

Nama : Achmad Pradita Dwi Firmansyah

NIM : 254107020130

Kelas / Absen : TI-1G / 01

JOBSITE III ARRAY OF OBJECTS

3.1 Membuat Array dari Object, Mengisi dan Menampilkan

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

3.1.1 Kode Program

```
public class MahasiswaDemo01 {  
    public static void main(String[] args) {  
        Mahasiswa01 [] arrayOfMahasiswa = new Mahasiswa01[3];  
        arrayOfMahasiswa[0] = new Mahasiswa01();  
        arrayOfMahasiswa[0].nim = "244107060033";  
        arrayOfMahasiswa[0].nama = "AGNES TITANIA KINANTI";  
        arrayOfMahasiswa[0].kelas = "SIB-1E";  
        arrayOfMahasiswa[0].ipk = 3.75;  
  
        arrayOfMahasiswa[1] = new Mahasiswa01();  
        arrayOfMahasiswa[1].nim = "2341720172";  
        arrayOfMahasiswa[1].nama = "ACHMAD MAULANA HAMZAH";  
        arrayOfMahasiswa[1].kelas = "TI-2A";  
        arrayOfMahasiswa[1].ipk = 3.36;  
  
        arrayOfMahasiswa[2] = new Mahasiswa01();  
        arrayOfMahasiswa[2].nim = "244107023006";  
        arrayOfMahasiswa[2].nama = "DIRHAMAWAN PUTRANTO";  
        arrayOfMahasiswa[2].kelas = "TI-2E";  
        arrayOfMahasiswa[2].ipk = 3.80;  
  
        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("-----");  
    }  
}
```

}

3.1.2 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)
```

```
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
```

3.1.3 Pertanyaan

1. Berdasarkan uji coba 3.2, apakah **class** yang akan dibuat **array of object** harus selalu memiliki **atribut** dan sekaligus **method**? Jelaskan!

Tidak, class tidak harus memiliki atribut dan juga method, class bisa saja hanya berisi atribut sudah cukup digunakan untuk membuat array of object.

2. Apa yang dilakukan oleh kode program berikut?

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

Kode tersebut merupakan deklarasi dan instansiasi array of object untuk membuat array of mahasiswa

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

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

Tidak, karena jika tidak ada konstruktor, pada java sudah terdapat konstruktor default sehingga kita dapat melakukan pemanggilan baris program tersebut.

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;
```

- Baris pertama kode program tersebut adalah pembuatan object Mahasiswa pada arrayOfMahasiswa pada indeks ke-0, kemudian dilakukan pengisian elemen array dengan menuliskan arrayOfMahasiswa[] . atributnya
5. Mengapa class **Mahasiswa** dan **MahasiswaDemo** dipisahkan pada uji coba 3.2?
Karena class MahasiswaDemo berisi fungsi main sehingga kita bisa membuat program lebih rapi, dan class Mahasiswa masih bisa digunakan pada class lain.

3.2 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 mengisikan atribut dari semua objek Mahasiswa

3.2.1 Kode Program

```
import java.util.Scanner;

public class MahasiswaDemo01 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Mahasiswa01[] arrayOfMahasiswa = new Mahasiswa01[3];
        String dummy;

        for (int i = 0; i < 1; i++) {
            arrayOfMahasiswa[i] = new Mahasiswa01();
            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("-----");
        }
        for (int i = 0; i < 1; 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.printf("IPK : %.2f \n", arrayOfMahasiswa[i].ipk);
            System.out.println("-----");
        }
    }
}
```

3.2.2 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-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)
```

```

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-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      : 3,80
-----
```

3.2.3 Pertanyaan

1. Tambahkan method `cetakInfo()` pada class `Mahasiswa` kemudian modifikasi kode program pada langkah no 3.

Class Mahasiswa :

```

void cetakInfo() {
    for (int i = 0; i < 3; i++) {
        System.out.println("Data Mahasiswa ke-" + (i + 1));
        System.out.println("NIM   : " + nim);
        System.out.println("Nama  : " + nama);
        System.out.println("Kelas : " + kelas);
        System.out.printf("IPK  : %.2f \n", ipk);
        System.out.println("-----");
    }
}
```

Class MahasiswaDemo :

```

for (int i = 0; i < 3; i++) {
    arrayOfMahasiswa[i].cetakInfo();
}
```

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;
```

Kode tersebut menyebabkan error karena belum dilakukan instansiasi object nya terlebih dahulu, sehingga program akan error memunculkan NullPointerException.

3.3 Constructor Berparameter

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

3.3.1 Kode Program

Class MataKuliah

```
public class MataKuliah01 {  
    public String kode;  
    public String nama;  
    public int sks;  
    public int jumlahJam;  
    public MataKuliah01 (String kode, String nama, int sks, int jumlahJam) {  
        this.kode = kode;  
        this.nama = nama;  
        this.sks = sks;  
        this.jumlahJam = jumlahJam;  
    }  
}
```

Class MataKuliahDemo

```
import java.util.Scanner;  
public class MataKuliahDemo01 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        MataKuliah01[] arrayOfMataKuliah01 = new MataKuliah01[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("-----");  
            arrayOfMataKuliah01[i] = new MataKuliah01(kode, nama, sks, jumlahJam);  
        }  
        for (int i = 0; i < 3; i++) {  
            System.out.println("Data Matakuliah ke-"+(i+1));  
            System.out.println("Kode      : " + arrayOfMataKuliah01[i].kode);  
            System.out.println("Nama      : " + arrayOfMataKuliah01[i].nama);  
        }  
    }  
}
```

```
        System.out.println("Sks      : " + arrayOfMataKuliah01[i].skls);
        System.out.println("Jumlah Jam : " + arrayOfMataKuliah01[i].jumlahJam);
        System.out.println("-----");
    }
}
}
```

3.3.2 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)
```

```
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 Pemograman
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 Pemograman
SkS       : 2
Jumlah Jam : 4
```

3.3.3 Pertanyaan

- Apakah suatu class dapat memiliki lebih dari 1 constructor? Jika iya, berikan contohnya

Class dapat memiliki lebih dari 1 konstruktor seperti konstruktor default dan konstruktor berparameter. Contoh: Public MataKuliah () dan public MataKuliah01 (String kode, String nama, int sks, int jumlahJam)

- Tambahkan method `tambahData()` pada class `MataKuliah`, kemudian gunakan method tersebut di class `MataKuliahDemo` untuk menambahkan data MataKuliah

Class MataKuliah :

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

Class MataKuliahDemo :

```
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("-----");  
    arrayOfMataKuliah01[i] = new MataKuliah01(kode, nama, sks, jumlahJam);  
}
```

- Tambahkan method `cetakInfo()` pada class `MataKuliah`, kemudian gunakan method tersebut di class `MataKuliahDemo` untuk menampilkan data hasil inputan di layar

Class MataKuliah:

```
void cetakInfo() {  
    System.out.println("Kode : " + kode);  
    System.out.println("Nama : " + nama);  
    System.out.println("Sks : " + sks);  
    System.out.println("Jumlah Jam : " + jumlahJam);  
    System.out.println("-----");  
}
```

Class MataKuliahDemo:

```
for (int i = 0; i < 3; i++) {  
    System.out.println("Data MataKuliah ke-" + (i + 1));  
    arrayOfMataKuliah01[i].cetakInfo();  
}
```

```
}
```

4. Modifikasi kode program pada class **MatakuliahDemo** agar panjang (jumlah elemen) dari **array of object Matakuliah** ditentukan oleh user melalui input dengan Scanner Class Matakuliah:

```
public class Matakuliah01 {  
    public String kode;  
    public String nama;  
    public String dummy;  
    public int sks;  
    public int jumlahJam;  
    public Matakuliah01() {  
    }  
    public Matakuliah01(String kode, String nama, int sks, int jumlahJam) {  
        this.kode = kode;  
        this.nama = nama;  
        this.sks = sks;  
        this.jumlahJam = jumlahJam;  
    }  
    void tambahData(String kode, String nama, int sks, int jumlahJam) {  
        this.kode = kode;  
        this.nama = nama;  
        this.sks = sks;  
        this.jumlahJam = jumlahJam;  
    }  
    void cetakInfo() {  
        System.out.println("Kode : " + kode);  
        System.out.println("Nama : " + nama);  
        System.out.println("Sks : " + sks);  
        System.out.println("Jumlah Jam : " + jumlahJam);  
        System.out.println("-----");  
    }  
}
```

Class MatakuliahDemo:

```
import java.util.Scanner;  
public class MatakuliahDemo01 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Masukkan Jumlah Matakuliah : ");  
        int jumlahMk = sc.nextInt();  
        sc.nextLine();  
        Matakuliah01[] arrayOfMataKuliah01 = new Matakuliah01[jumlahMk];  
        String kode, nama, dummy;  
        int sks, jumlahJam;  
        for (int i = 0; i < jumlahMk; 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("-----");
arrayOfMataKuliah01[i] = new Matakuliah01(kode, nama, sks, jumlahJam);
}
for (int i = 0; i < jumlahMk; i++) {
    System.out.println("Data Matakuliah ke-" + (i + 1));
    arrayOfMataKuliah01[i].cetakInfo();
}
}
```