

LAPORAN PRAKTIKUM PBO
Class Design Hints



Disusun oleh :
Muhamad Rafli Nur Ikhsan
201511048
D-3 Teknik Informatika 2B

Jurusan Teknik Komputer dan Informatika
Program studi D3 Teknik Informatika
Politeknik Negeri Bandung

1. Kasus 2

- SS hasil akhir program

```
Bala-Bala[20]    Rp.1000.0
Gehu[20]        Rp.1000.0
Molen[20]       Rp.1000.0
Masukan makanan yang dibeli : Molen
Masukan jumlah : 10
Bala-Bala[20]    Rp.1000.0
Gehu[20]        Rp.1000.0
Molen[10]       Rp.1000.0

Process finished with exit code 0
```

- Amati desain setiap class, Apakah desain class tersebut sudah memenuhi konsep OOP yang benar? Jika tidak, coba anda perbaiki dengan mengacu pada Design Hint di Buku Chapter 4.10. Stiap perubahan yang dibuat harus dibubuhi penjelasan serta argumentasi yang jelas.

Belum, karena desain tersebut tidak memenuhi poin-poin di Class Design Hints seperti yang ada di buku, poin-poin tersebut adalah :

- Always keep data private

Dalam desain yang ada data data dideklarasikan dalam bentuk public, sehingga harus diubah ke bentuk private.

- Don't use too many basic types in a class

Dalam desain yang ada terlalu banyak basic types dalam sebuah kelas, sehingga basic types tersebut harus dibuat lebih sederhana untuk membuat class tersebut lebih mudah untuk dipahami.

- Break up classes that have too many responsibilities

Dalam desain yang ada Class Restourant terlalu banyak responsibilities di dalamnya yang membuat desain program itu tidak sederhana. Saya membuat Class baru untuk memisahkan method yang berfungsi mengurus makanannya, sedangkan Class restaurant berfungsi untuk menampilkan makanannya.

- Make the names of your classes and methods reflect their responsibilities

Dalam desain yang ada walaupun judul Classnya adalah Restourant tetapi di dalamnya mengandung method-method yang mengurus tentang detail makannya. Akan lebih sederhana jika method yang mengatur tentang detail makanan dipisahkan dengan Class Restourant tersebut.

- Ada kebutuhan untuk mengembangkan aplikasi tersebut, dengan menambah fitur pemesanan dan mengurangi setiap stok yang ada.

Untuk fitur pemesanan dan pengurangan saya membuat fitur dimana kita harus menginputkan makanan apa yang ingin kita pesan dan berapa jumlah dari pesanan tersebut. Seperti contoh dari cara memesannya sebagai berikut:

```

System.out.printf("Masukan makanan yang dibeli : ");
makan = sc.nextLine();
System.out.printf("Masukan jumlah : ");
banyak = sc.nextInt();
menu.mesen(makan,banyak);
menu.tampilMenuMakanan();

```

Di sini untuk memesan makanan diharuskan untuk menginputkan makanan yang akan dipesan yang akan ditampung oleh variabel makan yang bertipekan string untuk menampung nama makanannya, dan variabel banyak yang bertipekan integer untuk menampung banyak makannya, yang nantinya akan di proses dalam method mesen yang isinya :

```

public void mesen(String nama, int jumlah){
    for(int i = 0; i <= id; i++){
        if(nama.equals(menuMakanan[i].getNama_makanan())){
            if(menuMakanan[i].getStok() >= jumlah){
                menuMakanan[i].kurang(jumlah);
            }
            else{
                System.out.println("Stok "+menuMakanan[i].getNama_makanan() + " tidak cukup");
            }
        }
        else{
            //
            //      System.out.println("Makanan tidak ada");
            //
        }
    }
}

```

- Permasalahan yang dihadapi

Pada awalnya tidak mengetahui konsep OOP yang benar itu seperti apa

- Solusi dari permasalahan yang dihadapi

Melihat konsep OOP yang benar di buku panduan dan referensi lainnya

2. Kasus 3

- SS hasil akhir program

```

Rinnegan
Mangekyou Sharingan
Sharingan
Susanoo

Process finished with exit code 0

```

- Pelajari dan jelaskan bagaimana urutan dan cara kerjanya!
 - Urutan output yang diinginkan adalah Rinnegan, Mangekyou Sharingan, Sharingan dan Susanoo. Output sementara yang tersedia adalah Sharingan dan Susanoo, sehingga harus menambahkan output Rinnegan dengan cara memanggil modul yang menampung data Rinnegan

```
void printDojutsu() {
    super.printDojutsu();
    System.out.println(this.Dojutsu);
}
```

Dengan ditambahkannya `super.printDojutsu()` maka method yang ada di Class Itachi akan terpanggil dimana dalam method tersebut mengoutputkan Dojutsu yang ada di Parent Class Itachi (Rikudo) dimana Dojutsu tersebut adalah Rinnegan. Sehingga outputnya akan menjadi :

```
Rinnegan
Rinnegan
Sharingan
Susanoo

Process finished with exit code 0
```

- Method `super.printDojutsu()` mengoutputkan 2 Dojutsu tambahan yang berisikan Rinnegan, untuk membangkitkan Dojutsu Mangekyou Sharingan dapat dilakukan dengan cara:

```
void printDojutsu() {
    System.out.println(this.Dojutsu);
    setDojutsu();
    System.out.println(this.Dojutsu);
}

private void setDojutsu() {
    this.Dojutsu = "Mangekyou Sharingan";
}
```

Pemanggilan method `setDojutsu()` dalam method `printDojutsu()` berfungsi untuk mengganti nilai `this.Dojutsu` yang semula "Rinnegan" menjadi "Mangekyou Sharingan" sehingga pada saat pengoutputan `this.Dojutsu` yang kedua nilainya sudah berubah menjadi "Mangekyou Sharingan".

- Permasalahan yang dihadapi
Tidak ada
- Solusi dari permasalahan yang dihadapi
Tidak ada