W4 OBJECT CLASS & ENCAPSULATION

LAPORAN

Diajukan untuk memenuhi Tugas Mata Kuliah Pemrograman Berorientasi Objek

Disusun Oleh
MUHAMMAD HAFIZH AULIANSYAH
211511047

PROGRAM DIPLOMA III TEKNIK INFORMATIKA
POLITEKNIK NEGERI BANDUNG
BANDUNG
2022

PERSOALAN

Link Repository : https://github.com/HafizhAuliansyah/211511047_M-Hafizh-A_Praktikum4.git

1.1. Kasus 1

Hasil Akhir :
Solusi agar variable “stok” dibungkus/ dilindungi sehingga tidak bisa dilakukan operasi aritmatika selain hanya tambah saja adalah mengubah stok menjadi private dan membuat funciton tambahStok() pada Barang.java
Berikut hasilnya :

Inventori.java

```
package kasus1;
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].getStok() + ")");
        System.out.println(barangs[1].nama_barang + "(" + barangs[1].getStok() + ")");
    }
    void pengadaan() {
        initBarang();
        barangs[0].tambahStok(20);
        //barangs[0].stok -= 30; // Bisa dikurangi dong ?
        //barangs[0].stok *= 30; // Dikali juga bisa dong
        showBarang();
    }

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

Barang.java

```java
package kasus1;

public class Barang {
    String kode_barang;
    String nama_barang;
    private int stok;

    public Barang(String kode, String nama, int stk){
        kode_barang = kode;
        nama_barang = nama;
        stok = stk;
    }
    public void tambahStok(int tambahan){
        stok += tambahan;
    }
    public void setStok(int stok){
        this.stok = stok;
    }
    public int getStok(){
        return stok;
    }
}
```

Output Program :

```
Output - Kasus1 (run)
run:
Baju(30)
Celana(20)
BUILD SUCCESSFUL (total time: 0 seconds)
```

Permasalahan : -

Solusi : -

1.2. Kasus 2

Hasil Akhir :

Menambahkan keyword this() pada constructor ke-2 di class Item

Item.java

```java
package kasus2;
public class Item {
    private String name;
    private Item() {
        name = "Ipin";
    }
    public Item(String name) {
        this();
        System.out.println(this.name);
    }
}
```

UpinIpin.java

```java
package kasus2;
public class UpinIpin {
    public static void main(String[] args) {
        Item name = new Item("upin");
    }
}
```

Output Program :

```
Output - Kasus2 (run)
run:
Ipin
BUILD SUCCESSFUL (total time: 0 seconds)
```

Permasalahan : Tidak tahu cara memanggil constructor dalam constructor di suatu class java

Solusi : Melakukan browsing dan menemukan caranya yaitu memakai keyword this();

1.3. Kasus 3

Hasil Akhir :

KelasDua.java

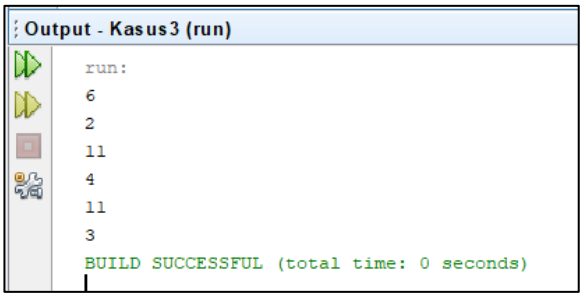
```java
public class KelasDua {
    {
        System.out.println(5);
    }
    public static void main(String[] args)
    {
        System.out.println(6);
        KelasSatu satu = new KelasSatu();
        KelasSatu dua = new KelasSatu(10);
    }
}
```

KelasSatu.java

```java
package kasus3;

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);
    }
}
```

Output Program :



Penjelasan Ourput :

Mengapa output seperti itu ?

- **System.out.println(5) tidak dipanggil karena merupakan instance field yang hanya akan dipanggil ketika pembuatan objek KelasDua**
- **Output dilanjutkan dimulai dari System.out.println(6) karena merupakan baris pertama dari fungsi main() di KelasDua**
- **Ouput dilanjutkan ke System.out.println(2) yang merupakan static instance field yang akan pertama dipanggil ketika pembuatan objek dari KelasSatu dan hanya sekali saja**
- **Output dilanjtukan ke System.out.println(11) yang merupakan instance field yang akan dipanggil setiap pembuatan objek dari KelasSatu**
- **Output dilanjutkan ke System.out.println(4) karena pemanggilan Constructor tanpa parameter saat deklarasi objek satu**
- **Ouput dilanjutkan ke System.out.println(11) karena terjadi pembuatan objek dari KelasSatu kembali yaitu dua**
- **System.out.println(2) tidak dipanggil kembali karena merupakan static**
- **Output terakhir adalah System.out.println(3) karena pemanggilan constructor berparameter dari kelas KelasSatu**

Permasalahan : -

Solusi : -