



## JOBSHEET III ARRAY OF OBJECTS

### 3.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Memahami dan menjelaskan fungsi array yang berisikan variabel object.
2. Mahasiswa mampu menerapkan instansiasi array of objects dalam Java
3. Mahasiswa mampu melakukan operasi terhadap elemen dalam array of objects

### 3.2 Membuat Array dari Object, Mengisi dan Menampilkan

Didalam praktikum ini, kita akan mempraktekkan bagaimana membuat array of objects, kemudian mengisi dan menampilkan array tersebut.

#### 3.2.1 Langkah-langkah Percobaan

1. Buat folder baru dengan nama Praktikum03.
2. Buat class **Mahasiswa**<NoPresensi>:

```
public class Mahasiswa {
    public String nim;
    public String nama;
    public String kelas;
    public float ipk;
}
```

3. Buat class **MahasiswaDemo**<NoPresensi> kemudian tambahkan fungsi main sebagai berikut;

```
public class MahasiswaDemo {
    public static void main(String[] args) {
        Mahasiswa[] arrayOfMahasiswa = new Mahasiswa[3];
    }
}
```

4. Kemudian isikan masing-masing atributnya:

```
public class MahasiswaDemo {
    public static void main(String[] args) {
        Mahasiswa[] arrayOfMahasiswa = new Mahasiswa[3];
        arrayOfMahasiswa[0] = new Mahasiswa();
        arrayOfMahasiswa[0].nim = "244107060033";
        arrayOfMahasiswa[0].nama = "AGNES TITANIA KINANTI";
        arrayOfMahasiswa[0].kelas = "SIB-1E";
        arrayOfMahasiswa[0].ipk = (float) 3.75;

        arrayOfMahasiswa[1] = new Mahasiswa();
        arrayOfMahasiswa[1].nim = "2341720172";
        arrayOfMahasiswa[1].nama = "ACHMAD MAULANA HAMZAH";
        arrayOfMahasiswa[1].kelas = "TI-2A";
        arrayOfMahasiswa[1].ipk = (float) 3.36;

        arrayOfMahasiswa[2] = new Mahasiswa();
        arrayOfMahasiswa[2].nim = "244107023006";
        arrayOfMahasiswa[2].nama = "DIRHAMAWAN PUTRANTO";
        arrayOfMahasiswa[2].kelas = "TI-2E";
        arrayOfMahasiswa[2].ipk = (float) 3.80;
    }
}
```

5. Cetak ke layar semua atribut dari objek `arrayOfMahasiswa`:

```
System.out.println("NIM      : "+ arrayOfMahasiswa[0].nim);
System.out.println("Nama      : "+ arrayOfMahasiswa[0].nama);
System.out.println("Kelas    : "+ arrayOfMahasiswa[0].kelas);
System.out.println("IPK      : "+ arrayOfMahasiswa[0].ipk);
System.out.println("-----");
System.out.println("NIM      : "+ arrayOfMahasiswa[1].nim);
System.out.println("Nama      : "+ arrayOfMahasiswa[1].nama);
System.out.println("Kelas    : "+ arrayOfMahasiswa[1].kelas);
System.out.println("IPK      : "+ arrayOfMahasiswa[1].ipk);
System.out.println("-----");
System.out.println("NIM      : "+ arrayOfMahasiswa[2].nim);
System.out.println("Nama      : "+ arrayOfMahasiswa[2].nama);
System.out.println("Kelas    : "+ arrayOfMahasiswa[2].kelas);
System.out.println("IPK      : "+ arrayOfMahasiswa[2].ipk);
System.out.println("-----");
```

6. Run program dan amati hasilnya.



### 3.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
NIM      : 244107060033
Nama     : AGNES TITANIA KINANTI
Kelas   : SIB-1E
IPK      : 3.75
-----
NIM      : 2341720172
Nama     : ACHMAD MAULANA HAMZAH
Kelas   : TI-2A
IPK      : 3.36
-----
NIM      : 244107023006
Nama     : DIRHAMAWAN PUTRANTO
Kelas   : TI-2E
IPK      : 3.8
-----
BUILD SUCCESSFUL (total time: 0 seconds)
```

### 3.2.3 Pertanyaan

1. Berdasarkan uji coba 3.2, apakah **class** yang akan dibuat **array of object** harus selalu memiliki **atribut** dan sekaligus **method**? Jelaskan!
2. Apa yang dilakukan oleh kode program berikut?

```
Mahasiswa[] arrayOfMahasiswa = new Mahasiswa[3];
```

3. Apakah class **Mahasiswa** memiliki konstruktor? Jika tidak, kenapa bisa dilakukan pemanggilan konstruktor pada baris program berikut?

```
arrayOfMahasiswa[0] = new Mahasiswa();
```

4. Apa yang dilakukan oleh kode program berikut?

```
arrayOfMahasiswa[0] = new Mahasiswa();
arrayOfMahasiswa[0].nim = "244107060033";
arrayOfMahasiswa[0].nama = "AGNES TITANIA KINANTI";
arrayOfMahasiswa[0].kelas = "SIB-1E";
arrayOfMahasiswa[0].ipk = (float) 3.75;
```

5. Mengapa class **Mahasiswa** dan **MahasiswaDemo** dipisahkan pada uji coba 3.2?

### 3.3 Menerima Input Isian Array Menggunakan Looping

Pada praktikum ini kita akan mengubah hasil program dari praktikum 3.2 sehingga program dapat menerima input dan menggunakan looping untuk mengisi atribut dari semua objek Mahasiswa

#### 3.3.1 Langkah-langkah Percobaan

1. Import scanner pada class **MahasiswaDemo**.

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

Note: Letakkan kode import dibawah kode package (jika ada).

2. Pada praktikum 3.2 poin nomor 4, modifikasi kode program sebagai berikut. Buat objek **Scanner** untuk menerima input, kemudian lakukan looping untuk menerima informasi panjang dan lebar:

```
Scanner sc = new Scanner(System.in);
Mahasiswa[] arrayOfMahasiswa = new Mahasiswa[3];
String dummy;

for(int i=0; i < 3; i++) {
    arrayOfMahasiswa[i] = new Mahasiswa();

    System.out.println("Masukkan Data Mahasiswa ke-" + (i + 1));
    System.out.print("NIM    : ");
    arrayOfMahasiswa[i].nim = sc.nextLine();
    System.out.print("Nama    : ");
    arrayOfMahasiswa[i].nama = sc.nextLine();
    System.out.print("Kelas : ");
    arrayOfMahasiswa[i].kelas = sc.nextLine();
    System.out.print("IPK    : ");
    dummy = sc.nextLine();
    arrayOfMahasiswa[i].ipk = Float.parseFloat(dummy);
    System.out.println("-----");
}
```

3. Pada praktikum 3.2 poin nomor 5, modifikasi kode program sebagai berikut. Lakukan looping untuk mengakses elemen array objek **mahasiswa** dan menampilkan informasinya ke layar:

```
for(int i=0; i < 3; i++) {
    System.out.println("Data Mahasiswa ke-" + (i + 1));
    System.out.println("NIM      : " + arrayOfMahasiswa[i].nim);
    System.out.println("Nama      : " + arrayOfMahasiswa[i].nama);
    System.out.println("Kelas    : " + arrayOfMahasiswa[i].kelas);
    System.out.println("IPK      : " + arrayOfMahasiswa[i].ipk);
    System.out.println("-----");
}
```

4. Run program dan amati hasilnya.

### 3.3.2 Verifikasi Hasil Percobaan

Contoh verifikasi hasil percobaan ini.

```
run:
Masukkan Data Mahasiswa ke-1
NIM : 244107060033
Nama : AGNES TITANIA KINANTI
Kelas : SIB-1E
IPK : 3.75
-----
Masukkan Data Mahasiswa ke-2
NIM : 2341720172
Nama : ACHMAD MAULANA HAMZAH
Kelas : TI-2A
IPK : 3.36
-----
Masukkan Data Mahasiswa ke-3
NIM : 244107023006
Nama : DIRHAMAWAN PUTRANTO
Kelas : TI-2E
IPK : 3.80
-----
Data Mahasiswa ke-1
NIM : 244107060033
Nama : AGNES TITANIA KINANTI
Kelas : SIB-1E
IPK : 3.75
-----
Data Mahasiswa ke-1
NIM : 244107060033
Nama : AGNES TITANIA KINANTI
Kelas : SIB-1E
IPK : 3.75
-----
Data Mahasiswa ke-2
NIM : 2341720172
Nama : ACHMAD MAULANA HAMZAH
Kelas : TI-2A
IPK : 3.36
-----
Data Mahasiswa ke-3
NIM : 244107023006
Nama : DIRHAMAWAN PUTRANTO
Kelas : TI-2E
IPK : 3.8
-----
BUILD SUCCESSFUL (total time: 1 minute 46 seconds)
```

### 3.3.3 Pertanyaan

1. Tambahkan method `cetakInfo()` pada class `Mahasiswa` kemudian modifikasi kode program pada langkah no 3.
2. Misalkan Anda punya `array baru` bertipe `array of Mahasiswa` dengan nama `myArrayOfMahasiswa`. Mengapa kode berikut menyebabkan error?

```
Mahasiswa[] myArrayOfMahasiswa = new Mahasiswa[3];
myArrayOfMahasiswa[0].nim = "244107060033";
myArrayOfMahasiswa[0].nama = "AGNES TITANIA KINANTI";
myArrayOfMahasiswa[0].kelas = "SIB-1E";
myArrayOfMahasiswa[0].ipk = (float) 3.75;
```

### 3.4 Constructor Berparameter

Pada praktikum ini kita akan melakukan instansiasi variable **array of object** dengan menggunakan **constructor** berparameter.

#### 3.4.1 Langkah-langkah Percobaan

1. Buatlah class baru dengan nama **Matakuliah<NoPresensi>** dengan **constructor** berparameter sebagai berikut;

```
public class Matakuliah {
    public String kode;
    public String nama;
    public int sks;
    public int jumlahJam;

    public Matakuliah(String kode, String nama, int sks, int jumlahJam) {
        this.kode = kode;
        this.nama = nama;
        this.sks = sks;
        this.jumlahJam = jumlahJam;
    }
}
```

2. Buatlah class baru **MatakuliahDemo<NoPresensi>** dan tambahkan fungsi **main()**. Kemudian sehingga instansiasi **array object Matakuliah** dilakukan menggunakan **constructor** berparameter sebagai berikut;

```
import java.util.Scanner;
public class MatakuliahDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Matakuliah[] arrayOfMatakuliah = new Matakuliah[3];
        String kode, nama, dummy;
        int sks, jumlahJam;

        for(int i=0; i < 3; i++) {
            System.out.println("Masukkan Data Matakuliah ke-" + (i + 1));
            System.out.print("Kode      : ");
            kode = sc.nextLine();
            System.out.print("Nama      : ");
            nama = sc.nextLine();
            System.out.print("Sks      : ");
            dummy = sc.nextLine();
            sks = Integer.parseInt(dummy);
            System.out.print("Jumlah Jam : ");
            dummy = sc.nextLine();
            jumlahJam = Integer.parseInt(dummy);
            System.out.println("-----");

            arrayOfMatakuliah[i] = new Matakuliah(kode, nama, sks, jumlahJam);
        }
    }
}
```

3. Run program kemudian amati hasilnya.

4. Modifikasi class **MatakuliahDemo** sehingga dapat menampilkan hasil inputan variable **array of object Matakuliah** di layer;

```
for(int i=0; i < 3; i++) {
    System.out.println("Data Matakuliah ke-" + (i + 1));
    System.out.println("Kode      : "+ arrayOfMatakuliah[i].kode);
    System.out.println("Nama      : "+ arrayOfMatakuliah[i].nama);
    System.out.println("Sks      : "+ arrayOfMatakuliah[i].sks);
    System.out.println("Jumlah Jam : "+ arrayOfMatakuliah[i].jumlahJam);
    System.out.println("-----");
}
```

5. Run program dan amati hasilnya.

### 3.4.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
Masukkan Data Matakuliah ke-1
Kode      : 12345
Nama      : Algoritma & Struktur Data
Sks      : 2
Jumlah Jam : 6
-----
Masukkan Data Matakuliah ke-2
Kode      : 54321
Nama      : Sistem Basis Data
Sks      : 2
Jumlah Jam : 4
-----
Masukkan Data Matakuliah ke-3
Kode      : 83652
Nama      : Dasar Pemrograman
Sks      : 2
Jumlah Jam : 4
-----
Data Matakuliah ke-1
Kode      : 12345
Nama      : Algoritma & Struktur Data
Sks      : 2
Jumlah Jam : 6
-----
Data Matakuliah ke-2
Kode      : 54321
Nama      : Sistem Basis Data
Sks      : 2
Jumlah Jam : 4
-----
Data Matakuliah ke-3
Kode      : 83652
Nama      : Dasar Pemrograman
Sks      : 2
Jumlah Jam : 4
-----
BUILD SUCCESSFUL (total time: 3 minutes 23 seconds)
```



### 3.4.3 Pertanyaan

1. Apakah suatu class dapat memiliki lebih dari 1 constructor? Jika iya, berikan contohnya
2. Tambahkan method `tambahData()` pada class `Matakuliah`, kemudian gunakan method tersebut di class `MatakuliahDemo` untuk menambahkan data Matakuliah
3. Tambahkan method `cetakInfo()` pada class `Matakuliah`, kemudian gunakan method tersebut di class `MatakuliahDemo` untuk menampilkan data hasil inputan di layar
4. Modifikasi kode program pada class `MatakuliahDemo` agar panjang (jumlah elemen) dari `array of object Matakuliah` ditentukan oleh user melalui input dengan Scanner

### 3.5 Tugas

1. Buatlah sebuah untuk menampilkan informasi tentang dosen. Program dapat menerima input semua informasi terkait dosen dan menampilkanya kembali ke layar. Program ini terdiri dari class `Dosen<NoPresensi>` dengan attribute/property sebagai berikut;

```
String kode
String nama
Boolean jenisKelamin
int usia
```

dengan method constructor sebagai berikut;

```
public dosen(String kode, String nama, Boolean jenisKelamin, int usia) {
    .....;
    .....;
}
```

Kemudian buatlah class `DosenDemo<NoPresensi>` untuk proses input dan menampilkan data beberapa dosen. Gunakan looping dengan `FOR` untuk pembuatan `array of object`. Gunakan looping dengan `FOREACH` untuk menampilkan data ke layar.





Contoh output sebagai berikut:

```
run:
Masukkan Data Dosen ke-1
Kode      : 05032001
Nama      : Aisyah Kamila
Jenis Kelamin : Wanita
Usia      : 24
-----
Masukkan Data Dosen ke-2
Kode      : 06041995
Nama      : Akmal Ahmad Ghozali
Jenis Kelamin : Pria
Usia      : 30
-----
Masukkan Data Dosen ke-3
Kode      : 23061975
Nama      : Wahyuningtyas
Jenis Kelamin : Wanita
Usia      : 50
-----
Data Dosen ke-1
Kode      : 05032001
Nama      : Aisyah Kamila
Jenis Kelamin : Wanita
Usia      : 24
-----
Data Dosen ke-2
Kode      : 06041995
Nama      : Akmal Ahmad Ghozali
Jenis Kelamin : Pria
Usia      : 30
-----
Data Dosen ke-3
Kode      : 23061975
Nama      : Wahyuningtyas
Jenis Kelamin : Wanita
Usia      : 50
-----
BUILD SUCCESSFUL (total time: 2 minute)
```

2. Tambahkan class baru **DataDosen**<**NoPresensi**> dengan beberapa method berikut;
  - a. **dataSemuaDosen**(**Dosen[] arrayOfDosen**) untuk menampilkan data semua dosen
  - b. **jumlahDosenPerJenisKelamin**(**Dosen[] arrayOfDosen**) untuk menampilkan data jumlah dosen per jenis kelamin (Pria / Wanita)
  - c. **rerataUsiaDosenPerJenisKelamin**(**Dosen[] arrayOfDosen**) untuk menampilkan rata-rata usia dosen per jenis kelamin (Pria / Wanita)
  - d. **infoDosenPalingTua**(**Dosen[] arrayOfDosen**) untuk menampilkan data dosen paling tua
  - e. **infoDosenPalingMuda**(**Dosen[] arrayOfDosen**) untuk menampilkan data dosen paling muda

Semua method tersebut harus bisa dipanggil/ditest dari class **DosenDemo**