Dasar-Dasar Pemrograman 2

Lab 05

Object-Oriented Programming (OOP) and more

Object Oriented Programming

Object Oriented Programming adalah suatu paradigma *programming* yang mengikuti konsep dari *object*. Dalam paradigma ini kita memodelkan objek di dunia nyata sebagai sebuah *class* di dalam program. Objek tersebut memiliki *variable, constructor,* dan *method*.

Sebagai contoh, kita akan membuat suatu *class* yang memodelkan objek **Mobil** di dunia nyata. Mobil dapat dilihat sebagai sebuah objek yang memiliki atribut **merek**, **model**, **warna**, dan **kelajuan**. Kelajuan mobil dapat dipercepat (**akselerasi**) maupun diperlambat (**deselerasi**). Berikut implementasi *class* yang dibuat.

```
public class Mobil{
    private String merek;
    private String model;
    private String warna;
    private double kelajuan;

    //class' Constructor
    public Mobil(String merek, String model, String warna){
        this.merek = merek;
        this.model = model;
        this.warna = warna;
        this.kelajuan = 0;
    }
```

```
    //class' method
    public void akselerasi(double perubahan){
        this.kelajuan += perubahan;
    }

    //class' method
    public void deselerasi(double perubahan){
        this.kelajuan -= perubahan;
    }
}
```

***Catatan**:*
*Penggunaan **this** pada kode di atas (Java) ekuivalen dengan penggunaan **self** pada kode Python.*

Constructor

Constructor adalah *method* untuk membuat instansiasi objek dari *class*. Constructor adalah *method* khusus yang akan dieksekusi pada saat pembuatan objek (*instance*). Pada umumnya, *method* ini berisi inisialisasi variabel untuk objek.

```
public class Mobil{

    //...inisialisasi variabel

    //class' Constructor
    public Mobil(String merek, String model, String warna){
        this.merek = merek;
        this.model = model;
        this.warna = warna;
        this.kelajuan = 0;
    }
}
```

Constructor di atas akan dijalankan jika objek mobil dibuat

```
Mobil mobil = new Mobil("Daihatsu", "Xenia", "Putih");
```

Setter & Getter

Setter adalah *method* yang dipakai suatu objek untuk mengubah *value* dari variabel yang dimiliki sebuah objek, sedangkan ***getter*** adalah *method* yang dipakai untuk mengambil *value* dari sebuah variabel yang dimiliki objek.

Setter dan *getter* digunakan bersamaan terhadap variabel yang disembunyikan oleh objek, biasanya dengan memakai tipe **private** pada atribut tersebut (Contohnya semua variabel pada kelas **Mobil** di atas). Hal ini dilakukan untuk menghindari akses secara langsung dari kelas lain yang bisa menyebabkan kebocoran dan perubahan terhadap data yang disimpan oleh suatu objek

Contoh pengaplikasian *setter* dan *getter*:

```
    //setter method
    public void setKelajuan(int laju){
        this.kelajuan = laju;
    }

    //getter method
    public double getKelajuan(){
        return this.kelajuan;
    }
```

Contoh penggunaan *setter* dan *getter*:

```
Mobil mobil = new Mobil("Daihatsu", "Xenia", "Putih");
System.out.println(mobil.getKelajuan());
mobil.setKelajuan(40);
System.out.println(mobil.getKelajuan());
```

Output yang dihasilkan:

```
0
40
```

toString()

toString adalah *method* yang digunakan untuk mengatur representasi suatu objek ke dalam String. Jika dalam suatu *class* tidak terdapat method **toString()**, maka saat melakukan print, yang muncul di output program adalah nilai kode hash dari objek tersebut

```
Mobil mobil = new Mobil("Daihatsu", "Xenia", "Putih");
System.out.println(mobil);
```

Output:
Mobil@d716361

Dengan *method* **toString()** kita dapat mengukur representasi String dari objek tersebut. Contohnya dengan menambahkan *method* **toString()** dalam *class* **Mobil**, seperti di bawah ini:

```
class Mobil{

   //......

   //toString method
   public String toString(){
      return "Mobil " + merek + " " + model + " berwarna " + warna;
   }
}
```

Output ketika kita melakukan print:
Mobil Daihatsu Xenia berwarna Putih

Fungsi Static & non-Static

Fungsi **static** merupakan suatu fungsi yang tidak perlu instansiasi objek untuk menggunakannya. Fungsi ini melekat ke suatu class sehingga untuk memanggilnya bisa dilakukan dengan **<NamaClass>.namaFungsi()**. Sedangkan fungsi **non-static** adalah fungsi yang membutuhkan suatu objek dari suatu class yang memiliki fungsi tersebut untuk menggunakannya.

```
public class Mobil{
      public static void main(String[] args){
            Mobil mobil = new Mobil();
            System.out.println(mobil.maju()); \\Maju
            System.out.println(Mobil.jalan()); \\Jalan
      }
      public static String jalan(){
            return "Jalan";
      }
      public String maju(){
            return "Maju";
      }
}
```

Variable Static dan non-Static

Sama halnya seperti fungsi, variabel static melekat pada suatu class bukan melekat pada suatu objek pada class tersebut sehingga untuk mengetahui value dari suatu variabel static dapat dilakukan dengan **<NamaClass>.namaVariabel**. Sedangkan variabel non-static akan melekat ke suatu objek dari class tersebut sehingga perlu dibuat objek dari class tersebut untuk menggunakannya.

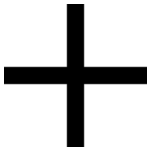
```
public class Mobil{
      public static String maju = "maju";
      public String jalan = "Jalan";
      public static void main(String[] args){
            Mobil mobil = new Mobil();
            System.out.println(mobil.jalan); \\Jalan
            System.out.println(Mobil.maju); \\maju
      }
}
```

Diagram UML (Diagram Kelas)

Diagram UML (*Unified Model Language*) merupakan suatu diagram yang digunakan untuk memodelkan desain program yang berbasis OOP (*Object Oriented Programming*). Diagram tersebut memiliki kapabilitas untuk menggambarkan susunan struktur suatu class di suatu program. Diagram UML tersusun dari sekumpulan tabel yang berisi keterangan dari suatu kelas dan dihubungkan dengan suatu notasi.

Notasi dari diagram UML:

Notasi	Nama	Keterangan
	Komposisi	Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi Composition terhadap class tempat dia bergantung tersebut.
	Agregasi	Jika sebuah class memiliki atribut dari class lain tetapi tidak sebaliknya.

	Dependensi	Sebuah class merupakan turunan dari class lainnya
	Private	Menyatakan sebuah fungsi atau variabel private
	Public	Menyatakan sebuah fungsi atau variable public

Struktur tabel diagram UML

NamaClass
Variabel
Fungsi

Contoh diagram UML

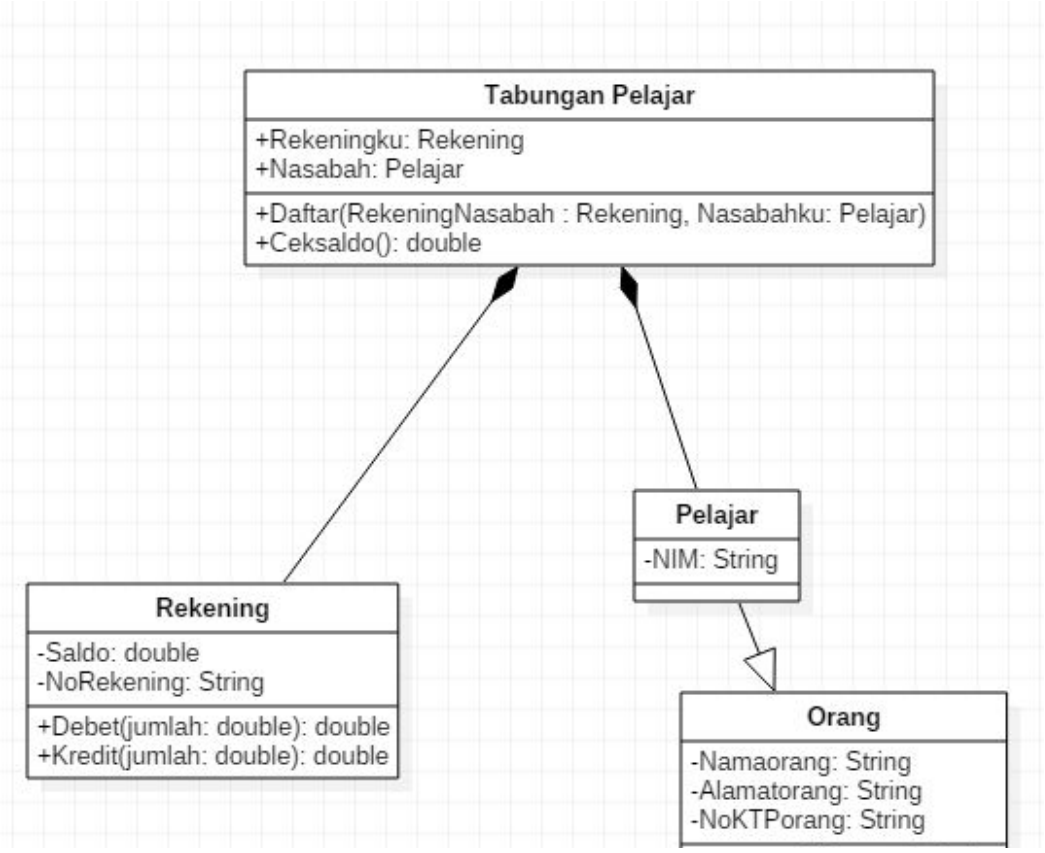


Diagram tersebut menjelaskan bahwa Tabungan Pelajar tidak mungkin ada jika tidak ada Rekening dan Pelajar, Pelajar merupakan turunan dari Orang. Variabel Saldo dan NoRekening di rekening merupakan variabel private. Sementara Rekeningku dan Nasabah di Tabungan Pelajar merupakan variabel public.

Soal Lab 05

Bantu Gang 0000 Bisa Kuliah Lagi!!!

Pada suatu hari di negeri Depok, hiduplah 4 orang sahabat yang sangat dekat. Mereka bernama Dek Depe, Pak Esde, Kak Pewe, dan Pak Oka. Saking dekatnya, mereka memutuskan untuk membuat gang. Uniknya, mereka berempat menyukai angka yang sama, yaitu angka 0. Sehingga, mereka menamainya Gang 0000.

Sayangnya, Gang 0000 baru saja membuat masalah kemarin di tempat kuliah lamanya, yaitu Pacilkom Uwiw. Masalah yang mereka buat sangatlah besar sampai-sampai akhirnya mereka harus di *drop out* dari Pacilkom Uwiw.

Saat ini, Gang 0000 berencana untuk mencari tempat kuliah barunya. Anda, sang pemilik academy X, ingin membantu Gang 0000 untuk bisa berkuliah lagi. Wah, baik sekali Anda! Namun, hanya dengan suatu syarat, yaitu nilai seleksi dari Gang 0000 bisa lebih dari atau sama dengan nilai *passing grade* yang sudah anda tetapkan.

Pada saat semester terakhir mereka di Pacilkom Uwiw, mata kuliah yang ditawarkan adalah DDP 2, PSD, PPSI, Matdas 1, Matdis 2, dan MPKT A. Setiap mata kuliah memiliki banyak SKS dan tingkat kesulitannya masing-masing sebagai berikut.

Nama matkul	Banyak SKS	Tingkat kesulitan (1 - 5)
DDP 2	4	3

PSD	4	3
PPSI	3	3
Matdas 1	3	2
Matdis 2	3	4
MPKT A	6	1

Setiap anggota Gang 0000 mengambil matkul yang berbeda-beda pada semester terakhirnya di Pacilkom Uwiw, sebagai berikut.

Nama	Matkul yang diambil
Dek Depe	DDP 2, PPSI, Matdis 2
Pak Esde	PSD, MPKT A, Matdas 1
Kak Pewe	DDP 2, PPSI, Matdis 2, PSD, MPKT A, Matdas 1
Pak Oka	PSD, MPKT A, PPSI

Setiap anggota Gang 0000 juga memiliki kekuatannya tersendiri dalam belajar dalam skala 1-5. Dek Depe memiliki kekuatan sebesar 4, Pak Esde memiliki kekuatan sebesar 2, Kak Pewe memiliki kekuatan sebesar 5, dan Pak Oka memiliki kekuatan sebesar 3.

Total waktu belajar untuk setiap anggota dapat dihitung dengan cara sebagai berikut :

(Kekuatan belajar)*(Banyak SKS)+(Tingkat kesulitan matkul)

Nilai seleksi untuk masuk academy X adalah, rata-rata waktu belajar (dalam jam) Gang 0000 pada saat semester terakhirnya di Pacilkom Uwiw. Apabila nilai Gang 0000 lebih besar atau sama dengan dari nilai *passing grade* yang sudah Anda tetapkan, maka Gang 0000 berhasil masuk academy Anda. Sebaliknya, apabila nilai mereka kurang dari nilai passing grade, maka Gang 0000 gagal masuk academy Anda.

Tentukan apakah Gang 0000 berhasil masuk academy Anda atau tidak.

SPESIFIKASI

Akan ada template yang terdiri dari 3 class, yaitu :

- Course
- Student
- Simulator

Course

Kelas **Course** memiliki 3 atribut, yaitu **name (String), sks (int),** dan **difficulty (int).** Tugas Anda adalah melengkapi TO DO dalam class ini yaitu membuat *constructor* dan *method setter* / *getter* apabila diperlukan. (**NOTE: Atribut class harus private. Jangan di ubah!**)

Student

Kelas **Student** memiliki 5 atribut, yaitu **courses (ArrayList<Course>), name (String), strength (int), totalStudyTime (int),** dan **numOfStudents (int).** Atribut **totalStudyTime** dan **numOfStudents** merupakan *static attribute*. Tugas Anda adalah melengkapi TO DO dalam method di class ini. (**NOTE: Atribut harus private**)

Class **Student** memiliki beberapa method, yaitu :

- void addCourse(Course course)

  Method ini akan menambahkan course ke daftar courses dari student.

- void printIntroduction()

  Method ini akan meng-*output*-kan introduction dari student ke layar dengan format :

  <name> memiliki kekuatan sebesar **<strength>**

- void printCourses()

  Method ini akan meng-*output*-kan course apa saja yang dipilih student ke layar dengan format :

  Matkul yang di ambil oleh **<name>** adalah:

  1. **<nama-matkul>** -- **<sks>** -- **<tingkat kesulitan>**
  2. dst

- void printStudyTime()

  Method ini akan meng-*output*-kan total waktu belajar dari student ke layar dengan format :

  Total waktu belajar **<name>** adalah **<total-waktu-belajar>** jam.

- int calculateStudyTime()

  Method ini akan mengembalikan total waktu belajar dari student

- static int getTotalStudyTime()

  Method ini merupakan method static yang akan mengembalikan total waktu belajar dari seluruh student

- static int getNumOfStudents()

  Method ini merupakan method static yang akan mengembalikan banyaknya object Student yang sudah dibuat sampai saat ini. (Seharusnya tidak akan pernah lebih dari 4 nilainya :P)

- static double calculateAverageStudyTime()

  Method ini merupakan method static yang akan mengembalikan rata-rata waktu belajar dari seluruh student (Dek Depe, Pak Esde, Kak Pewe, Pak Oka)

NOTE : Static attribute dan static method harus di keep static ya. Jangan diubah ataupun dihilangkan :)

Simulator

Untuk memeriksa kebenaran program Anda, silahkan jalankan class Simulator. Namun sebelum itu lengkapi dulu //TO DO yang ada.

Contoh *output* dari program :

```
Masukkan passing grade untuk academy anda: 50
Dek Depe memiliki kekuatan sebesar 4
Matkul yang diambil oleh Dek Depe adalah:
1. DDP 2 -- 4 SKS -- 3
2. PPSI -- 3 SKS -- 3
3. Matdis 2 -- 3 SKS -- 4
Total waktu belajar Dek Depe adalah 50 jam.
Pak Esde memiliki kekuatan sebesar 2
Matkul yang diambil oleh Pak Esde adalah:
1. PSD -- 4 SKS -- 3
2. MPKT A -- 6 SKS -- 1
3. Matdas 1 -- 3 SKS -- 2
Total waktu belajar Pak Esde adalah 32 jam.
Kak Pewe memiliki kekuatan sebesar 5
Matkul yang diambil oleh Kak Pewe adalah:
1. DDP 2 -- 4 SKS -- 3
2. PPSI -- 3 SKS -- 3
3. Matdis 2 -- 3 SKS -- 4
4. PSD -- 4 SKS -- 3
5. MPKT A -- 6 SKS -- 1
6. Matdas 1 -- 3 SKS -- 2
Total waktu belajar Kak Pewe adalah 131 jam.
Pak Oka memiliki kekuatan sebesar 3
Matkul yang diambil oleh Pak Oka adalah:
1. PSD -- 4 SKS -- 3
2. MPKT A -- 6 SKS -- 1
3. PPSI -- 3 SKS -- 3
Total waktu belajar Pak Oka adalah 46 jam.
Total waktu belajar Gang 0000 (jam):
259
Rata-rata waktu belajar Gang 0000 (jam):
64.75
Gang 0000 berhasil masuk ke academy anda!
```

```
Masukkan passing grade untuk academy anda: 75
Dek Depe memiliki kekuatan sebesar 4
Matkul yang diambil oleh Dek Depe adalah:
1. DDP 2 -- 4 SKS -- 3
2. PPSI -- 3 SKS -- 3
3. Matdis 2 -- 3 SKS -- 4
Total waktu belajar Dek Depe adalah 50 jam.
Pak Esde memiliki kekuatan sebesar 2
Matkul yang diambil oleh Pak Esde adalah:
1. PSD -- 4 SKS -- 3
2. MPKT A -- 6 SKS -- 1
3. Matdas 1 -- 3 SKS -- 2
Total waktu belajar Pak Esde adalah 32 jam.
Kak Pewe memiliki kekuatan sebesar 5
Matkul yang diambil oleh Kak Pewe adalah:
1. DDP 2 -- 4 SKS -- 3
2. PPSI -- 3 SKS -- 3
3. Matdis 2 -- 3 SKS -- 4
4. PSD -- 4 SKS -- 3
5. MPKT A -- 6 SKS -- 1
6. Matdas 1 -- 3 SKS -- 2
Total waktu belajar Kak Pewe adalah 131 jam.
Pak Oka memiliki kekuatan sebesar 3
Matkul yang diambil oleh Pak Oka adalah:
1. PSD -- 4 SKS -- 3
2. MPKT A -- 6 SKS -- 1
3. PPSI -- 3 SKS -- 3
Total waktu belajar Pak Oka adalah 46 jam.
Total waktu belajar Gang 0000 (jam):
259
Rata-rata waktu belajar Gang 0000 (jam):
64.75
Gang 0000 gagal masuk academy anda :'(
```

NOTE : Nilai passing grade akan anda input

Catatan

1. Jangan lupa berdoa sebelum mengerjakan :)
2. Disarankan menggunakan template :)
3. Mengubah hal yang sudah diperingatkan adalah kesalahan yang **fatal**, akan ada pengurangan nilai yang signifikan untuk hal itu. (Mengubah private attribute menjadi tidak private, mengubah/menghilangkan static attribute dan static method)
4. Menghitung banyak anggota Gang 0000, menghitung total waktu belajar untuk seluruh anggota Gang 0000, dan menghitung rata-rata waktu belajar Gang 0000 **HARUS** dihitung pada **static method** di dalam class **Student**.

Komponen Penilaian

- 20% Implementasi *class* Course
- 40% Implementasi *class* Student
- 30% Implementasi *class* Simulator
- 10% Dokumentasi dan kerapian kode

Good luck! Jangan lupa untuk bantu kami!! -Gang 0000