LAPORAN PRAKTIKUM
PERTEMUAN-4

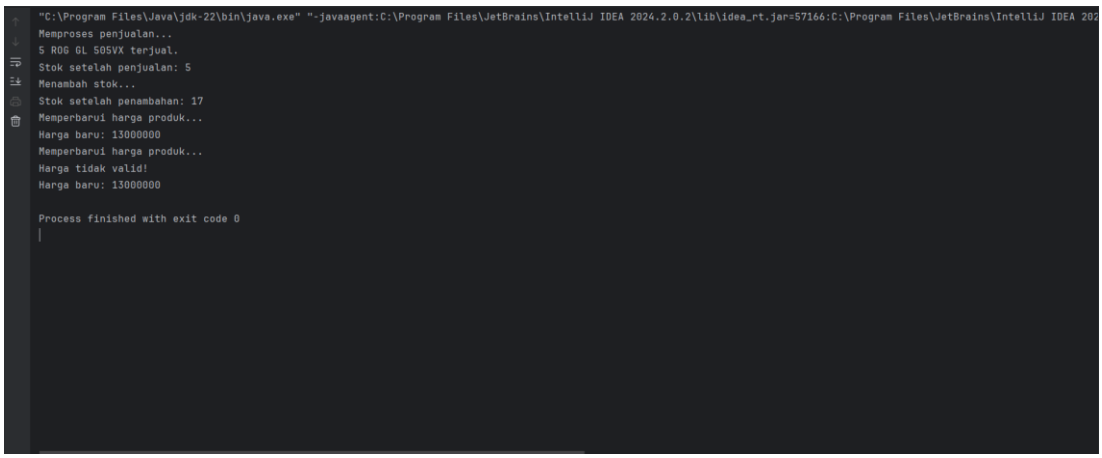
Diajukan untuk memenuhi salat satu tugas praktikum Mata kuliah Pemrograman Berorientasi Objek

Disusun Oleh:
Daiva Raditya Pradipa (231511039)

Jurusan Teknik Komputer dan Informatika
Program Studi D-3 Teknik Informatika
Politeknik Negeri Bandung
2024

Soal Praktikum-4

1. Soal 1 Product and Sales



Pertanyaan

1. Modifikasi agar output harga baru dalam format string tidak 1.4E7
   Untuk mengatasi masalah ini, kita dapat memodifikasi konversi output data pada method updateProductPrice(). Pada bagian yang menampilkan harga produk menggunakan method getPrice() dari class Product, konversi nilai yang di return method getPrice ke tipe data long untuk memastikan harga tidak ditampilkan dalam format eksponensial (seperti 1.4E7), melainkan sebagai angka biasa (seperti 14.000.000).

Leason learn

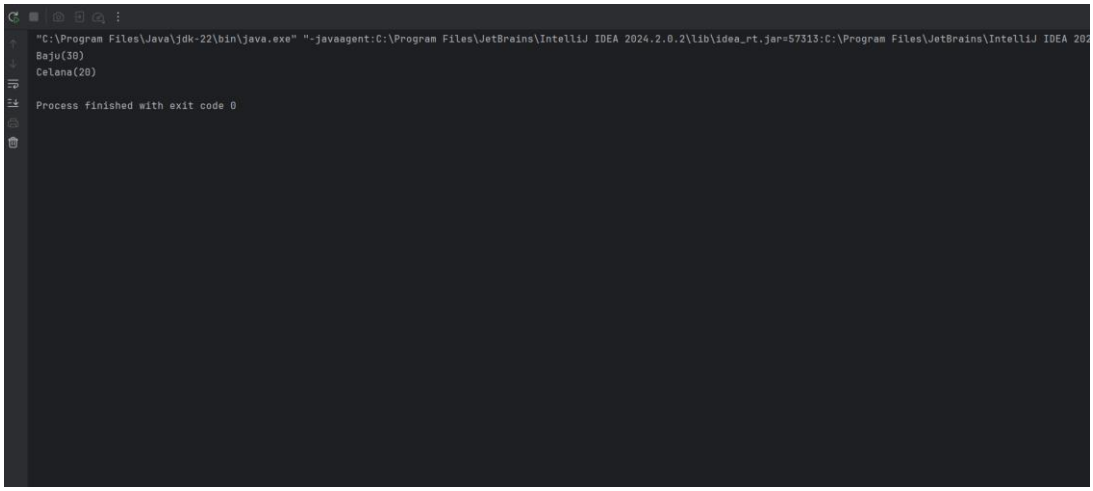
1. Konversi tipe data merupakan salah satu solusi yang dapat digunakan untuk mengatasi masalah nilai numerik yang ditampilkan dalam format eksponensial (misalnya, 1.4E7). Dengan mengonversi tipe *double* ke tipe *long*, kita bisa memastikan hasilnya ditampilkan sebagai angka penuh, seperti 14.000.000.

Masalah: -

Solusi: -

Nama teman yang membantu: -

2. Soal 2 Barang dan Inventori



Pertanyaan:

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

   Untuk memastikan variabel stok yang ada pada class Barang dibungkus atau dilindungi sehingga tidak bisa dilakukan operasi aritmatika kecuali pernambahan, kita dapat menggunakan konsep enkapsulasi dan mengimplemetnasikan getter dan setter pada class Barang. Pada class barang ubah modifie atribut stok menjadi private, dan buat lah getter dan setter untuk atribut stok seperi code berikut

```java
package com.polban.jtk.inventory;

public class Barang {  4 usages  new *
    String kode_barang;  1 usage
    String nama_barang;  3 usages
    private int stok;  3 usages
    public Barang(String kode, String nama, int stk) {  2 usages  new *
        kode_barang = kode;
        nama_barang = nama;
        stok = stk;
    }

    public void setStok(int jumlah) {  1 usage  new *
        if (jumlah > 0) {
            this.stok += jumlah; // Hanya penambahan yang diizinkan
        } else {
            System.out.println("Error: Stok hanya bisa ditambah!");
        }
    }

    public int getStok() {  2 usages  new *
        return stok;
    }
}
```

   Dengan adanya getter dan setter ini kita dapat memastikan perubahan value pada atribut stok dapat dijaga sesuai aturan yang ingin di tetapkan. Misalkan disini kita ingin memastikan bahwa pada atribut tok hanya bisa dilakukan proses penambahan saja, maka dari itu pada setter atribut stok kita dapat membuat parameter untuk meneirma jumlah penambahan stok. Kemudian pada logic method kita memastikan bahwa nilai jumlah penambahan stok yang didapatkan melalui parameter itu bernilai > 0. Hal ini dilakukan untuk menghindari penginputan nilai dengan nilai negatif yang dapat memungkinkan operasi pengurangan pada atribut stok. Apabila parameter jumlah memiliki nilai negative maka program akan menampilkan text yang mengindikasikan bahwa stok hanya bisa ditambah. Sedangkan apabila

parameter bernilai positif (>0) maka program akan menambah jumlah stok yang ada dengan nilai dari parameter jumlah. Dengan menggunakan cara ini, kita dapat memastikan bahwa operasi yang digunakan pada atribut stok pada class Barang hanya bisa melakuka operasi penambahan sedangkan operasi lainnya tidak bisa dilakukan.

Permasalahan: -

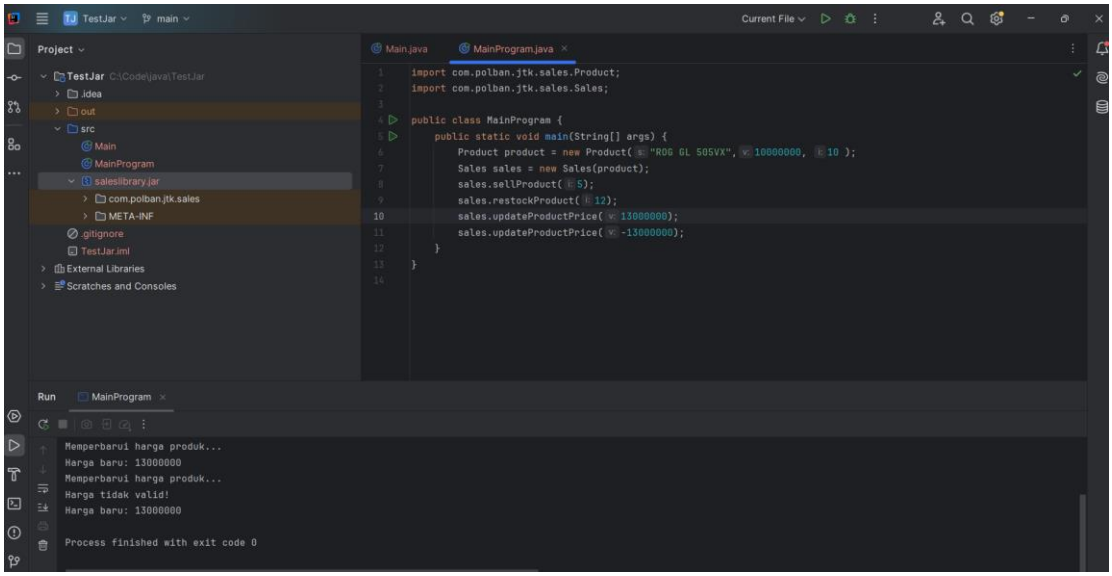
Solusi:-

Leason learn:

1. Konsep enkapsulasi dapat digunakan untuk memastikan bahwa perubahan nilai pada suatu atribut di suatu class sesuai dengan aturan yang kita inginkan dengan mengimplementasikan getter dan setter pada atribut yang kita ingin pastikan perubahan nilainya.

Teman yang membantu: -

3. Build and Import JAR file

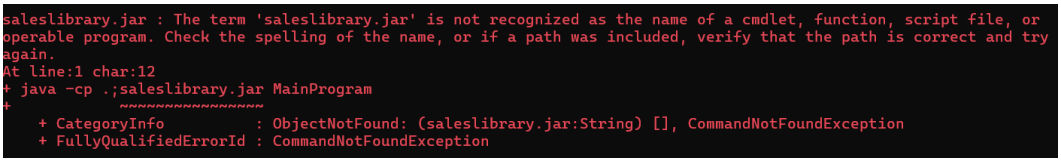


Penjelasan:

Pada soal ketiga ini, kita diharuskan untuk mengubah package yang telah kita buat pada 2 soal sebelumnya menjadi bentuk jar. Sehingga package yang berisi class class tersebut dapat digunakan pada project atau direktori lain tanpa secara langsung kita mengcopy package tersebut (copy hardcode).

Permasalahan:

1. Command jar tidak terbaca pada system



2. Package dalam jar tidak terbaca pada project lain walaupun kita telah mengimport class yang kita butuhkan pada main program dan jar telah berada pada direktori yang sama pada main.

Solusi:

1. Untuk mengatasi masalah command .jar yang tidak terbaca pada sistem operasi, kita perlu menambahkan path direktori bin dari instalasi Java ke dalam environment variables sistem operasi kita.
   a. Buka Control Panel dan pilih "System" atau klik kanan pada "This PC" dan pilih "Properties".
   b. Pilih "Advanced system settings", lalu klik tombol "Environment Variables".
   c. Pada bagian "System variables", cari variabel bernama Path dan pilih "Edit".
   d. Tambahkan direktori bin dari instalasi Java (misalnya C:\Program Files\Java\jdkX.X.X\bin) ke dalam daftar path tersebut, lalu klik "OK".
   e. Setelah itu, restart terminal atau command prompt Anda agar perubahan PATH tersebut diterapkan.
2. Untuk mengatasi package dalam jar tidak terbaca pada pada project lain walaupun kita telah mengimport class yang kita butuhkan pada main program dan jar telah berada pada direktori yang sama pada main.Kita hanya perlu mengimport file jar tersebut pada IDE intelij dengan cara sebagai berikut
   a. Klik kanan pada folder project Anda, pilih "Open Module Settings"
   b. Di dalam jendela Project Structure, pilih "Modules" di panel sebelah kiri.
   c. Pilih module yang ingin Anda tambahkan file .jar-nya.
   d. Di bagian "Dependencies", klik ikon plus (+), lalu pilih "JARs or directories".
   e. Arahkan ke lokasi file .jar yang ingin Anda tambahkan, lalu klik "OK".
   f. Pastikan file .jar sudah muncul di daftar Dependencies, kemudian klik "Apply" dan "OK".

Leason learn:

1. Kita dapat meng-compile sebuah package yang berisi berbagai class ke dalam bentuk file .jar, sehingga memungkinkan kita untuk mengakses class-class tersebut di project lain tanpa perlu menyalin kode sumber (hardcode) secara langsung ke project tersebut. Dengan menggunakan file .jar, kita dapat dengan mudah mengakses kode secara modular di berbagai project.

Link github source code week4: Tugas-PBO/Week-4 at main · RaditZX/Tugas-PBO (github.com)