= LAPORAN PRAKTIKUM
PERTEMUAN 4
Prinsip Perancangan Kelas

Alif Alpian Sahrul Muharom (20102007)

Dosen :

Agus Priyanto S.kom, M.kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2022

I. TUJUAN

Mahasiswa diharapkan mampu memahamai tentang perancangan kelas yang baik serta menerjemahkannya dalam bahasa pemrograman.

II. TOOL

a. CodeBlocks / Borland C++ / Visual C++

b. NetBeans IDE 13

c. Java SE Development Kit 18

III. DASAR TEORI

Beberapa prinsip-prinsip yang perlu diperhatikan dalam perancangan suatu kelas, antara lain :

a. Destructor

Destructor adalah fungsi anggota kelas yang akan dijalankan secara otomatis pada saat objek hilang (bertujuan melakukan pembebasan memori yang telah dipakai). Namun dalam java tidak ada fungsi destructor karena menggunakan sistem garbage collector.

b. Visibilitas Bagi Atributes Dan Methods

Dalam suatu kelas diperlukan kontrol akses untuk mengatur siapa saja yang dapat mengakses, atau mengubah nilai dari atributes atau methods dalam kelas tersebut, sehingga penyalahgunaan penggunaan atributes atau methods dalam kelas tersebut dapat dihindar. Untuk menentukan kontrol akses dari atributes atau methods dalam kelas perlu ditambahkan acces specifier terhadap atributes atau methods tersebut. Beberapa kontrol akses yang sering digunakan dalam perancangan suatu kelas antara lain :

a) Public

Kontrol akses yang memberikan akses penuh kepada atributes atau methods, dapat diakses oleh objek dalam kelas itu sendiri atau objek dari kelas lain.

b) Private

Kontrol akses yang memberikan akses tertutup kepada atributes atau methods, hanya dapat diakses oleh objek dalam kelas itu sendiri dan objek dari kelas lain tidak dapat mengaksesnya (seolah-olah atributes atau methods tersebut tersembunyi dalam kelas (encapsulation)).

c) Protected

Kontrol akses yang memberikan akses terbatas kepada atributes atau methods, hanya dapat diakses oleh objek dalam kelas itu sendiri dan objek dari kelas lain yang merupakan turunan dari kelas yang memliki atributes atau methods tersebut .

c. Fungsi Accessor Dan Fungsi Mutator

a) Fungsi Accessor

Merupakan fungsi untuk mendapatkan property dari suatu objek, mengembalikan nilai atau value dari suatu atribut (get).

b) Fungsi Mutator

Merupakan fungsi untuk mengubah property dari suatu objek, mengubah nilai atau value dari sebuah atribut (set).

d. Method Overloading Dan Operator Overloading

Method overloading adalah membuat dua atau lebih methods dengan nama yang sama dalam satu kelas. Method-method tersebut harus dapat dibedakan antara satu dengan yang lain dalam jumlah dan atau tipe datanya. Operator overloading adalah meredefinisi operator (dalam C++) untuk memudahkan operasi terhadap data, khususnya yang melibatkan objek. Konsep ini diilhami oleh operasi yang sering kita alami sehari- hari. Misalnya, tanda '+' dapat dipakai untuk menjumlahkan dua buah bilangan integer atau real, lalu bagaimana bila kita ingin melakukan operasi berikut:
4 apel dan 6 jeruk + 14 apel dan 10 jeruk
matriks A + matriks B
Untuk melakukan operasi di atas, digunakan konsep operator overloading.

e. Melewatkan Argumen/Parameter Ke Method

Ada 2 cara melewatkan argumen/parameter ke method :

a) Melewatkan secara nilai (pass by value)

Diterapkan pada argumen bertipe data primitif (byte, short, int, float, dll). Prosesnya adalah compiler hanya menyalin isi memori (yang telah dialokasikan untuk suatu variabel), dan kemudian menyampaikan salinan tersebut kepada method yang bersangkutan (isi memory merupakan data sesungguhnya yang akan dioperasikan), karena yang disampaikan hanya salinan dari isi memori, maka perubahan yang terjadi pada variabel akibat proses didalam method tidak mempengaruhi nilai variabel asalnya di dalam memori.

b) Melewatkan secara referensi (pass by reference)

Diterapkan pada argumen bertipe data array atau objek. Isi memori pada variabel array atau objek merupakan penunjuk alamat memori yang mengandung data sesungguhnya yang akan dioperasikan. Dengan kata lain, variabel array atau objek menyimpan alamat memori bukan isi memori, maka perubahan yang terjadi pada variabel akibat proses didalam method akan mempengaruhi nilai variabel asalnya.

f. Responbility Driven Design

Bahwa dalam perancangan kelas, semua fungsi/methods yang ada harus mencerminkan perilaku lengkap yang dimiliki kelas tersebut. Semua fungsi/method tersebut bertanggung jawab terhadap maintenanance atribut yang dimiliki kelas.

IV. GUIDED

A. Main

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.Alpiann.Pertemuan4.Guided.Project;


import java.util.Scanner;


/**
 *
 * @author Lenovo
 */
public class Main {
    public static void main(String[] args) {
        Buku novel, fiksi;


        novel = new Buku();
        fiksi = new Buku();


        System.out.println();


        Scanner inputBilangan = new Scanner(System.in);
        Scanner inputKalimat = new Scanner(System.in);


        System.out.println("masukkan judul buku: ");
        novel.setJudul(inputKalimat.nextLine());
```

```
        System.out.println("masukkan pengarang: ");
        novel.setPengarang(inputKalimat.nextLine());

        novel.setInfo(0.2f, 45000);

        System.out.println("masukkan jumlah halaman: ");
        novel.setJumlahHalaman(inputBilangan.nextInt());

        novel.cetak();

        System.out.println();

        System.out.println("masukkan judul buku: ");
        fiksi.setJudul(inputKalimat.nextLine());

        System.out.println("masukkan pengarang: ");
        fiksi.setPengarang(inputKalimat.nextLine());

        fiksi.setInfo(79000);

        System.out.println("masukkan jumlah halaman: ");
        fiksi.setJumlahHalaman(inputBilangan.nextInt());
        fiksi.cetak();
    }
}
```

B. Buku

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.Alpiann.Pertemuan4.Guided.Project;


/**
 *
 * @author Lenovo
 */


public class Buku {
    private String pengarang;
    private String judul;
    private int jumlahHalaman;
    private float diskon;
    private double harga;


    //methods
    public Buku() {
        System.out.println("konstruktor buku dijalankan....");
    }


    public void setPengarang(String pengarang){
        this.pengarang = pengarang;
    }
```

```
    public String getPengarang(){
        return pengarang;
    }

    public String getJudul() {
        return judul;
    }

    public void setJudul(String judul) {
        this.judul = judul;
    }

    public int getJumlahHalaman() {
        return jumlahHalaman;
    }

    public void setJumlahHalaman(int jumlahHalaman) {
        this.jumlahHalaman = jumlahHalaman;
    }
    public float getDiskon() {
        return diskon;
    }

    public void setDiskon(float diskon) {
        this.diskon = diskon;
    }

    public double getHarga() {
        return harga;
    }
```

```
    public void setHarga(double harga) {
        this.harga = harga;
    }

    public void setInfo(float diskon, double harga){
        this.diskon = diskon;
        this.harga = harga - (harga * diskon);
    }

    public void setInfo(double harga) {
        this.diskon = 0.1f;
        this.harga = harga - (harga * diskon);
    }

    public void cetak() {
        System.out.println("judul buku : " +
this.getJudul());
        System.out.println("pengarang buku : " +
this.getPengarang());
        System.out.println("jumlah halaman buku : " +
this.getJumlahHalaman() + "halaman");
        System.out.println("Diskon buku : " +
this.getDiskon());
        System.out.println("Harga buku : " +
this.getHarga());
    }
}
```

Penjelasan Program : Sebuah contoh program penerapan fungsi konstructor, accessor , mutator, method overloading,dan penggunaaan libraary java.util.Scanner.

V. UNGUIDED

A. Main

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
to edit this template
 */
package com.Alpiann.Pertemuan4.Unguided;

import java.util.Scanner;

/**
 *
 * @author Lenovo
 */
public class Main {
    public static void main(String[] args){
        Pegawai Pegawai1;

        Pegawai1 = new Pegawai();

        System.out.println();

        Scanner inputBil = new Scanner(System.in);
        Scanner inputKal = new Scanner(System.in);

        System.out.println("Masukkan NIP : ");
        Pegawai1.setNIP(inputBil.nextInt());

        System.out.println("Masukkan Nama : ");
        Pegawai1.setNama(inputKal.nextLine());

        System.out.println("Masukkan Alamat : ");
        Pegawai1.setAlamat(inputKal.nextLine());
```

```
        System.out.println("Masukkan Jumlah Hari Lembur : ");
        Pegawai1.setJmlhhrLembur(inputBil.nextInt());

        System.out.println("Masukkan Gaji Pokok : ");
        Pegawai1.setGajiPokok(inputBil.nextDouble());

        Pegawai1.setTotalGaji(0);

        Pegawai1.cetak();

    }
}
```

B. Pegawai

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
to edit this template
 */
package com.Alpiann.Pertemuan4.Unguided;

import java.util.Scanner;

/**
 *
 * @author Lenovo
 */
public class Main {
    public static void main(String[] args){
        Pegawai Pegawai1;

        Pegawai1 = new Pegawai();

        System.out.println();
```

```
        Scanner inputBil = new Scanner(System.in);
        Scanner inputKal = new Scanner(System.in);

        System.out.println("Masukkan NIP : ");
        Pegawai1.setNIP(inputBil.nextInt());

        System.out.println("Masukkan Nama : ");
        Pegawai1.setNama(inputKal.nextLine());

        System.out.println("Masukkan Alamat : ");
        Pegawai1.setAlamat(inputKal.nextLine());

        System.out.println("Masukkan Jumlah Hari Lembur : ");
        Pegawai1.setJmlhhrLembur(inputBil.nextInt());

        System.out.println("Masukkan Gaji Pokok : ");
        Pegawai1.setGajiPokok(inputBil.nextDouble());

        Pegawai1.setTotalGaji(0);

        Pegawai1.cetak();

    }
}
```

Penjelasan Program : Sebuah program yang dapat mencatat data pegawai, memiliki sebuah konstructor awal yang ketika dijalankan akan menampilkan kata “Konstructor pegawai dijalankan” , menggunakan fungsi mutator dan accessor, dimana fungsi mutator untuk menginputkan data pada atribut menggunakan sebuah library java.util.Scanner, kemudian fungsi accessor digunakan untuk memanggil data yang diinputkan pada atribut, memiliki method overloading pada setTotalGaji, dimana ada 2 method setTotalGaji dalam satu kelas, yang dibedakan dari tipe data dan inputannyaa.

VI. KESIMPULAN

Dengan mempelajari ini kita dapat memahami prinsip prinsip perancangan kelas dan mengimplementasikan perancangan kelas yang baik dan benar