W4 - Instruksi Praktikum PBO

Class Object Encapsulation

Kerjakan 2 soal dibawah ini dengan mengikuti ketentuan sebagai berikut:

1. Isi sheet monitoring berdasarkan ketentuan yang ada di sheet tersebut.
2. Source code setiap pengerjaan soal, simpan di Github, lampirkan komentar dari hasil pengerjaan tersebut.
3. Buat laporan hasil pengerjaan berbentuk dokumen, upload laporan di folder Hasil Praktikum di folder hasil praktikum, laporan harus mencakup:
    1. Cover.
    2. Persoalan yang telah dikerjakan. Setiap persoalan, harus menjawab beberapa deskripsi berikut ini:
        1. Screenshoot hasil akhir program.
        2. Screenshoot setiap jawaban soal yang dipertanyakan.
        3. Permasalahan yang dihadapi.
        4. Solusi dari permasalahan yang dihadapi.
        5. Nama teman yang membantu memecahkan permasalahan di persoalan ini.

Soal 1

Kelas Product menggunakan Enkapsulasi

```
// Kelas Product
class Product {
    // Atribut private untuk menyembunyikan informasi
    private String productName;
    private double price;
    private int stock;

    // Constructor
    public Product(String productName, double price, int stock)
{
        this.productName = productName;
        this.price = price;
        this.stock = stock;
    }

    // Getter dan Setter untuk mengontrol akses ke atribut
private
    public String getProductName() {
        return productName;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        if (price > 0) {
            this.price = price;
        } else {
            System.out.println("Harga tidak valid!");
        }
    }

    public int getStock() {
        return stock;
    }

    public void addStock(int amount) {
        if (amount > 0) {
            this.stock += amount;
        } else {
```

```
            System.out.println("Jumlah stok tidak valid!");
        }
    }

    public void sellProduct(int quantity) {
        if (quantity > 0 && quantity <= stock) {
            stock -= quantity;
            System.out.println(quantity + " " + productName + "
terjual.");
        } else {
            System.out.println("Jumlah stok tidak cukup untuk
penjualan.");
        }
    }
}
```

Kelas Sales yang Mengelola Penjualan

```
// Kelas Sales untuk mengelola transaksi penjualan
class Sales {
    private Product product;

    // Constructor
    public Sales(Product product) {
        this.product = product;
    }

    // Metode untuk melakukan penjualan
    public void sellProduct(int quantity) {
        System.out.println("Memproses penjualan...");
        product.sellProduct(quantity);
        System.out.println("Stok setelah penjualan: " +
product.getStock());
    }

    // Metode untuk menambah stok produk
    public void restockProduct(int quantity) {
        System.out.println("Menambah stok...");
        product.addStock(quantity);
        System.out.println("Stok setelah penambahan: " +
product.getStock());
    }

    // Metode untuk memperbarui harga produk
    public void updateProductPrice(double newPrice) {
```

```
        System.out.println("Memperbarui harga produk...");
        product.setPrice(newPrice);
        System.out.println("Harga baru: " + product.getPrice());
    }
}
```

Program Utama untuk Menguji Enkapsulasi

```
public class Main {
    public static void main(String[] args) {
        ….
    }
}
```

Lakukan pengujian Enkapsulasi dengan melakukan beberapa hal berikut :

1. Buatlah object product dengan quantity produk 10
2. Buat package com.polban.jtk.sales dan simpan class class ke dalam package tersebut
3. Buatlah object sales
4. Lakukan penjualan produk dengan quantity 5
5. Lakukan restock produk
6. Lakukan memperbaharui harga produk
7. Coba lakukan memperbaharui harga produk dengan nilai negative
8. Modifikasi agar output harga baru dalam format string tidak 1.4E7

Sehingga output yang diharapkan sebagai berikut :

```
PS C:\Users\jtk1409233>  & 'D:\APPLICATION\jdk-11.0.10\bin\java.exe' '-cp' 'C:\User
ws_c2856\jdt_ws\jdt.ls-java-project\bin' 'Main'
Memproses penjualan...
3 Laptop terjual.
Stok setelah penjualan: 7
Menambah stok...
Stok setelah penambahan: 12
Memperbarui harga produk...
Harga baru: 1.4E7
Memperbarui harga produk...
Harga tidak valid!
Harga baru: 1.4E7
PS C:\Users\jtk1409233>
```

Soal 2

```
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;
     }
}
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();
}
}
```

Instruksi Kasus 2

1. Salin ulang baris kode dan lakukan eksekusi terhadap 2 class tersebut, dimana:
    a. Class Barang berfungsi untuk mendefiniskan struktur data yang diperlukan oleh Objek Barang.
    b. 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. Kelas tersebut tersimpan pada package, buatlah package dengan format com.polban.jtk.inventory
3. 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).
4. Carilah solusi, agar variable “stok” dibungkus/ dilindungi sehingga tidak bisa dilakukan operasi aritmatika selain hanya tambah saja.
5. Untuk memecahkan kasus tersebut Bacalah Chapter 4.2.3.

Soal 3

Build dan Import Jar file

1. Compile dan Buat Library (.jar) pada soal no 1 dengan menjalankan perintah javac -d . Product.java Sales.java
2. Buat file JAR:
3. jar cvf saleslibrary.jar com.polban.jtk.sales/*.class
4. buat main program terpisah import jar file tersebut pada program
5. code main program yang terdapat pada soal no 1 copy paste ke program baru yang melaukan import jar file.