PRAKTIKUM 8

ENKAPSULASI

A. TUJUAN PEMBELAJARAN

1. Menerapkan konsep enkapsulasi pada class
2. Mendeklarasikan suatu constructor

B. DASAR TEORI

Kita dapat menyembunyikan information dari suatu class sehingga anggota-anggota class tersebut tidak dapat diakses dari luar. Adapun caranya adalah cukup dengan memberikan akses kontrol private ketika mendeklarasikan suatu atribut atau method. Contoh:

```
private int nrp;
```

Encapsulation (Enkapsulasi) adalah suatu cara untuk menyembunyikan implementasi detail dari suatu class. Enkapsulasi mempunyai dua hal mendasar, yaitu :

1. information hiding
2. menyediakan suatu perantara (method) untuk pengaksesan data

Contoh:

```
public class Siswa {
    private int nrp;

    public void setNrp(int n) {
          nrp=n;
    }
}
```

Contructor (konstruktor) adalah suatu method yang pertama kali dijalankan pada

saat pembuatan suatu obyek. Konstruktor mempunyai ciri yaitu :

1. mempunyai nama yang sama dengan nama class
2. tidak mempunyai return type (seperti void, int, double dll)

Contoh:

```
public class Siswa {
      private int nrp;
      private String nama;

      public Siswa(int n, String m) {
            nrp=n;
            nama=m;
      }
}
```

Suatu class dapat mempunyai lebih dari 1 konstruktor dengan syarat daftar parameternya tidak boleh ada yang sama. Contoh :

```
public class Siswa {
      private int nrp;
      private String nama;

      public Siswa(String m) {
            nrp=0;
            nama="";
      }

      public Siswa(int n, String m) {
            nrp=n;
            nama=m;
      }
}
```

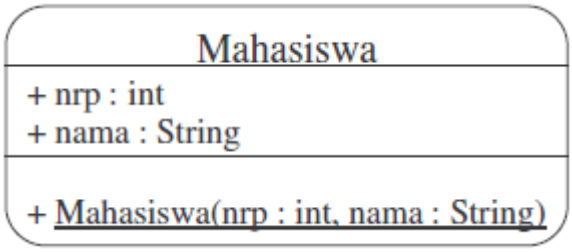
C. TUGAS PENDAHULUAN

1. Apakah yang dimaksud dengan enkapsulasi?
2. Apakah yang dimaksud dengan constructor?
2. Apakah yang dimaksud dengan overloading constructor?

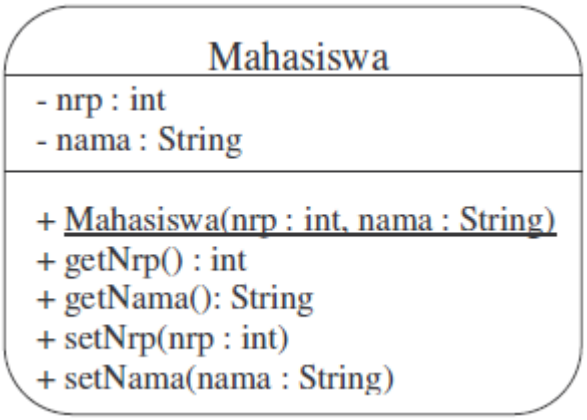
D. PERCOBAAN

Percobaan 1 : Melakukan enkapsulasi pada suatu class

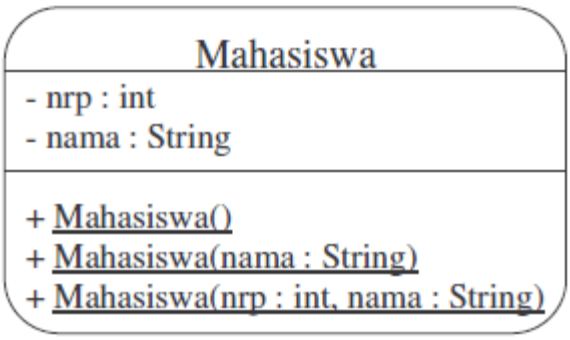
Implementasikan UML class diagram Mahasiswa sebelum dan setelah dilakukan enkapsulasi!



Jika enkapsulasi dilakukan pada class diagram diatas, maka akan berubah menjadi:



Percobaan 2 : Melakukan overloading constructor



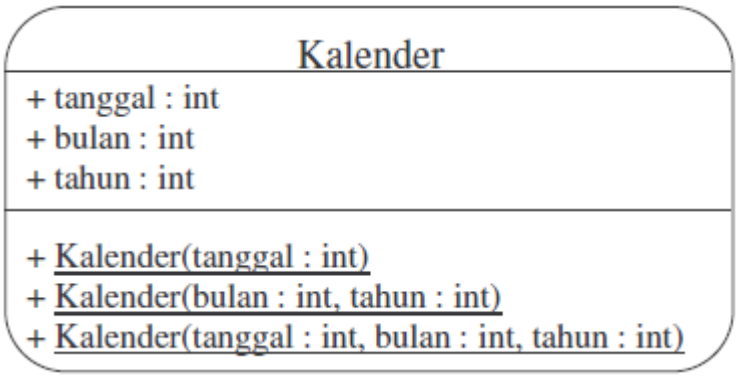
Dari class diagram tersebut, dapat diimplementasikan ke dalam program sebagai berikut:

```
public class Mahasiswa {
    private int nrp;
```

```
    private String nama;

    public Mahasiswa() {
        nrp=0;
        nama=”“;
    }

    public Mahasiswa(String nama) {
        nrp=0;
        this.nama=nama;
    }

    public Mahasiswa(int nrp, String nama) {
        this.nrp=nrp;
        this.nama=nama;
    }
}
```

E. LATIHAN

Latihan 1: Mengimplementasikan UML class diagram dalam program untuk class Kalender



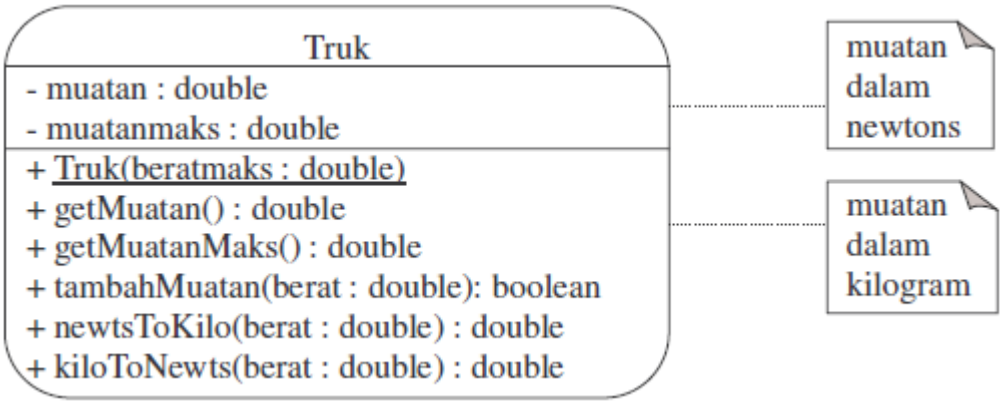
Dari class diagram diatas, desainlah suatu class yang memenuhi konsep enkapsulasi. Untuk nilai inisialisasi, dipakai 1-1-2000. Pakailah kata kunci *this* untuk mempersingkat pengkodean. Tulislah listing program berikut ini sebagai pengetesan.

```
public class TesKalender {
  public static String getTime(Kalender kal) {
      String tmp;
      tmp = kal.getTanggal() + "-" +
            kal.getBulan() + "-" +
            kal.getTahun();
      return tmp;
  }
  public static void main(String args[]) {
      Kalender kal = new Kalender(8);
      System.out.println("Waktu awal : " + getTime(kal));
      kal.setTanggal(9);
      System.out.println("1 hari setelah waktu awal : "+getTime(kal));
      kal = new Kalender(6, 2003);
      System.out.println("Waktu berubah : " + getTime(kal));
      kal.setBulan(7);
      System.out.println("1 bulan setelah itu : " + getTime(kal));
      kal = new Kalender(20, 10, 2004);
      System.out.println("Waktu berubah : " + getTime(kal));
      kal.setTahun(2005);
      System.out.println("1 tahun setelah itu : " + getTime(kal));
  }
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda.

```
Waktu awal : 8-1-2000
1 hari setelah waktu awal : 9-1-2000
Waktu berubah : 1-6-2003
1 bulan setelah itu : 1-7-2003
Waktu berubah : 20-10-2004
1 tahun setelah itu : 20-10-2005
```

Latihan 2 : Mengimplementasikan UML class diagram dalam program untuk class Truk



Keterangan : 1 kilogram = 9,8 newtons

Transformasikan class diagram diatas ke dalam bentuk program! Tulislah listing program berikut ini sebagai pengetesan.

```
public class TesTugas2{
    public static void main(String args[]){
        boolean status;

        Truk truk = new Truk(900);
        System.out.println("Muatan maksimal = "+truk.getMuatanMaks());
        status = truk.tambahMuatan(500.0);
        System.out.println("Tambah muatan : 500");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");

        status = truk.tambahMuatan(300.0);
        System.out.println("Tambah muatan : 300");
        if (status)
            System.out.println("Ok");
```

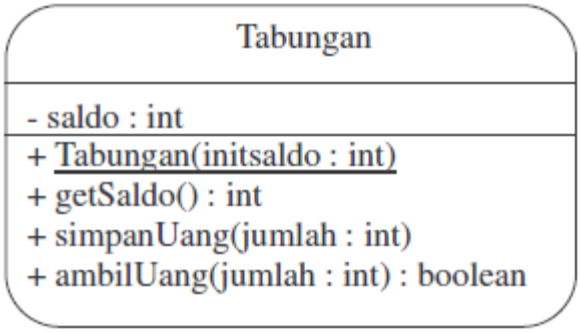
```
        else
            System.out.println("Gagal");

        status = truk.tambahMuatan(150.0);
        System.out.println("Tambah muatan : 150");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");

        status = truk.tambahMuatan(50.0);
        System.out.println("Tambah muatan : 50");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");
        System.out.println("Muatan sekarang = " + truk.getMuatan());
    }
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```
Muatan maksimal : 900.0
Tambah muatan : 500  ok
Tambah muatan : 300  ok
Tambah muatan : 150  gagal
Tambah muatan : 50  ok
Muatan sekarang = 849.9999999999999
```

F. TUGAS

Tugas 1. Menerapkan konsep enkapsulasi pada kelas Tabungan yang terdapat di Tugas 1. Bab 7. Pengenalan Pemrograman Berbasis Obyek.



Kembangkan kelas Tabungan diatas sehingga memungkinkan pengguna untuk memilih satuan mata uang yang berbeda (USD, AUD, IDR) ketika mengambil atau menyimpan uang. Saldo tabungan disimpan dalam satuan IDR oleh sistem. Beri nama kelas anda dengan nama MultiTabungan.java. Diasumsikan bahwa:

1 AUD = 10.000 IDR

1 USD = 9.000 IDR

Buat kelas baru untuk mengetes kelas MultiTabungan yang anda buat!

G. LAPORAN RESMI

Kumpulkan hasil latihan dan tugas di atas. Tambahkan analisa dalam laporan resmi.