

Pemrograman Berorientasi Object

Tugas 5 – Class Design Hints



Disusun oleh :

Nadhifah Nur Shadrina

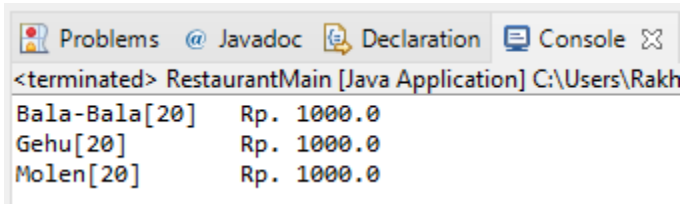
201511021

D3-2A

POLITEKNIK NEGERI BANDUNG
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
PROGRAM STUDI D3 TEKNIK INFORMATIKA
2021-2022

Kasus 2

1. Hasil Akhir Program



```
<terminated> RestaurantMain [Java Application] C:\Users\Rakh
Bala-Bala[20]      Rp. 1000.0
Gehu[20]           Rp. 1000.0
Molen[20]          Rp. 1000.0
```

2. Apakah design class tersebut sudah memenuhi konsep OOP yang benar?

Design class dari program untuk kasus 2 ini belum memenuhi konsep OOP yang benar meskipun beberapa konsep sudah memenuhi seperti,

- Terdapat inialisasi dari setiap data yang digunakan dalam class Restaurant.
- Penggunaan nama variable yang sesuai dengan representasinya.

Namun, ada hal yang perlu diperbaiki yaitu,

- Pada class Restaurant belum adanya encapsulation (accessor dan mutators).
- Pada class Restaurant terlalu banyak responsibility jadi perlunya dilakukan pemisahan class.
- Penggunaan atribut dengan modifier “private”

Karena pada class Restaurant menggunakan konsep Encapsulation, dimana atribut data harus dijaga dengan modifier “private”. Artinya, jika suatu atribut dideklarasikan dengan menggunakan “private” maka atribut tersebut tidak dapat diakses diluar class Restaurant.

Sebelum :

```
public class Restaurant {
    /* Menset semua atribut data menjadi private */
    public String[] nama_makanan;
    public double[] harga_makanan;
    public int[] stok;
    public static byte id=0;
```

Sesudah :

```
private String[] nama_makanan;
private double[] harga_makanan;
private int[] stok;
private static byte id=0;
```

Apakah dengan design program yang ada dapat dikembangkan ?

```
package Kasus2;

public class Restaurant {

    // public static byte id=0;
    private Makanan[] menuMakanan;

    private static byte id=0;

    Restaurant(){
        menuMakanan = new Makanan[10];
    }

    public void tambahMenuMakanan(String nama, double harga, int stok) {
        this.menuMakanan[id] = new Makanan(nama,harga,stok);
        // this.harga_makanan[id] = harga;
        // this.stok[id] = stok;
    }

    public void tampilMenuMakanan(){
        for(int i =0; i<=id;i++){
            if(!isOutOfStock(i)){
                System.out.println(menuMakanan[i].getNama() + "["+menuMakanan[i].getStock()+"]"+"\\tRp. "+menuMakanan[i].getHarga());
            }
        }
    }

    public boolean isOutOfStock(int id){
        if(menuMakanan[id].getStock() == 0){
            return true;
        }else{
            return false;
        }
    }

    public static void nextId(){
        id++;
    }
}

package Kasus2;

public class Makanan {
    /* Menget semua atribut data menjadi "private" */
    private String nama_makanan;
    private double harga_makanan;
    private int stok;
    //private static byte id=0;

    public Makanan(String nama, double harga, int stok) {
        this.nama_makanan = nama;
        this.harga_makanan = harga;
        this.stok = stok;
    }

    public String getNama() {
        return this.nama_makanan;
    }

    public void setNama(String namaMakanan) {
        this.nama_makanan= namaMakanan;
    }

    public double getHarga() {
        return this.harga_makanan;
    }

    public void setNama(double hargaMakanan) {
        this.harga_makanan= hargaMakanan;
    }

    public int getStock() {
        return stok;
    }

    public void sisaStok(int stokk) {
        this.stok = stok - stokk;
    }
}
```

```

package Kasus2;

public class RestaurantMain {
    public static void main(String[] args) {

        Restaurant menu = new Restaurant();

        // menambahkan menu makanan
        menu.tambahMenuMakanan("Bala-Bala", 1_000, 20);
        Restaurant.nextId();
        menu.tambahMenuMakanan("Gehu", 1_000, 20);
        Restaurant.nextId();
        menu.tambahMenuMakanan("Tahu", 1_000, 0);
        Restaurant.nextId();
        menu.tambahMenuMakanan("Molen", 1_000, 20);

        menu.tampilMenuMakanan();

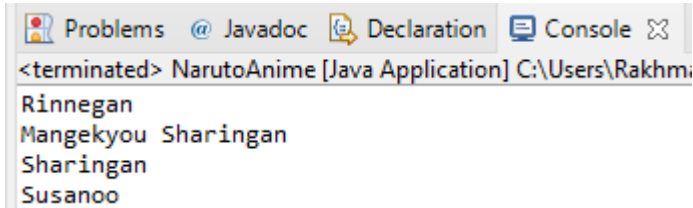
    }
}

```

3. Permasalahan yang dihadapi
 - Belum dapat menambahkan fitur pemesanan dan mengurangi stok
 - Kesulitan saat melakukan modifikasi program.
4. Solusi
 - Mencoba –coba kembali

Kasus 3

1. Hasil akhir program



```
<terminated> NarutoAnime [Java Application] C:\Users\Rakhm...
Rinnegan
Mangekyou Sharingan
Sharingan
Susanoo
```

2. Bagaimana cara agar sasuke dapat menguasai Dojutsu & KekkeiGenkai (*menampilkan output*)

- Rinnegan
- Mangekyou Sharingan
- Sharingan
- Susanoo.

Penjelasan :

Program pada kasus 3 ini menerapkan bagaimana penggunaan keyword super dalam method overriding.

- 1) Melakukan modifikasi pada class Itachi dan class Sasuke. Dimana class Sasuke merupakan inherit dari class Itachi dan class Itachi merupakan inherit dari class Rikudo. Artinya class Rikudo disini berperan sebagai superclass/parent.
- 2) Untuk menampilkan urutan output seperti di atas, Dalam method main dibuat object baru dari class Sasuke bernama s dan melakukan pemanggilan terhadap printDojutsu().
- 3) Ketika pemanggilan method printDojutsu() dengan keyword super, maka akan dipanggil method printDojutsu() yang merupakan parent dari subclassnya.
- 4) Dalam method printDojutsu() pada class Itachi yang akan menampilkan “Rinnegan” , " Mangekyou Sharingan” dan “Sharingan” secara berurutan.
- 5) Selanjutnya dalam method main dibuat object baru dengan nama i dan melakukan pemanggilan terhadap method printKekkeiGenkai() yang ada dalam class Itachi, dan menampilkan “Susanoo”.

Sebelum modifikasi :

```
package Kasus3;

// class turunan dari Rikudo
public class Itachi extends Rikudo {
    private String KekkeiGenkai = "Susanoo";
    private String Dojutsu = super.Dojutsu;

    void printKekkeiGenkai() {
        System.out.println(this.KekkeiGenkai);
    }

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

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

package Kasus3;

// class turunan dari Itachi
public class Sasuke extends Itachi {
    String Dojutsu = "Sharingan";

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

Sesudah modifikasi :

```
package Kasus3;

// class turunan dari Rikudo
public class Itachi extends Rikudo {
    private String KekkeiGenkai = "Susanoo";
    private String Dojutsu = super.Dojutsu;

    void printKekkeiGenkai() {
        System.out.println(this.KekkeiGenkai);
    }

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

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

package Kasus3;

// class turunan dari Itachi
public class Sasuke extends Itachi {
    String Dojutsu = "Sharingan";

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

3. Permasalahan yang dihadapi

- Belum mengetahui penggunaan konsep inheritance dengan baik.
- Belum mengetahui keyword super.

4. Solusi

- Memahami kembali dan melakukan pencarian referensi terkait dengan konsep inheritance.
- Mencari tahu penggunaan dari keyword super yang ada pada program.

Refersi :

<https://beginnersbook.com/2014/07/super-keyword-in-java-with-example/>