

# **LAPORAN TUGAS PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

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



Disusun Oleh :

Benny Yoga Suhardi 211511035

**PROGRAM STUDI D3 TEKNIK INFORMATIKA  
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA  
POLITEKNIK NEGERI BANDUNG  
TAHUN 2022**

## KASUS 1

```
1 package Kasus1;
2
3 public class Makanan {
4     private String nama_makanan;
5     private double harga_makanan;
6     private int stok;
7
8     public String getNama_makanan() {
9         return nama_makanan;
10    }
11
12    public double getHarga_makanan() {
13        return harga_makanan;
14    }
15
16    public int getStok() {
17        return stok;
18    }
19
20    public Makanan(String namaMakanan, double hargaMakanan, int Stok) {
21        this.nama_makanan = namaMakanan;
22        this.harga_makanan = hargaMakanan;
23        this.stok = Stok;
24    }
25
26    public void TampilMakanan() {
27        System.out.print(this.nama_makanan + "["+stok+"]"+"\\tRp. "+harga_makanan);
28    }
29
30    public void kurangStok(int jumlah){
31        this.stok = this.stok - jumlah;
32    }
33 }
```

## Class Restorant

```
1 package Kasus1;
2
3 public class Restoran {
4     private Makanan[] food;
5     private static byte id=0;
6
7     public static byte getId() {
8         return id;
9     }
10
11    public static void setId(byte id) {
12        Restoran.id = id;
13    }
14
15    public Restoran() {
16        food = new Makanan[10];
17    }
18
19    public void tambahMenuMakanan(String nama, double harga, int stok) {
20        this.food[id] = new Makanan(nama,harga,stok);
21    }
22
23    public void tampilMenuMakanan() {
24        for(int i = 0; i <= id; i++) {
25            if(!isOutOfStock(i)) {
26                System.out.print(i+1 + " ");
27                this.food[i].TampilMakanan();
28                System.out.println("");
29            }
30        }
31    }
32 }
```

```

3 public void pemesanan(int idMakanan, int jumlah) {
4     for(int i = 0; i <= id; i++) {
5         if(i == idMakanan-1) {
6             this.food[i].kurangStok(jumlah);
7         }
8     }
9 }
10
11 public boolean isOutOfStock(int id) {
12     if(food[id].getStok() == 0) {
13         return true;
14     }
15     else {
16         return false;
17     }
18 }
19
20 public static void nextId() {
21     id++;
22     setId(id);
23 }
24 }
25

```

## Main Class

```

1 package Kasus1;
2
3 public class RestoranMain {
4
5     public static void main(String[] args) {
6         Restoran menu = new Restoran();
7         menu.tambahMenuMakanan("Bala-Bala", 1_000, 20);
8         Restoran.nextId();
9         menu.tambahMenuMakanan("Gehu", 1_000, 20);
10        Restoran.nextId();
11        menu.tambahMenuMakanan("Tahu", 1_000, 00);
12        Restoran.nextId();
13        menu.tambahMenuMakanan("Molen", 1_000, 20);
14
15        menu.tampilMenuMakanan();
16
17        System.out.println("");
18
19        menu.pemesanan(4,3);
20        menu.pemesanan(2,1);
21        menu.tampilMenuMakanan();
22    }
23 }
24 }
25

```

## Result

```

1 Bala-Bala[20] Rp. 1000.0
2 Gehu[20]      Rp. 1000.0
4 Molen[20]     Rp. 1000.0

1 Bala-Bala[20] Rp. 1000.0
2 Gehu[19]      Rp. 1000.0
4 Molen[17]     Rp. 1000.0

```

- **Pemenuhan Konsep OOP**

Merujuk kepada buku Core Java Volume 1 ada beberapa konsep dari OOP yang harus dipenuhi mungkin di bawah ini yang menurut saya adalah beberapa core dari konsep OOP dari buku tersebut diantaranya

1. Keep Data Private

```
public class Restaurant {  
    public String[] nama_makanan;  
    public double[] harga_makanan;  
    public int[] stok;  
    public static byte id=0;
```

Pada saat awal deklarasi menggunakan variable yang bersifat public sesuai dengan Konsep tersebut bahwa data tetap bersifat private maka beberapa variable tersebut diubah menjadi

```
private String nama_makanan;  
private double harga_makanan;  
private int stok;
```

2. Not all fields need individual field accessors and mutators.

Dengan penggunaanya Encapsulasi membungkusnya menjadi private maka untuk mengatur setiap variabelnya tidak bisa langsung harus dengan menggunakan accessors dan juga mutators dengan getter dan setter untuk setiap pengelolaannya

```
public String getNama_makanan() {  
    return nama_makanan;  
}  
  
public double getHarga_makanan() {  
    return harga_makanan;  
}  
  
public int getStok() {  
    return stok;  
}
```

3. Break up classes

Yang sebelumnya ada 2 class yaitu main class dan juga restaurant tetapi di dalam class restaurant itu dapat dibagi lagi kedalam class yang lebih spesifik yaitu class makanan dimana di dalamnya terdapat variable

```
private String nama_makanan;  
private double harga_makanan;  
private int stok;
```

- **Fitur Pemesanan**

```
public void pemesanan(int idMakanan, int jumlah) {  
    for(int i =0;i<=id;i++) {  
        if(i == idMakanan-1) {  
            this.food[i].kurangStok(jumlah);  
        }  
    }  
}
```

Di dalam prosedur ini terdapat 2 parameter yaitu id dan juga jumlah dimana pertama tama akan mengecek apakah id tersebut ada dalam list makanan kemudian jika id tersebut ada di dalam list makanan maka stok dari makanan tersebut akan dikurangi berdasarkan jumlah dari pilihan user dari stok makanannya dengan memanggil method milik makanan

```
public void kurangStok(int jumlah){  
    this.stok = this.stok - jumlah;  
}
```

## KASUS 2

### Class Penjualan

```
1 package Kasus2;
2
3 public class Penjualan {
4     private String[] nama_produk;
5     private int[] quantity;
6     private int[] harga_total;
7     private static int id = 0;
8     int x;
9
10    public Penjualan() {
11        this.nama_produk = new String[10];
12        this.quantity = new int[10];
13        this.harga_total = new int[10];
14    }
15
16    public void TambahPesanan(String namaProduk, int Qty, int hargaTotal) {
17        this.nama_produk[id] = namaProduk;
18        this.quantity[id] = Qty;
19        this.harga_total[id] += hargaTotal;
20        id++;
21    }
22
23    public void PesananSaya() {
24        double totalPesanan = 0;
25        for(int i = 0 ; i<id;i++) {
26            System.out.println(this.nama_produk[i] + " " + this.quantity[i]);
27            totalPesanan += this.harga_total[i];
28        }
29        System.out.println("Pesanan Anda : " + totalPesanan);
30    }
31 }
32
```

### Class Produk

```
1 package Kasus2;
2
3 public class Produk {
4     private String namaProduk;
5     private int harga;
6     private int stok;
7     private int id;
8     private static int nextid =0;
9
10    {
11        this.id = nextid;
12        nextid ++;
13    }
14
15    public String getNamaProduk() {
16        return namaProduk;
17    }
18    public void setNamaProduk(String namaProduk) {
19        this.namaProduk = namaProduk;
20    }
21    public int getHarga() {
22        return harga;
23    }
24    public void setHarga(int harga) {
25        this.harga = harga;
26    }
27    public int getStok() {
28        return stok;
29    }
30 }
```

```

27 public int getStok() {
28     return stok;
29 }
30 public void setStok(int stok) {
31     this.stok = stok;
32 }
33 public int getId() {
34     return id;
35 }
36
37 public Produk(String nama, int harga, int stok) {
38     this.namaProduk = nama;
39     this.stok = stok;
40     this.harga = harga;
41 }
42
43 public void print() {
44     System.out.println((this.id+1) + " " + this.namaProduk +
45         " = Rp." + this.harga + " (" + this.stok + ")");
46 }
47 }

```

## Class Main

```

2 import java.util.Scanner;
3
4
5 public class Utama {
6
7     public static Produk[] generateProduk() {
8         Produk[] listProduk = new Produk[10];
9         String[] namaProduk = {
10             "Combro", "Bala-bala", "Gehu", "Kacang", "Bacang", "Nasi Kuning",
11             "Nasi Uduk", "Kopikap", "Aqua", "Ale-ale"
12         };
13
14         int[] hargaProduk = {
15             1000, 1000, 1500, 1000, 1000, 6000, 6000, 1000, 3000, 1000
16         };
17
18         int[] stokProduk = {
19             3, 3, 15, 10, 10, 5, 10, 15, 10, 20
20         };
21
22         for(int i=0; i < 10; i++) {
23             listProduk[i] = new Produk(namaProduk[i], hargaProduk[i], stokProduk[i]);
24         }
25
26         return listProduk;
27     }
28 }

```

```

9 public static void main(String[] args) {
10     Produk[] listProduk = generateProduk();
11     Penjualan listPenjualan = new Penjualan();
12
13     System.out.println("Daftar Menu Makanan");
14     System.out.println("=====");
15     for(int i=0; i<10; i++) {
16         listProduk[i].print();
17     }
18     System.out.println("\n");
19     Scanner scan = new Scanner(System.in);
20     char pilihan = 'y';
21     byte jumlahPembelian = 1;
22
23     while((pilihan == 'y' || pilihan == 'Y') && (jumlahPembelian <=10)){
24         System.out.print("Masukkan Id Pilihan Makanan = ");
25         int id = scan.nextInt();
26         System.out.print("Masukkan Jumlah Pembelian = ");
27         int jumlah = scan.nextInt();
28         String nama = listProduk[id-1].getNamaProduk();
29         int harga = listProduk[id-1].getHarga();
30         listPenjualan.TambahPesanan(nama, jumlah, harga*jumlah);
31         System.out.print("\nApakah akan memesan kembali = ");
32         pilihan = scan.next().charAt(0);
33         jumlahPembelian++;
34     }
35     listPenjualan.PesananSaya();
36 }
37
38 }

```

## Hasil Program

```
Daftar Menu Makanan
=====
1 Combro = Rp.1000 (3)
2 Bala-bala = Rp.1000 (3)
3 Gehu = Rp.1500 (15)
4 Kacang = Rp.1000 (10)
5 Bacang = Rp.1000 (10)
6 Nasi Kuning = Rp.6000 (5)
7 Nasi Uduk = Rp.6000 (10)
8 Kopikap = Rp.1000 (15)
9 Aqua = Rp.3000 (10)
10 Ale-ale = Rp.1000 (20)

Masukkan Id Pilihan Makanan = 1
Masukkan Jumlah Pembelian = 3

Apakah akan memesan kembali = y
Masukkan Id Pilihan Makanan = 6
Masukkan Jumlah Pembelian = 1

Apakah akan memesan kembali = n
Combro 3
Nasi Kuning 1
Pesanan Anda : 9000.0
```

Link Github : [BennyYoga/PBO\\_Praktikum \(github.com\)](https://github.com/BennyYoga/PBO_Praktikum)