

LAPORAN TUGAS MINGGU 5

Object & Class 2

Pemorgaman Berorientasi Objek

Resume ini disusun untuk memenuhi Tugas Mata Kuliah Pemrograman Berorientasi Objek



Disusun oleh:
Muhammad Rivan Rivaldi 211511048

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

1. Restaurant.java

```
public class Restaurant {  
    private Makanan[] menuMakanan;  
    private static byte id=0;  
  
    public Restaurant(){  
        menuMakanan = new Makanan[10];  
    }  
  
    public void tampilMenuMakanan() {  
        for(int i=0;i<=id;i++) {  
            if(!isOutOfStock(i)) {  
                //diubah  
                this.menuMakanan[i].TampilMakanan();  
            }  
        }  
    }  
  
    public void tambahMenuMakanan(String nama, double harga, int stok) {  
        this.menuMakanan[id] = new Makanan(nama, harga, stok);  
    }  
  
    public boolean isOutOfStock(int id) {  
        return menuMakanan[id].getStok() == 0;  
    }  
  
    public static void nextId() {  
        id++;  
    }  
}
```

```
public void Pemesanan(String namaMakanan, int jumlah) {  
    for(int i=0;i<=id;i++) {  
        if(namaMakanan.equals(menuMakanan[i].getNamaMakanan())) {  
            if(menuMakanan[i].getStok() >= jumlah) {  
                System.out.println(jumlah + " " + menuMakanan[i].getNamaMakanan() + " Terjual");  
                menuMakanan[i].kurangStok(jumlah);  
            }  
            else System.out.println("Stok" + menuMakanan[i].getNamaMakanan() + " tidak cukup!");  
        }  
    }  
}
```

2. 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("Gebu", 1_000, 20);  
        Restaurant.nextId();  
        menu.tambahMenuMakanan("Tahu", 1_000, 0);  
        Restaurant.nextId();  
        menu.tambahMenuMakanan("Molen", 1_000, 20);  
  
        //Show pesanan  
        menu.Pemesanan("Molen", 6);  
        System.out.println("\n===== Stok Makanan Tersisa =====");  
        menu.tampilMenuMakanan();  
    }  
}
```

3. Makanan.java

```
public class Makanan {  
    private String nama_makanan;  
    private double harga_makanan;  
    private int stok;  
  
    //Make a Constructor  
    public Makanan(String _namaMakanan, double _hargaMakanan, int _stok) {  
        this.nama_makanan = _namaMakanan;  
        this.harga_makanan = _hargaMakanan;  
        this.stok = _stok;  
    }  
  
    //Encapsulation  
    public String getNamaMakanan() {  
        return nama_makanan;  
    }  
  
    public int getStok() {  
        return stok;  
    }  
  
    public void TampilMakanan() {  
        System.out.println(nama_makanan + "[" + stok + "]" + "Rp" + harga_makanan);  
    }  
  
    //Stok Decrease  
    public void kurangStok(int jumlah) {  
        stok -= jumlah;  
    }  
}
```

Mengacu pada buku Java Fundamental Core 10th Edition disebutkan bahwa “*always keep data private*” sedangkan pada soal syntax yang diinisiasikan berupa public. Setelah itu ada beberapa yang saya ubah yakni namaMakanan, hargaMakanan, dan stokMakanan kedalam kelompok Makanan karena dalam buku dijelaskan “*don’t use too many basic types in a class*”. Tidak lupa untuk melakukan enkapsulasi contohnya getStok dan getNamaMakanan, mengingat “*not all fields need individual field accessors and mutators*”.

Pada syntax sebelumnya didalam class Restaurant tidak terfokus kedalam sistematis pengimplementasian dari dunia nyata, yang seharusnya Restaurant tidak sampai menggunakan method-method yang berhubungan dengan makanannya langsung, oleh karena itu saya membuat class Makanan.