



Jurusan Teknik Komputer dan Informatika

Politeknik Negeri Bandung

Studi Kasus 4

Pertemuan ke-4

Terdapat 2 Class. Class Barang dan Inventori

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 + ")");  
    }  
}
```

Lanjutan Inventori.java

```
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();  
}  
}
```

Instruksi 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.

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

KASUS2

```
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);  
    }  
}
```

Baca sub Chapter 4.3.4 untuk menyelesaikan kasus ini.

KASUS 3

```
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);
    }
}
```

Instruksi 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