

Pemrograman Berorientasi Obyek II

Semester Gasal





Tujuan

- Mampu memahami konsep enkapsulasi, inheritance dan polimorfisme
- Mampu membedakan kelas, obyek, variabel, dan berbagai jenis tipe akses
- Mampu membuat kelas dan interface
- Mampu membuat gambar kelas UML dari listing program



Soal

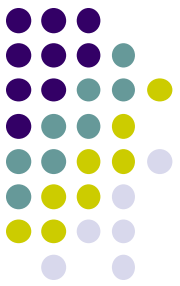
- Kerjakan secara mandiri soal ini!
- Tentukan kelas dan obyek dari listing di bawah ini!
- Buatlah diagram kelas berdasar listing kode dibawah ini!

```

public class Penduduk{
    String nama;
    public Penduduk(){
    }
    public Penduduk(String data){
        nama = data;}
}
class Mahasiswa extends Penduduk{
    String nim;
    public Mahasiswa(String dataNama, String dataNim){
        nama = dataNama; nim= dataNim;}
}
class Kelas{
    int kapasitasKelas=40;
    int jumlahMahasiswa=0;
    Mahasiswa peserta[] = new Mahasiswa[kapasitasKelas];
    String namaKelas;
    public Kelas (String dataNama){
        namaKelas = dataNama;}
    public void tambahMahasiswa(Mahasiswa dataMahasiswa){
        peserta[jumlahMahasiswa]=dataMahasiswa; jumlahMahasiswa++; }
}
class Testing{
public static void main (String args[]){
    Mahasiswa budi = new Mahasiswa("Budi","125314001");
    Kelas pbo2 = new Kelas("PBO2");
    pbo2.tambahMahasiswa(budi);

    for (int i=0;i<pbo2.jumlahMahasiswa;i++){
        System.out.println("Nama =" +pbo2.peserta[i].nama);}
}

```





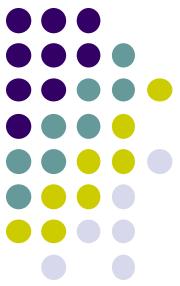
Kelas dan Obyek

- Obyek
 - Sesuatu yang berarti
 - dalam penyelesaian masalah
 - Instance of class
- Kelas
 - Deskripsi atau penjelasan kelakuan
 - Sekelompok obyek
 - Sama atribut dan fungsi/method



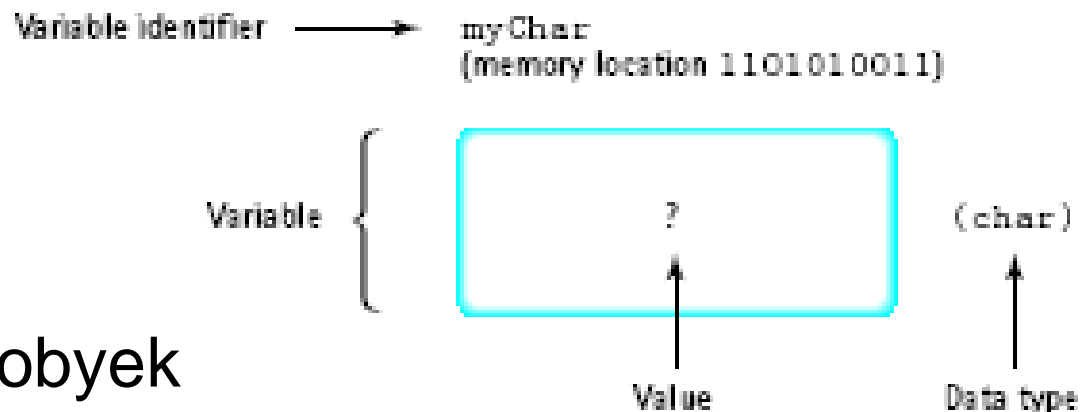
Kata “This”

- Kata “This” merupakan presentasi dari sebuah obyek yang dibentuk. Umumnya digunakan untuk mengakses data atau fungsi kelas



Variabel vs Obyek

- Deklarasi
- Variabel
- Contoh deklarasi variabel
 - `char myChar;`
- Variabel ekuivalen obyek
- Tipe ekuivalen dengan kelas
- Variabel akan menjadi obyek setelah di-instance-kan



Kelas dan Obyek



```
Student sari;= new Student();
```

Kelas

Obyek



Atribut dan Fungsi/Method

- Atribut
 - data yang dimiliki obyek
- Method
 - Sebuah sub program
 - Menyatakan satu kelakuan sebuah kelas

Atribut dan Fungsi/Method



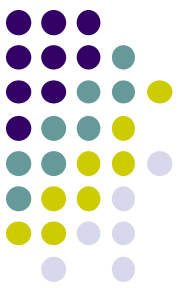
sari.nama; ← Atribut

sari.getBreakfast(); ← Method



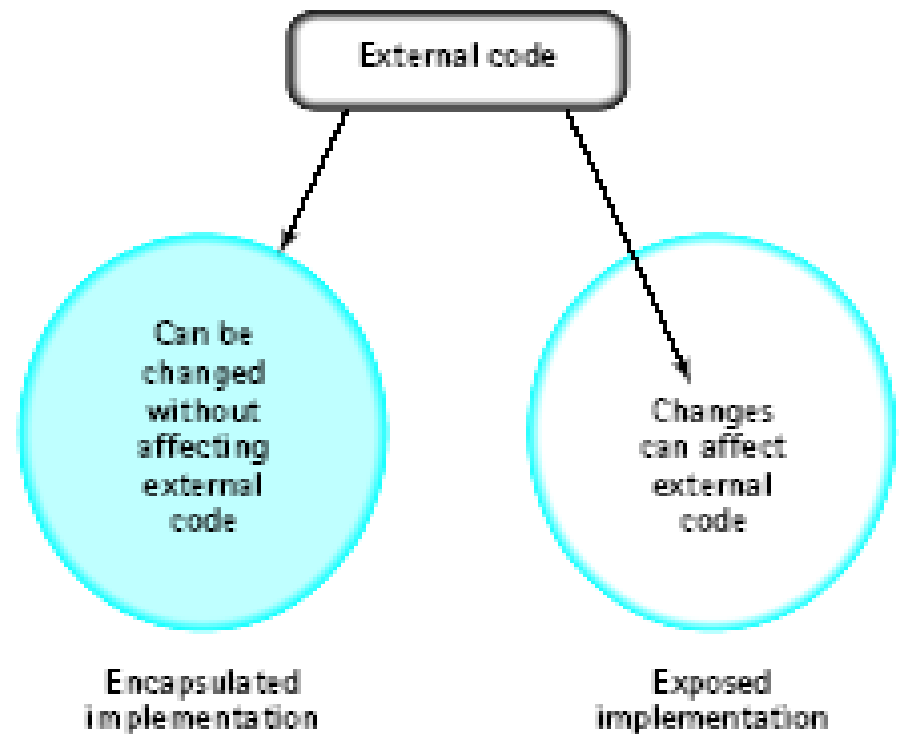
Konvensi Nama

- Gunakan kata bahasa Inggris
- Untuk obyek/variabel diawali huruf kecil
 - calculator
 - lengthOfWord
- Untuk kelas diawali huruf besar
 - CalculatorForm
- Untuk konstanta, semua huruf besar
 - BOOK_TITLE



Encapsulasi

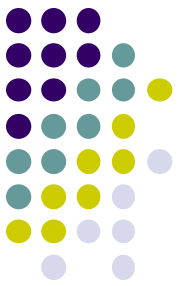
- Berasal dari kata: capsule
- Encapsulasi:
 - Proses perancangan kelas
 - penyembunyian data/atribut dan fungsi
- Fungsi: Reuse dan Modifiability





Encapsulation

- Tujuan
 - Penyembunyian data
 - Data hanya dapat diakses oleh method yang berhak
 - Tipe akses: private, protected, public, final, static
- Tipe akses
 - Private: hanya dapat diakses oleh obyek tsb
 - Protected: dapat diakses oleh obyek kelas turunan
 - Public: dapat diakses semua obyek
 - Final: konstanta
 - Static: dapat diakses oleh kelas



Tata Cara Pembuatan Kelas

Class

```
ClassModifier . . . class Identifier  
  
{  
    ClassDeclaration . . .  
}
```



Tata Cara Pembuatan Kelas

- ClassModifier
 - Public
 - Private
 - Default (kosong)
 - Abstract



Deklarasi Kelas

ClassDeclaration

```
graph LR; A[ClassDeclaration] --- B[FieldDeclaration]; A --- C[MethodDeclaration];
```

FieldDeclaration

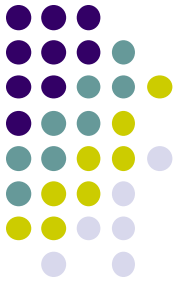
MethodDeclaration



Contoh Kelas

```
public class Name
{
    static final String PUNCT = ", "; // Class constant
    // Instance variables
    private String first;
    private String middle;
    private String last;
    // Declare methods that implement responsibilities of
    Name
}
```

Diagram Kelas



Name	
+PUNCT	: String <static><final>
+first	: String
+middle	: String
+last	: String
// method	



Tipe Akses Data dan Fungsi

- Public
- Private
- Protected
- Default (kosong)
- Final : data berupa konstanta
- Static : data diakses oleh kelas atau semua obyek yang berkelas sama



Fungsi/Method Kelas

- Terbagi atas: constructor, transformer, observer
- Constructor
 - Membuat obyek (instance of class)
- Transformer (set)
 - Mengubah state obyek
- Observer (get)
 - Melihat state obyek tanpa mengubah state



Fungsi/Method Kelas

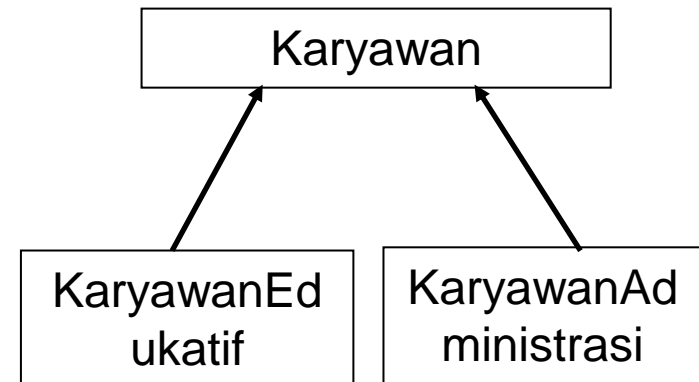
```
public class TrafficLight
{
    // Instance variables
    String currentColor;
    int timeRemaining;

    // Constructor
    public TrafficLight()
    {
    }
    // Observer methods
    public String knowCurrentColor()
    {
    }
    public int knowTimeRemaining()
    {
    }
    // Transformers
    public void changeColor()
    {
    }
    public void decrementTimeRemaining()
    {
    }
}
```

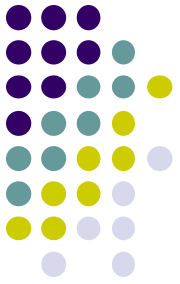
Inheritance



- Inheritance
 - Mekanisme suatu kelas dapat mengakses data dan fungsi kelas lain
- Superclass
 - Kelas yang dapat diturunkan
- Subclass
 - Kelas turunan superclass
- Tugas Implementasikan diagram kelas di samping



Kata “Super”



- Kata “super” merupakan penunjuk untuk kelas dasar/induk



Polymorphisme

- Polymorph
 - Kemampuan duplikasi fungsi/method
- Override
 - Kemampuan definisi ulang fungsi/method di subclass
- Tugas buatlah fungsi dengan menggunakan prinsip polymorfisme pada kelas Karyawan dan turunannya



Kelas Abstrak

- Kelas abstrak ditandai dengan kata kunci *abstract*
- Kelas abstrak merupakan kelas yang tidak dapat diinstanskan dalam sebuah obyek



Interface

- Interface merupakan sebuah “kontrak” sebuah kelas
- Jika sebuah kelas mengimplementasikan sebuah interface maka kelas tersebut wajib membuat fungsi sesuai yang dideklarasikan oleh interface



Kasus I

- Buatlah aplikasi yang akan membantu petugas perpustakaan USD. Aplikasi yang akan dibuat mampu melakukan pencarian dan peminjaman koleksi perpustakaan (buku, cd dan majalah).
- Semua koleksi memiliki id koleksi, judul, penerbit, tahun penerbitan. Ketentuan koleksi adalah sebagai berikut :
 - Koleksi berupa CD/DVD memiliki tambahan data ISBN dan format (audio, tutorial, multimedia)
 - Koleksi berupa buku memiliki tambahan data ISBN dan jumlah halaman
 - Koleksi berupa majalah memiliki tambahan data : ISSN, volume, series

Kasus II



- Peminjam dapat seorang mahasiswa, dosen, atau masyarakat umum. Semua peminjam akan disimpan data nama dan alamat. Ketentuan peminjam sebagai berikut :
 - Mahasiswa memiliki tambahan data : nomor mahasiswa
 - Dosen memiliki tambahan data : nomor induk pegawai
 - Masyarakat umum memiliki tambahan data : nomor KTP
 - Seorang peminjam dapat meminjam beberapa koleksi . Jika pengembalian koleksi lebih dari tanggal yang ditetapkan maka akan dikenakan denda. Aturan peminjaman dan denda akan ditetapkan dalam pertemuan praktikum.



Kasus

- Tugas anda:
 - Buatlah diagram kelas dari kasus diatas!
 - Setiap kelas dilengkapi dengan atribut/properti dan fungsi/method
- Ada pertanyaan?