

# **LAPORAN PRAKTIKUM JAVA OBJECT ORIENTED PROGRAMMING**

*Laporan praktikum ini disusun untuk memenuhi Tugas Mata Kuliah Pemrograman  
Berorientasi Objek Praktek*



Disusun oleh:

Wildan Setya Nugraha

211511032

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

**2022**

## DAFTAR ISI

DAFTAR ISI	i
SOAL 1	1
SOAL 2	4

# SOAL 1

## 1. Screenshot Program

```
1 package Task_1;
2
3 public class Restaurant {
4     private String [] nama_makanan;
5     private double[] harga_makanan;
6     private int[] stok;
7     private static byte id = 0;
8
9     public Restaurant() {
10         nama_makanan = new String [10];
11         harga_makanan = new double[10];
12         stok = new int[10];
13     }
14
15
16 // Setter
17 public void tambahMenuMakanan(String nama, double harga, int stok) {
18     this.nama_makanan[id] = nama;
19     this.harga_makanan[id] = harga;
20     this.stok[id] = stok;
21 }
22
23 // Getter
24 public void tampilMenuMakanan() {
25     for(int i = 0; i<=id; i++) {
26         if(!isOutOfStock(i)) {
27             System.out.println(nama_makanan[i] + "[" + stok[i] + "]" + "\tRp. " + harga_makanan[i]);
28         }
29     }
30 }
31
32 private boolean isOutOfStock(int id) {
33     if(stok[id] == 0) {
34         return true;
35     } else {
36         return false;
37     }
38 }
39
40 public static void nextId() {
41     id++;
42 }
43
44 public void pesanMakanan(String namaMakanan, int jumlahPesanan) {
45     for(int i=0; i<=id; i++) {
46         if (this.nama_makanan[i].equalsIgnoreCase(namaMakanan)) {
47             System.out.print("Makanan tersedia");
48             if(!isOutOfStock(i) && jumlahPesanan <= this.stok[i]) {
49                 this.stok[i] -= jumlahPesanan;
50                 System.out.println(", Jumlah Bayaran: " + (jumlahPesanan*this.harga_makanan[i]));
51             } else if(isOutOfStock(i)) {
52                 System.out.println(", tapi stoknya tidak ada");
53             } else {
54                 System.out.println(". Maaf, jumlah pesanan anda melebihi stok");
55             }
56         }
57     }
58 }
59 }
60 }
```

```

1 package Task_1;
2
3 public class RestaurantMain {
4     public static void main(String[] args){
5         Restaurant menu = new Restaurant();
6
7         menu.tambahMenuMakanan("Bala-bala", 1_000, 20);
8         Restaurant.nextId();
9         menu.tambahMenuMakanan("Gehu", 1_000, 20);
10        Restaurant.nextId();
11        menu.tambahMenuMakanan("Tahu", 1_000, 0);
12        Restaurant.nextId();
13        menu.tambahMenuMakanan("Molen", 1_000, 20);
14
15        menu.tampilMenuMakanan();
16
17        // Pesan makanan
18        System.out.println();
19        menu.pesanMakanan("Molen", 5);
20
21        System.out.println();
22        menu.tampilMenuMakanan();
23    }
24 }

```

## 2. Screenshot Hasil Program

```

Bala-bala[20]    Rp. 1000.0
Gehu[20]        Rp. 1000.0
Molen[20]       Rp. 1000.0

Makanan tersedia, Jumlah Bayaran: 5000.0

Bala-bala[20]    Rp. 1000.0
Gehu[20]        Rp. 1000.0
Molen[15]       Rp. 1000.0

```

## 3. Penjelasan Program

Pada kasus ini telah diberikan kode nya. Namun kode yang diberikan belum sesuai dengan kaidah enkapsulasi. Oleh karena itu, kita diminta untuk bagaimana caranya agar kode tersebut memenuhi kaidah enkapsulasi sesuai dari pedoman buku yang diberikan.

Dalam menyelesaikan kasus ini, saya melakukan penambahan **private** pada setiap atribut di kelas restaurant, karena dengan begitu atribut tersebut tidak akan dapat diakses oleh semua kelas, sehingga untuk mengakses kelas tersebut harus menggunakan method tambahan yaitu **getter** dan **setter**.

Setelah melakukan proses perubahan kode sesuai dengan pedoman enkapsulasi, selanjutnya adalah bagaimana cara menambah method pemesanan. Penambahan method pemesanan ini saya lakukan dengan cara membuat algoritma, yang dimana logika dari algoritmanya yaitu saya membandingkan nama produk yang diinginkan dengan nama produk yang telah ada pada daftar menu. Saya juga menyadari bahwasannya dengan membandingkannya pada nama produk maka banyak kemungkinan yang terjadi, sehingga hasil program tidak sesuai dengan harapan. Salah satu contohnya adalah ketika nama produknya bala bala tapi di daftar menu nama produknya bala-bala, hal tersebutlah yang membuat program belum dikatakan sempurna untuk proses pemesanan.

Kembali lagi ke logika proses pemesanan. Setelah dari perbandingan, langkah selanjutnya yaitu melihat stok barang dan jumlah barang yang dibeli dengan stok barang yang ada itu apakah melebihi. Jika stok barang ada dan jumlah barang yang dibeli tidak melebihi dari stok barang, maka tinggal menghitung total harga yang harus dibayar.

#### 4. Permasalahan yang Dihadapi

Permasalahan yang dihadapi ketika mengerjakan kasus ini adalah masih bingung dalam menganalisis suatu program sehingga menjadi program yang mengandung konsep enkapsulasi yang sesuai dengan pedoman.

#### 5. Solusi Permasalahan yang Dihadapi

Harus banyak-banyak eksplorasi mengenai contoh program yang telah mengandung enkapsulasi yang baik dan benar.

#### 6. Nama Teman yang Membantu Memecahkan Permasalahan di Persoalan ini

Tidak ada

## SOAL 2

### 1. Screenshot Program

Produk.java ×

```
1 package Task_2;
2
3 public class Produk {
4     private int idProduk = 0;
5     private String [] nama_produk;
6     private double [] harga;
7     private int [] qty;
8
9     public Produk() {
10         nama_produk = new String[11];
11         harga = new double[11];
12         qty = new int[11];
13     }
14
15     public void setQty (int index, int jumlah) {
16         this.qty[index] -= jumlah;
17     }
18
19     public String getNama_produk(int index) {
20         return nama_produk[index];
21     }
22
23
24
25     public double getHarga(int index) {
26         return harga[index];
27     }
28
29
30
31     public void tambahMenuMakanan(String nama, double harga, int stok) {
32         idProduk += 1;
33         this.nama_produk[idProduk] = nama;
34         this.harga[idProduk] = harga;
35         this.qty[idProduk] = stok;
36     }
37
38     void tampilMenu() {
39         System.out.println("+++++");
40         System.out.println(" Data Makanan ");
41         System.out.println("+++++");
42         System.out.println("NoNama Makanan Harga JumlahProduk ");
43         System.out.println("-----");
44         for (int a=1; a<=idProduk; a++) {
45             System.out.println(a + " " + nama_produk[a] + " " + harga[a] + " " + qty[a]);
46         }
47         System.out.println("-----");
48     }
49 }
50
51 }
```

## Penjualan.java

```

1 package Task_2;
2 import java.util.Scanner;
3
4 public class Penjualan {
5     private String nama_produk;
6     private int quantity;
7     private double harga_total = 0;
8     private int [] idPesanan;
9     private int id2 = 0;
10
11     public Penjualan() {
12         idPesanan = new int[10];
13     }
14
15     public void pesanMakanan(Produk produk) {
16         Scanner sc = new Scanner(System.in);
17
18         System.out.println();
19         System.out.println("----Form Pemesanan Makanan----");
20
21         while(true) {
22             System.out.print("Apakah anda yakin ingin memesan ? (y/n) :");
23             String pesanan = sc.next();
24
25             if(pesanan.equalsIgnoreCase("n")) {
26                 break;
27             } else {
28                 id2 += 1;
29
30                 System.out.print("No. Makanan : ");
31
32                 idPesanan [id2] = sc.nextInt();
33
34                 nama_produk = produk.getNama_produk(idPesanan[id2]);
35                 System.out.println("Nama Makanan : " + nama_produk);
36
37                 System.out.println("Harga Makanan : " + produk.getHarga(idPesanan[id2]));
38
39                 System.out.print("Jumlah Pesan : ");
40                 quantity = sc.nextInt();
41                 produk.setQty(idPesanan[id2], quantity);
42
43                 harga_total = quantity * produk.getHarga(idPesanan[id2]);
44                 System.out.println("Total Harga : " + harga_total);
45
46             }
47             System.out.println();
48         }
49         sc.close();
50     }
51 }
52

```

```

1 package Task_2;
2
3 public class Utama {
4     public static void main(String[] args) {
5         Produk p = new Produk();
6         Penjualan pj = new Penjualan();
7
8         p.tambahMenuMakanan("Batagor", 5_000, 30);
9         p.tambahMenuMakanan("Roti Bakar", 12_000, 40);
10        p.tambahMenuMakanan("Indomie+Telor", 10_000, 20);
11        p.tambahMenuMakanan("Kwetiaw", 12_000, 50);
12        p.tambahMenuMakanan("Nasi Goreng", 12_000, 70);
13        p.tambahMenuMakanan("Air Mineral", 3_000, 30);
14        p.tambahMenuMakanan("Teh Manis", 4_000, 20);
15        p.tambahMenuMakanan("Jus Alpukat", 8_000, 50);
16        p.tambahMenuMakanan("Teh Botol", 5_000, 60);
17        p.tambahMenuMakanan("Kopi", 3_000, 40);
18
19        p.tampilMenu();
20
21        pj.pesanMakanan(p);
22
23        p.tampilMenu();
24    }
25
26 }

```



## 2. Screenshot Hasil Program

```
+++++
Data Makanan
+++++
NoNama Makanan  Harga  JumlahProduk
-----
1 Batagor  5000.0 30
2 Roti Bakar  12000.0 40
3 Indomie+Telor  10000.0 20
4 Kwetiaw  12000.0 50
5 Nasi Goreng  12000.0 70
6 Air Mineral  3000.0 30
7 Teh Manis  4000.0 20
8 Jus Alpukat  8000.0 50
9 Teh Botol  5000.0 60
10 Kopi  3000.0 40
-----

----Form Pemesanan Makanan----
Apakah anda yakin ingin memesan ? (y/n) :y
No. Makanan : 3
Nama Makanan : Indomie+Telor
Harga Makanan : 10000.0
Jumlah Pesan : 5
Total Harga : 50000.0
```

```
+++++
Data Makanan
+++++
NoNama Makanan  Harga  JumlahProduk
-----
1 Batagor  5000.0 30
2 Roti Bakar  12000.0 40
3 Indomie+Telor  10000.0 15
4 Kwetiaw  12000.0 50
5 Nasi Goreng  12000.0 70
6 Air Mineral  3000.0 30
7 Teh Manis  4000.0 20
8 Jus Alpukat  8000.0 50
9 Teh Botol  5000.0 60
10 Kopi  3000.0 40
-----
```

## 3. Penjelasan Program

Pada kasus kali ini, kita diminta untuk membuat program pesan makanan. Sebenarnya program ini hampir sama dengan kasus sebelumnya yang restaurant. Namun, perbedaan yang ada pada kasus kali ini ada pada inputan pesan makanan yang menggunakan **Scanner**. Hal-hal yang perlu diperhatikan dalam membuat program ini yaitu ada pada cara mengenkapsulasi program sesuai pedoman. Pada program ini saya menggunakan **private** untuk setiap atribut di class dan menggunakan metode **getter**

**setter** agar atribut yang private tersebut dapat diakses di luar class. Terdapat suatu hal yang saya lakukan di program ini adalah menggunakan dependency (uses-a) pada objek produk di class pemesanan yang ada di parameter method pesanMakanan. Hal itu dimaksud agar saya dapat mengakses nama produk, harga, dan stok yang telah dideklarasikan pada class produk.

4. Permasalahan yang Dihadapi

Permasalahan yang dihadapi ketika mengerjakan kasus ini yaitu ada pada penentuan logika prosesnya.

5. Solusi Permasalahan yang Dihadapi

Solusinya adalah dengan mencari kasus yang serupa pada internet dan memodifikasi kasus yang serupa tersebut, sehingga sesuai dengan apa yang ditugaskan.

6. Nama Teman yang Membantu Memecahkan Permasalahan di Persoalan ini

Tidak ada