

LAPORAN PRATIKUM

“PEKAN 5”

Disusun Untuk Memenuhi Tugas Mata Kuliah Algoritma Pemograman

DOSEN PENGAMPU:

Wahyudi, Dr. S.T. M.T.



DISUSUN OLEH:

Karimah Irsyadiyah (2411533018)

UNIVERSITAS ANDALAS

T.A 2024/2025

Daftar Pustaka

DOSEN PENGAMPU:	1
Wahyudi, Dr. S.T. M.T.	1
DISUSUN OLEH:	1
BAB I PENDAHULUAN.....	3
1.1 Latar Belakang	3
1.2 Tujuan	3
1.3 Alat dan Bahan.....	3
BAB II PEMBAHASAN	4
2.1 Langkah-langkah Praktikum di IDE Eclipse	4

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perulangan *for* adalah salah satu metode dalam bahasa Java yang digunakan untuk menjalankan proses pengulangan berdasarkan pola tertentu hingga kondisi yang ditentukan terpenuhi. Metode ini sangat berguna dalam berbagai situasi, seperti mencetak pola, menghitung nilai tertentu, atau menjalankan logika berulang secara efisien. Sebagai contoh, jika kita ingin membuat sebuah pola berbentuk piramida, kita dapat memanfaatkan perulangan *for* untuk menghasilkan pola tersebut dengan mudah.

1.2 Tujuan

Praktikum ini bertujuan untuk mempelajari dan memahami:

1. Memahami konsep dasar dari perulangan *for* dalam bahasa Java.
2. Mengetahui cara kerja, fungsi, dan implementasi perulangan *for* dalam membuat program.
3. Mengaplikasikan perulangan *for* secara efektif untuk menyelesaikan berbagai permasalahan dalam pemrograman.

1.3 Alat dan Bahan

- Perangkat Keras: Laptop/PC
- Perangkat Lunak:

Java Development Kit (JDK)

IDE Eclipse

BAB II

PEMBAHASAN

2.1 Langkah-langkah Praktikum di IDE Eclipse

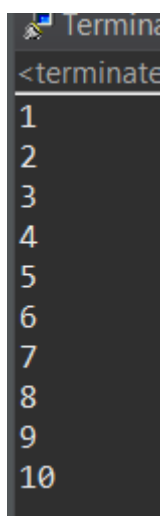
- **Perulangan For 1**

Pada contoh pertama, perulangan *for* digunakan untuk mencetak bilangan integer dari 1 hingga 10 secara vertikal. Berikut langkah-langkahnya:

- a. Buat *new package* dan *new class* seperti biasa.
- b. Buat dengan *PerulanganFor1*, lalu masukkan fungsi *for* untuk menginisiasi bilangan dari 1 hingga 10.
- c. Gunakan `System.out.println(i);` untuk mencetak output secara vertikal.
- d. Jalankan program untuk melihat hasilnya.

```
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  
12 }
```

Hasil Output: Bilangan dari 1 hingga 10 tercetak secara vertikal.



```
Terminal  
<terminated>  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

- **Perulangan For 2**

Pada contoh kedua, perulangan *for* digunakan hampir sama dengan contoh pertama. Perbedaannya terletak pada outputnya, di mana `System.out.println(i+" ");` digunakan untuk mencetak bilangan secara horizontal dengan spasi di antaranya.

```
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
12 }
```

Hasil Output: Bilangan dari 1 hingga 10 tercetak secara horizontal dengan spasi.

```
<terminated> PerulanganFor2 [Java Applic
1 2 3 4 5 6 7 8 9 10
```

- **Perulangan For 3**

Pada contoh ketiga, perulangan *for* digunakan untuk menjumlahkan bilangan yang sama dengan 1 tetapi kurang dari 10. Berikut langkah-langkahnya:

- a. Buat kelas baru seperti biasa.
- b. Masukkan fungsi *for* untuk menentukan kondisi.
- c. Tambahkan fungsi *if* untuk memastikan hanya bilangan tertentu yang dihitung.
- d. Gunakan `System.out.println` untuk mencetak hasil penjumlahan ke konsol.

```

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 }

```

Hasil Output: Penjumlahan bilangan dari 1 hingga 9 berhasil dicetak.

```

<terminated> PerulanganFor3 [Java Application] D:\Program
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 55

```

• Perulangan For 4

Pada contoh ini, perulangan *for* digunakan untuk menjumlahkan bilangan hingga angka yang dimasukkan oleh pengguna. Langkah-langkahnya:

- Buat kelas baru dan tambahkan *import java.util.Scanner* untuk menerima input dari pengguna.
- Inisiasi scanner dan masukkan angka yang diinginkan oleh pengguna.
- Gunakan fungsi *for* untuk melakukan penjumlahan hingga angka tersebut.
- Cetak hasilnya menggunakan *System.out.println*.

```

1 package pekan5;
2 import java.util.Scanner;
3
4 public class PerulanganFor4 {
5
6     public static void main(String[] args) {
7         int angka;
8         Scanner console = new Scanner(System.in);
9         System.out.print("Input angka berapa? : ");
10        angka = console.nextInt();
11        console.close();
12
13        int jumlah = 0;
14        for(int i=1; i<=angka; i++) {
15            System.out.print(i);
16            jumlah = jumlah + i;
17            if(i<angka) {
18                System.out.print(" + ");
19            }
20        }
21        System.out.println();
22        System.out.println("Jumlah = "+jumlah);
23
24    }
25

```

Hasil Output: Penjumlahan bilangan hingga angka yang dimasukkan oleh pengguna tercetak di konsol.

```

Input angka berapa? : 10
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah = 55

```

- **Perulangan For 5**

Pada contoh ini, perulangan *for* digunakan untuk mencetak pola tertentu. Berikut adalah kode program dan hasil output yang mencetak pola dengan struktur yang telah ditentukan.

```

1 package perkuliahan;
2
3 public class PerulanganFor5 {
4
5     public static void main(String[] args) {
6         for(int i = 1; i <= 5; i++) {
7             for(int j = 1; j <= (-1 * i + 5); j++) {
8                 System.out.print(".");
9             }
10            System.out.println(i);
11        }
12    }
13 }
14
15 }

```

```

<terminated>
....1
...2
..3
.4
5

```

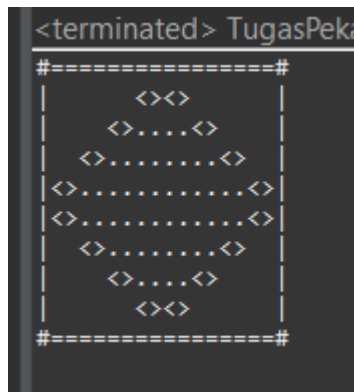
- **Tugas For**

Pada tugas ini, kita ditugaskan untuk membuat program menggunakan perulangan *for* yang menghasilkan pola berbentuk persegi dengan belah ketupat di dalamnya. Kode program ditulis untuk memenuhi kebutuhan tersebut dan menghasilkan pola sesuai desain.


```

1 package pekan5;
2
3 public class TugasPekan5BelahKetupat {
4     public static void main(String[] args) {
5         int a= 10;
6         int b;
7         int c;
8         c=a;
9
10        for(int i =1; i<=10;i++) {
11            b=c+1;
12            for(int j = 1; j<= 18;j++) {
13                if(i==1 || i==10) {
14                    if(j==1 || j==18 ) {
15                        System.out.print("#");
16                    }else {
17                        System.out.print("=");
18                    }
19                }
20                if(i>1 && i<10) {
21                    if(j==1 || j ==18) {
22                        System.out.print("|");
23                    }else {
24                        if(j == c ) {
25                            System.out.print("<");
26                        }else if(j== b) {
27                            System.out.print(">");
28                        }
29                        else if(j==(18-c)) {
30                            System.out.print("<");
31                        }else if(j==(19-c)) {
32                            System.out.print(">");
33                        }else if(j>b && j<(18-c)){
34                            System.out.print(".");
35                        }else{
36                            System.out.print(" ");
37                        }
38                    }
39                }
40
41            }
42            System.out.println();
43            a-=2;
44            if(a<=0) {
45                c=(-1*a)+2;
46            }else {
47                c=a;
48            }
49        }
50    }
51 }
52 }

```



1.4 Kesimpulan

Melalui praktikum kali ini, kita telah mempelajari konsep dasar, cara penggunaan, dan fungsi dari perulangan *for* dalam bahasa Java. Perulangan *for* terbukti sangat bermanfaat untuk menciptakan pola berulang, melakukan perhitungan hingga bilangan tertentu, serta menyelesaikan berbagai tugas pemrograman dengan efisien. Kombinasi perulangan *for* dengan fungsi *if* memungkinkan kita untuk menangani kondisi yang lebih kompleks, sehingga kode program menjadi lebih efektif dan fleksibel.