



JOBSHEET 7

Perulangan 2

1. Tujuan

- Mahasiswa memahami **konsep** perulangan bersarang
- Mahasiswa dapat menjelaskan format penulisan perulangan bersarang (*nested loop*)
- Mahasiswa dapat mengimplementasikan *flowchart* perulangan bersarang menggunakan bahasa pemrograman Java

2. Alat dan Bahan

- PC/Laptop
- JDK
- Java IDE

3. Praktikum

3.1 Percobaan 1: review perulangan yang lalu

1. Percobaan ini ditujukan me-review kembali perulangan yang sudah dibahas pada pertemuan sebelumnya. Pada percobaan 1 akan dibuat program untuk membuat tampilan * sebanyak N kali ke arah **samping**.
2. Buat class baru dengan nama **Star** dan simpan dalam file **StarNoAbsen.java**
3. Buat fungsi/method **main()** di dalamnya.
4. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner. Jadi tambahkan sintaks import di baris atas sendiri program.

```
import java.util.Scanner;
```

5. Di dalam fungsi **main()** yang telah dibuat, deklarasikan objek **Scanner** dengan nama **sc**.

```
Scanner sc = new Scanner(System.in);
```

6. Pada baris selanjutnya, tampilkan instruksi untuk memasukan nilai yang akan disimpan ke variabel **N**.

```
System.out.print("Masukkan nilai N = ");
int N = sc.nextInt();
```



7. Pada baris selanjutnya, buat sintaks perulangan dengan for seperti di bawah ini.

```
for(int i=1; i<=N; i++){
    System.out.print("*");
}
```

Catatan: perlu diperhatikan, bahwa yang digunakan adalah perintah **print**, bukan **println** karena kita ingin menampilkan tanpa ada baris baru

8. Compile dan jalan program!
9. Amati hasilnya, maka hasilnya harusnya akan serupa dengan tampilan di bawah ini.

```
Masukkan Nilai N = 5
*****
```

Pertanyaan

1. Jika pada perulangan for, inisialisasi **i=1** diubah menjadi **i=0**, apa yang akibatnya? Mengapa bisa demikian?
2. Jika pada perulangan for, kondisi **i <= N** diubah menjadi **i > N**, apa akibatnya? Mengapa bisa demikian?
3. Jika pada perulangan for, kondisi step **i++** diubah menjadi **i--** apa akibatnya? Mengapa bisa demikian?

3.2 Percobaan 2 : Bintang Persegi

1. Pada percobaan ke-2 akan dilakukan percobaan tentang *nested loop*. Kasus yang akan diselesaikan adalah untuk membuat tampilan bujursangkar *, dengan panjang sisi sebanyak N. Misalkan N dimasuki 5, maka hasilnya adalah:

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

2. Kalau diamati lebih lanjut, sebenarnya mirip dengan kasus percobaan 1 bukan? Jika di percobaan 1, misal input N bernilai 5, maka yang akan dihasilkan adalah ***** (kita bisa anggap ini sebagai **inner loop** yang mencetak 5 bintang *****), maka untuk



kasus percobaan 2 ini bukankah hasil dari percobaan 1 tersebut hanya perlu diulang lagi sebanyak N kali? (dengan menambahkan **outer loop** untuk mengulangi proses **inner loop** sebanyak N kali.)

4. Buat class Square dan simpan dengan nama file **SquareNoAbsen.java**
5. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner. Jadi tambahkan sintaks import di baris atas sendiri program.

```
import java.util.Scanner;
```

6. Buat method main(), dan isikan kode program yang sama dengan isi method main() di percobaan 1.

```
Scanner sc = new Scanner(System.in);
System.out.print("Masukkan nilai N = ");
int N = sc.nextInt();
for(int i=1; i<=N; i++){
    System.out.print("*");
}
```

7. Compile dan jalankan program! Dan pastikan program jalan seperti saat percobaan 1.
8. Perhatikan sintaks perulangan yang digunakan untuk mencetak * sebanyak N kali ke arah samping. Di step-6 di atas kode **for** (kota merah) kita jadikan sebagai **inner loop**.
9. Kita looping lagi inner loop sebanyak N kali untuk menghasilkan output seperti tahap 1. Maka perlu ditambahkan perulangan luar (**outer loop**).

```
for(int iOuter=1; iOuter<=N; iOuter++){
    for(int i=1; i<=N; i++){
        System.out.print("*");
    }
    System.out.println();
}
```

10. Simpan perubahan, compile dan jalankan program!
11. Amati hasilnya, maka hasilnya harusnya akan serupa dengan tampilan di bawah ini.

Masukkan Nilai N = 5

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



Pertanyaan

1. Perhatikan perulangan luar. Jika pada sintaks **for**, inisialisasi **iOuter=1** diubah menjadi **iOuter=0**, apa yang akibatnya? Mengapa bisa demikian?
2. Kembalikan program semula dimana inisialisasi **iOuter=1**. Kemudian perhatikan perulangan dalam, Jika pada sintaks **for**, inisialisasi **i=1** diubah menjadi **i=0**, apa yang akibatnya? Mengapa bisa demikian?
3. Jadi, apakah perbedaan kegunaan antara perulangan luar dengan perulangan yang berada didalamnya?
4. Mengapa perlu ditambahkan sintaks **System.out.println();** di bawah perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?
5. Silakan commit dan push ke repository Anda.

3.3 Percobaan 3: Mengisi dan menampilkan array

1. Buatlah kelas baru dengan nama NestedLoop_NIM
2. Buatlah fungsi main()
3. Dalam fungsi main() tambahkan deklarasi untuk Scanner dan deklarasi array 2 dimensi dengan jumlah baris 5 dan kolom 7.
4. Tambahkan kode baris seperti berikut

```
for (int i = 0; i < temps.length; i++) {
    System.out.println("Kota ke-" + i);
    for (int j = 0; j < temps[0].length; j++) {
        System.out.print("Hari ke-" + (j + 1) + ": ");
        temps[i][j] = scanner.nextDouble();
    }
    System.out.println();
}
```

5. Selanjutnya, tambahkan juga beberapa baris kode berikut

```
for (int i = 0; i < temps.length; i++) {
    System.out.print("Kota ke-" + (i + 1) + ": ");
    for (int j = 0; j < temps[0].length; j++) {
        System.out.print(temps[i][j] + " ");
    }
    System.out.println();
}
```

6. Compile dan jalankan program.
7. Amati jalannya alur program tersebut.

Catatan: Contoh program di atas adalah contoh kasus untuk menampung data suhu pada sebuah kota, cek kembali slide pada minggu lalu.

Pertanyaan

1. Jelaskan alur program di atas!



2. Silakan modifikasi program di atas pada bagian untuk menampilkan array menggunakan foreach!
3. Modifikasi program di atas sehingga bisa menampilkan nilai rata-rata masing-masing kota!
4. Silakan commit dan push ke repository Anda.

3.4 Percobaan 4: Mencari Nilai Terendah, Nilai Tertinggi, dan Menghitung Rata-rata pada Array 2 Dimensi

Waktu Percobaan: 90 menit

Pada percobaan ini dilakukan pembuatan kode program untuk mencari nilai terendah, nilai tertinggi, dan menghitung rata-rata dari data yang tersimpan di dalam array 2 dimensi. Data yang disimpan merupakan data hasil pemantauan peralatan klimatologi yang digunakan oleh BMKG untuk mencatat suhu, persentase kelembaban, kecepatan angin, dan index UV setiap tiga jam sekali dalam setengah hari (12 jam).

Data yang dicatat berdasarkan deskripsi tersebut ditunjukkan pada tabel berikut.

Jam	Suhu	Persentase kelembaban	Kecepatan angin	Index UV
3	23	100	0	0
6	24	100	1	0
9	32	83	4	5
12	33	78	5	7

1. Buka text editor. Buat file baru, beri nama **ArrayDuaNoAbsen.java**
2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi **main()**
3. Tambahkan library Scanner
4. Buat deklarasi **Scanner** dengan nama **scan**
5. Buat array bertipe integer dengan nama **data** dengan kapasitas baris 4 elemen dan kolom 4 elemen

```
int data[][] = new int[4][4];
```
6. Buat array 1 dimensi bertipe int dengan nama **jam** untuk menyimpan data jam

```
int jam[] = {3, 6, 9, 12};
```
7. Buat array 1 dimensi bertipe int dengan nama **min** untuk menyimpan data nilai minimum dari setiap hasil pencatatan parameter dan inisialisasi variabel tersebut dengan nilai 100. Catatan: *Inisialisasi nilai minimum tidak boleh 0 untuk menghindari kesalahan, jika data yang disimpan tidak ada yang lebih kecil dari 0*

```
int min[] = {100, 100, 100, 100};
```
8. Buat array 1 dimensi bertipe double dengan nama **total** dan **rata** dan inisialisasi variabel tersebut dengan nilai 0

```
double total[] = {0, 0, 0, 0};  
double rata[] = {0, 0, 0, 0};
```

9. Dengan menggunakan perulangan, buat input untuk mengisi elemen dari array **data**. Dalam hal ini digunakan **switch case** untuk membedakan informasi yang ditampilkan berkaitan dengan parameter suhu, persentase kelembaban, kecepatan angin, dan index UV.

```
for (int i = 0; i < data.length; i++) {  
    System.out.printf("Jam %s\n", jam[i]);  
    for (int j = 0; j < data[0].length; j++) {  
        switch (j) {  
            case 0:  
                System.out.print("Masukkan suhu: ");  
                data[i][j] = scan.nextInt();  
                break;  
            case 1:  
                System.out.print("Masukkan kelembaban: ");  
                data[i][j] = scan.nextInt();  
                break;  
            case 2:  
                System.out.print("Masukkan kecepatan angin: ");  
                data[i][j] = scan.nextInt();  
                break;  
            case 3:  
                System.out.print("Masukkan index UV: ");  
                data[i][j] = scan.nextInt();  
                break;  
        }  
    }  
}
```

10. Di dalam setiap **switch case**, tambahkan kondisi untuk membandingkan nilai minimum pada variabel **min** dengan data yang baru dimasukkan. Perhitungan variabel **total** juga dilakukan di dalam **switch case** ini

```
for (int i = 0; i < data.length; i++) {
    System.out.printf("Jam %s\n", jam[i]);
    for (int j = 0; j < data[0].length; j++) {
        switch (j) {
            case 0:
                System.out.print("Masukkan suhu: ");
                data[i][j] = scan.nextInt();
                if (min[j] > data[i][j]) {
                    min[j] = data[i][j];
                }
                total[j] += data[i][j];
                break;
            case 1:
                System.out.print("Masukkan kelembaban: ");
                data[i][j] = scan.nextInt();
                if (min[j] > data[i][j]) {
                    min[j] = data[i][j];
                }
                total[j] += data[i][j];
                break;
            case 2:
                System.out.print("Masukkan kecepatan angin: ");
                data[i][j] = scan.nextInt();
                if (min[j] > data[i][j]) {
                    min[j] = data[i][j];
                }
                total[j] += data[i][j];
                break;
            case 3:
                System.out.print("Masukkan index UV: ");
                data[i][j] = scan.nextInt();
                if (min[j] > data[i][j]) {
                    min[j] = data[i][j];
                }
                total[j] += data[i][j];
                break;
        }
        rata[j] = total[j] / data[0].length;
    }
}
```

Sementara itu, perhitungan rata-rata pada variabel **rata** dilakukan di luar **switch case** namun tetap di dalam inner loop

Perhatikan bahwa variabel yang digunakan sebagai indeks pada array **min** dan **total** adalah indeks **j** (bukan indeks **i**) karena pencarian nilai minimum dan perhitungan rata-rata berdasarkan kolom (bukan baris)

11. Dengan perulangan, tampilkan semua isi elemen dari array **data**. Tampilkan juga nilai rata-rata suhu dan persentase kelembaban terendah

```
System.out.println("-----");
for (int i = 0; i < data.length; i++) {
    for (int j = 0; j < data[0].length; j++) {
        System.out.print(data[i][j] + " ");
    }
    System.out.println("");
}
System.out.println("-----");
System.out.printf("Rata-rata suhu: %.2f\n", rata[0]);
System.out.printf("Persentase kelembaban terendah: %s\n", min[1]);
```

12. Jalankan program tersebut. Cocokkan hasil *running* program yang sudah Anda buat apakah sudah sesuai dengan tampilan berikut

```
Jam 3
Masukkan suhu: 23
Masukkan kelembaban: 100
Masukkan kecepatan angin: 0
Masukkan index UV: 0
Jam 6
Masukkan suhu: 24
Masukkan kelembaban: 100
Masukkan kecepatan angin: 1
Masukkan index UV: 0
Jam 9
Masukkan suhu: 32
Masukkan kelembaban: 83
Masukkan kecepatan angin: 4
Masukkan index UV: 5
Jam 12
Masukkan suhu: 33
Masukkan kelembaban: 78
Masukkan kecepatan angin: 5
Masukkan index UV: 7
-----
23 100 0 0
24 100 1 0
32 83 4 5
33 78 5 7
-----
Rata-rata suhu: 28.00
Persentase kelembaban terendah: 78
```

Pertanyaan!

1. Tambahkan kode program untuk menampilkan semua data hanya pada kolom kecepatan angin!
2. Modifikasi kode program untuk menentukan nilai maksimum dari setiap parameter suhu, persentase kelembaban, kecepatan angin, dan index UV!
3. Berdasarkan hasil modifikasi program sesuai soal nomor 2, tambahkan kode program untuk menampilkan nilai maksimum index UV!

4. Tugas individu dan kelompok

1. Buatlah program untuk mencetak tampilan segitiga angka seperti di bawah ini berdasarkan input N (nilai N minimal 3). Contoh N = 5

```

1
12
123
1234
12345
    
```

2. Buatlah program untuk mencetak tampilan persegi angka seperti di bawah ini berdasarkan input N (nilai N minimal 3). Contoh N = 3, dan N = 5

```

          5 5 5 5 5
          5      5
3 3 3      5      5
3   3      5      5
3 3 3      5 5 5 5 5
    
```

3. Tahun 2024 Politeknik Negeri Malang menjadi host event nasional Porseni, ada beberapa cabang olahraga yang dipertandingkan seperti **badminton, tenis meja, basket, dan bola voly**. Setiap cabor mengirimkan **5 atlet** terbaiknya dari seluruh politeknik seluruh Indonesia untuk mengikuti kegiatan 2 tahunan tersebut. Buatlah penyimpanan data untuk menampilkan **informasi nama atlet** dari berbagai cabang yang telah disebutkan dengan diurutkan secara **ascending**.

4. Terdapat sejumlah karakter yang tersimpan pada array satu dimensi bertipe char.

Data yang disimpan di dalam array tersebut sesuai **nama lengkap Anda**. Contoh:

```
char[] karakter = {'A', 'l', 'b', 'e', 'r', 't', 'E', 'i', 'n', 's',
                  't', 'e', 'i', 'n'};
```

Konversi variabel karakter tersebut ke dalam array 2 dimensi berukuran 8 baris 5 kolom dengan memanfaatkan loop dan/atau kondisi pemilihan. Apabila karakter pada nama lengkap Anda kurang dari 40, maka isi bagian yang kosong dengan karakter pertama, kedua, dan seterusnya sehingga semua indeks array menjadi terisi penuh seperti ilustrasi berikut.

A	l	b	e	r
t	E	i	n	s
t	e	i	n	A
l	b	e	r	t
E	i	n	s	t
e	i	n	A	l
b	e	r	t	E
i	n	s	t	e

Catatan: Anda tidak diperkenankan melakukan konversi secara manual



5. Implementasikan flowchart dari fitur-fitur yang telah Anda buat pada tugas teori sebelumnya tentang nested loop!
6. Jangan lupa, semoga kode program harus di-push ke repository Anda.