

# **LAPORAN TUGAS MINGGU 4**

## **PRAKTIKUM OBJECT, CLASS, & ENCAPSULATION**

*Laporan ini disusun untuk memenuhi tugas mata kuliah Pemrograman Berorientasi Objek*



Disusun oleh :  
Yumi Febriana                      211511063

**PROGRAM DIPLOMA III TEKNIK INFORMATIKA**  
**POLITEKNIK NEGERI BANDUNG**  
**BANDUNG**  
**2022**

## DAFTAR ISI

### Daftar isi

A.	KASUS 1.....	3
	SOURCE CODE SOAL .....	3
	JAWABAN .....	4
B.	KASUS 2.....	6
	SOURCE CODE SOAL .....	6
	JAWABAN .....	6
C.	KASUS 3.....	7
	SOURCE CODE SOAL .....	7
	JAWABAN .....	8

LINK GITHUB :

<https://github.com/yumifree/PBO-P-W4.git>

## A. KASUS 1

1. Salin ulang baris kode dan lakukan eksekusi terhadap 2 class tersebut, dimana: • Class Barang berfungsi untuk mendefinisikan struktur data yang diperlukan oleh Objek Barang. • Class Inventori berfungsi untuk mendefinisikan pembuatan objek-objek barang dan menampilkan objek barang yang telah dibuat dan pengadaan barang baru untuk menambah stok barang. Class inventori juga adalah Main Classnya.
2. Pada Class Inventori, method pengadaan berfungsi untuk melakukan pengadaan barang dan penambahan stok. Dengan struktur data yang ada, program sudah mampu mengakomodir fungsi tersebut. Namun kendalanya, data stok masih bisa dimanipulasi dengan proses aritmatika selain penambahan (seperti kali, bagi, atau kurang).
3. Carilah solusi, agar variable “stok” dibungkus/ dilindungi sehingga tidak bisa dilakukan operasi aritmatika selain hanya tambah saja.
4. Untuk memecahkan kasus tersebut Bacalah Chapter 4.2.3.

### SOURCE CODE SOAL

#### **Barang.java**

```
public class Barang {  
    String kode_barang;  
    String nama_barang;  
    int stok;  
  
    public Barang(String kode, String nama, int stk) {  
        kode_barang = kode;  
        nama_barang = nama;  
        stok = stk;  
    }  
}
```

#### **Inventori.java**

```
public class Inventori {  
    Barang[] barangs;  
  
    void initBarang() {  
        barangs = new Barang[2];  
        barangs[0] = new Barang("001", "Baju", 10);  
        barangs[1] = new Barang("002", "Celana", 20);  
    }  
  
    void showBarang() {  
        System.out.println(barangs[0].nama_barang + "(" + barangs[0].stok + ")");  
        System.out.println(barangs[1].nama_barang + "(" + barangs[1].stok + ")");  
    }  
}
```

```

void pengadaan() {
    initBarang();

    barangs[0].stok += 20;
    barangs[0].stok -= 30; //Bisa juga dikurangi dong?
    barangs[0].stok *= 30; //dikali juga bisa dong??
    showBarang();
}

public static void main(String[] args) {
    Inventori beli = new Inventori();
    beli.pengadaan();
}
}

```

## JAWABAN

### a. Source Code

```

Barang.java
package com.mycompany.barang;

/**
 * @author Yumi Febriana
 */
public class Barang {
    String kode_barang;
    String nama_barang;
    private int stok;

    public int getStok(){
        return stok;
    }

    public void setStok(int stok){
        this.stok = stok;
    }

    public Barang(String kode, String nama, int stk){
        kode_barang = kode;
        nama_barang = nama;
        stok = stk;
    }

    public void addStok(int stok){
        this.stok+=stok;
    }
}

```

```

    public static void main(String[] args){
        Inventory beli = new Inventory();
        beli.pengadaan();
    }
}

```

#### Inventory.java

```

package com.mycompany.barang;

/**
 * @author Yumi Febriana
 */
public class Inventory {
    Barang[] barangs;

    void initBarang(){
        barangs = new Barang [2];
        barangs [0] = new Barang ("001", "Baju",10);
        barangs [1] = new Barang("002", "Celana",20);
    }

    void showBarang(){
        System.out.println(barangs[0].nama_barang + "(" + barangs[0].getStok() + ")");
        System.out.println(barangs[1].nama_barang + "(" + barangs[0].getStok() + ")");
    }
    void pengadaan(){
        initBarang();

        barangs[0].addStok(20);
        showBarang();
    }
}

```

#### b. Result

```

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Barang ---
Baju(30)
Celana(30)

```

-----  
**BUILD SUCCESS**  
 -----

#### c. Problem

Saya terkadang merasa bingung dan terbalik membedakan get dan set

#### d. Solution

Lebih belajar memahami dan memperhatikan Ketika dosen menjelaskan

#### e. Teman yang membantu

Teman yang presentasi pada jam pelajaran dimulai

## B. KASUS 2

Modifikasi code pada class item agar output yang dihasilkan adalah “ipin” , catatan modifikasi hanya diizinkan 1 langkah perubahan

### SOURCE CODE SOAL

```
public class UpinIpin {
    public static void main(String[] args) {
        Item name = new Item("upin");
    }
}

public class Item {
    private String name;
    private Item() {
        name = "Ipin";
    }
    public Item(String name) {
        System.out.println(this.name);
    }
}
```

### JAWABAN

- a. Source code

#### UpinIpin.java

```
package com.mycompany.upinipin;
/**
 * @author Yumi Febriana
 */
public class UpinIpin {
    public static void main(String[] args) {
        Item name = new Item("upin");
    }
}
```

#### Item.java

```
package com.mycompany.upinipin;
/**
 * @author Yumi Febriana
 */
public class Item {
    private String name;
    private Item() {
        name = "Ipin";
    }
}
```

```
}  
  
public Item(String name) {  
    this();  
    System.out.println(this.name);  
}  
}
```

b. Result

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ UpinIpin ---  
Ipin
```

```
-----  
BUILD SUCCESS  
-----
```

c. Problem

Tidak ada

d. Solution

Tidak ada

e. Teman yang membantu

Teman yang presentasi pada saat jam pelajaran dimulai

### C. KASUS 3

1. Salin ulang baris kode dan lakukan eksekusi terhadap 2 Class Tersebut.
2. Lakukan Pengamatan terhadap outputnya.
3. Cari tahu, bagaimana urutan konstruksi objek tersebut, dan mengapa urutannya seperti itu?
4. Untuk memecahkan kasus tersebut Bacalah Chapter 4.3.4

SOURCE CODE SOAL

```

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);
    }
}

```

```

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);
    }
}

```

## KASUS 3

### JAWABAN

#### a. Source Code

##### **KelasSatu.java**

```
package com.mycompany.kelassatu;
```

```

/**
 * @author Yumi Febriana
 */
public 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);
    }
}

```

##### **KelasDua.java**

```
package com.mycompany.kelassatu;
```

```

/**
 * @author Yumi Febriana
 */

```



```

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);
    }
}

```

b. Result

```

--- exec-maven-plugin:3.0.0:exec (default-cli) @ KelasDua ---
6
2
11
4
11
3
-----
BUILD SUCCESS
-----

```

c. Hasil Pengamatan

- Output yang pertama kali tampil adalah angka 6. Karena merupakan angka yang pertama kali dipanggil pada modul main.
- Output kedua yang tampil adalah angka 2. Karena membuat objek baru dari class KelasSatu dengan nama variable 'satu' dan angka 2 berada pada class KelasSatu juga merupakan static, dimana angka 2 hanya akan muncul atau tampil sekali.
- Output ketiga yang tampil adalah angka 11. Karena objek dibuat dari class KelasSatu dengan nama variable 'satu'.
- Output keempat yang tampil adalah angka 4 karena objek dibuat dari class KelasSatu dengan nama variable 'satu' dan tanpa parameter sehingga saat objek dibuat akan memanggil constructor tanpa parameter dan di constructor tersebut berisi perintah untuk menampilkan angka 4.
- Output yang tampil kelima adalah angka 11, karena membuat objek baru dari class KelasSatu dengan nama variable 'dua'.
- Output yang tampil keenam adalah angka 3 karena objek dibuat dari class KelasSatu dengan nama variable 'dua' dengan parameter sehingga saat objek dibuat akan memanggil constructor dengan parameter dan di constructor yang menggunakan parameter berisi perintah untuk menampilkan angka 3.

d. Problem

Saat menggunakan netbeans, saya membuat main class yang terdapat pada KelasSatu sedangkan pada soal, public static void main (String[]args) ada pada KelasDua sehingga ketika di run tidak berhasil

e. Solution

Saya mengubah main class menjadi KelasDua

- f. Teman yang membantu  
Tidak ada