

16TKO3043 PEMROGRAMAN BERORIENTASI OBJEK

W5 – Studi Kasus 5

LAPORAN



Oleh:

M. Fatur Maulidan Azzahra 211511020

PROGRAM STUDI D-III

**TEKNIK INFORMATIKA JURUSAN TEKNIK INFORMATIKA DAN
KOMPUTER POLITEKNIK NEGERI BANDUNG**

2022

Kasus 1 -

- Tangkapan Layar Hasil Akhir Program

```
Main-Class-Kasus1

package restaurant;
public class RestaurantMain {
    public static void main(String[] args){
        Restaurant restaurant = new Restaurant();
        restaurant.tambahMenuMakanan("Gehu", 1000, 20);
        restaurant.tampilMenuMakanan();
        restaurant.mengurangiStokMakanan(0);
        restaurant.tampilMenuMakanan();
        restaurant.menambahkanStokMakanan(0, 100);
        restaurant.tampilMenuMakanan();
    }
}
```

```
Class-Kasus1

package restaurant;
public class Menu {
    private String nama_makanan;
    private double harga_makanan;
    private int stok;
    private static byte id = 0;

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

    public String getNama_makanan() {
        return nama_makanan;
    }

    public double getHarga_makanan() {
        return harga_makanan;
    }

    public int getStok() {
        return stok;
    }

    public static byte getId(){
        return id;
    }

    public static void nextId(){
        id += 1;
    }
}
```

```
Method-Class-Kasus1

package restaurant;
public class Restaurant {
    Menu[] menu = new Menu[10];

    public void tambahMenuMakanan(String nama_makanan, double harga_makanan, int stok){
        byte id = Menu.getId();
        menu[id] = new Menu(nama_makanan, harga_makanan, stok);
        Menu.nextId();
    }

    public void mengurangiStokMakanan(int pilih){
        menu[pilih] = new
Menu(menu[pilih].getNama_makanan(), menu[pilih].getHarga_makanan(), menu[pilih].getStok() - 1);
    }

    public void menambahkanStokMakanan(int pilih, int stok){
        menu[pilih] = new
Menu(menu[pilih].getNama_makanan(), menu[pilih].getHarga_makanan(), menu[pilih].getStok() + stok);
    }

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

    public void tampilMenuMakanan(){
        for(int i = 0; i <= Menu.getId() - 1; i++){
            if(!isOutOfStock(i)){
                System.out.println(menu[i].getNama_makanan() + "[" + menu[i].getStok() + "]" + "\tRp.
" + menu[i].getHarga_makanan());
            }
        }
    }
}
```

- **Tangkapan Layar Setiap Jawaban Soal yang Dipertanyakan**

```
run:
Gehu[20]          Rp. 1000.0
Gehu[19]          Rp. 1000.0
Gehu[119]         Rp. 1000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Permasalahan yang dihadapi**

- Amati desain setiap class, Apakah desain class tersebut sudah memenuhi konsep OOP yang benar? Jika tidak, coba anda perbaiki dengan mengacu pada Design Hint 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.

- **Solusi dari Permasalahan yang dihadapi**
 1. Membangun dependency kemudian encapsulation yang merujuk pada chapter 4.10 namun, di bagian ini saya masih mengalami buntu.
- **Nama teman yang membantu memecahkan permasalahan di persoalan ini**

(Referensi sumber)

(Tidak ada)

Kasus 2

- **Tangkapan Layar Hasil Akhir Program**

```

package restaurantmakan;
import java.util.Scanner;
public class RestaurantMakan {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int element = 20;
        Restaurant res = new Restaurant(element);
        int pilihan;
        boolean y = true;
        res.daftarMenuMakanan();

        while(y){
            System.out.println("=====");
            System.out.println("1. Menambahkan Daftar Makanan");
            System.out.println("2. Pesan Makanan");
            System.out.println("3. Menambah Stok Makanan");
            System.out.println("");
            System.out.print("Silahkan pilih pilihan diatas terlebih dahulu : ");
            pilihan = scan.nextInt();
            switch (pilihan){
                case 1:
                    res.inputMenambahMakanan();
                    break;
                case 2:
                    res.pesanMakanan();
                    break;
                case 3:
                    res.menambahStokMakanan();
                    break;
                default:
                    System.out.println("Tidak ada pilihan");
                    break;
            }
        }
    }
}

```

```
package restaurantmakan;

public class Produk {
    private String namaProduk;
    private double harga;
    private int qty;
    private static int id = 0;

    public String getNamaProduk() {
        return namaProduk;
    }

    public void setNamaProduk(String namaProduk) {
        this.namaProduk = namaProduk;
    }

    public double getHarga() {
        return harga;
    }

    public void setHarga(double harga) {
        this.harga = harga;
    }

    public int getQty() {
        return qty;
    }

    public void setQty(int qty) {
        this.qty = qty;
    }

    public static int getId(){
        return id;
    }

    public static void nextId(){
        id += 1;
    }
}
```



```
package restaurantmakan;

import java.util.Scanner;

public class Restaurant {
    Produk[] produk;
    Scanner scan = new Scanner(System.in);

    Restaurant(int idElement){
        produk = new Produk[idElement];
    }

    public void tambahMakanan(String namaMakanan, int harga, int stok){
        int id = Produk.getId();
        produk[id] = new Produk();

        produk[id].setNamaProduk(namaMakanan);
        produk[id].setHarga(harga);
        produk[id].setQty(stok);

        Produk.nextId();
    }

    public void inputMenambahMakanan(){
        String namaMakanan;
        int harga;
        int stok;
        System.out.print("Nama Makanan : ");
        namaMakanan = scan.next();
        System.out.print("Harga : ");
        harga = scan.nextInt();
        System.out.print("Stok : ");
        stok = scan.nextInt();

        tambahMakanan(namaMakanan,harga,stok);
    }

    public void daftarMenuMakanan(){
        tambahMakanan("Batagor", 5000, 2);
        tambahMakanan("Roti Bakar", 12000, 10);
        tambahMakanan("Indomie+telor", 10000, 10);
        tambahMakanan("Kwetiaw", 12000, 10);
        tambahMakanan("Nasi Goreng", 12000, 10);
        tambahMakanan("Air Mineral", 3000, 10);
        tambahMakanan("Teh Manis", 4000, 10);
        tambahMakanan("Jus Alpukat", 8000, 10);
        tambahMakanan("Teh Botol", 5000, 10);
        tambahMakanan("Kopi", 3000, 0);
        tambahMakanan("Susu", 3000, 10);
    }
}
```



```
public void pesanMakanan(){
    boolean done = true;
    boolean selesai = false;
    int totalBayar = 0;
    int hitung = 0;
    int pilihMakanan;
    int[] pesan = new int[3];
    printMakanan();
    while(done){
        System.out.print("Silahkan Pilih Menu : ");
        pilihMakanan = scan.nextInt();
        if(isStokEmpty(pilihMakanan - 1)){
            if(validasiPesanan(pilihMakanan - 1)){
                hitung += 1;
                pesan[hitung - 1] = pilihMakanan - 1;
                produk[pilihMakanan - 1].setQty(produk[pilihMakanan - 1].getQty() - 1);
                System.out.println("Stok saat ini : "+produk[pilihMakanan - 1].getQty());
                if(hitung == 3){
                    break;
                }
                done = cekLanjutPemesanan();
            } else {
                done = cekLanjutPemesanan();
            }
        } else {
            System.out.println("Produk tersebut habis");
        }
    }
    for(int i = 0; i < hitung; i++){
        pesananStruk(pesan[i]);
        totalBayar += produk[pesan[i]].getHarga();
    }

    while(!selesai){
        System.out.println("Total yang perlu anda bayar adalah = "+totalBayar);
        System.out.print("Masukkan Uang anda = ");
        int a = scan.nextInt();
        if(a >= totalBayar){
            selesai = true;
            int kembali = a - totalBayar;
            System.out.println("Kembalian anda adalah = "+ kembali);
        } else {
            System.out.println("Uang anda kurang");
        }
    }
}
```

```

public boolean validasiPesanan(int pilihMakanan){
    if(pilihMakanan < Produk.getId()){
        return true;
    } else {
        return false;
    }
}

public boolean cekLanjutPemesanan(){
    String check;
    System.out.print("Apakah masih ingin memesan? [y/n] : ");
    check = scan.next();
    return "y".equals(check) || "Y".equals(check);
}

public void pesananStruk(int id){
    System.out.println("Anda Memilih pesanan ini : ");
    System.out.println("======(id+1)=====");
    System.out.println("Nama_Produk : "+produk[id].getNamaProduk());
    System.out.println("Nama_Produk : "+produk[id].getHarga());
    System.out.println("");
}

public void menambahStokMakanan(){
    printMakanan();
    int id = 0;
    int stok = 0;
    System.out.print("Makanan/Minuman apa yang ingin ditambahkan : ");
    id = scan.nextInt();
    System.out.print("Berapa banyak stok : ");
    stok = scan.nextInt();
    produk[id-1].setQty(produk[id-1].getQty() + stok);
}

public boolean isStokEmpty(int id){
    if(produk[id].getQty() == 0){
        return false;
    } else {
        return true;
    }
}

public void printMakanan(){
    System.out.println("=====");
    System.out.println("Selamat Datang di Restaurant kami");
    System.out.println("=====");
    System.out.println("=====Menu=====");
    System.out.println("=====");
    for(int i = 0; i < Produk.getId(); i++){
        if(isStokEmpty(i)){
            System.out.println(i+1 + ". " + produk[i].getNamaProduk() + "          = Rp " +
            produk[i].getHarga() + " Stok : "+produk[i].getQty());
        } else {
            System.out.println(i+1 + " "+produk[i].getNamaProduk()+ " Stok makanan/minuman ini
            sedang habis");
        }
    }
}
}

```

- **Tangkapan Layar Setiap Jawaban Soal yang Dipertanyakan**

```
run:
```

```
=====
```

1. Menambahkan Daftar Makanan
2. Pesan Makanan
3. Menambah Stok Makanan

Silahkan pilih pilihan diatas terlebih dahulu : |

```
=====
```

```
Selamat Datang di Restaurant kami
```

```
=====
```

```
=====Menu=====
```

```
=====
```

1. Batagor = Rp 5000.0 Stok : 2
2. Roti Bakar = Rp 12000.0 Stok : 10
3. Indomie+telor = Rp 10000.0 Stok : 10
4. Kwetiaw = Rp 12000.0 Stok : 10
5. Nasi Goreng = Rp 12000.0 Stok : 10
6. Air Mineral = Rp 3000.0 Stok : 10
7. Teh Manis = Rp 4000.0 Stok : 10
8. Jus Alpukat = Rp 8000.0 Stok : 10
9. Teh Botol = Rp 5000.0 Stok : 10
- 10 Kopi Stok makanan/minuman ini sedang habis
11. Susu = Rp 3000.0 Stok : 10

Ketika memilih no 1

Silahkan pilih pilihan diatas terlebih dahulu : 1

Nama Makanan : Telor

Harga : 3000

Stok : 10

Ketika memilih no 2

```
=====
Selamat Datang di Restaurant kami
=====
=====Menu=====
=====
1. Batagor          = Rp 5000.0 Stok : 2
2. Roti Bakar       = Rp 12000.0 Stok : 10
3. Indomie+telor    = Rp 10000.0 Stok : 10
4. Kwetiaw          = Rp 12000.0 Stok : 10
5. Nasi Goreng      = Rp 12000.0 Stok : 10
6. Air Mineral      = Rp 3000.0 Stok : 10
7. Teh Manis        = Rp 4000.0 Stok : 10
8. Jus Alpukat      = Rp 8000.0 Stok : 10
9. Teh Botol        = Rp 5000.0 Stok : 10
10 Kopi Stok makanan/minuman ini sedang habis
11. Susu            = Rp 3000.0 Stok : 10
12. Telor           = Rp 3000.0 Stok : 10
Silahkan Pilih Menu : |
```

```
Silahkan Pilih Menu : 1
Stok saat ini : 1
Apakah masih ingin memesan? [y/n] : y
Silahkan Pilih Menu : 1
Stok saat ini : 0
Apakah masih ingin memesan? [y/n] : y
Silahkan Pilih Menu : 1
Produk tersebut habis
Silahkan Pilih Menu : |
```

Anda Memilih pesanan ini :

=====1=====

Nama_Produk : Batagor

Nama_Produk : 5000.0

Anda Memilih pesanan ini :

=====1=====

Nama_Produk : Batagor

Nama_Produk : 5000.0

Anda Memilih pesanan ini :

=====2=====

Nama_Produk : Roti Bakar

Nama_Produk : 12000.0

Total yang perlu anda bayar adalah = 22000

Masukkan Uang anda = |

Total yang perlu anda bayar adalah = 22000

Masukkan Uang anda = 20000

Uang anda kurang

Total yang perlu anda bayar adalah = 22000

Masukkan Uang anda = 22000

Kembalian anda adalah = 0

=====

1. Menambahkan Daftar Makanan

2. Pesan Makanan

3. Menambah Stok Makanan

Silahkan pilih pilihan diatas terlebih dahulu :

Ketika memilih pilihan 3

```
=====
Selamat Datang di Restaurant kami
=====
=====Menu=====
=====
1 Batagor Stok makanan/minuman ini sedang habis
2. Roti Bakar          = Rp 12000.0 Stok : 9
3. Indomie+telor       = Rp 10000.0 Stok : 10
4. Kwetiaw             = Rp 12000.0 Stok : 10
5. Nasi Goreng         = Rp 12000.0 Stok : 10
6. Air Mineral         = Rp 3000.0 Stok : 10
7. Teh Manis           = Rp 4000.0 Stok : 10
8. Jus Alpukat         = Rp 8000.0 Stok : 10
9. Teh Botol           = Rp 5000.0 Stok : 10
10 Kopi Stok makanan/minuman ini sedang habis
11. Susu               = Rp 3000.0 Stok : 10
12. Telor              = Rp 3000.0 Stok : 10
Makanan/Minuman apa yang ingin ditambahkan : |
Makanan/Minuman apa yang ingin ditambahkan : 1
Berapa banyak stok : 20
```

Tampil Menu

```
=====
Selamat Datang di Restaurant kami
=====
=====Menu=====
=====
1. Batagor          = Rp 5000.0 Stok : 20
2. Roti Bakar       = Rp 12000.0 Stok : 9
3. Indomie+telor    = Rp 10000.0 Stok : 10
4. Kwetiaw          = Rp 12000.0 Stok : 10
5. Nasi Goreng      = Rp 12000.0 Stok : 10
6. Air Mineral      = Rp 3000.0 Stok : 10
7. Teh Manis        = Rp 4000.0 Stok : 10
8. Jus Alpukat      = Rp 8000.0 Stok : 10
9. Teh Botol        = Rp 5000.0 Stok : 10
10 Kopi Stok makanan/minuman ini sedang habis
11. Susu            = Rp 3000.0 Stok : 10
12. Telor           = Rp 3000.0 Stok : 10
Silahkan Pilih Menu : |
```

- **Permasalahan Yang Dihadapi**

Buat program pemesanan makanan sederhana dengan mengikuti ketentuan sebagai berikut :

- Input menggunakan scanner
- Buat menu ditampilkan dilayar untuk menampilkan item makanan beserta harganya (minimal 10 produk)
- Terdapat kelas produk yang berisi variable nama_produk, harga, qty,
- Terdapat kelas Penjualan yang berisi nama_produk, quantity serta harga_total
- Tampilkan menu yang dipesan beserta total bayar nya,
- Konfirmasi untuk memesan item makanan lain

- **Solusi dari Permasalahan yang dihadapi**

Mengikuti Instruksi tersebut

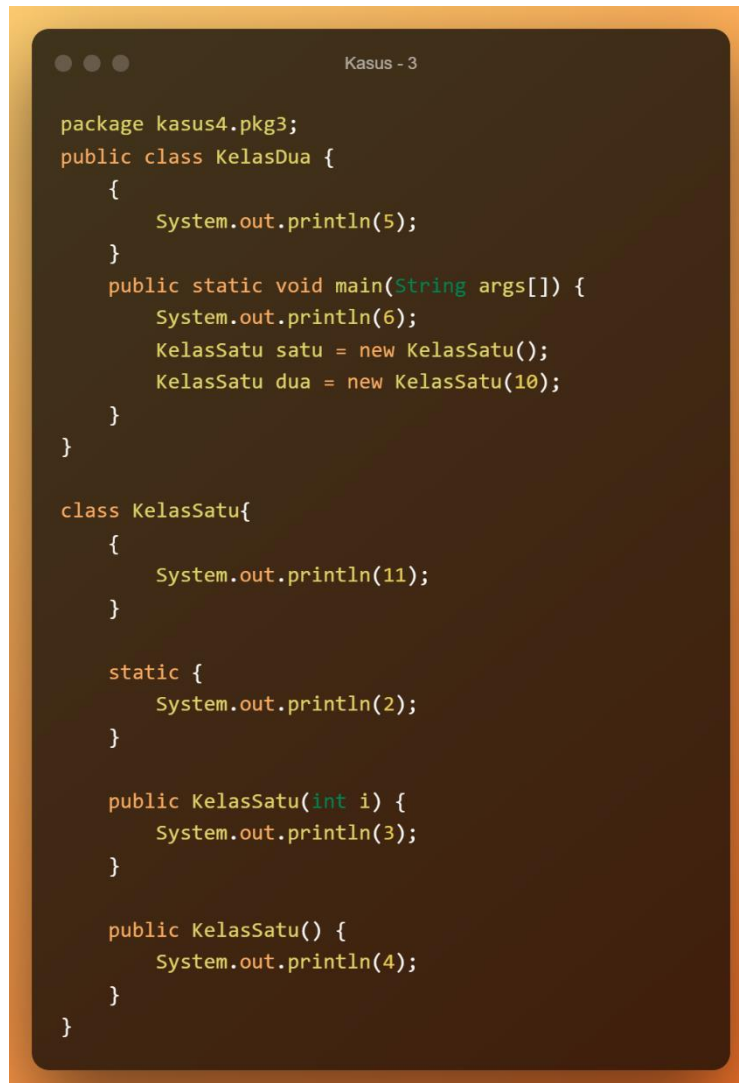
- **Nama teman yang membantu memecahkan permasalahan di persoalan ini**

(Referensi sumber)

Stack Overflow

Soal 3 – Berhitung

- Tangkapan Layar Hasil Akhir Program



```
Kasus - 3

package kasus4.pkg3;
public class KelasDua {
    {
        System.out.println(5);
    }
    public static void main(String args[]) {
        System.out.println(6);
        KelasSatu satu = new KelasSatu();
        KelasSatu dua = new KelasSatu(10);
    }
}

class KelasSatu{
    {
        System.out.println(11);
    }

    static {
        System.out.println(2);
    }

    public KelasSatu(int i) {
        System.out.println(3);
    }

    public KelasSatu() {
        System.out.println(4);
    }
}
```

- Tangkapan Layar Setiap Jawaban Soal yang Dipertanyakan

run :

6

2

11

4

11

3

- **Permasalahan Yang Dihadapi**

Cari tahu, bagaimana urutan konstruksi objek tersebut, dan mengapa urutannya seperti itu?

- **Solusi dari Permasalahan yang dihadapi**

Yang pertama kali di execute adalah initialize block yang terdiri dari 2 jenis yaitu static, dan default yang tidak menggunakan access modifier apa – apa. Kemudian masuk ke Constructor dan selesai Static → Initialize Block → Constructor

- **Nama teman yang membantu memecahkan permasalahan di persoalan ini**

(Referensi sumber)

Aldrin Rayhan Putra

