	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LAB SHEET PEMROGRAMAN 2		
	Semester 2	Review OOP (CLASS DESIGN)	4 x 50mnt
	No. LST/EKA/PTI208/01	Revisi : 02	Feb 2010
			Hal 1 dari 7

A. Kompetensi

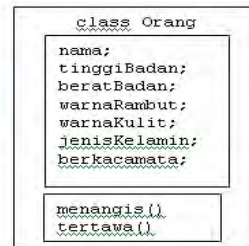
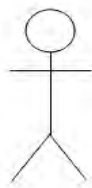
Setelah mengikuti mata kuliah ini, mahasiswa diharapkan memiliki kemampuan membuat suatu Class dan Object

B. Dasar Teori

➤ Mengenal Objek & Class

Contoh pemisalan objek orang nyata menjadi kode program dalam class Orang :

```
class Orang {
    String nama;           //nama orang
    int tinggiBadan;      //dalam cm
    int beratBadan;       //dlm kg
    String warnaRambut;   //hitam, pirang, coklat
    String warnaKulit;    //sawoMatang, hitam, putih
    String jenisKelamin;  //pria atau wanita
    boolean berkacamata;  //bila berkacamata berarti true
}
```




Memindahkan orang dari dunia nyata menjadi class Orang

Class dapat diumpamakan seperti spesifikasi atau blueprint. Dalam hal ini, Tuhan menciptakan manusia dengan spesifikasi tertentu. Jadi dapat diumpamakan bahwa Tuhan memiliki class Orang yang kemudian membuat banyak objek dari class Orang tsb, dan contoh objek tersebut adalah Anda sendiri.

Objek dalam pemrograman adalah objek yang dibuat dari class tertentu. Dari definisi class Orang di atas, kita bisa membuat objek-objek berdasar class tersebut. Objek-objek yang dibuat perlu disimpan dalam variabel yang akan menyimpan referensi/address dari objek yang dibuat. Proses pembuatan objek sering disebut sebagai instansiasi class, sedangkan objeknya disebut sebagai instance dari class.

➤ Method

Dibuat oleh : Herman DS,	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
-----------------------------	--	------------------

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LAB SHEET PEMROGRAMAN 2		
	Semester 2	Review OOP (CLASS DESIGN)	4 x 50mnt
	No. LST/EKA/PTI208/01	Revisi : 02	Feb 2010
			Hal 2 dari 7

Selain memiliki atribut (STATE) yang diimplementasikan sebagai data member di atas, manusia juga dapat melakukan suatu aksi atau pekerjaan tertentu (BEHAVIOR). Contoh aksi/behavior yang umum adalah menangis dan tertawa. Kedua behavior tsb bisa dipindahkan ke dalam bahasa pemrograman menjadi method sbb :

```

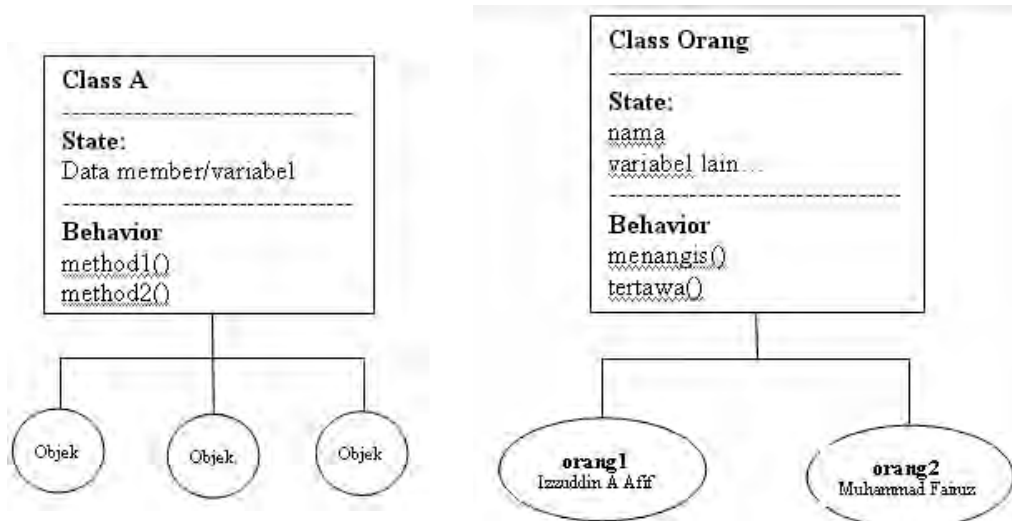
void menangis () {
    System.out.println("hik..hikk..hik...");
}

void tertawa() {
    System.out.println("ha..ha..ha..ha..");
}

```

Method merupakan perwujudan aksi atau tindakan dari dunia nyata di dalam pemrograman komputer. Method dalam dunia pemrograman juga "pasti melakukan sesuatu aksi", misalnya menampilkan String di konsol. Terdapat 3 macam metode yang meliputi:


1. Metode kelas: Metode ini dapat dieksekusi walaupun tidak terdapat objek dalam kelas tersebut. Metode kelas ini dideklarasikan dengan menggunakan modifier static.
2. Metode objek: Metode ini hanya dapat dieksekusi sehubungan dengan objek tertentu.
3. Metode main: Metode ini digunakan pada saat aplikasi java dimulai, menggunakan keyword static.



Ilustrasi perbedaan antara

Ilustrasi pembuatan

Dibuat oleh : Herman DS,	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
-----------------------------	--	------------------

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LAB SHEET PEMROGRAMAN 2		
	Semester 2	Review OOP (CLASS DESIGN)	4 x 50mnt
	No. LST/EKA/PTI208/01	Revisi : 02	Feb 2010
Hal 3 dari 7			

class dan objek

objek dari class

Dari gambar di atas dapat dipahami bahwa suatu class dapat memiliki banyak objek, dan setiap objek akan mewarisi data member dan method yang sama dari class. Untuk membuat objek Orang dari class Orang, gunakan keyword new sbb :

```
Orang orang1 = new Orang("Izzuddin A Afif");
Orang orang2 = new Orang("Muhammad Fairuz");
```

Setiap objek dapat memiliki state atau nilai data member yang berbeda (hanya nama dan tipe variabel yang sama). Pada saat objek dibuat, objek berisi semua variabel yang terdapat dalam kelas tersebut. Terdapat dua macam variabel dalam kelas, yaitu:

- ✓ Variabel kelas: variabel yang dideklarasikan di dalam kelas dan diberi modifier static. Sifat dari variabel static ini adalah melekat pada kelas dan tidak terpengaruh oleh proses instansiasi.
- ✓ Variabel objek(instans): variabel yang dideklarasikan di dalam kelas dan tidak memiliki modifier static. Berbeda dengan variabel kelas, pada proses instansiasi kelas menjadi objek, variabel instans ini akan ikut dimiliki dan dapat diakses oleh objek.

➤ Variabel this

Kata kunci this dipergunakan pada pembuatan kelas dan digunakan untuk menyatakan objek sekarang.

```
void isiData(String warnaMobil,int tahunProduksiMobil){
    warna = warnaMobil;
    tahunProduksi = tahunProduksiMobil;
}
```

Bisa ditulis menjadi:

```
void isiData(String warna,int tahunProduksi){
    this.warna = warnaMobil;
    this.tahunProduksi = tahunProduksi;
}
```

Metode konstruktor atau biasa disebut konstruktor saja adalah metode yang dapat digunakan untuk memberikan nilai awal saat objek diciptakan. Metode ini akan dipanggil secara otomatis oleh java ketika new dipakai untuk menciptakan instan kelas.

Dibuat oleh : Herman DS,	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
-----------------------------	---	------------------



Konstruktor mempunyai sifat:

- namanya sama dengan nama kelas
- tidak memiliki nilai balik (termasuk tidak boleh ada kata kunci void)

➤ **Overloading terhadap Konstruktor**

Overloading terhadap konstruktor merupakan suatu mekanisme pembuatan konstruktor yang memiliki bentuk lebih dari satu. Dalam hal ini pembeda antara konstruktor dengan konstruktor yang lain berupa jumlah parameter atau tipe parameter.

```
class mobil{
    private String warna;
    private int tahunProduksi;

    public mobil(String warnaMobil,int tahunProduksiMobil){
        warna=warnaMobil;
        tahunProduksi=tahunProduksiMobil;
    }

    public mobil(){
    }

    public void info(){
        System.out.println("warna:"+this.warna);
        System.out.println("tahun:"+this.tahunProduksi);
    }
}

public class konstruktor{
    public static void main(String args[]){
        mobil mobilku=new mobil("merah",2003);
        mobilku.info();

        mobil mobilmu = new mobil();
        mobilmu.info();
    }
}
```

Hasil

```
warna:merah
tahun:2003
warna:null
tahun:0
Finished executing
```

➤ **Overloading pada metode**

Overloading juga dapat dikenali pada metode non konstruktor

```
class matematika{
    static public double kuadrat(double nilai){
        return nilai*nilai;
    }

    static public int kuadrat(int nilai){
        return nilai*nilai;
    }

    static public double kuadrat(String nilai){
        double bilangan;
        bilangan=double.parseDouble(nilai);
    }
}
```



FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA

LAB SHEET PEMROGRAMAN 2

Semester 2	Review OOP (CLASS DESIGN)	4 x 50mnt
No. LST/EKA/PTI208/01	Revisi : 02	Feb 2010
		Hal 5 dari 7

Hasil

```
625.0  
625  
625.0|  
Finished executing
```

C. Alat/ Bahan

1. Lab. Sheet Pemrograman Java 2
2. PC / Laptop with OS installed
3. JDK 1.5 or latest
4. J-Creator or text editor

D. Langkah Kerja

1. Baca dan pahami dasar teori di atas.
2. Lakukan kompilasi dan eksekusi terhadap contoh-contoh source code atau program yang ada di dasar teori dan LAMPIRAN.
3. Kerjakan tugas individu di bawah.

E. Tugas Individu

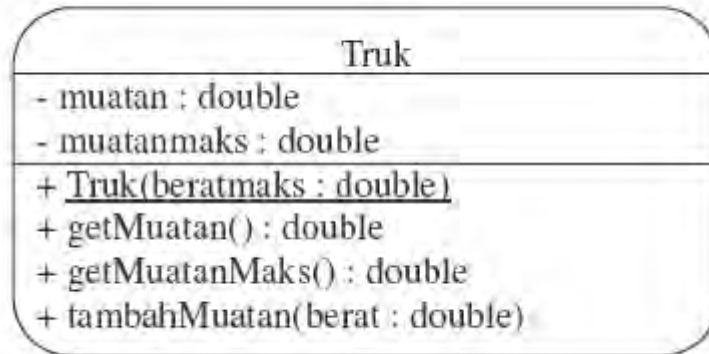
Upload tugas minggu 1 ke www.besmart.uny.ac.id dengan format **NIM_nama** dalam bentuk **zip/rar**. Bentuk laporan berupa dokumen (**pdf/doc/docx**)

1. Mengimplementasikan class diagram dalam program untuk class Truk seperti berikut ini;

Dibuat oleh :
Herman DS,

Dilarang memperbanyak sebagian atau seluruh isi dokumen
tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh :




Buatlah program sehingga sehingga contoh hasilnya seperti di bawah ini: (dimana tambah muatan berupa input interaktif)

```
Muatan maksimal : 1000.0
Tambah muatan : 500
Tambah muatan : 350
Tambah muatan : 100
Tambah muatan : 150
Muatan sekarang = 950.0
```

2. Perhatikan source code berikut:

```
import java.util.Scanner;
class ProcessData1 {
    public static void main(String args[]) {
        Scanner myScanner = new Scanner(System.in);
        double amount;
        boolean taxable;
        double total;

        System.out.print("Amount: ");
        amount = myScanner.nextDouble();
        System.out.print("Taxable? (true/false) ");
        taxable = myScanner.nextBoolean();
        if (taxable)
            { total = amount * 1.05; }
        else
            { total = amount; }
    }
}
```

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LAB SHEET PEMROGRAMAN 2		
	Semester 2	Review OOP (CLASS DESIGN)	4 x 50mnt
	No. LST/EKA/PTI208/01	Revisi : 02	Feb 2010
			Hal 7 dari 7

```

System.out.print("Total: ");
System.out.println(total);
}
}

```

Ubahlah class **ProcessData1** sehingga menjadi seperti berikut:

```

import java.util.Scanner;
class ProcessData2 {
    public static void main(String args[]) {
        Scanner myScanner = new Scanner(System.in);
        Purchase Pemrosesan = new Purchase();
        .....
        .....
    }
}

```

Buatlah sebuah class **Purchase** dari class **ProcessData1** sehingga hasil output tetap sama (*hasil ProcessData1 = ProcessData2*). Capture hasil dari ProcessData1 dan ProcessData2, sertakan juga source code masing-masing

F. Lampiran

1. File: CH32-Class Design Example.doc
2. File: CH33-Encapsulation.doc

Dibuat oleh : Herman DS,	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
-----------------------------	---	------------------