

LAPORAN PRAKTIKUM
PEMOGRAMAN BERORIENTASI OBJECT
“Fundamental Programming Structures in Java”



Disusun oleh :
Alifah Fisalsabilawati
201511035
2B – D3 Teknik Informatika

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

Kasus 2

- Amati desain setiap class, Apakah desain class tersebut sudah memenuhi konsep OOP yang benar?

Menurut buku Java Fundamental Core 10 th Edition disebutkan ada 7 konsep yang sesuai dengan konsep OOP yaitu :

1. *Always keep data private*
Ketika data dirubah menjadi private, perubahan dalam representasi tidak akan mempengaruhi pengguna kelas, selain itu kita juga bisa lebih mudah mendeteksi bug.
2. *Always initialize data.*
Karena Java tidak akan menginisialisasi variabel lokal untuk kita, tetapi akan menginisialisasi block instance dari objek. Maka dari itu inisialisasi semua variabel secara eksplisit, baik sesuai dengan default atau dengan menyetel default di semua konstruktor.
3. *Don't use too many basic types in a class.*
Buatlah syntax sesederhana mungkin, hal ini bertujuan untuk membuat kelas Anda lebih mudah dipahami dan diubah.
4. *Not all fields need individual field accessors and mutators.*
untuk mencegah data agar tidak mudah berubah, maka dilakukanlah enkapsulasi.
5. *Break up classes that have too many responsibilities.*
Jangan terlalu banyak menuliskan syntax yang memiliki banyak responbility, maka dari itu kita harus kelompokkan mana saja method yang sesuai dengan class nya.
6. *Make the names of your classes and methods reflect their responsibilities.*
Semua variabel harus memiliki nama yang bermakna dengan apa yang mereka wakili. Penamaan class yang baik biasanya menggunakan kata benda. Selain class method juga harus memiliki nama yang bermakna agar tidak tertukar dengan nama class. Biasanya nama method diawali dengan get dan dilanjutkan dengan nama class nya, sebagai contoh getMakanan
7. *Prefer immutable classes*

Dan program yang ada pada kasus 2 belum lah sesuai dengan konsep OOP maka dari itu saya membuat class baru yaitu Makanan.java karena pada Restaurant.java terlalu banyak method yang membahas makanan dan merubah data yang awalnya public menjadi private. Berikut merupakan program yang sudah di ubah menggunakan konsep OOP

Restaurant.java

```
1 package kasuscoba;
2
3 public class Restaurant {
4     private Makanan[] menuMakanan;
5     private static byte id=0;
6
7     public Restaurant() {
8         menuMakanan = new Makanan[10];
9     }
10
11     public void tampilMenuMakanan()
12     {
13         for(int i = 0; i <= id; i++)
14         {
15             if(!isOutOfStock(i))
16             {
17                 this.menuMakanan[i].TampilMakanan();
18             }
19         }
20     }
21
22     public void tambahMenuMakanan(String nama, double harga, int stok) {
23         this.menuMakanan[id] = new Makanan (nama, harga, stok);
24     }
25
26
27     public boolean isOutOfStock(int id){
28
29         return menuMakanan[id].getStok() == 0;
30     }
31
32     public static void nextId(){
33         id++;
34     }
35
36     public void Pemesanan (String namaMakanan, int jumlah)
37     {
38         for (int i = 0 ; i <= id; i++)
39         {
40             if(namaMakanan.equals(menuMakanan[i].getNama_makanan()))
41             {
42                 if(menuMakanan[i].getStok() >= jumlah)
43                 {
44                     System.out.println(jumlah + " " + menuMakanan[i].getNama_makanan() + " " + "Terjual!");
45                     menuMakanan[i].kurangStok(jumlah);
46                 }
47                 else System.out.println("Stok" + menuMakanan[i].getNama_makanan() + "tidak cukup !");
48             }
49         }
50     }
51 }
52
```

Makanan.java

```
1 package kasuscoba;
2
3 public class Makanan {
4     private String nama_makanan;
5     private double harga_makanan;
6     private int stok;
7
8     public Makanan (String _namaMakanan, double _hargaMakanan, int _stok)
9     {
10         this.nama_makanan = _namaMakanan;
11         this.harga_makanan = _hargaMakanan;
12         this.stok = _stok;
13     }
14
15     public String getNama_makanan() {
16         return nama_makanan;
17     }
18
19     public int getStok() {
20         return stok;
21     }
22
23     public void TampilMakanan() {
24         System.out.println(nama_makanan + "[" + stok + "]" + "Rp." + harga_makanan);
25     }
26
27     public void kurangStok(int jumlah) {
28         stok -= jumlah;
29     }
30 }
```

RestaurantMain.java

```
1 package kasuscoba;
2
3 import java.util.Scanner;
4
5 public class RestaurantMain {
6     public static void main(String[] args) {
7         Restaurant menu = new Restaurant();
8         Scanner sc = new Scanner(System.in);
9         String makanan;
10        int pesanan;
11
12        System.out.println("====Menu makanan====");
13        menu.tambahMenuMakanan("Bala-Bala", 1_000, 20);
14        Restaurant.nextId();
15        menu.tambahMenuMakanan("Gehu", 1_000, 20);
16        Restaurant.nextId();
17        menu.tambahMenuMakanan("Tahu", 1_000, 0);
18        Restaurant.nextId();
19        menu.tambahMenuMakanan("Molen", 1_000, 20);
20        menu.tampilMenuMakanan();
21        System.out.println("=====");
22
23        System.out.println("Masukan nama makanan :");
24        makanan = sc.nextLine();
25        System.out.println("Jumlah makanan yang dibeli :");
26        pesanan = sc.nextInt();
27        System.out.println("=====");
28        menu.Pemesanan(makanan, pesanan);
29        System.out.println("Stok makanan tersisa");
30        menu.tampilMenuMakanan();
31    }
32 }
```

Output

```
<terminated> RestaurantMain (2) [Java Application] C:\F
====Menu makanan====
Bala-Bala[20]Rp.1000.0
Gehu[20]Rp.1000.0
Molen[20]Rp.1000.0
=====
Masukan nama makanan :
Gehu
Jumlah makanan yang dibeli :
10
=====
10 Gehu Terjual!
Stok makanan tersisa
Bala-Bala[20]Rp.1000.0
Gehu[10]Rp.1000.0
Molen[20]Rp.1000.0
```

- Ada kebutuhan untuk mengembangkan aplikasi tersebut, dengan menambah fitur pemesanan dan mengurangi setiap stok yang ada. Apakah dengan desain program yang ada dapat dikembangkan?

Saya menambahkan fitur pemesanan yang berfungsi untuk memesan makanan dan mengurangi stok makanan ketika makanan berhasil di pesan.

- Permasalahan yang dihadapi
Jumlah stok yang tidak berkurang terus
- Solusi dari permasalahan yang dihadapi
Saat menuliskan program ternyata saya salah memasukkan variabel, yang harusnya saya menuliskan namaMakanan tapi saya malah menuliskan menuMakanan

Kasus 3

- Salin ulang baris kode dan lakukan eksekusi terhadap program tersebut
- Jika dirunning, akan menampilkan output awal sebagai berikut:

```
<terminated> NarutoAnir
Sharingan
Susanoo
```

- Bagaimana cara agar Sasuke dapat menguasai Dojutsu & KekkeiGenkai (*menampilkan output*) sebagai berikut:

```
Rineggan
Mangekyou Sharingan
Sharingan
Susanoo
```

- Pelajari dan jelaskan bagaimana urutan dan cara kerjanya!
Awalnya output yang muncul hanyalah Sharingan dan Susanoo, untuk menampilkan Rineggan dan Mangekyou Sharingan hal yang pertama saya lakukan adalah menambahkan syntax *super.printDojutsu()* di Sasuke.java seperti dibawah ini

```
3 public class Sasuke extends Itachi {
4     String Dojutsu = "Sharingan";
5
6     void printDojutsu() {
7         super.printDojutsu(); //untuk menampilkan super class dari dojutsu yaitu Rinegan
8         System.out.println(this.Dojutsu);
9     }
10 }
```

Setelah saya lakukan pencarian ternyata *super.print* merupakan syntax untuk menampilkan atau print dari super class. Karena Sasuke.java merupakan extend dari Itachi.java dan Itachi.java juga ternyata extend dari Rikudo.java yang di dalam class nya

berisi `protected String Dojutsu = "Rinnegan"`; dan di wariskan ke Itachi.java yang selanjutnya di Itachi.java di panggil dua kali pada `void printDojutsu ()` yang menyebabkan output program seperti berikut

```
<terminated> NarutoAnime [Java A  
Rinnegan  
Rinnegan  
Sharingan  
Susanoo
```

Maka dari itu langkah selanjutnya saya menambahkan syntax `setDojutsu ()` di Itachi.java seperti program dibawah , hal ini bertujuan untuk menampilkan Mangekyou Sharingan yang ternyata ada pada dalam `void setDojutsu()`

```
void printDojutsu() {  
    System.out.println(this.Dojutsu);  
    setDojutsu(); //untuk memanggil void setDojutsu yang isi nya memanggil Mangekyou Sharingan  
    System.out.println(this.Dojutsu);  
}
```

Dan akhirnya output final dari program diatas adalah seperti berikut

```
<terminated> NarutoAnime [Java Appli  
Rinnegan  
Mangekyou Sharingan  
Sharingan  
Susanoo
```