

Laporan Praktikum ASD

Nama: Ahmad Adrian Ramadhan

NIM: 254107020118

Kelas: TI-1G

1. 3.2 Membuat dan Menampilkan *array of object*

Tujuan: Membuat dan menampilkan *array* yang berisi objek beserta atributnya
Praktikum ini berisi dua *class*, isi *class* pertama yang berjudul “Mahasiswa” adalah sebagai berikut:

```
package Praktikum3;

public class mahasiswa2 {
    public String nim;
    public String nama;
    public String kelas;
    public float ipk;
}
```

Sedangkan *class* kedua yang berisi *main* berjudul “Mahasiswa Demo” adalah sebagai berikut:

```
package Praktikum3;

public class mahasiswaDemo2 {
    public static void main(String[] args) {
        mahasiswa2[] arayofMahasiswa = new mahasiswa2[3];

        arayofMahasiswa[0] = new mahasiswa2();
        arayofMahasiswa[0].nim = "254822310";
        arayofMahasiswa[0].nama = "Prabu Pranotocoro";
        arayofMahasiswa[0].kelas = "TI-1F";
        arayofMahasiswa[0].ipk = 3.80f;

        arayofMahasiswa[1] = new mahasiswa2();
        arayofMahasiswa[1].nim = "254738137492";
        arayofMahasiswa[1].nama = "Ahmad Maulana";
        arayofMahasiswa[1].kelas = "TI-1H";
        arayofMahasiswa[1].ipk = 3.54f;

        arayofMahasiswa[2] = new mahasiswa2();
        arayofMahasiswa[2].nim = "254313718432";
        arayofMahasiswa[2].nama = "Purtanto";
    }
}
```

```

        arayofMahasiswa[2].kelas = "SIB-1D";
        arayofMahasiswa[2].ipk = 3.92f;

        System.out.println("NIM:      " + arayofMahasiswa[0].nim);
        System.out.println("Nama:     " + arayofMahasiswa[0].nama);
        System.out.println("Kelas:    " + arayofMahasiswa[0].kelas);
        System.out.println("IPK:      " + arayofMahasiswa[0].ipk);
        System.out.println("=====

");
        System.out.println("NIM:      " + arayofMahasiswa[1].nim);
        System.out.println("Nama:     " + arayofMahasiswa[1].nama);
        System.out.println("Kelas:    " + arayofMahasiswa[1].kelas);
        System.out.println("IPK:      " + arayofMahasiswa[1].ipk);
        System.out.println("=====

");
        System.out.println("NIM:      " + arayofMahasiswa[2].nim);
        System.out.println("Nama:     " + arayofMahasiswa[2].nama);
        System.out.println("Kelas:    " + arayofMahasiswa[2].kelas);
        System.out.println("IPK:      " + arayofMahasiswa[2].ipk);
        System.out.println("=====

");
    }
}

```

Ketika program dijalankan, maka output yang tampil adalah sebagai berikut:

```

NIM:      254822310
Nama:     Prabu Pranotocoro
Kelas:    TI-1F
IPK:      3.8
=====
NIM:      254738137492
Nama:     Ahmad Maulana
Kelas:    TI-1H
IPK:      3.54
=====
NIM:      254313718432
Nama:     Purtanto
Kelas:    SIB-1D
IPK:      3.92
=====

```

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[] arayofMahasiswa = new mahasiswa[3];
```

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

```
arayofMahasiswa[0] = new mahasiswa2();
```

- Apa yang dilakukan oleh kode program berikut?

```
arayofMahasiswa[0] = new mahasiswa2();
arayofMahasiswa[0].nim = "254822310";
arayofMahasiswa[0].nama = "Prabu Pranotocoro";
arayofMahasiswa[0].kelas = "TI-1F";
arayofMahasiswa[0].ipk = 3.80f;
```

- Mengapa class Mahasiswa dan MahasiswaDemo dipisahkan pada uji coba 3.2?

Jawab

- Umumnya, class tidak harus memiliki method maupun atribut atau disebut class kosong yang berfungsi sebagai *placeholder*. Namun, untuk menampilkan output class perlu setidaknya satu atribut untuk ditampilkan.
- Kode berfungsi untuk membuat array dari objek “Mahasiswa2” sebanyak tiga indeks.
- Tidak ada konstruktor, Karena java akan membuat konstruktor default secara otomatis jika tidak ada konstruktor
- Kode berfungsi untuk mengisi atribut pada objek di indeks ke-0 pada array
- Untuk membuktikan jika class yang berisi objek dapat diakses meskipun berbeda class

3.3 Menerima input isian array dengan looping

Tujuan: mengubah hasil program dari praktikum 3.2 sehingga program dapat menerima input dan menggunakan looping untuk mengisikan atribut dari semua objek Mahasiswa

Modifikasi terdapat pada class “MahasiswaDemo2” yang mengubah data yang bersifat statis menjadi dinamis(dapat menerima input) dan menggunakan perulangan sehingga kode program menjadi lebih pendek. Kode program baru adalah sebagai berikut:

```
package Praktikum3;

import java.util.Scanner;

public class mahasiswaDemo2 {
    public static void main(String[] args) {
        mahasiswa2[] arayofMahasiswa = new mahasiswa2[3];
        Scanner scanner = new Scanner(System.in);
        String dummy = null;

        for (int i = 0; i < arayofMahasiswa.length; i++) {
            arayofMahasiswa[i] = new mahasiswa2();
            System.out.println("Masukkan data mahasiswa ke-"+
                +(i+1));
        }
    }
}
```

```

        System.out.print("NIM: ");
        arayofMahasiswa[i].nim = scanner.nextLine();

        System.out.print("Nama:   ");
        arayofMahasiswa[i].nama = scanner.nextLine();

        System.out.print("Kelas:   ");
        arayofMahasiswa[i].kelas = scanner.nextLine();

        System.out.print("IPK:   ");
        dummy = scanner.nextLine();
        arayofMahasiswa[i].ipk = Float.parseFloat(dummy);
        System.out.println("=====");
    }

    scanner.nextLine();
}
mahasiswa2.ctkInf(arayofMahasiswa, 3);

scanner.close();

}
}

```

Tampilan output pertama akan menanyakan data yang akan dimasukkan. Output yang tampil sebagai berikut

```

Masukkan data mahasiswa ke-1
NIM: 736513
Nama: ABC
Kelas: TI-1K
IPK: 3.33
=====

Masukkan data mahasiswa ke-2
NIM: 723682354
Nama: GHI
Kelas: SIB-1N
IPK: 3.56
=====

Masukkan data mahasiswa ke-3
NIM: 328652323
Nama: JKL
Kelas: TI-1R
IPK: 3.77
=====
```

Setelah menginput data, array akan tampil seperti berikut

```
NIM:    736513
Nama:   ABC
Kelas:  TI-1K
IPK:    3.33
=====
NIM:    723682354
Nama:   GHI
Kelas:  SIB-1N
IPK:    3.56
=====
NIM:    328652323
Nama:   JKL
Kelas:  TI-1R
IPK:    3.77
=====
```

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?

Jawab:

1. Class Mahasiswa2:

```
public static void ctkInf(mahasiswa2[] arayofMahasiswa ,int pgj){
    for (int i = 0; i < pgj; i++) {
        System.out.println("NIM:      " + arayofMahasiswa[i].nim);
        System.out.println("Nama:     " + arayofMahasiswa[i].nama);
        System.out.println("Kelas:    " + arayofMahasiswa[i].kelas);
        System.out.println("IPK:      " + arayofMahasiswa[i].ipk);
        System.out.println("=====");
    }
}
```

Class Mahasiswa Demo2:

```
mahasiswa2.ctkInf(arayofMahasiswa, 3);
```

2. Karena referensi array pada java harus jelas, tidak boleh ada perbedaan nama. Java menganggap “arrayofMahasiswa” dan “myArrayOfMahasiswa” adalah dua array yang berbeda sehingga error pun terjadi.

3.4 Constructor berparameter

Tujuan: Melakukan inisialisasi array objek dengan konstruktor berparameter

Konstruktor berada pada class “Matakuliah2” dengan kode sebagai berikut:

```
package Praktikum3;

public class matakuliah2 {
    public String kode;
    public String nama;
    public int sks;
    public int jmlJam;

    public matakuliah2(String kode, String nama, int sks, int jmlJam){
        this.kode = kode;
        this.nama = nama;
        this.sks = sks;
        this.jmlJam = jmlJam;
    }
}
```

Konstruktor akan dipanggil di class “mataKuliahDemo2” dengan kode sebagai berikut:

```
package Praktikum3;

import java.util.Scanner;

public class mahasiswaDemo2 {
    public static void main(String[] args) {
        mahasiswa2[] arayofMahasiswa = new mahasiswa2[3];
        Scanner scanner = new Scanner(System.in);
        String dummy = null;

        for (int i = 0; i < arayofMahasiswa.length; i++) {
            arayofMahasiswa[i] = new mahasiswa2();
            System.out.println("Masukkan data mahasiswa ke- "+(i+1));
            System.out.print("NIM: ");
            arayofMahasiswa[i].nim = scanner.nextLine();

            System.out.print("Nama:   ");
            arayofMahasiswa[i].nama = scanner.nextLine();

            System.out.print("Kelas:   ");
            arayofMahasiswa[i].kelas = scanner.nextLine();

            System.out.print("IPK:   ");
            dummy = scanner.nextLine();
        }
    }
}
```

```
        arayofMahasiswa[i].ipk = Float.parseFloat(dummy);
        System.out.println("=====");
    }

    scanner.nextLine();
}
mahasiswa2.ctkInf(arayofMahasiswa, 3);

scanner.close();

}
}
```

Nama konstruktor harus sama persis dengan nama class, jika tidak, java akan menganggap konstruktor sebagai method biasa yang dapat menyebabkan error

Hal pertama yang dilakukan program adalah menginputkan data. Tampilan terminal akan menjadi seperti berikut:

```
Masukkan Data Kuliah ke-1
Kode:      opo
Nama:      uerygfdg
SKS:       2
Jumlah Jam: 4
Masukkan Data Kuliah ke-2
Kode:      jkl
Nama:      uhdfgdfhdf
SKS:       5
Jumlah Jam: 10
Masukkan Data Kuliah ke-3
Kode:      mnk
Nama:      fdgndffbvc
SKS:       2
Jumlah Jam: 4
```

Setelah menginputkan tiga data mata kuliah, program akan menampilkan data yang telah diinput. Data yang ditampilkan akan menjadi seperti berikut:

```
Kode:      opo
nama:     uverygfdg
SKS:      2
Jumlah jam: 4
=====
data mata kuliah ke-2
Kode:      jkl
nama:     uhdfgdfhdf
SKS:      5
Jumlah jam: 10
=====
data mata kuliah ke-2
Kode:      jkl
nama:     uhdfgdfhdf
SKS:      5
Jumlah jam: 10
```

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

Jawab:

1. Karena dalam bahasa pemrograman java terdapat suatu konsep yang dikenal dengan "*constructor overloading*". Dalam java, satu class bisa memiliki konstruktor yang banyak agar memudahkan pembuatan objek dengan kondisi yang berbeda. Contohnya saat penyimpanan data mata kuliah, dengan constructor overloading, data mata kuliah bisa disimpan meskipun hanya dengan satu parameter saja. Hal ini ditujukan untuk fleksibilitas pemrograman. Java membedakan konstruktor berdasarkan jumlah dan tipe parameternya. Jika keduanya sama namun signature datanya berbeda, maka java akan "bingung" memilih sehingga terjadi error