

## BAB IX

### PERULANGAN BERSARANG

---

#### 9.1. Pengantar

Perintah perulangan sangat bermanfaat karena banyak sekali persoalan yang dapat diselesaikan memakai perintah ini. Namun demikian, masih banyak persoalan lain yang hanya mungkin diselesaikan dengan efektif dengan cara menempatkan suatu perintah perulangan ke dalam perulangan yang lain. Perulangan yang berada dalam suatu perulangan disebut perulangan bersarang atau *nested loop*.

Ada banyak persoalan yang penyelesaiannya memakai perulangan bersarang yang bertingkat-tingkat lebih dari 2 tingkat. Paling mudah membayangkan perulangan bersarang ini adalah dengan melihat susunan *gear* (gigi) belakang sepeda balap. Contoh lain struktur perulangan bersarang yang mudah kita bayangkan adalah jarum jam non-digital yang memuat 3 tangkai penunjuk. Dalam jam model ini, ada 3 jarum penunjuk yakni penunjuk detik, menit dan jam. Setelah jarum penunjuk detik berputar sekali yakni 60 detik maka jarum menit baru akan bergeser ke kanan satu menit. Proses ini terus berulang sampai jarum menit berputar 360 derajat (satu lingkaran penuh) yang artinya 60 menit atau 1 jam. Setelah itu barulah jarum penunjuk jam bergeser 1 jam ke kanan. Memakai bahasa sehari-hari, perputaran 3 jarum jam tersebut dapat digambarkan sebagai berikut

Jarum jam berputar ke kanan 1 jam dari 1, 2, ..., 12

Jarum menit berputar ke kanan 1 menit, dari 1, 2, ..., 60

Jarum detik berputar ke kanan 1 detik, dari 1, 2, ..., 60

Dengan memakai perintah perulangan `for()`, simulasi perputaran 3 jarum jam dapat ditulis sebagai berikut:

```
for (int jam = 1; jam <= 12; jam++) {  
    System.out.println(jam);  
    for (int menit = 1; menit <= 60; menit++) {  
        System.out.println(menit);
```

```

        for (int detik = 1; detik <= 60; detik++) {
            System.out.println(detik);
        }
    }
}

```

Karena pada dasarnya semua perintah perulangan yakni **while**, **do-while** dan **for** dapat digunakan bersamaan maka pembentukan perulangan bersarang juga dapat dilakukan memakai ketiga perintah tersebut maupun campuran ketiganya.

## 9.2. Contoh Perulangan Bersarang

Salah satu contoh penggunaan perulangan bersarang yang mudah untuk dibayangkan adalah program untuk menampilkan daftar bilangan hasil perkalian antara bilangan yang berada di baris tertentu dengan semua bilangan yang berada di kolom tertentu. Berikut adalah contoh daftar bilangan tersebut.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
5	10	15	20	25	30	35	40	45	50

Daftar di atas dapat dibayangkan sebagai rangkaian 5 baris yang masing-masing terdiri atas 10 kolom di mana besarnya bilangan pada baris dan kolom tertentu merupakan hasil perkalian antara dua bilangan di baris dan kolom yang sesuai. Bilangan pada baris ke  $i$  dan kolom ke  $j$  adalah  $i * j$ . Untuk itu program yang sesuai untuk menampilkan daftar bilangan di atas adalah sebagai berikut:

```

for ( int kolom = 1; kolom <= 5; kolom++) {
    for (int baris = 1; baris <= 10; baris++)
        System.out.printf("%5d", baris * kolom);
    System.out.println();
}

```

### 9.3. Praktikum

Tulis, jalankan serta pahami beberapa program berikut:

#### 1. Blok bintang

```
package loop;
public class Loop1 {
    public static void main(String[] args) {
        for (int j =1; j<= 5; j++){
            for(int i=1; i<=5; i++)
                System.out.print("*");
            System.out.println();
        }
    }
}
```

#### 2. Blok bintang segitiga

```
package loop;
public class Loop2 {
    public static void main(String[] args) {
        for (int j =1; j<= 5; j++){
            for(int i=1; i<=j; i++)
                System.out.print("*");
            System.out.println();
        }
    }
}
```

#### 3. Blok bintang terbalik

```
package loop;
public class Loop 3 {
    public static void main(String[] args) {
        for (int j =1; j<= 10; j++){
            for(int i=1; i<= 10 - j; i++)System.out.print("*");
            System.out.println();
        }
    }
}
```

#### 4. Pohon bintang

```
package loop;
public class Loop4 {
    public static void main(String[] args) {
        for (int j=1; j<= 10; j=j+2){
            for(int k=8; k >= j; k=k-2) System.out.print(" ");
            for(int i=1; i<=j; i++)System.out.print("*");
            System.out.println();
        }
    }
}
```

#### 5. Tabel perkalian

```
package loop;
public class Loop5 {
    public static void main(String[] args) {
        for (int baris =1; baris<= 10; baris++){
            for(int kolom=1; kolom <= 10; kolom++)System.out.printf("%5d",
baris*kolom);
            System.out.println();
        }
    }
}
```

6. Buat program yang dapat menampilkan matrik identitas  $N \times N$  yakni  $N$  baris dan  $N$  kolom di mana  $N$  dimasukkan lewat keyboard. Matrik identitas adalah matrik yang semua elemennya berupa bilangan nol (0) kecuali elemen diagonal utama yang bernilai 1. Dengan kata lain jika  $a_{ij}$  adalah elemen baris ke  $i$  dan kolom ke  $j$  matrik maka  $a_{ij} = 0$  untuk  $i \neq j$  dan  $a_{ij} = 1$  bila  $i = j$ . Untuk matrik  $5 \times 5$  maka bentuknya adalah

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

7. Buat program yang dapat menampilkan diagram batang mendatar dari data usia penduduk yang berjumlah N di mana usia dan N dimasukkan lewat keyboard. Data penduduk tersebut disajikan memakai format berikut:

0 – 20 : \*\*\*  
21 – 40 : \*\*\*\*\*  
41 – 60 : \*\*  
61 – 80 : \*\*\*\*\*  
80 – 100 : \*\*

Di dalam diagram di atas, satu bintang mewakili satu penduduk yang berusia pada rentang data tersebut. Contoh input dan output program Anda adalah sbb:

Masukkan jumlah penduduk : 15

Usia warga 1 : 20  
Usia warga 2 : 25  
Usia warga 3 : 10  
Usia warga 4 : 45  
Usia warga 5 : 50  
Usia warga 6 : 70  
Usia warga 7 : 75  
Usia warga 8 : 60  
Usia warga 9 : 40  
Usia warga 10 : 30  
Usia warga 11 : 35  
Usia warga 12 : 15  
Usia warga 13 : 40  
Usia warga 14 : 50  
Usia warga 15 : 55

Distribusi usia 15 penduduk di atas adalah:

0 – 20 : \*\*\*  
21 – 40 : \*\*\*\*\*  
41 – 60 : \*\*\*\*\*  
61 – 80 : \*\*  
80 – 100 :

8. Buat program untuk menampilkan counter waktu seperti berikut

Detik : 1 2 3 4 5 .....60

Menit: 1

Detik : 1 2 3 4 5 .....60

Menit: 2

.....

.....

Menit: 60

Jam : 1

Detik : 1 2 3 4 5 .....60

.....

Demikian seterusnya dan program baru berhenti ketika jam mencapai 12.

9. Paket Statistik Usia Sederhana

Dengan memakai perintah do .... while () tampilkan menu berikut di layar

Menu:

1. Memasukkan data usia
2. Rata-rata
3. Deviasi standar
4. Diagram batang
5. Keluar

Selanjutnya di dalam do -while berisi program-program untuk keperluan 5 menu di atas.

Sebagai catatan, diagram batang yang anda buat menggunakan 5 kelompok data dan datanya harus dalam rentang 0 sampai 100. Artinya kelas datanya adalah

0 – 20 :

21 – 40 :

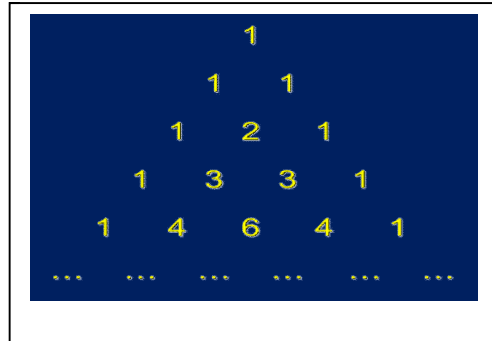
41 – 60 :

61 – 80 :

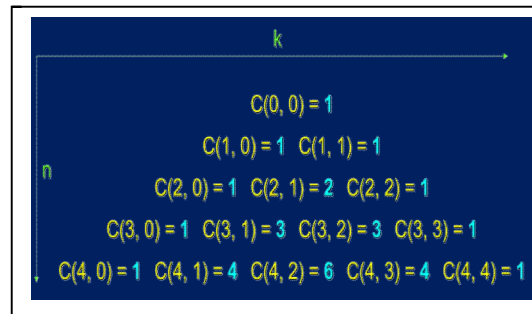
81 –100 :

## 10. Segitiga Pascal

Segitiga Pascal adalah daftar bilangan bulat yang disusun berdasarkan jumlah dua bilangan sebelah kiri dan kanan dari deret sebelumnya seperti dalam gambar di bawah



Angka-angka tersebut di atas juga dapat dihitung memakai rumus Kombinasi ( $C(n,r)$ ) seperti dalam diagram berikut



$C(n,r)$  dapat dihitung memakai rumus  $n! / [(n-r)! r!]$  di mana  $n! = 1 \times 2 \times 3 \times \dots \times n$

Sebagai contoh,  $C(4,2) = 4! / [(4-2)! 2!] = (1 \times 2 \times 3 \times 4) / [(1 \times 2) (1 \times 2)] = 6$

Tugas Anda adalah menampilkan bilangan dari segitiga Pascal untuk baris atau  $n$  tertentu yang dimasukkan lewat keyboard.

Jika nilai  $n$  adalah 3 maka program Anda menampilkan angka 1 3 3 1 dan angka-angka ini dapat dihitung masing-masing dengan  $C(3,0)$ ,  $C(3,1)$ ,  $C(3,2)$ ,  $C(3,3)$ .

Dengan kata lain, berdasarkan input besarnya  $n$ , program Anda harus menghitung dan menampilkan nilai dari  $C(n,0)$ ,  $C(n,1)$ ,  $C(n,2)$ , ...,  $C(n,(n-1))$ ,  $C(n,n)$ .

Jika Anda sudah selesai mengerjakan tugas ini, program dapat Anda modifikasi sehingga mampu menghasilkan semua bilangan dalam segitiga Pascal mulai dari  $n = 0$  sampai  $n$  bernilai tertentu yang dimasukkan lewat keyboard.