

JOB SHEET 9

Array 1

1. Tujuan

- Mahasiswa mampu memahami pembuatan Array 1 dimensi dan pengaksesan elemennya di Java
- Mahasiswa mampu membuat program dengan penggunaan Array satu dimensi

2. Praktikum

2.1 Percobaan 1: Mengisi Elemen Array

Waktu percobaan: 20 menit

1. Buat folder baru pada repositori lokal Anda, beri nama **jobsheet9**. Buat class Java baru dengan nama **ArrayBilanganXX.java**. (XX=nomor presensi)
2. Buat array bertipe integer dengan nama **bil** dengan kapasitas 4 elemen.

```
int[] bil = new int[4];
```

3. Isi masing-masing elemen array **bil** tadi dengan angka 5, 13, -7, 17.

```
bil[0] = 5;  
bil[1] = 13;  
bil[2] = -7;  
bil[3] = 17;
```

4. Tampilkan ke layar semua isi elemennya:

```
System.out.println(bil[0]);  
System.out.println(bil[1]);  
System.out.println(bil[2]);  
System.out.println(bil[3]);
```

5. Cocokkan dan amati hasilnya dengan gambar berikut ini:

```
5  
13  
-7  
17
```

6. Commit program Anda ke Github dengan pesan "Percobaan 1"

Pertanyaan

1. Jika isi masing-masing elemen array **bil** diubah dengan angka 5.0, 12867, 7.5, 2000000.
Apa yang terjadi? Mengapa bisa demikian?



2. Modifikasi kode program di atas dengan melakukan inisialisasi elemen array sekaligus pada saat deklarasi array.
3. Ubah statement pada langkah No 4 menjadi seperti berikut

```
for (int i = 0; i < 4; i++){
    System.out.println(bil[i]);
}
```

Apa keluaran dari program? Jelaskan maksud dari statement tersebut.

4. Jika kondisi pada statement for-loop di atas diubah menjadi: `i <= 4`, apa keluaran dari program? Mengapa demikian?
5. **Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 1"**

2.2 Percobaan 2: Studi Kasus Nilai Mahasiswa di SIAKAD - Meminta Inputan Pengguna untuk Mengisi Elemen Array

Waktu percobaan: 40 menit

1. Buat file Java kemudian simpan dengan nama ArrayNilaiXX.java. (XX=nomor presensi)
2. Buatlah struktur dasar java (membuat class dan method main).
3. Tambahkan import library Scanner.
4. Buat deklarasi array bertipe integer dengan nama nilaiAkhir dan berkapasitas 10 elemen seperti di bawah ini :

```
int[] nilaiAkhir = new int[10];
```

5. Buatlah struktur perulangan untuk menerima input dan mengisi elemen array nilaiAkhir, seperti berikut:

```
for (int i = 0; i < 10; i++){
    System.out.print("Masukkan nilai akhir ke-"+i+" : ");
    nilaiAkhir[i] = sc.nextInt();
}
```

6. Menggunakan struktur perulangan, tampilkan semua isi elemen dari array nilaiAkhir, seperti berikut:

```
for (int i = 0; i < 10; i++){
    System.out.println("Nilai akhir ke-"+i+" adalah "+nilaiAkhir[i]);
}
```

7. Jalankan program. Amati dan cocokkan dengan output berikut:

```
Masukkan nilai akhir ke-0 : 78
Masukkan nilai akhir ke-1 : 89
Masukkan nilai akhir ke-2 : 94
Masukkan nilai akhir ke-3 : 85
Masukkan nilai akhir ke-4 : 79
Masukkan nilai akhir ke-5 : 87
Masukkan nilai akhir ke-6 : 93
Masukkan nilai akhir ke-7 : 72
Masukkan nilai akhir ke-8 : 86
Masukkan nilai akhir ke-9 : 91
Nilai akhir ke-0 adalah 78
Nilai akhir ke-1 adalah 89
Nilai akhir ke-2 adalah 94
Nilai akhir ke-3 adalah 85
Nilai akhir ke-4 adalah 79
Nilai akhir ke-5 adalah 87
Nilai akhir ke-6 adalah 93
Nilai akhir ke-7 adalah 72
Nilai akhir ke-8 adalah 86
Nilai akhir ke-9 adalah 91
```

8. Commit program Anda ke Github dengan pesan “Percobaan 2”

Pertanyaan

1. Ubah statement pada langkah nomor 5 menjadi seperti berikut ini:

```
for (int i = 0; i < nilaiAkhir.length; i++){
    System.out.print("Masukkan nilai akhir ke-"+i+" : ");
    nilaiAkhir[i] = sc.nextInt();
}
```

Jalankan program. Apakah terjadi perubahan? Mengapa demikian?

2. Apa yang dimaksud dengan kondisi: `i < nilaiAkhir.length` ?

3. Ubah statement pada langkah nomor 6 menjadi seperti berikut ini, sehingga program hanya menampilkan nilai Mahasiswa yang lulus saja (yaitu mahasiswa yang memiliki nilai > 70):

```
for (int i = 0; i < nilaiAkhir.length; i++){
    if (nilaiAkhir[i] > 70){
        System.out.println("Mahasiswa ke-"+i+" lulus!");
    }
}
```

Jalankan program dan jelaskan alur program!



4. Modifikasi program agar menampilkan status kelulusan semua mahasiswa berdasarkan nilai, yaitu dengan menampilkan status mana mahasiswa yang lulus dan tidak lulus, seperti ilustrasi output berikut:

```
Masukkan nilai akhir ke-0 : 87
Masukkan nilai akhir ke-1 : 65
Masukkan nilai akhir ke-2 : 78
Masukkan nilai akhir ke-3 : 95
Masukkan nilai akhir ke-4 : 92
Masukkan nilai akhir ke-5 : 58
Masukkan nilai akhir ke-6 : 89
Masukkan nilai akhir ke-7 : 67
Masukkan nilai akhir ke-8 : 85
Masukkan nilai akhir ke-9 : 78
Mahasiswa ke-0 lulus!
Mahasiswa ke-1 tidak lulus!
Mahasiswa ke-2 lulus!
Mahasiswa ke-3 lulus!
Mahasiswa ke-4 lulus!
Mahasiswa ke-5 tidak lulus!
Mahasiswa ke-6 lulus!
Mahasiswa ke-7 tidak lulus!
Mahasiswa ke-8 lulus!
Mahasiswa ke-9 lulus!
```

5. **Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 2”**

2.3 Percobaan 3: Studi Kasus Nilai Mahasiswa di SIAKAD - Melakukan Operasi Aritmatika terhadap Elemen Array

Waktu percobaan: 75 menit

Pada percobaan ini, akan dilakukan percobaan untuk menjumlahkan array. Program akan menerima input sebanyak 10 nilai mahasiswa. Kemudian program akan menampilkan nilai rata-rata dari 10 mahasiswa.

1. Buat file Java, kemudian simpan dengan nama `ArrayRataNilaiXX.java`. (XX = nomor presensi).
2. Buatlah struktur dasar java (membuat class dan method main).
3. Import dan deklarasikan Scanner untuk keperluan input.
4. Buat array `nilaiMhs` bertipe integer dengan kapasitas 10. Kemudian deklarasikan variable `total` dan `rata2` seperti gambar berikut ini:

```
int[] nilaiMhs= new int[10];
double total = 0;
double rata2;
```

- Isi array nilaiMhs dengan nilai dari input pengguna, sebagai berikut:

```
for (int i = 0; i < nilaiMhs.length; i++){
    System.out.print("Masukkan nilai mahasiswa ke-"+(i+1)+" : ");
    nilaiMhs[i] = sc.nextInt();
}
```

- Gunakan perulangan untuk menghitung jumlah keseluruhan nilai dalam array nilaiMhs, sebagai berikut:

```
for (int i = 0; i < nilaiMhs.length; i++){
    total += nilaiMhs[i];
}
```

- Kemudian hitung nilai rata-rata dengan cara nilai total dibagi jumlah elemen dari array nilaiMhs:

```
rata2 = total/nilaiMhs.length;
System.out.println("Rata-rata nilai = "+rata2);
```

- Amati hasilnya sebagai berikut:

```
Masukkan nilai mahasiswa ke-1 : 80
Masukkan nilai mahasiswa ke-2 : 90
Masukkan nilai mahasiswa ke-3 : 87
Masukkan nilai mahasiswa ke-4 : 67
Masukkan nilai mahasiswa ke-5 : 58
Masukkan nilai mahasiswa ke-6 : 90
Masukkan nilai mahasiswa ke-7 : 78
Masukkan nilai mahasiswa ke-8 : 85
Masukkan nilai mahasiswa ke-9 : 70
Masukkan nilai mahasiswa ke-10 : 60
Rata-rata nilai = 76.5
```

- Commit program Anda ke Github dengan pesan "Percobaan 3"

Pertanyaan

- Modifikasi kode program pada praktikum percobaan 3 di atas (ArrayRataNilaiXX.java) agar program dapat menampilkan banyaknya mahasiswa yang lulus, yaitu mahasiswa yang memiliki lebih besar dari 70 (>70).
- Modifikasi program pada praktikum percobaan 3 di atas (ArrayRataNilaiXX.java) sehingga program menerima jumlah elemen berdasarkan input dari pengguna dan mengeluarkan output seperti berikut ini:



```
Masukkan jumlah mahasiswa : 5
Masukkan nilai mahasiswa ke-1 : 80
Masukkan nilai mahasiswa ke-2 : 60
Masukkan nilai mahasiswa ke-3 : 90
Masukkan nilai mahasiswa ke-4 : 85
Masukkan nilai mahasiswa ke-5 : 65
Rata-rata nilai lulus = 85.0
Rata-rata nilai tidak lulus = 62.5
```

3. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 3”

2.4 Percobaan 4: Searching

Waktu percobaan: 45 menit

1. Buka text editor, buat file Java, kemudian simpan dengan nama SearchNilaiXX.java. (XX = nomor absen).
2. Tambahkan kode berikut ini:

```
1  public class SearchNilai {
    Run | Debug
2  public static void main(String[] args) {
3      int[] arrNilai = {80, 85, 78, 96, 90, 82, 86};
4      int key = 90;
5      int hasil = 0;
6
7      for (int i = 0; i < arrNilai.length; i++){
8          if (key == arrNilai[i]){
9              hasil = i;
10             break;
11         }
12     }
13     System.out.println();
14     System.out.println("Nilai "+key+" ketemu di indeks ke-"+hasil);
15     System.out.println();
16 }
17 }
```

3. Jalankan program tersebut. Amati apa yang terjadi. Cocokkan dengan output berikut:

```
Nilai 90 ketemu di indeks ke-4
```

4. Commit program Anda ke Github dengan pesan “Percobaan 4”

Pertanyaan

1. Jelaskan maksud dari statement **break;** pada baris ke-10 kode program percobaan 4 di atas.



2. Modifikasi kode program pada percobaan 4 di atas sehingga program dapat menerima input berupa banyaknya elemen array nilai, isi array, dan sebuah nilai (key) yang ingin dicari. Lalu cetak ke layar indeks posisi elemen dari nilai (key) yang dicari. Contoh hasil program:

```
Masukkan banyaknya nilai yang akan diinput: 6
Masukkan nilai mahasiswa ke-1: 80
Masukkan nilai mahasiswa ke-2: 90
Masukkan nilai mahasiswa ke-3: 75
Masukkan nilai mahasiswa ke-4: 83
Masukkan nilai mahasiswa ke-5: 78
Masukkan nilai mahasiswa ke-6: 92
Masukkan nilai yang ingin dicari: 78
```

```
Nilai 78 ketemu, merupakan nilai mahasiswa ke-5
```

3. Modifikasi program pada percobaan 4 di atas, sehingga program akan memberikan pesan "Nilai yang dicari tidak ditemukan" jika nilai yang dicari (key) tidak ada di dalam array. Contoh tampilan program sebagai berikut:

```
Masukkan banyaknya nilai yang akan diinput: 6
Masukkan nilai mahasiswa ke-1: 80
Masukkan nilai mahasiswa ke-2: 90
Masukkan nilai mahasiswa ke-3: 75
Masukkan nilai mahasiswa ke-4: 82
Masukkan nilai mahasiswa ke-5: 95
Masukkan nilai mahasiswa ke-6: 70
Masukkan nilai yang ingin dicari: 85
```

```
Nilai yang dicari tidak ditemukan
```

4. **Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 4"**

3. Tugas

Waktu pengerjaan: 150 menit

1. Anda diminta untuk membuat program yang dapat menyimpan dan mengelola nilai mahasiswa. Nilai berupa bilangan bulat. Program harus menyediakan fitur untuk:
 - memasukkan banyaknya nilai mahasiswa yang akan diinput,
 - memasukkan setiap nilai mahasiswa,
 - menghitung nilai rata-rata,
 - menampilkan nilai tertinggi dan nilai terendah, serta



- menampilkan semua nilai yang telah dimasukkan.
- 2. Buat program yang dapat mengelola pemesanan makanan dan minuman di sebuah kafe. Program akan memungkinkan pengguna untuk memasukkan pesanan, menghitung total biaya pesanan, dan menampilkan daftar pesanan yang telah dibuat.
 - Input:
 - o jumlah pesanan (input dari pengguna).
 - o nama makanan/minuman dan harga untuk masing-masing pesanan (input dari pengguna)
 - Proses:
 - o simpan data pesanan dalam array satu dimensi untuk nama pesanan; dan array satu dimensi terpisah untuk harga.
 - o hitung total biaya dari semua pesanan yang dimasukkan.
 - o tampilkan daftar pesanan yang telah dimasukkan bersama dengan total biaya.
 - Output:
 - o daftar pesanan dan total biaya dari semua pesanan.
- 3. Masih menggunakan kasus pada pemesanan makanan di kafe, buatlah program yang memungkinkan pengguna untuk memesan makanan dari menu yang tersedia di kafe. Program harus menyimpan daftar nama makanan dalam sebuah array dan memberikan opsi untuk mencari makanan yang diinginkan menggunakan metode linear search.
 - Input:
 - o daftar menu makanan yang telah ditentukan sebelumnya dalam bentuk array. Nama-nama makanan telah di-inisialisasi saat deklarasi array. Misal:

```
String[] menu = {"Nasi Goreng", "Mie Goreng", "Roti Bakar", "Kentang Goreng", "Teh Tarik", "Cappucino", "Chocolate Ice"};
```
 - o nama makanan yang ingin dicari (input dari pengguna).
 - Proses:
 - o program mencari nama makanan yang dimasukkan pengguna menggunakan algoritma linear search.
 - o jika makanan ditemukan, program akan menginformasikan pengguna bahwa makanan tersebut tersedia. Jika tidak ditemukan, program akan memberi tahu pengguna bahwa makanan yang dicari tidak ada di menu.
 - Output:



-
- Tampilkan hasil pencarian kepada pengguna.