

Studi Kasus 5

Pertemuan ke-5

Diketahui: Terdapat 2 class yaitu Restaurant.java dan RestaurantMain.java

Restaurant.java

```
public class Restaurant {  
    public String[] nama_makanan;  
    public double[] harga_makanan;  
    public int[] stok;  
    public static byte id=0;  
  
    public Restaurant() {  
        nama_makanan = new String[10];  
        harga_makanan = new double[10];  
        stok = new int[10];  
    }  
}
```

Lanjutan Restaurant.java

```
public void tambahMenuMakanan(String nama, double harga, int stok) {  
    this.nama_makanan[id] = nama;  
    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(nama_makanan[i] +"["+stok[i]+""]"+"\\tRp. "+harga_makanan[i]);  
        }  
    }  
}
```

Lanjutan Restaurant.java

```
public boolean isOutOfStock(int id){  
    if(stok[id] == 0){  
        return true;  
    }else{  
        return false;  
    }  
}  
  
public static void nextId(){  
    id++;  
}  
}
```

RestaurantMain.java

```
public class RestaurantMain {  
    public static void main(String[] args) {  
        Restaurant menu = new Restaurant();  
        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();  
    }  
}
```

Instruksi Kasus 2

- Amati desain setiap class, Apakah desain class tersebut sudah memenuhi konsep OOP yang benar? Jika tidak, coba anda perbaiki dengan mengacu pada Design Hnt di Buku Chapter 4.10. Setiap perubahan yang dibuat harus dibubuhi penjelasan serta argumentasi yang jelas.
- Ada kebutuhan untuk mengembangkan aplikasi tersebut, dengan menambah fitur pemesanan dan mengurangi setiap stok yang ada. Apakah dengan desain program yang ada dapat dikembangkan? Jika Sulit kemukakan alasannya dan bandingkan dengan desain class hasil modifikasi anda.
- Setiap jawaban ditulis dalam Google doc dan di simpan di Folder Pertemuan 5

Diketahui: Terdapat 4 class Rikudo, Itachi, Sasuke, dan NarutoAnime (Main Class).

Rikudo.java

```
public class Rikudo {  
    protected String Dojutsu = "Rinnegan";  
    protected String Senjutsu = "Sage Mode";  
}
```

Itachi.java

```
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";  
    }  
}
```


Sasuke.java

```
public class Sasuke extends Itachi {  
    String Dojutsu = "Sharingan";  
  
    void printDojutsu() {  
        System.out.println(this.Dojutsu);  
    }  
}
```

NarutoAnime.java

```
public class NarutoAnime {  
    public static void main(String[] args) {  
        Sasuke s = new Sasuke();  
        s.printDojutsu();  
        Itachi i = new Sasuke();  
        i.printKekkeiGenkai();  
    }  
}
```

Instruksi Kasus 3

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

Sharingan

Susanoo

Pertanyaan Kasus 3

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

Rinnegan

Mangekyou Sharingan

Sharingan

Susanoo

- Pelajari dan jelaskan bagaimana urutan dan cara kerjanya!
- Kondisi yang harus dipenuhi:
 - Tidak boleh menambahkan `System.out.println()` di class mana pun
 - Tidak boleh membuat function baru di class mana pun
 - Class `NarutoAnime` tidak boleh diubah sedikit pun
 - Tidak boleh mengubah encapsulation
- Setiap jawaban ditulis dalam Google doc dan di simpan di Folder Pertemuan 5

Clue: Chapter 5.1