PERTEMUAN KE - 5
ENKAPSULASI dan KOMPOSISI

A. TUJUAN

Dapat memahami konsep enkapsulasi

Dapat memahami konsep komposisi

Dapat membuat program yang menggunakan enkapsulasi

Dapat membuat program yang menggunakan komposisi

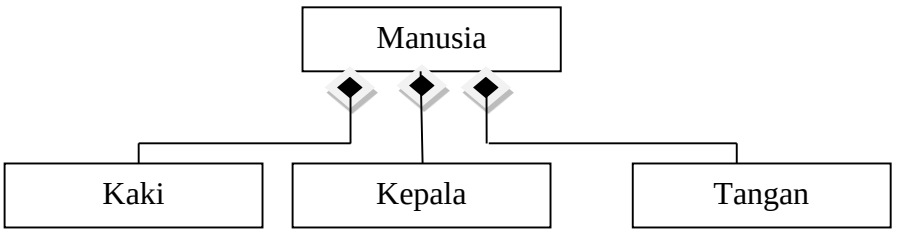
B. TEORI SINGKAT

Enkapsulasi

Adalah salah satu konsep fundamental dalam *object oriented*, lainnya adalah pewarisan, polymorfisme dan abstrak. Enkapsulasi adalah suatu cara pengemasan data dan fungsi dalam sebuah kelas yang terlindungi dari akses secara sembarangan dari pihak luar. Oleh karena itu disebut sebagai proses pengkapsulan, dimana seperti obat bubuk yang terlindungi oleh bungkus kapsul supaya tidak mudah terkontaminasi bakteri. Teknik melakukan enkapsulasi adalah dengan memberikan hak akses private untuk atribut kelas, lalu untuk mengaksesnya dilakukan melalui method yang diberi hak akses public. Jika atribut dideklarasikan dengan hak akses private maka tidak dapat diakses dari luar kelas, maka data tersebut terlindungi. Keuntungan menggunakan enkapsulasi adalah dapat memodifikasi suatu implementasi program tanpa perlu membongkat kode aslinya.

Komposisi

Dalam *object oriented*, satu kelas dapat berelasi dengan kelas lain untuk menggunakan fungsionalitas *method* yang disediakan oleh kelas lain. Hubungan relasi antar kelas diantaranya adalah komposisi. Dua buah kelas dikatakan memiliki relasi komposisi jika salah satu kelas bersifat mempunyai(*own*) kelas yang lain dan apabila pemilik kelas (*owner*) dihapuskan maka kelas yang lain tidak bisa berfungsi. Contohnya kelas manusia mempunyai kelas kepala, kelas tangan, kelas kaki. Bila kelas manusia tersebut dihapus maka kelas kepala, tangan, kaki tidak bisa berfungsi.



C. PRAKTIK

Praktik Enkapsulasi

1. Ketikkan kode program berikut ini

```
public class Pegawai {
    private String nama;
    private String jabatan;
    private int gaji;

    public String getJabatan() { return jabatan; }
    public void setJabatan(String jabatan) {  this.jabatan = jabatan; }
    public long getGaji() {  return gaji; }
    public void setGaji(int gaji) { this.gaji= gaji; }
    public String getNama() { return nama;  }
    public void setNama(String nama) { this.nama = nama;  }

    public void cetakPegawai(){
    System.out.println("");
    System.out.println("Nama Pegawai:"+this.nama);
    System.out.println("Jabatan:"+this.jabatan);
    System.out.println("Gaji Pokok \t:"+this.gaji);
    }
}
```

2. Atribut nama,jabatan, gaji memiliki hak akses private untuk mempertahankan prinsip enkapsulasi. Mengakses atribut tersebut harus menggunakan method yang hak aksesnya bukan private.

```
public class TestEnkapsulasi{
public static void main(String[] args) {
   Pegawai dataPeg=new Pegawai();
   dataPeg.setNama("Budi");
   dataPeg.setJabatan("Supervisor");
   dataPeg.setGaji(4000000);
   dataPeg.cetakPegawai();
}
}
```

Praktik Komposisi

3. Ketikkan program berikut ini

```
public class Perusahaan {
    private String nmPerush;
    private String alamat;
    private Pegawai peg; //komposisi dengan relasi has-a
    public Perusahaan(){
        this.nmPerush="PT. Lancar Jaya";
        this.alamat="Yogyakarta";
        this.peg=new Pegawai();
        peg.setNama("David");
```

```
        peg.setJabatan("Manager");
        peg.setGaji(5000000);
    }
    public void cetakPerush(){
    System.out.println("Nama Perusahaan:"+this.nmPerush);
    System.out.println("Alamat :"+this.alamat);
    peg.cetakPegawai();
    }
}
```

Program no.3 berelasi komposisi dengan program no.1

4. Ketikkan program berikut ini

```
public class TestKomposisi{
public static void main(String[] args) {
    Perusahaan kantor1=new Perusahaan();
    kantor1.cetakPerush();
}}
```

D. LATIHAN

- *Latihan diberikan oleh dosen pengampu pada saat praktikum.*
- *Dikerjakan di laboratorium pada jam praktikum.*

E. TUGAS

- *Tugas diberikan oleh dosen pengampu pada akhir praktikum.*
- *Dikerjakan di rumah dan dilampirkan pada laporan.*