

16TKO3043 PEMROGRAMAN BERORIENTASI OBJEK

W4 – Studi Kasus 4

LAPORAN



Oleh:

M. Fatur Maulidan Azzahra 211511020

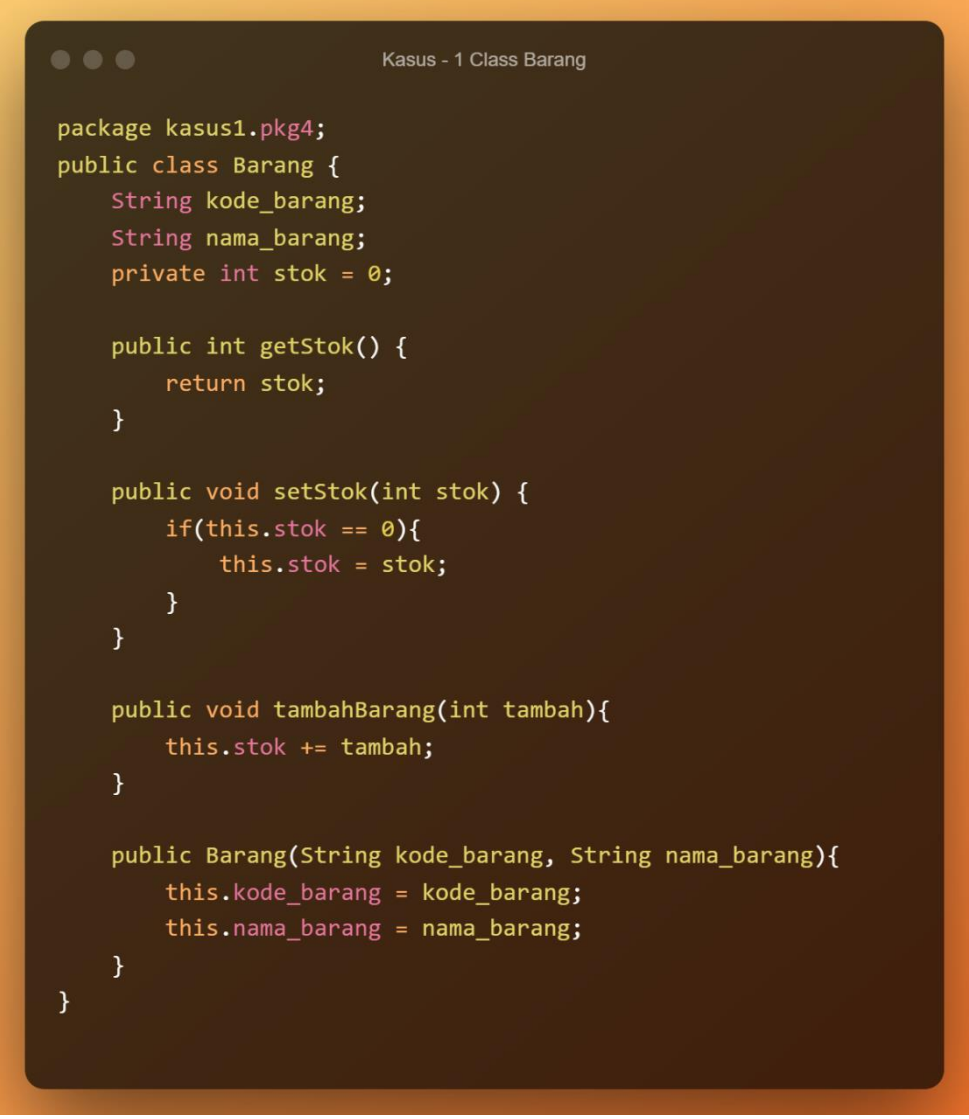
PROGRAM STUDI D-III

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

2022

Kasus 1 -

- Tangkapan Layar Hasil Akhir Program

A screenshot of a Java IDE window titled "Kasus - 1 Class Barang". The window has a dark background with orange borders. The code is written in Java and defines a class named "Barang" within the package "kasus1.pkg4". The class has two private String attributes: "kode_barang" and "nama_barang", and one private int attribute: "stok" initialized to 0. It includes three public methods: "getStok()" which returns the "stok" value, "setStok(int stok)" which sets the "stok" value only if it is currently 0, and "tambahBarang(int tambah)" which increments the "stok" by the given value. A constructor "Barang(String kode_barang, String nama_barang)" is also present, which initializes the "kode_barang" and "nama_barang" attributes.

```
package kasus1.pkg4;
public class Barang {
    String kode_barang;
    String nama_barang;
    private int stok = 0;

    public int getStok() {
        return stok;
    }

    public void setStok(int stok) {
        if(this.stok == 0){
            this.stok = stok;
        }
    }

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

    public Barang(String kode_barang, String nama_barang){
        this.kode_barang = kode_barang;
        this.nama_barang = nama_barang;
    }
}
```

Kasus - 1 Class Inventori

```
package kasus1.pkg4;
public class Inventori {
    Barang[] barangs;

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

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

    void pengadaanBarang(){
        initBarang();
        // barangs[0].setStok(barangs[0].getStok() + 20);
        // barangs[0].setStok(barangs[0].getStok() - 20);
        barangs[0].setStok(barangs[0].getStok() * 20);
        // barangs[0].tambahBarang(12);
        showBarang();
    }
}
```

Kasus - 1 Class Main

```
package kasus1.pkg4;
public class Kasus14 {
    public static void main(String[] args) {
        Inventori invent = new Inventori();
        invent.pengadaanBarang();
    }
}
```

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

run:

Baju (10)

Celana (20)

- **Permasalahan yang dihadapi**

Carilah solusi, agar variable “stok” dibungkus/ dilindungi sehingga tidak bisa dilakukan operasi aritmatika selain hanya tambah saja.

- **Solusi dari Permasalahan yang dihadapi**

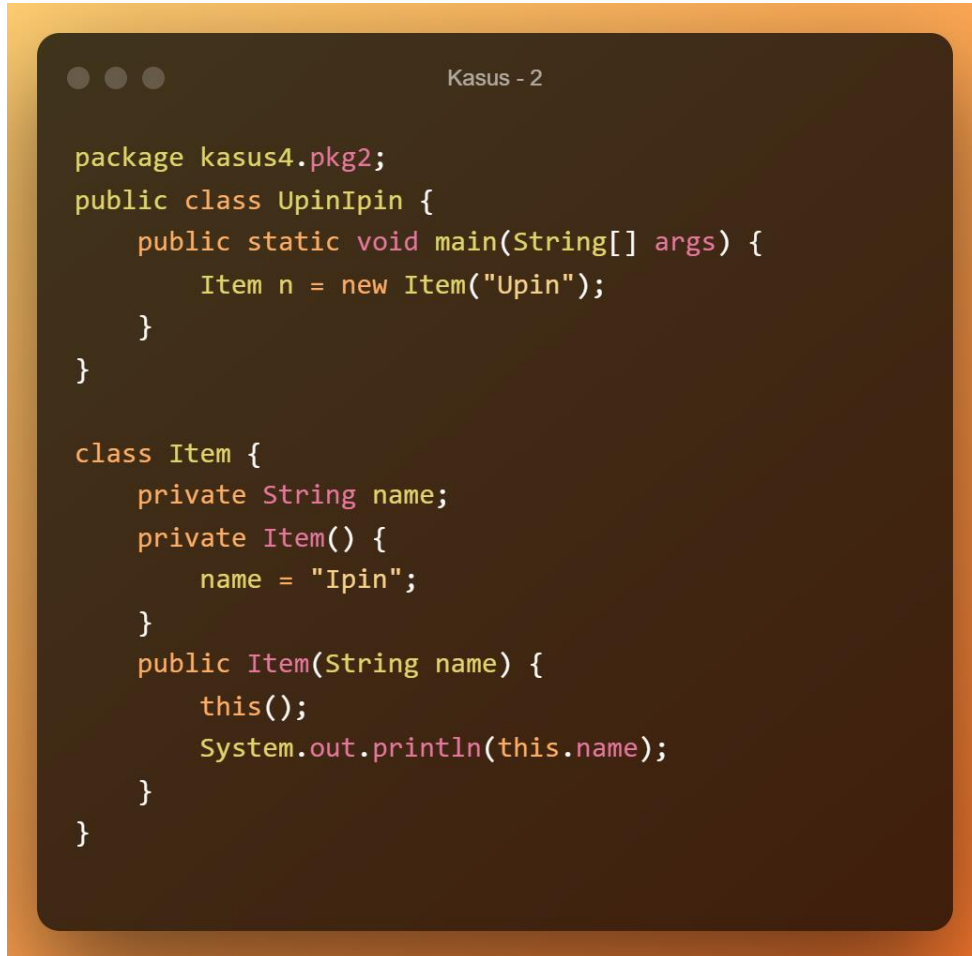
1. Membuat terlebih dahulu variable stok menjadi private supaya tidak dapat diganti sebarang mungkin oleh user – user yang lain
2. Membuat encapsulation dari variable stok supaya user perlu melewati set untuk merubah data.
3. Saya membuat value dari stok tersebut menjadi static supaya user tidak semena – mena mengganti value dari stok tersebut
4. Membuat method tambahBarang supaya user melewati method ini jika ingin menambahkan barang.

- **Nama teman yang membantu memecahkan permasalahan di persoalan ini (Referensi sumber)**

(Tidak ada)

Soal 2 – Input & Output (2)

- Tangkapan Layar Hasil Akhir Program



```
package kasus4.pkg2;
public class UpinIpin {
    public static void main(String[] args) {
        Item n = new Item("Upin");
    }
}

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

- Tangkapan Layar Setiap Jawaban Soal yang Dipertanyakan

run:
Ipin

- **Permasalahan Yang Dihadapi**

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

- **Solusi dari Permasalahan yang dihadapi**

Menambahkan this() pada constructor Item yang memiliki parameter, sehingga this() tersebut memanggil constructor dirinya sendiri yang tidak memiliki parameter.

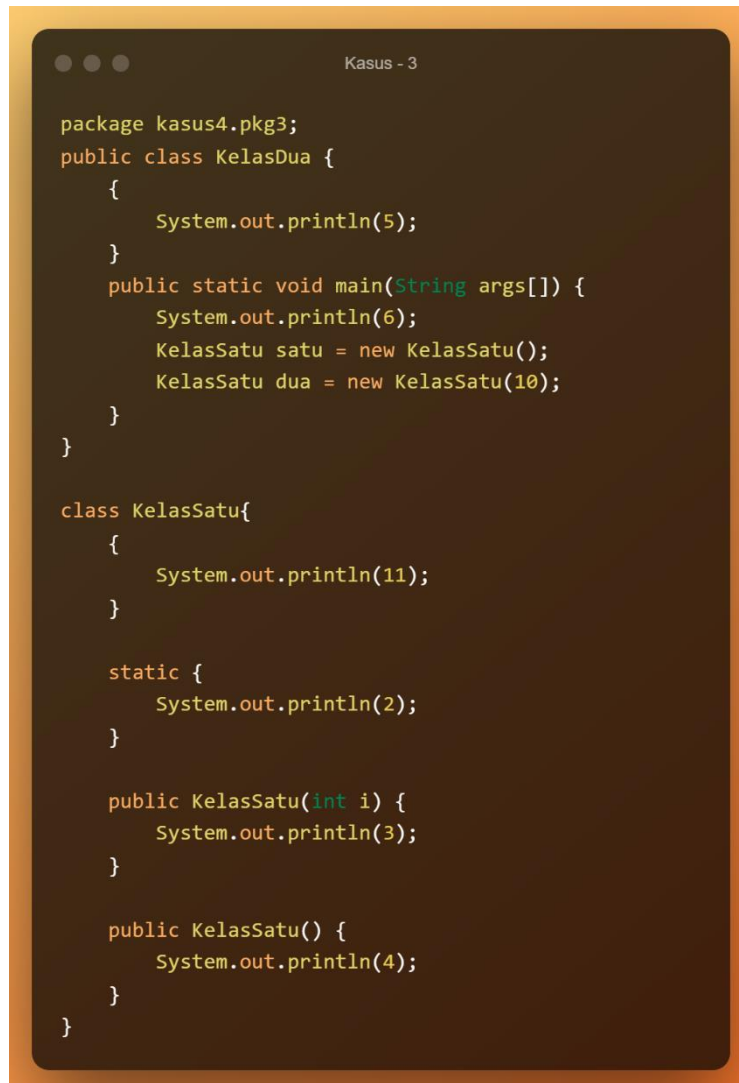
- **Nama teman yang membantu memecahkan permasalahan di persoalan ini**

(Referensi sumber)

M. Ardi Nur Insan

Soal 3 – Berhitung

- Tangkapan Layar Hasil Akhir Program



```
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

