

**LAPORAN PRATIKUM PEKAN 5**

**ALGORITMA DAN PEMROGRAMAN**

**disusun Oleh:**

**ALIF MUHAMMAD IHSAN**

**NIM 2511531004**

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

**Asisten Laboratorium: Muhammad Zaki Al Hafiz**



**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**PADANG**

**2025**

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Swt atas limpahan rahmat dan karunia-Nya sehingga praktikum pemrograman Java ini dapat diselesaikan dengan baik dan laporan ini dapat disusun sebagai bentuk pertanggungjawaban akademik.

Laporan praktikum ini disusun untuk mendokumentasikan proses pembelajaran yang telah dilakukan, khususnya dalam memahami struktur dasar pemrograman Java, penggunaan metode, serta penerapan logika prosedural dalam bentuk program sederhana. Praktikum ini diharapkan dapat memperkuat pemahaman peserta terhadap konsep-konsep dasar pemrograman serta meningkatkan kemampuan teknis dalam menulis kode yang terstruktur dan efisien.

Penulis menyadari bahwa penyusunan laporan ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada dosen pembimbing, asisten praktikum, serta rekan-rekan yang telah memberikan dukungan dan masukan selama pelaksanaan praktikum.

Akhir kata, penulis berharap laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pembaca dalam memahami dasar-dasar pemrograman Java. Kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Padang, November 2025

Alif Muhammad Ihsan

## **Daftar Isi**

<b>KATA PENGANTAR.....</b>	i
<b>BAB I.....</b>	1
<b>PENDAHULUAN .....</b>	1
<b>1.1    Latar Belakang.....</b>	1
<b>1.2    Tujuan.....</b>	1
<b>1.3    Manfaat Praktikum .....</b>	2
<b>BAB II.....</b>	3
<b>PEMBAHASAN.....</b>	3
<b>2.1    Kode Program.....</b>	3
<b>2.2    Langkah Kerja.....</b>	11
<b>2.3    Analisis Hasil .....</b>	19
<b>BAB III .....</b>	25
<b>PENUTUP .....</b>	25
<b>3.1    Kesimpulan.....</b>	25
<b>3.2    Saran.....</b>	25
<b>DAFTAR PUSTAKA .....</b>	26

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Dalam era perkembangan teknologi informasi yang semakin pesat, pemahaman terhadap dasar-dasar pemrograman menjadi hal yang esensial bagi setiap individu yang ingin berkontribusi dalam bidang rekayasa perangkat lunak. Salah satu bahasa pemrograman yang banyak digunakan dalam dunia akademik maupun industri adalah Java. Bahasa ini dikenal karena sifatnya yang berorientasi objek, portabel, serta memiliki struktur sintaks yang relatif mudah dipahami oleh pemula.

Praktikum ini dilaksanakan sebagai bagian dari upaya untuk memperkuat pemahaman mahasiswa terhadap konsep dasar pemrograman Java, khususnya dalam hal struktur program, penggunaan metode, serta logika eksekusi program. Melalui kegiatan praktikum, peserta diharapkan mampu menerapkan teori yang telah dipelajari dalam bentuk implementasi nyata, sehingga tercipta keterkaitan antara aspek konseptual dan praktik langsung.

#### **1.2 Tujuan**

1. Memahami struktur dasar program Java, termasuk penggunaan metode, pemanggilan fungsi, dan alur eksekusi program secara sistematis.
2. Melatih kemampuan dalam menyusun algoritma prosedural, melalui implementasi langkah-langkah logis yang merepresentasikan proses nyata, seperti pembuatan kue.
3. Meningkatkan keterampilan teknis dalam menulis kode yang terstruktur dan efisien, serta membiasakan diri dengan praktik penulisan program yang baik dan mudah dipahami.

4. Mendorong kemampuan berpikir kritis dan analitis, khususnya dalam mengidentifikasi urutan proses, menyederhanakan logika, dan menerjemahkan aktivitas sehari-hari ke dalam bentuk algoritma.

### **1.3 Manfaat Praktikum**

1. **Memperkuat pemahaman konsep dasar pemrograman Java**, sehingga peserta mampu mengaitkan teori dengan implementasi nyata.
2. **Mengasah keterampilan penyusunan algoritma prosedural**, melalui pembiasaan merinci setiap langkah kerja sebelum menerjemahkannya ke dalam kode.
3. **Meningkatkan kemampuan menulis kode yang modular dan efisien**, dengan memanfaatkan metode sebagai unit kerja yang terpisah dan terstruktur.
4. **Memberikan pengalaman langsung dalam penerjemahan proses sehari-hari ke dalam program**, sehingga peserta dapat melihat relevansi pemrograman dalam konteks kehidupan nyata.
5. **Melatih kedisiplinan, kerapian, dan kemampuan dokumentasi**, yang menjadi landasan penting bagi profesional di bidang rekayasa perangkat lunak.

## BAB II

### PEMBAHASAN

#### 2.1 Kode Program

a. Perulangan For 1

package Pekan5;

```
public class PerulanganFor {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

#### Hasil Output

1

2

3

4

5

6

7

8

9

10

b. Perulangan For 2

```
package Pekan5;
```

```
public class PerulanganFor2 {
```

```
    public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        for (int i= 1; i <= 10; i++) {
```

```
            System.out.print(i + " ");
```

```
}
```

```
}
```

Hasil Output

1 2 3 4 5 6 7 8 9 10

c. Perulangan For 3

```
package Pekan5;
```

```
public class PerulanganFor3 {
```

```
    public static void main(String[] args) {
```

```

// TODO Auto-generated method stub

int jumlah = 0 ;
for (int i = 1; i <= 10; i++) {
    System.out.print(i);
    jumlah = jumlah + i;
    if ( i < 10) {
        System.out.print(" + ");
    }
}
System.out.println();
System.out.println("Jumlah = " + jumlah);

}

```

#### Hasil Output

1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10  
Jumlah = 55

#### d. Perulangan For 4

```

package Pekan5;
import java.util.Scanner;
public class PerulanganFor4 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int jumlah = 0;
        int batas;

```

```

Scanner input = new Scanner(System.in);
System.out.print("Masukkan Nilai Batas");
batas = input.nextInt();
input.close();

for (int i = 1; i <= batas; i++) {
    System.out.print(i);
    jumlah = jumlah + i;
    if (i < batas) {
        System.out.print(" + ");
    } else {
        System.out.print(" = ");
    }
}

System.out.println(jumlah);

}

```

Hasil Outpu  
 Masukkan Nilai Batas 3  
 $1 + 2 + 3 = 6$

e. Nested for 0

```

package Pekan5;

public class nestedfor0 {

    public static void main(String[] args) {

```

```
// TODO Auto-generated method stub

for (int line = 1; line <= 5; line++) {
    for (int j = 1; j <= (-1 * line + 5); j++) {
        System.out.print(".");
    }

    System.out.println(line);
    System.out.println();
}

}
```

Hasil output

....1

...2

..3

.4

5

f. Nested for 1

```
package Pekan5;
```

```
public class nestedfor1 {
```

```

public static void main(String[] args) {
    // TODO Auto-generated method stub

    for (int i = 1; i <= 5; i++) {
        for (int j = 1; j <= 5; j++) {
            System.out.print("*");

        }
        System.out.println();
    }

}

```

Hasil output

```

*****
*****
*****
*****
*****

```

g. Nested For 2

```
package Pekan5;
```

```

public class NesteedFor2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        for (int i = 0; i <= 5; i++) {
            for (int j = 0; j <= 5; j++) {

```

```

        System.out.print(i + j + " ");
    }
    System.out.println();
}
}
}

```

Hasil output

```

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

```

h. Nested For 3

```

package Pekan5;

public class Pekan5 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int n = 5;
        for (int line = 1; line <= n; line++) { // untuk buat jadi 5 baris
            // untuk buat titik dari kiri
            for (int j = 1; j <= (-1 * line + 5); j++) {
                System.out.print(".");
            }
        }
    }
}

```

```
// cetak angka  
System.out.print(line);  
// cetak titik dari kanan  
for (int j = 1; j < line; j++ ) {  
    System.out.print(".");  
  
}  
// untuk pindah baris  
System.out.println();  
  
}  
}  
  
}
```

Hasil output

....1

...2.

..3..

.4...

5....

## 2.2 Langkah Kerja

Langkah-langkah yang dilakukan dalam praktikum ini adalah sebagai berikut:

### A. Perulangan For 1

#### 1. Inisialisasi Variabel Penghitung

- **Sintaks:** int i = 1
- **Penjelasan:** Program memulai perulangan dengan menetapkan nilai awal variabel *i* sebesar 1. Variabel ini berfungsi sebagai penghitung iterasi.

#### 2. Pemeriksaan Kondisi Perulangan

- **Sintaks:** i <= 10
- **Penjelasan:** Sebelum setiap iterasi, program memeriksa apakah nilai *i* masih kurang dari atau sama dengan 10. Jika kondisi ini benar, maka blok perulangan akan dijalankan. Jika salah, perulangan berhenti.

#### 3. Eksekusi Blok Perulangan

- **Sintaks:** System.out.println(i);
- **Penjelasan:** Selama kondisi terpenuhi, program akan mencetak nilai *i* ke layar. Setiap iterasi menghasilkan satu baris output berupa angka dari 1 hingga 10.

#### 4. Inkrementasi Nilai Penghitung

- **Sintaks:** i++
- **Penjelasan:** Setelah satu siklus perulangan selesai, nilai *i* akan ditambah satu. Proses ini disebut *inkrementasi*, dan bertujuan untuk mendekati batas akhir perulangan.

#### 5. Pengulangan Proses

- Program kembali ke langkah 2 untuk memeriksa apakah *i* masih memenuhi syarat. Jika ya, langkah 3 dan 4 diulang. Jika tidak, perulangan berhenti dan program selesai.

## B. Perulangan For 2

### 1. Inisialisasi Variabel Penghitung

- **Sintaks:** int  $i = 1$
- **Penjelasan:** Program memulai perulangan dengan menetapkan nilai awal variabel  $i$  sebesar 1. Variabel ini berfungsi sebagai penghitung iterasi.

### 2. Pemeriksaan Kondisi Perulangan

- **Sintaks:**  $i \leq 10$
- **Penjelasan:** Sebelum setiap iterasi, program memeriksa apakah nilai  $i$  masih kurang dari atau sama dengan 10. Jika benar, maka blok perulangan dijalankan. Jika salah, perulangan berhenti.

### 3. Eksekusi Blok Perulangan

- **Sintaks:** System.out.print( $i + " "$ );
- **Penjelasan:** Selama kondisi terpenuhi, program mencetak nilai  $i$  ke layar, diikuti dengan spasi. Berbeda dari versi sebelumnya yang menggunakan println, kali ini menggunakan print, sehingga semua angka dicetak dalam satu baris.

### 4. Inkrementasi Nilai Penghitung

- **Sintaks:**  $i++$
- **Penjelasan:** Setelah satu siklus perulangan selesai, nilai  $i$  bertambah satu. Proses ini disebut *inkrementasi*, dan bertujuan untuk mendekati batas akhir perulangan.

### 5. Pengulangan Proses

- Program kembali ke langkah 2 untuk memeriksa apakah  $i$  masih memenuhi syarat. Jika ya, langkah 3 dan 4 diulang. Jika tidak, perulangan berhenti dan program selesai.

## C. Perulangan For 3

### 1. Deklarasi dan Inisialisasi Variabel Akumulator

- **Sintaks:** int jumlah = 0;
- **Penjelasan:** Program membuat variabel *jumlah* untuk menyimpan hasil penjumlahan dari angka 1 sampai 10. Nilai awalnya ditetapkan nol.

### 2. Perulangan for

- **Sintaks:** for (int i = 1; i <= 10; i++)
- **Penjelasan:**
- *i* dimulai dari 1.
- Perulangan terus berjalan selama *i* kurang dari atau sama dengan 10.
- Setelah setiap iterasi, *i* bertambah satu.

### 3. Menampilkan Nilai dan Operator Tambah

- **Sintaks:**

```
System.out.print(i); if (i < 10) { System.out.print(" + "); }
```

- **Penjelasan:**

- Program mencetak nilai *i* ke layar.
- Jika *i* belum mencapai 10, maka program mencetak tanda tambah (+) setelah angka.
- Tujuannya agar tampilan menjadi: 1 + 2 + 3 + ... + 10

### 4. Menjumlahkan Nilai ke Akumulator

- **Sintaks:** jumlah = jumlah + i;
- **Penjelasan:** Nilai *i* ditambahkan ke variabel *jumlah* setiap kali perulangan berjalan. Ini disebut proses *akumulasi*.

### 5. Akhir Perulangan dan Output Total

- Setelah perulangan selesai, program mencetak baris kosong (System.out.println();) untuk pindah ke baris baru.
- Kemudian mencetak hasil akhir penjumlahan: Jumlah = 55

## D. Perulangan For 4

### 1. Deklarasi Variabel

- **Sintaks:**

Int jumlah = 0; int batas;

- **Penjelasan:**

- *jumlah* digunakan sebagai akumulator untuk menyimpan hasil penjumlahan.
- *batas* digunakan untuk menyimpan nilai batas atas yang dimasukkan oleh pengguna.

### 2. Input dari Pengguna

- **Sintaks:**

```
Scanner input = new Scanner(System.in); System.out.print("Masukkan  
Nilai Batas"); batas = input.nextInt(); input.close();
```

- **Penjelasan:**

- Program meminta pengguna memasukkan nilai batas.
- Nilai tersebut disimpan dalam variabel *batas*.
- Scanner ditutup setelah input diterima untuk efisiensi dan keamanan.

### 3. Perulangan for

- **Sintaks:** for (int i = 1; i <= batas; i++)

- **Penjelasan:**

- Perulangan dimulai dari angka 1 hingga nilai *batas*.
- Setiap iterasi mencetak angka dan menambahkan nilainya ke variabel *jumlah*.

### 4. Menampilkan Proses Penjumlahan

- **Sintaks:**

```
System.out.print(i); jumlah = jumlah + i; if (i < batas) { System.out.print(" + "); } else { System.out.print(" = "); }
```

- **Penjelasan:**

- Program mencetak angka saat ini (*i*).

- Menambahkan nilai  $i$  ke *jumlah*.
- Jika  $i$  belum mencapai batas, tampilkan tanda tambah (+).
- Jika  $i$  sudah mencapai batas, tampilkan tanda sama dengan (=).

## 5. Menampilkan Hasil Akhir

- **Sintaks:** System.out.println(jumlah);
- **Penjelasan:** Setelah perulangan selesai, program mencetak hasil total penjumlahan.

## E. Nested For 0

### 1. Perulangan Luar (Baris)

- **Sintaks:** for (int line = 1; line <= 5; line++)
- **Penjelasan:** Perulangan ini mengatur jumlah baris yang akan ditampilkan, dimulai dari baris ke-1 hingga baris ke-5.

### 2. Perulangan Dalam (Titik)

- **Sintaks:** for (int j = 1; j <= (-1 \* line + 5); j++)
- **Penjelasan:**
  - Perulangan ini mencetak karakter titik (.) sebanyak tertentu di setiap baris.
  - Rumus  $(-1 * line + 5)$  menghasilkan jumlah titik yang menurun seiring bertambahnya nilai *line*.
  - Baris 1:  $(-1*1 + 5) = 4$  titik
  - Baris 2:  $(-1*2 + 5) = 3$  titik
  - Baris 3:  $(-1*3 + 5) = 2$  titik
  - Baris 4:  $(-1*4 + 5) = 1$  titik
  - Baris 5:  $(-1*5 + 5) = 0$  titik

### 3. Menampilkan Nomor Baris

- **Sintaks:** System.out.println(line);

- **Penjelasan:** Setelah mencetak titik-titik, program mencetak nomor baris sesuai nilai *line*. Angka ini muncul setelah titik-titik di baris yang sama.

#### 4. Pindah Baris

- **Sintaks:** System.out.println();
- **Penjelasan:** Perintah ini digunakan untuk membuat baris baru agar tampilan output tidak menumpuk.

### F. Nested For 1

#### 1. Perulangan Luar (Baris)

- **Sintaks:** for (int i = 1; i <= 5; i++)
- **Penjelasan:** Perulangan ini mengatur jumlah baris yang akan dicetak. Dimulai dari baris ke-1 hingga baris ke-5.

#### 2. Perulangan Dalam (Kolom)

- **Sintaks:** for (int j = 1; j <= 5; j++)
- **Penjelasan:** Di dalam setiap baris, program mencetak karakter bintang (\*) sebanyak 5 kali. Perulangan ini berjalan dari kolom ke-1 hingga kolom ke-5.

#### 3. Mencetak Karakter Bintang

- **Sintaks:** System.out.print("\*");
- **Penjelasan:** Karakter bintang dicetak tanpa pindah baris, sehingga membentuk satu baris penuh berisi lima bintang.

#### 4. Pindah ke Baris Baru

- **Sintaks:** System.out.println();
- **Penjelasan:** Setelah mencetak lima bintang dalam satu baris, program pindah ke baris berikutnya untuk melanjutkan perulangan luar.

## G. Nested For 2

### 1. Perulangan Luar (Baris)

- **Sintaks:** for (int i = 0; i <= 5; i++)
- **Penjelasan:** Perulangan ini mengatur jumlah baris yang akan dicetak. Nilai *i* dimulai dari 0 dan terus bertambah hingga 5. Total ada 6 baris.

### 2. Perulangan Dalam (Kolom)

- **Sintaks:** for (int j = 0; j <= 5; j++)
- **Penjelasan:** Di dalam setiap baris, program mencetak enam angka. Nilai *j* dimulai dari 0 hingga 5.

### 3. Menampilkan Penjumlahan *i* + *j*

- **Sintaks:** System.out.print(i + j + " ");
- **Penjelasan:** Program mencetak hasil penjumlahan antara *i* dan *j*, dipisahkan dengan spasi. Karena *i* dan *j* berubah setiap iterasi, hasilnya membentuk pola angka yang meningkat.

### 4. Pindah ke Baris Baru

- **Sintaks:** System.out.println();
- **Penjelasan:** Setelah satu baris selesai dicetak, program pindah ke baris berikutnya agar tampilan output rapi dan terstruktur.

## H. Nested For 3

### 1. Deklarasi Variabel

- **Sintaks:** int n = 5;
- **Penjelasan:** Menetapkan jumlah baris yang akan dicetak, yaitu sebanyak 5.

### 2. Perulangan Luar (Baris)

- **Sintaks:** for (int line = 1; line <= n; line++)
- **Penjelasan:** Mengatur iterasi dari baris ke-1 hingga baris ke-5. Variabel *line* menunjukkan nomor baris saat ini.

### **3. Perulangan Dalam 1 – Titik dari Kiri**

- **Sintaks:** for (int j = 1; j <= (-1 \* line + 5); j++)
- **Penjelasan:**
  - Mencetak titik (.) di sebelah kiri angka.
  - Jumlah titik berkurang seiring bertambahnya nilai *line*.
  - Baris 1: 4 titik
  - Baris 2: 3 titik
  - Baris 3: 2 titik
  - Baris 4: 1 titik
  - Baris 5: 0 titik

### **4. Mencetak Angka Tengah**

- **Sintaks:** System.out.print(line);
- **Penjelasan:** Menampilkan angka sesuai nomor baris di tengah antara titik kiri dan titik kanan.

### **5. Perulangan Dalam 2 – Titik dari Kanan**

- **Sintaks:** for (int j = 1; j < line; j++)
- **Penjelasan:**
  - Mencetak titik (.) di sebelah kanan angka.
  - Jumlah titik bertambah seiring bertambahnya nilai *line*.
  - Baris 1: 0 titik
  - Baris 2: 1 titik
  - Baris 3: 2 titik
  - Baris 4: 3 titik
  - Baris 5: 4 titik

### **6. Pindah ke Baris Baru**

- **Sintaks:** System.out.println();
- **Penjelasan:** Setelah satu baris selesai dicetak, program pindah ke baris berikutnya agar tampilan output rapi.

## 2.3 Analisis Hasil

### A. Perulangan For 1

Program ini menggunakan struktur perulangan for untuk mencetak angka 1 sampai 10 ke layar. Kode berada dalam paket Pekan5 dan kelas PerulanganFor berisi metode main sebagai titik eksekusi.

Sintaks for (int i = 1; i <= 10; i++) berarti:

- Inisialisasi: i dimulai dari 1
- Kondisi: selama i kurang dari atau sama dengan 10
- Iterasi: i bertambah 1 setiap siklus
- Aksi: System.out.println(i) mencetak nilai i ke layar

Hasilnya adalah angka 1 sampai 10 tercetak satu per satu di baris baru. Struktur ini efisien untuk tugas berulang dan sangat umum digunakan dalam pemrograman dasar.

#### Referensi:

- Oracle Java Documentation: [The for Statement](#)
- Nugroho, Adi. *Pemrograman Java*. Andi Offset
- KBBI Daring. Badan Bahasa Kemdikbud

### B. Perulangan For 2

Program ini menggunakan struktur perulangan for untuk mencetak angka 1 sampai 10 dalam satu baris. Kode berada dalam paket Pekan5 dan kelas PerulanganFor2 berisi metode main sebagai titik eksekusi.

Sintaks for (int i = 1; i <= 10; i++) berarti:

- Inisialisasi: i dimulai dari 1
- Kondisi: selama i kurang dari atau sama dengan 10
- Iterasi: i bertambah 1 setiap siklus
- Aksi: System.out.print(i + " ") mencetak nilai i di satu baris, dipisahkan oleh spasi

Hasilnya adalah:

1 2 3 4 5 6 7 8 9 10

Semua angka tercetak dalam satu baris karena menggunakan print, bukan println.

**Referensi:**

- Oracle Java Documentation: [The for Statement](#)
- Nugroho, Adi. *Pemrograman Java*. Andi Offset
- KBBI Daring. Badan Bahasa Kemdikbud

C. Perulangan For 3

Program ini menggunakan perulangan for untuk mencetak deret angka 1 sampai 10 yang dijumlahkan, serta menampilkan totalnya. Kode berada dalam paket Pekan5 dan kelas PerulanganFor3 berisi metode main.

Sintaks for (int i = 1; i <= 10; i++) berarti:

- Inisialisasi: i dimulai dari 1
- Kondisi: selama i kurang dari atau sama dengan 10
- Iterasi: i bertambah 1 setiap siklus
- Aksi:
  - System.out.print(i) mencetak angka
  - jumlah = jumlah + i menambahkan nilai i ke variabel jumlah
  - if (i < 10) menambahkan simbol + di antara angka, kecuali setelah angka terakhir

Hasilnya:

1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 Jumlah = 55

Program ini menunjukkan cara menghitung total dari deret angka menggunakan perulangan dan logika kondisional.

**Referensi:**

- Oracle Java Documentation: [The for Statement](#)
- Nugroho, Adi. *Pemrograman Java*. Andi Offset
- KBBI Daring. Badan Bahasa Kemdikbud

#### D. Perulangan For 4

Program ini mencetak deret angka dari 1 sampai nilai batas yang dimasukkan oleh pengguna, lalu menghitung total jumlahnya. Kode berada dalam paket Pekan5 dan kelas PerulanganFor4 berisi metode main.

Alur program:

- Scanner input digunakan untuk membaca nilai batas dari pengguna
- Perulangan for (`int i = 1; i <= batas; i++`) mencetak angka dan menjumlahkannya
- `System.out.print(" + ")` ditampilkan di antara angka, dan = jumlah di akhir

Contoh output jika pengguna memasukkan 5:

Masukkan Nilai Batas5  $1 + 2 + 3 + 4 + 5 = 15$

Program ini menggabungkan input dinamis dengan logika perulangan dan penjumlahan, cocok untuk latihan pemahaman dasar Java.

#### Referensi:

- Oracle Java Documentation: [The for Statement](#)
- Nugroho, Adi. *Pemrograman Java*. Andi Offset
- KBBI Daring. Badan Bahasa Kemdikbud

#### E. Nested For 0

Program ini menggunakan perulangan bersarang (nested for) untuk mencetak pola titik(.) dan angka dalam format tertentu. Kode berada dalam paket Pekan5 dan kelas nestedfor0 berisi metode main.

Struktur perulangan:

- Perulangan luar (line) berjalan dari 1 sampai 5
- Perulangan dalam (j) mencetak titik sebanyak  $(-1 * \text{line} + 5)$  kali
- Setelah titik, angka line dicetak di akhir baris
- `System.out.println()` tambahan digunakan untuk membuat baris kosong antar output

Contoh hasil output:

....1 ...2 ..3 .4 5

Pola ini menunjukkan bagaimana perulangan bersarang bisa digunakan untuk membentuk tampilan visual yang terstruktur, berguna dalam pembuatan menu, grafik teks, atau simulasi tampilan konsol.

**Referensi:**

- Oracle Java Documentation: [Nested Loops](#)
- Nugroho, Adi. *Pemrograman Java*. Andi Offset
- KBBI Daring. Badan Bahasa Kemdikbud

F. Nested For 1

Program ini menggunakan perulangan bersarang (nested for) untuk mencetak pola kotak bintang berukuran 5x5. Kode berada dalam paket Pekan5 dan kelas nestedfor1 berisi metode main.

Struktur perulangan:

- Perulangan luar (i) mengatur jumlah baris (1 sampai 5)
- Perulangan dalam (j) mengatur jumlah kolom (1 sampai 5)
- System.out.print("\*") mencetak bintang tanpa pindah baris
- System.out.println() digunakan untuk pindah ke baris berikutnya setelah satu baris selesai

Hasil output:

```
***** ***** ***** ***** *****
```

Program ini menunjukkan cara membuat pola berbentuk persegi menggunakan perulangan bersarang, cocok untuk latihan logika visual dan kontrol struktur.

**Referensi:**

- Oracle Java Documentation: [Nested Loops](#)
- Nugroho, Adi. *Pemrograman Java*. Andi Offset
- KBBI Daring. Badan Bahasa Kemdikbud

## G. Nested For 2

Program ini menggunakan perulangan bersarang (nested for) untuk mencetak hasil penjumlahan dari dua variabel penghitung (i dan j). Kode berada dalam paket Pekan5 dan kelas NesteedFor2 berisi metode main.

Struktur perulangan:

- Perulangan luar (i) berjalan dari 0 sampai 5
- Perulangan dalam (j) juga berjalan dari 0 sampai 5
- System.out.print(i + j + " ") mencetak hasil penjumlahan i + j di satu baris
- System.out.println() digunakan untuk pindah ke baris baru setelah satu baris selesai

Contoh hasil output:

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

Program ini menunjukkan bagaimana nested loop dapat digunakan untuk membentuk tabel numerik atau simulasi operasi matematika dua dimensi.

**Referensi:**

- Oracle Java Documentation: [Nested Loops](#)
- Nugroho, Adi. *Pemrograman Java*. Andi Offset
- KBBI Daring. Badan Bahasa Kemdikbud

## H. Nested For 3

Program ini menggunakan perulangan bersarang (nested for) untuk mencetak pola titik (.) dari kiri dan angka di ujung kanan. Kode berada dalam paket Pekan5 dan kelas nestedFor3 berisi metode main.

Struktur perulangan:

- Variabel n = 5 menentukan jumlah baris
- Perulangan luar (line) berjalan dari 1 sampai 5
- Perulangan dalam mencetak titik sebanyak (-1 \* line + 5)

- Setelah titik, angka line dicetak di ujung kanan
- System.out.println() digunakan untuk pindah ke baris baru

Contoh hasil output:

....1 ...2 ..3 .4 5

Program ini cocok untuk latihan nested loop dan logika tampilan berbasis baris, serta melatih pemahaman aritmetika dalam kontrol perulangan.

**Referensi:**

- Oracle Java Documentation: [Nested Loops](#)
- Nugroho, Adi. *Pemrograman Java*. Andi Offset
- KBBI Daring. Badan Bahasa Kemdikbud

## BAB III

## PENUTUP

### 3.1 Kesimpulan

- Struktur perulangan for sangat efektif untuk mencetak data berulang, baik berupa angka, simbol, maupun pola visual.
- Penggunaan System.out.print() dan System.out.println() menentukan format tampilan output, apakah dalam satu baris atau berpindah baris.
- Perulangan bersarang (nested for) memungkinkan pembentukan pola dua dimensi seperti kotak, segitiga, atau bentuk simetris.
- Logika aritmetika seperti  $i + j$ ,  $(-1 * \text{line} + 5)$ , dan kondisi if memperkaya kontrol tampilan dan perhitungan dalam program.
- Integrasi dengan Scanner menunjukkan kemampuan program untuk menerima input dinamis dari pengguna.

Secara keseluruhan, sintaks-sintaks tersebut mencerminkan pemahaman dasar yang kuat terhadap struktur perulangan dan logika kontrol dalam Java.

### 3.2 Saran

Agar penguasaan materi semakin mantap, berikut beberapa saran:

- Lanjutkan eksplorasi ke **struktur perulangan lain** seperti while dan do-while untuk memahami perbedaan kontrol alur.
- Coba kombinasikan perulangan dengan **array** atau **fungsi** agar program lebih modular dan fleksibel.
- Latih pembuatan **pola dinamis** berdasarkan input pengguna, misalnya tinggi segitiga atau ukuran kotak.
- Gunakan komentar yang jelas dan konsisten agar kode mudah dipahami saat revisi atau debugging.

## **DAFTAR PUSTAKA**

- Nugroho, Adi. (2005). *Pemrograman Java*. Yogyakarta: Andi Offset.
- Badan Pengembangan dan Pembinaan Bahasa. (n.d.). *Kamus Besar Bahasa Indonesia (KBBI) Daring*. Diakses dari <https://kbbi.kemdikbud.go.id>
- Deitel, H. M., & Deitel, P. J. (2012). *Java: How to Program*. Pearson Education.
- Savitch, W. (2014). *Absolute Java*. Pearson.