

# Dasar-Dasar Pemrograman 2

## Lab 3

### Objects and Classes



FAKULTAS  
ILMU  
KOMPUTER

---

#### Riwayat Versi

Versi	Timestamp	Keterangan	Warna
1	22-02-2024; 20.55 WIB	Rilis Pertama	-

# DekDepeNG



(Generated with: PIXLR AI Image Generator)

<p><u>NilaiController</u></p> <p>-kodeMatkul: String -nilai: int</p> <p>+ NilaiController(kodeMatkul: String) + getKodeMatkul(): string + getNilai(): int + setNilai(nilai: int) :void</p>	<p><u>Siswa</u></p> <p>-npm: String -listNilai: NilaiController[]</p> <p>+ Siswa(npm: String) + ambilMatkul(mataKuliah: MataKuliah): String + tampilkanNilai(): String + getListNilai() : NilaiController[] + getNpm(): String</p>
<p><u>MataKuliah</u></p> <p>-kodeMatkul: String -listSiswa: Siswa[] -kapasitas: int -jumlahSiswa: int</p> <p>+ MataKuliah(kodeMatkul: String, kapasitas: int) + getKodeMatkul(): string + getListSiswa(): Siswa[] + getKapasitas: int + setJumlahSiswa(jumlahSiswa:int): void + getJumlahSiswa(): int</p>	<p><u>Dosen</u></p> <p>-idDosen: String -mataKuliah: MataKuliah</p> <p>+ Dosen(idDosen: String, mataKuliah: MataKuliah) + beriNilai(npm:string, nilai:int): String + getIdDosen(): String + getMataKuliah(): MataKuliah</p>

## Deskripsi

Setelah Dek Depe belajar mengenai Object and Class, ia tertantang untuk membuat sebuah aplikasi. Terinspirasi dari sebuah aplikasi yang bernama S\*AK-NG, ia ingin membuat sebuah sistem untuk pengecekan nilai mata kuliah. Ia berharap bahwa aplikasi yang akan dikembangkan ini dapat bermanfaat bagi siswa serta dosen untuk membantu melakukan pengecekan serta penginputan nilai.

Tetapi, Dek Depe tahu bahwa membuat sebuah aplikasi akan sangatlah sulit jika dilakukan sendiri. Oleh karena itu, Dek Depe meminta bantuanmu untuk membuat sebuah sistem yang dapat melakukan pengecekan nilai bagi siswa. Konon katanya, jika

kamu sering dan tulus membantu Dek Depe, maka kelak kamu akan memperoleh nilai yang bagus untuk atribut dari object MataKuliah DekDepeDua hehehehe :D

## Ketentuan Program

Terdapat tiga perintah yang dapat dijalankan pada program DekDepeNG:

- **BERINILAI [ID\_DOSEN] [NPM] [NILAI]**  
Query ini berfungsi untuk memberikan **[NILAI]** kepada siswa **[NPM]** untuk mata kuliah yang diajarkan oleh **[ID\_DOSEN]**
- **CEKNILAI [NPM]**  
Query ini berfungsi untuk mengiterasi semua mata kuliah yang diambil oleh siswa **[NPM]** dan menampilkan mata kuliah yang diambil beserta nilainya
- **AMBILMATKUL [NPM] [KODE\_MATKUL]**  
Query ini berfungsi untuk siswa dengan **NPM** tersebut untuk mengambil mata kuliah dengan kode **[KODE\_MATKUL]**

Terdapat empat *class* yang ada pada file yang berbeda:

### 1. DekDepeNG

Class DekDepeNG adalah class utama dari program ini. Class ini digunakan untuk menjalankan program yang sudah dibuat dan menyimpan list **Dosen** dan list **Siswa**.

Method yang ada pada class ini selain method main adalah:

- `void beriNilai(String idDosen, String npm, int nilai)`  
Method ini digunakan untuk memberikan **[nilai]** kepada siswa dengan npm **[npm]** pada mata kuliah yang dibuka oleh **[idDosen]**.
- `void cekNilai(String npm)`  
Method ini digunakan untuk mengecek semua mata kuliah yang diambil oleh siswa **[npm]** dan menampilkan mata kuliah yang diambil beserta nilainya.

### 2. Dosen

Class ini merepresentasikan dosen yang terdaftar pada DekDepeNG. Class ini memiliki dua *attribute*, yaitu **String idDosen** dan **MataKuliah mataKuliah**. **Setiap dosen yang terdaftar hanya dapat membuka satu mata kuliah dan tidak ada mata kuliah yang dipegang oleh dua atau lebih dosen**. Semua attribute memiliki **visibility private**.

Method yang ada pada class ini adalah:

- `String beriNilai(String npm, int nilai)`  
Method ini digunakan untuk memberikan **[nilai]** kepada siswa dengan npm **[npm]** pada mata kuliah yang dibuka oleh dosen tersebut. **Sebelum dosen berhasil memberi nilai kepada siswa dengan npm [npm], ada beberapa hal yang harus divalidasi:**
  1. **Jika siswa dengan npm [npm] ditemukan di list siswa tetapi ia tidak mengambil mata kuliah yang diajar oleh dosen tersebut, maka dosen gagal untuk memberikan nilai bagi siswa tersebut**
  2. **Jika siswa dengan npm [npm] ditemukan di list siswa dan mengambil mata kuliah yang diajar oleh dosen tersebut, maka dosen berhasil untuk memberikan nilai bagi siswa tersebut.**

### 3. Siswa

Class ini merepresentasikan siswa yang terdaftar pada DekDepeNG. Class ini memiliki dua *attribute*, yaitu **String npm** dan **NilaiController[] listNilai**. Semua attribute memiliki **visibility private**.

Method yang ada pada class ini adalah:

- `String ambilMatkul(MataKuliah matkul)`  
Method ini digunakan untuk mengambil mata kuliah yang telah dibuka oleh dosen-dosen yang bersangkutan. Pada method ini, *constructor* untuk `NilaiController` akan dipanggil dan object `NilaiController` akan dibuat yang berfungsi untuk menampung kode mata kuliah beserta nilainya. **Sebelum siswa berhasil mengambil [matkul], ada beberapa hal yang harus divalidasi:**
  1. **Jika kapasitas dari mata kuliah sudah habis, maka siswa gagal untuk mengambil mata kuliah tersebut.**
  2. **Jika kapasitas dari mata kuliah masih ada, siswa baru dapat mengambil mata kuliah tersebut.**
- `String tampilkanNilai()`  
Method ini digunakan untuk menampilkan pasangan mata kuliah dengan nilainya dengan format **KodeMatkul - Nilai**

### 4. MataKuliah

Class ini merepresentasikan mata kuliah yang dibuka oleh dosen yang bersangkutan, **tetapi class ini tidak dipergunakan untuk menyimpan nilai siswa**. Class ini memiliki dua *attribute*, yaitu **String kodeMatkul**, **int kapasitas**, dan **Siswa[] listSiswa**. Semua attribute memiliki **visibility private**.

### 5. NilaiController

Class ini merepresentasikan nilai yang dimiliki oleh suatu siswa terhadap suatu mata kuliah. Class ini memiliki dua *attribute*, yaitu **String kodeMatkul** dan **int nilai**. Semua attribute memiliki **visibility private**.



### Template Lab

Untuk mempermudah pengerjaan lab ini, kamu dapat melihat template Lab03 [disini](#)

## Format Masukan

- Baris pertama berisi banyak **[Dosen]** yang akan mengajar pada mata kuliah yang saling berbeda (**Tidak ada dua mata kuliah yang diajar oleh satu dosen**)
- **[Dosen]** baris berikutnya berisi keterangan idDosen kemudian kode dari mata kuliah yang akan dibuka lalu kapasitas maksimal dari kelas yang akan dibuka.
- Baris berikutnya berisi banyak **[Siswa]**
- **[Siswa]** baris berikutnya berisi keterangan npm dari siswa tersebut
- Baris berikutnya berisi banyak query yang akan dikerjakan
- **[Banyak Query]** baris berikutnya berisikan format query yang akan dikerjakan (**untuk format query sudah dijelaskan di Ketentuan Program**)

## Batasan

- Jumlah siswa berkisar  $1 \leq x \leq 100$
- Jumlah query dipastikan  $\geq 1$
- idDosen dan npm siswa dijamin unik
- Kode mata kuliah yang akan diambil oleh siswa dijamin sudah ada pada masukan untuk mata kuliah
  - Query yang diujikan dijamin terdapat pada jenis query (**case sensitive!**).
  - Nilai yang diberikan untuk query **BERINILAI** dijamin  $0 \leq \text{nilai} \leq 100$
- Query yang menerima idDosen dan npm dijamin valid untuk nilai dari idDosen/npm
- Mata kuliah yang akan diambil siswa dijamin unik dan tidak duplikat

## Format Keluaran

- **BERINILAI [ID\_DOSEN] [NPM] [NILAI]**
  - Saat berhasil:  
**[ID\_DOSEN] berhasil memberikan nilai kepada siswa dengan NPM [NPM]**
  - Saat gagal:  
**[ID\_DOSEN] gagal memberikan nilai kepada siswa dengan NPM [NPM]**
- **CEKNILAI [NPM]**
  - Jika siswa memiliki minimal 1 mata kuliah yang diambil:  
**(for each Mata Kuliah):**

**Kode matkul [Kode\_matkul] memiliki nilai [Nilai]**  
(nilai memiliki default 0 jika belum dinilai oleh dosen)

- Jika siswa belum mengambil mata kuliah:  
**Siswa belum mengambil mata kuliah :v**

- **AMBILMATKUL [NPM] [KODE\_MATKUL]**

- Saat berhasil:  
**Siswa dengan NPM [NPM] berhasil mengambil matkul dengan kode [KODE\_MATKUL]**
- Saat gagal:  
**Siswa dengan NPM [NPM] gagal mengambil matkul dengan kode [KODE\_MATKUL]**

### Keterangan Tambahan

- Diperbolehkan untuk menambah *attribute/method* pada setiap class jika dirasa diperlukan tetapi **tidak diperbolehkan** untuk mengurangi *attribute/method* yang sudah ditentukan
- Wajib menggunakan Array Primitif!

### Test Case

#### Input 1

```
2
Dosen1 MK001 30
Dosen2 MK002 25
3
101
102
103
3
AMBILMATKUL 101 MK001
BERINILAI Dosen1 101 85
CEKNILAI 101
```

#### Output 1

```
Siswa dengan NPM 101 berhasil mengambil matkul dengan kode MK001
Dosen1 berhasil memberikan nilai kepada siswa dengan NPM 101
```

Kode matkul MK001 memiliki nilai 85

## Input 2

```
1
Dosen1 MK001 1
3
301
302
303
5
AMBILMATKUL 301 MK001
BERINILAI Dosen1 301 85
CEKNILAI 301
AMBILMATKUL 302 MK001
CEKNILAI 302
```

## Output 2

```
Siswa dengan NPM 301 berhasil mengambil matkul dengan kode MK001
Dosen1 berhasil memberikan nilai kepada siswa dengan NPM 301
Kode matkul MK001 memiliki nilai 85
Siswa dengan NPM 302 gagal mengambil matkul dengan kode MK001
Siswa belum mengambil mata kuliah :v
```

## Komponen Penilaian

- 60% Kebenaran hasil keluaran program
- 20% Kebenaran implementasi method-method pada class yang ada
- 10% Standar penulisan kode sesuai dengan [Code Conventions for the Java Programming Language](#)
- 10% Dokumentasi dan kerapian kode



## **Pengumpulan Berkas**

Kumpulkan berkas .java yang telah di-zip dengan format penamaan seperti berikut.  
[ KodeAsdos ] \_ [ Kelas ] \_ [ NPM ] \_ [ NamaLengkap ] \_ Lab3 . zip

Contoh:

**DDP\_A\_1234567890\_DekDePe\_Lab3.zip**