, sehingga dapat melatih kemampuan logika dan efisiensi dalam menulis kode program.

Dalam penyusunan laporan ini, penulis tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh sebab itu, penulis menyampaikan rasa terima kasih yang tulus kepada:

1. Dosen pengampu mata kuliah *Praktikum Algoritma dan Pemrograman* atas ilmu, bimbingan, dan arahannya selama proses pembelajaran.
2. Para asisten praktikum yang telah dengan sabar memberikan penjelasan serta membantu kelancaran kegiatan praktikum.

Penulis menyadari bahwa laporan ini masih memiliki banyak kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun guna penyempurnaan laporan di masa mendatang.

Akhir kata, semoga laporan ini dapat memberikan manfaat bagi penulis maupun pembaca dalam memahami konsep perulangan pada pemrograman Java.

Wassalamu’alaikum Warahmatullahi Wabarakatuh

Padang, 30 oktober 2025

Abdul jabbar

**DAFTAR ISI**

[BAB I PENDAHULUAN 4](#_Toc212809485)

[1.1 Latar Belakang 4](#_Toc212809486)

[1.2 Tujuan 4](#_Toc212809487)

[1.3 Manfaat 4](#_Toc212809488)

[BAB II PEMBAHASAN 5](#_Toc212809489)

[2.1 perulangan for 5](#_Toc212809490)

[2.2 Perulangan nested for 6](#_Toc212809491)

[2.3 pengaplikasian perulangan for 6](#_Toc212809492)

[2.3.1 Kode Program 7](#_Toc212809493)

[2.3.2 Kode Program 7](#_Toc212809494)

[2.3.3 Kode Program 9](#_Toc212809495)

[2.3.4 Kode Program 10](#_Toc212809496)

[2.4 Pengaplikasian nested for 11](#_Toc212809497)

[2.4.1 Kode program 11](#_Toc212809498)

[2.4.2 Kode Program 12](#_Toc212809499)

[2.4.3 Kode Program 13](#_Toc212809500)

[BAB III PENUTUP 14](#_Toc212809501)

[3.1 Kesimpulan 14](#_Toc212809502)

[3.2 Saran 14](#_Toc212809503)

[3.3 Daftar Pustaka 14](#_Toc212809504)

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Dalam bidang pemrograman, kemampuan untuk menjalankan sebuah blok kode secara berulang dengan aturan yang jelas merupakan aspek fundamental dalam membangun program yang efektif dan adaptif. Salah satu struktur kontrol yang sering digunakan untuk mengatur proses pengulangan tersebut adalah perulangan for. Struktur ini memberikan kendali penuh kepada programmer untuk menentukan banyaknya pengulangan, sehingga sangat bermanfaat dalam berbagai penerapan seperti pengolahan data, pembentukan pola, maupun manipulasi elemen dalam array.

Kegiatan praktikum ini bertujuan untuk memberikan pemahaman yang lebih mendalam mengenai mekanisme kerja perulanganfor dan nestedfor, serta penerapannya dalam bahasa pemrograman Java. Melalui berbagai latihan yang disajikan, mahasiswa diharapkan tidak hanya memahami sintaks dan konsep dasarnya, tetapi juga mampu mengembangkan kemampuan berpikir logis dan kreatif dalam menyelesaikan permasalahan yang memerlukan efisiensi dan ketepatan logika.

## 1.2 Tujuan

1. Memahami struktur sintaks dan prinsip kerja perulangan for dalam bahasa pemrograman Java.
2. Mampu menerapkan konsep nestedfor untuk memecahkan persoalan yang melibatkan struktur data bertingkat.
3. Melatih kemampuan analisis logika dan penyusunan algoritma melalui berbagai studi kasus terkait perulangan.
4. Mengasah keterampilan dalam melakukan debugging serta memahami alur eksekusi program secara sistematis.

## 1.3 Manfaat

1. Meningkatkan kemampuan menulis kode yang efisien, rapi, dan mudah dipahami menggunakan struktur perulangan.
2. Memperluas kemampuan pemecahan masalah yang membutuhkan proses iteratif kompleks, seperti pembuatan pola dan pengolahan array dua dimensi.
3. Menjadi dasar penting dalam memahami struktur kontrol dan algoritma tingkat lanjut di masa mendatang.
4. Memberikan wawasan praktis untuk mengembangkan aplikasi yang memerlukan pengolahan data secara berulang dan terorganisir

# BAB II PEMBAHASAN

## 2.1 perulangan for

Struktur perulangan (atau dalam bahasa inggris disebut loop) adalah instruksi kode program yang bertujuan untuk mengulang beberapa baris perintah.

Dalam merancang perulangan for, setidaknya kita harus mengetahui 3 komponen:

1. Kondisi awal perulangan.
2. Kondisi pada saat perulangan.
3. Kondisi yang harus dipenuhi agar perulangan berhenti.

Berikut format dasar struktur perulangan for dalam bahasa Java:

for (start; condition; increment)

{

   // kode program

   // kode program

}

Start adalah kondisi pada saat 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 variabel counter (ini tidak harus, boleh juga memakai variabel lain).

Condition adalah kondisi yang harus dipenuhi agar perulangan berjalan. Selama kondisi ini terpenuhi, maka *compiler* bahasa Java akan terus melakukan perulangan. Misalnya *condition* ini berisi perintah i < 7, maka selama *variabel counter* i berisi angka yang kurang dari 7, terus lakukan perulangan.

Increment adalah bagian yang dipakai untuk memproses variabel counter agar bisa memenuhi kondisi akhir perulangan. Bagian ini akan selalu di eksekusi di setiap perulangan**.**

Disebut *increment* karena biasanya berisi operasi increment seperti i++, yang sebenarnya sama dengan i = i + 1. Maksudnya, dalam setiap perulangan naikkan variabel i sebanyak 1 angka. Namun kita juga bisa memberikan nilai lain seperti i = i + 2 sehingga variabel counter akan naik 2 angka setiap perulangan

Sebagai tambahan, terdapat istilah iterasi (*iteration*), yang berarti 1 kali perulangan. Istilah ini cukup sering dipakai ketika membahas tentang struktur perulangan.

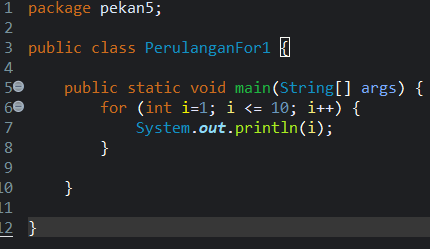
## 2.2 Perulangan nested for

Perulangan nested for dalam bahasa Java adalah perulangan *for* yang berada di dalam perulangan *for* lainnya. Dengan kata lain, satu perulangan dijalankan di dalam perulangan lain. Konsep ini digunakan ketika kita perlu melakukan proses berulang yang melibatkan dua tingkat atau lebih, seperti mencetak pola bintang, mengakses elemen dalam array dua dimensi, atau membuat tabel data. Cara kerjanya adalah, perulangan luar (*outer loop*) akan berjalan satu kali, lalu perulangan dalam (*inner loop*) akan dieksekusi sepenuhnya hingga selesai sebelum kembali ke perulangan luar untuk iterasi berikutnya. Misalnya, pada contoh kode for (int i = 1; i <= 3; i++) { for (int j = 1; j <= 5; j++) { System.out.print("\* "); } System.out.println(); }, perulangan luar mengatur jumlah baris sedangkan perulangan dalam mengatur jumlah kolom. Hasilnya akan menampilkan pola bintang berbentuk persegi dengan tiga baris dan lima kolom. Dengan demikian, nested for sangat berguna untuk memproses data yang memiliki struktur bertingkat atau berulang di dalam elemen yang lebih besar.

## 2.3 pengaplikasian perulangan for

Pada pekan 5 mahasiswa kelas A informatika praktikum mengenai kode pemprograman dalam Bahasa java yaitu perulangan for dan prulangan nested for dengan ini dapat membantu mahasiswa dalam memahami kode pemprograman tersebut

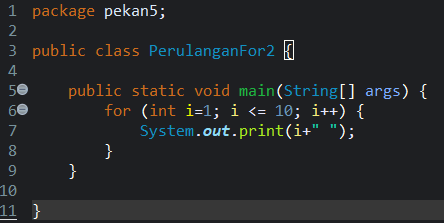
### 2.3.1 Kode Program



Program Java di atas merupakan contoh penggunaan perulangan **for** yang berfungsi untuk menampilkan angka dari 1 hingga 10.

1. Pada bagian pertama terdapat deklarasi package pekan5; yang menunjukkan bahwa program ini disimpan di dalam paket bernama *pekan5*. Selanjutnya, dibuat sebuah kelas bernama **PerulanganFor1** yang berisi seluruh kode program. Di dalam kelas tersebut terdapat metode utama public static void main(String[] args).
2. yang merupakan titik awal eksekusi program Java. Di dalam metode main terdapat struktur perulangan for (int i = 1; i <= 10; i++), yang berarti variabel i dimulai dari 1 dan akan terus bertambah satu setiap kali perulangan berlangsung hingga mencapai nilai 10. Setiap kali perulangan dijalankan, perintah System.out.println(i); akan menampilkan nilai i ke layar dan berpindah ke baris berikutnya. Dengan demikian, ketika program dijalankan, output yang dihasilkan adalah deretan angka dari 1 sampai 10 yang ditampilkan secara berurutan ke bawah.

### 2.3.2 Kode Program



Selanjutnya kode program yang mirip dengan sebelum nya yang membedakan nya yaitu print dan println sehingga terdapat perbedaan baris pada output nya dan baris System.out.print(i + " "); akan mencetak nilai i diikuti satu spasi.

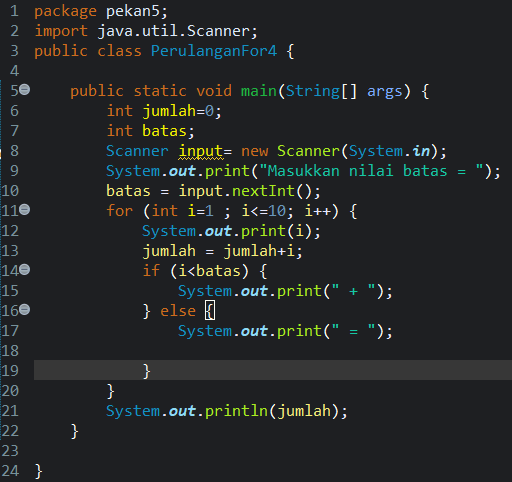
1. Menentukan paket Baris package pekan5; menunjukkan bahwa file Java ditempatkan dalam paket pekan5. Pastikan file disimpan di folder bernama pekan5 agar struktur paket sesuai dengan lokasi file.
2. Mendeklarasikan kelas utama public class PerulanganFor2 { mendefinisikan nama kelas PerulanganFor2. Nama file harus PerulanganFor2.java karena nama file harus sama dengan nama kelas publik.
3. Membuat metode main public static void main(String[] args) { adalah titik awal eksekusi program. Semua instruksi yang ingin dijalankan diletakkan di dalam blok main.
4. Menulis perulangan for for (int i = 1; i <= 10; i++) { terdiri dari tiga bagian:
5. Inisialisasi: int i = 1 → menetapkan nilai awal penghitung i.
6. Kondisi: i <= 10 → perulangan akan berlanjut selama kondisi ini bernilai benar.
7. Increment: i++ → menambah i sebesar 1 setiap akhir iterasi.
8. Perintah yang dieksekusi tiap iterasi Di dalam blok for ada System.out.print(i + " "); yang mencetak nilai i di baris yang sama diikuti satu spasi. Perbedaan penting: print tidak memindahkan baris, sehingga semua angka akan dicetak pada satu baris terpisah oleh spasi.Menyimpan dan mengompilasi Simpan file PerulanganFor2.java di folder pekan5. Dari command line (posisi di folder yang berisi folder pekan5) jalankan: javac pekan5/PerulanganFor2.java untuk mengompilasi.Menjalankan program Setelah sukses dikompilasi, jalankan program dengan perintah: java pekan5.PerulanganFor2. Perhatikan pemanggilan kelas saat menjalankan menggunakan nama paket penuh.Hasil eksekusi Program akan menampilkan angka 1 2 3 4 5 6 7 8 9 10 pada satu baris, setiap angka dipisahkan satu spasi karena penggunaan System.out.print(i + " ");.

### 2.3.3 Kode Program



merupakan pengembangan dari perulangan sebelumnya karena tidak hanya menampilkan deretan angka, tetapi juga menghitung total penjumlahannya. Pada bagian pertama, terdapat deklarasi package pekan5; yang menunjukkan bahwa program ini disimpan dalam paket bernama *pekan5*. Selanjutnya, dibuat kelas utama dengan nama PerulanganFor3, dan di dalamnya terdapat metode main sebagai titik awal eksekusi program. Di dalam metode main, dideklarasikan variabel int jumlah = 0; yang berfungsi untuk menampung hasil penjumlahan dari setiap angka yang ditampilkan. Struktur perulangan for (int i = 1; i <= 10; i++) digunakan untuk mengulangi proses dari angka 1 hingga 10. Pada setiap iterasi, program menampilkan nilai i dengan perintah System.out.print(i);, kemudian menambahkan nilai tersebut ke variabel jumlah menggunakan jumlah = jumlah + i;. Selain itu, terdapat percabangan if (i < 10) yang berfungsi untuk menampilkan tanda “+” di antara angka-angka, sehingga tanda tersebut tidak muncul setelah angka terakhir. Setelah perulangan selesai, perintah System.out.println(); digunakan untuk berpindah ke baris baru, kemudian System.out.println("jumlah = " + jumlah); menampilkan hasil total penjumlahan semua angka. Dengan demikian, ketika program dijalankan, hasil yang muncul adalah deretan angka dari 1 hingga 10 yang dipisahkan tanda “+”, diikuti dengan hasil penjumlahannya, yaitu jumlah = 55.

### 2.3.4 Kode Program



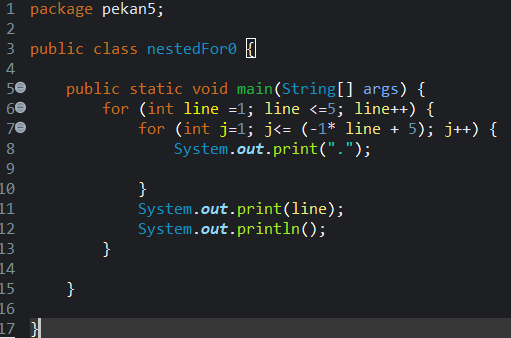
merupakan pengembangan dari program sebelumnya dengan menambahkan fitur input dari pengguna menggunakan kelas Scanner. Pada bagian awal, program mendeklarasikan package pekan5; untuk menandakan bahwa file ini berada di dalam paket *pekan5*. Selanjutnya, terdapat perintah import java.util.Scanner; yang berfungsi untuk mengimpor kelas Scanner, sehingga program dapat menerima masukan dari pengguna melalui keyboard. Di dalam kelas PerulanganFor4, terdapat metode main sebagai titik awal eksekusi program. Dua variabel dideklarasikan di awal, yaitu jumlah dengan nilai awal 0 untuk menyimpan total penjumlahan angka, dan batas yang akan diisi berdasarkan input pengguna. Program kemudian membuat objek Scanner input = new Scanner(System.in); dan menampilkan pesan Masukkan nilai batas = agar pengguna memasukkan angka batas. Nilai yang dimasukkan disimpan ke dalam variabel batas dengan perintah batas = input.nextInt();.

Selanjutnya, program menggunakan perulangan for (int i = 1; i <= 10; i++) untuk menampilkan angka dari 1 hingga 10. Setiap kali perulangan berjalan, program mencetak nilai i dengan System.out.print(i); dan menambahkan nilai tersebut ke variabel jumlah menggunakan jumlah = jumlah + i;. Setelah itu, terdapat struktur percabangan if (i < batas) yang berfungsi untuk menentukan tanda yang akan ditampilkan setelah angka. Jika nilai i masih kurang dari batas, maka program menampilkan tanda “+”, tetapi jika nilai i sudah sama atau lebih besar dari batas, maka program menampilkan tanda “=”. Dengan demikian, tanda yang muncul di antara angka-angka akan bergantung pada input yang diberikan pengguna. Setelah perulangan selesai, program menampilkan hasil akhir dengan System.out.println(jumlah); yang mencetak total penjumlahan seluruh angka dari 1 sampai 10.

## 2.4 Pengaplikasian nested for

Pada pekan 5 mahasiswa kelas A informatika praktikum mengenai kode pemprograman dalam Bahasa java yaitu perulangan for dan prulangan nested for dengan ini dapat membantu mahasiswa dalam memahami kode pemprograman tersebut.

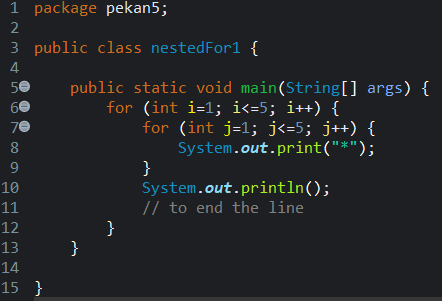
### 2.4.1 Kode program



perulanganbersarang **(**nestedfor**)** dalam bahasa Java, yang berfungsi untuk mencetak pola tertentu dengan kombinasi titik (“.”) dan angka. Pada bagian awal, terdapat deklarasi package pekan5; yang menunjukkan bahwa program ini berada di dalam paket bernama *pekan5*. Selanjutnya, dibuat kelas dengan nama nestedFor0, dan di dalamnya terdapat metode main sebagai titik awal eksekusi program.

Program ini menggunakan dua perulangan for yang bersarang. Perulangan pertama, yaitu for (int line = 1; line <= 5; line++), berfungsi sebagai perulanganluar(outer loop) yang mengatur jumlah baris yang akan dicetak. Nilai variabel line dimulai dari 1 hingga 5, sehingga program akan mencetak total lima baris. Di dalam perulangan luar terdapat perulangan dalam (inner loop), yaitu for (int j = 1; j <= (-1 \* line + 5); j++). Perulangan ini berfungsi untuk mencetak titik (“.”) sebanyak jumlah yang bergantung pada nilai variabel line.Ekspresi (-1 \* line + 5) digunakan untuk mengatur banyaknya titik pada setiap baris. Ketika nilai line semakin besar, jumlah titik akan berkurang. Misalnya, pada baris pertama (line = 1), hasilnya (-1 \* 1 + 5) = 4, sehingga empat titik akan dicetak. Pada baris kedua (line = 2), jumlah titik menjadi tiga, dan seterusnya hingga baris kelima hanya mencetak satu angka tanpa titik. Setelah perulangan dalam selesai, program mencetak nilai variabel line menggunakan System.out.print(line); dan kemudian menambahkan perintah System.out.println(); agar kursor berpindah ke baris baru untuk mencetak baris berikutnya.

### 2.4.2 Kode Program



ada gambar merupakan contoh penggunaan perulangan bersarang (nested for loop) dalam bahasa Java untuk mencetak pola bintang berbentuk persegi. Program diawali dengan deklarasi package pekan5; yang menunjukkan bahwa file ini termasuk dalam paket bernama *pekan5*. Selanjutnya, terdapat deklarasi kelas public class nestedFor1 sebagai wadah utama program. Di dalam kelas ini, terdapat metode main yang didefinisikan dengan sintaks public static void main(String[] args), yang menjadi titik awal eksekusi program Java.

Pada bagian isi metode main, terdapat dua perulangan for yang saling bersarang. Perulangan pertama for (int i = 1; i <= 5; i++) berfungsi sebagai perulangan luar (outer loop) yang mengatur jumlah baris yang akan dicetak. Nilai variabel i dimulai dari 1 dan meningkat hingga 5, sehingga program akan mencetak total lima baris. Di dalam perulangan luar terdapat perulangan kedua for (int j = 1; j <= 5; j++), yang berfungsi sebagai perulangan dalam (inner loop) untuk mencetak karakter bintang "\*" pada setiap baris.

Setiap kali perulangan dalam dijalankan, perintah System.out.print("\*"); akan mencetak satu karakter bintang tanpa berpindah ke baris baru. Setelah perulangan dalam selesai dijalankan (artinya lima bintang sudah dicetak pada satu baris), perintah System.out.println(); akan dieksekusi untuk memindahkan kursor ke baris berikutnya, sehingga baris selanjutnya dapat dicetak dengan pola yang sama.

### 2.4.3 Kode Program

A screen shot of a computer code

AI-generated content may be incorrect.

pada gambar merupakan contoh penerapan **perulangan bersarang (nested for loop)** dalam bahasa Java yang digunakan untuk mencetak hasil penjumlahan dari dua variabel perulangan, yaitu i dan j. Program diawali dengan deklarasi package pekan5; yang menandakan bahwa file ini berada dalam paket bernama *pekan5*. Selanjutnya, didefinisikan kelas public class nestedFor2 sebagai wadah utama program, yang di dalamnya terdapat metode main dengan sintaks public static void main(String[] args). Metode ini merupakan titik awal eksekusi program.

Pada bagian isi metode main, terdapat dua perulangan **for** yang saling bersarang. Perulangan luar for (int i = 0; i <= 5; i++) berfungsi untuk mengontrol jumlah baris output, di mana variabel i dimulai dari 0 hingga 5. Setiap kali perulangan luar dijalankan, program akan mengeksekusi perulangan dalam for (int j = 0; j <= 5; j++), yang berfungsi untuk mengontrol jumlah kolom pada setiap baris.

Di dalam perulangan dalam, terdapat perintah System.out.print(i + j + " "); yang mencetak hasil penjumlahan antara nilai i dan j diikuti dengan spasi. Karena System.out.print tidak membuat baris baru, semua hasil penjumlahan pada satu baris akan dicetak berdampingan. Setelah perulangan dalam selesai, perintah System.out.println(); dijalankan untuk memindahkan kursor ke baris berikutnya sehingga baris baru dapat dicetak.

# BAB III PENUTUP

## 3.1 Kesimpulan

Perulangan for merupakan struktur kontrol dalam pemrograman yang digunakan untuk mengeksekusi suatu blok kode secara berulang dengan jumlah iterasi yang sudah diketahui. Struktur ini sangat efisien untuk proses yang membutuhkan pengulangan dengan pola tertentu, seperti mencetak urutan angka, menghitung total nilai, atau memproses elemen dalam array.

Sementara itu, nested for (perulangan bersarang) adalah perulangan for yang ditempatkan di dalam perulangan for lainnya. Teknik ini digunakan untuk mengelola proses berulang dua dimensi atau lebih, seperti mencetak pola berbentuk tabel, segitiga, atau matriks.

Dengan memahami dan menggunakan for serta nested for, programmer dapat menulis kode yang lebih sistematis, efisien, dan mudah digunakan untuk menyelesaikan berbagai permasalahan logika dan pola dalam dunia pemrograman.

## 3.2 Saran

Dalam menggunakan perulangan for dan nested for, disarankan agar programmer selalu memperhatikan kondisi awal, syarat berhenti, dan increment/decrement yang digunakan agar tidak terjadi infinite loop (perulangan tak terbatas). Selain itu, penggunaan nested for sebaiknya disesuaikan dengan kebutuhan, karena semakin dalam tingkat perulangan, semakin besar pula kompleksitas waktu eksekusi program. Untuk meningkatkan efisiensi, programmer juga perlu memahami logika pola yang ingin dibuat sebelum menuliskan kode. Dengan latihan yang konsisten dan pemahaman logika yang baik, penggunaan perulangan for dan nested for dapat membantu dalam menyelesaikan berbagai permasalahan pemrograman dengan cara yang efektif dan terstruktur.

## 3.3 Daftar Pustaka

[1] Dunia Ilkom, Tutorial Belajar Java Part 33: Perulangan FOR Bahasa Java,2020, <https://www.duniailkom.com/tutorial-belajar-java-perulangan-for-bahasa-java/>

[2]Codegym,LoopBersarangJava <https://codegym.cc/id/groups/posts/id.665.loop-bersarang-java>