**LAPORAN HASIL PRAKTIKUM**

**MATA KULIAH :**

**TEKNIK PEMROGRAMAN**

Logo, company name

Description automatically generated

Disusun Oleh :

|  |  |
| --- | --- |
| **NIM** | **: 221524061** |
| **NAMA** | **: Thoriq Muhammad Fadhli** |
| **KELAS** | **: 1BD4** |

**Disusun pada : Minggu, 5 Februari 2023**

**DAFTAR ISI**

[Kasus 1 3](#_Toc126501782)

[Kasus 2 5](#_Toc126501783)

[Kasus 3 6](#_Toc126501784)

**Link GitHub :** [**https://github.com/tmfadhi12/tekpro/tree/master/Pertemuan3**](https://github.com/tmfadhi12/tekpro/tree/master/Pertemuan3)

# Kasus 1

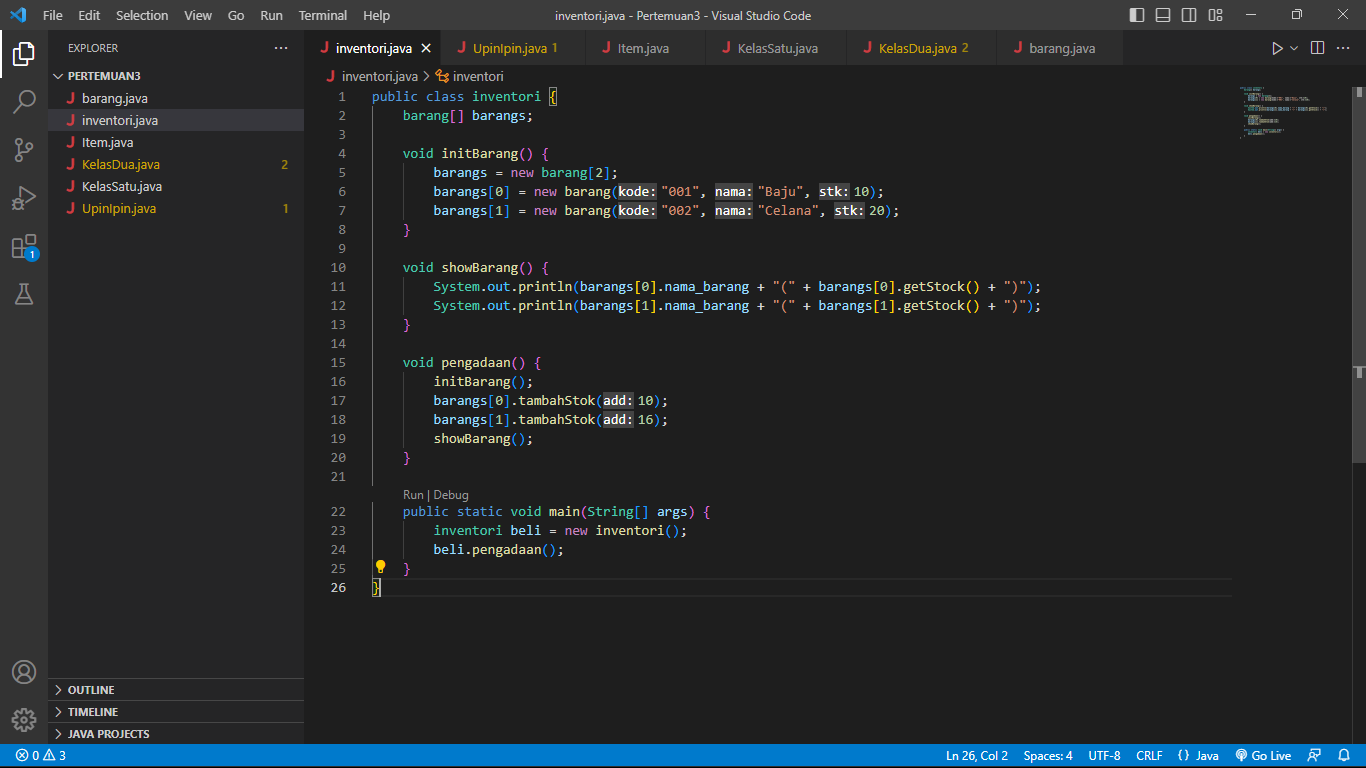
Hasil Akhir Program :

**barang.java**

A screenshot of a computer

Description automatically generated with medium confidence

**inventori.java**



**Permasalahan Kasus :**

Carilah Solusi Agar variable stok dilindungi sehingga tidak bisa dilakukan operasi aritmatika selain hanya tambah saja.

**Solusi :**

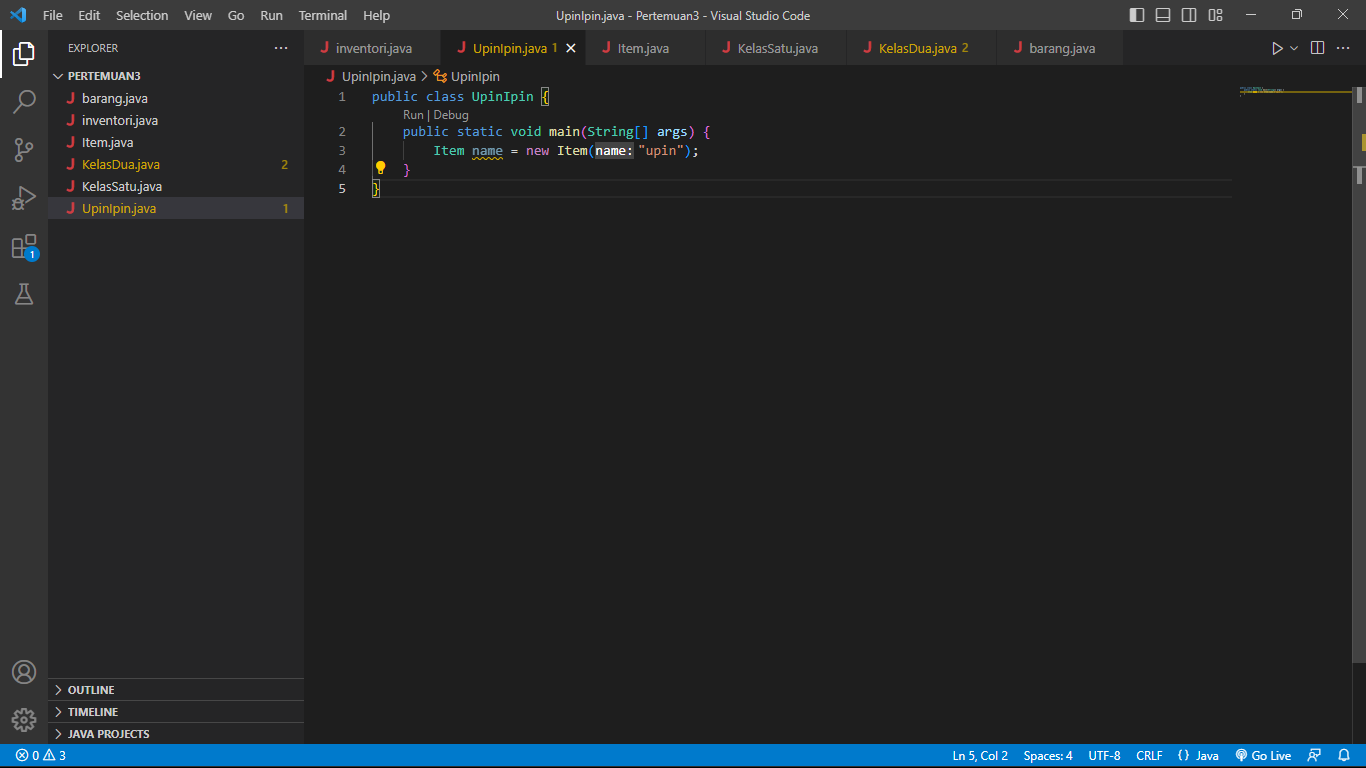
Menggunakan metode *encapsulation*, sehingga stok merupakan private variable dan hanya bisa diubah melalui fungsi tambahStok(), dan hanya melakukan operasi tambah saja.

**Teman yang membantu :** Faris Abulkhoir.

# Kasus 2

Hasil Akhir Program :

UpinIpin.java



Item.java

A screenshot of a computer

Description automatically generated with medium confidence

**Permasalahan Kasus :** Tampilan Null dan seharusnya menampilkan Ipin.

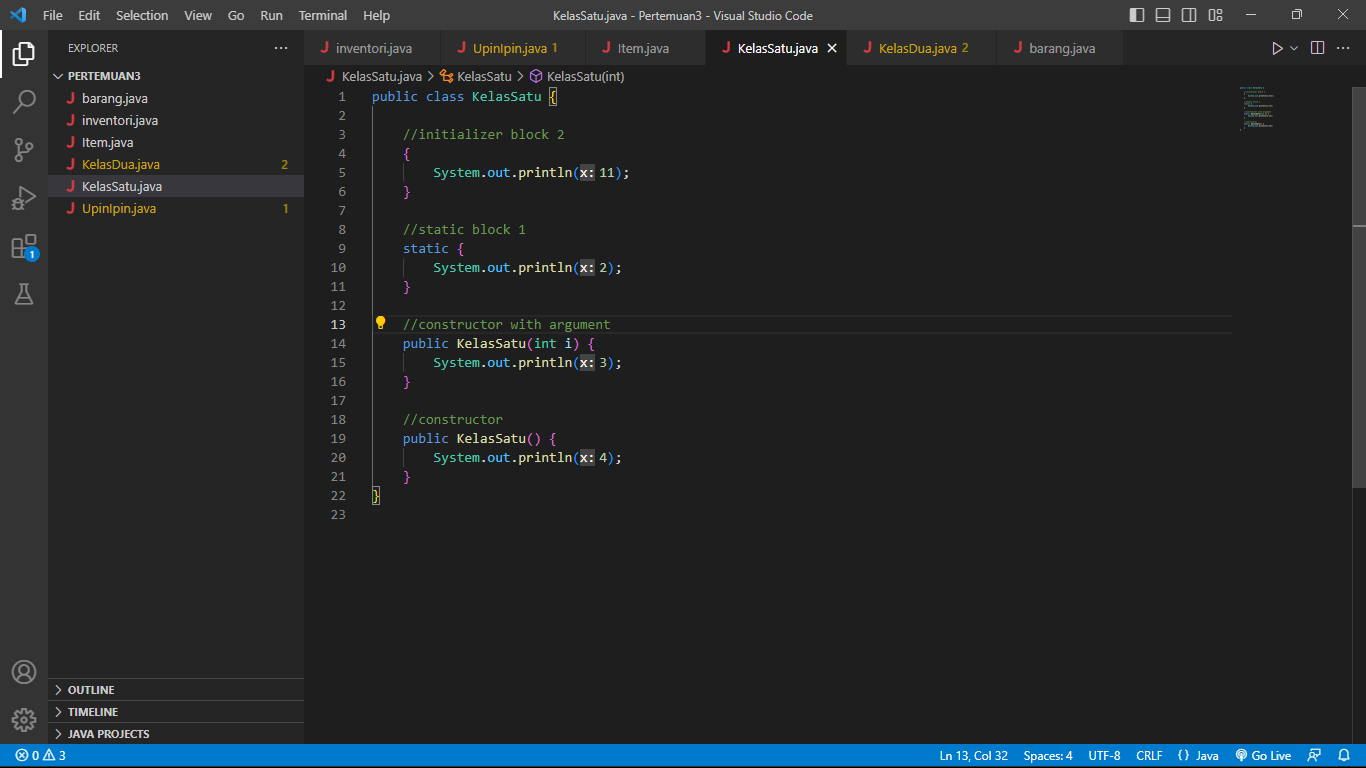
**Solusi :**

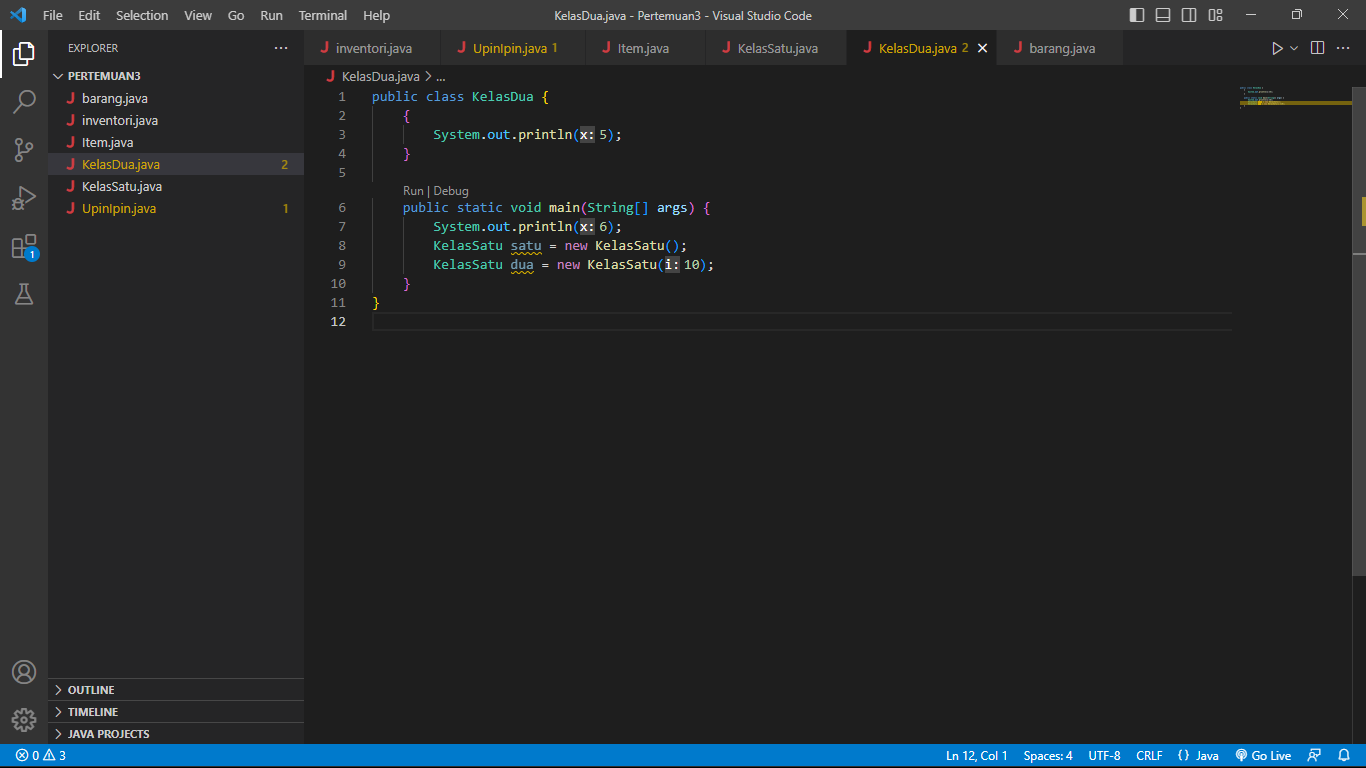
Menambahkan this() pada constructor public Item(). Sehingga constructor private Item terpanggil dikarenakan nama constructor private Item() sama dengan nama class Item().

**Teman yang membantu :** Faris Abulkhoir

# Kasus 3

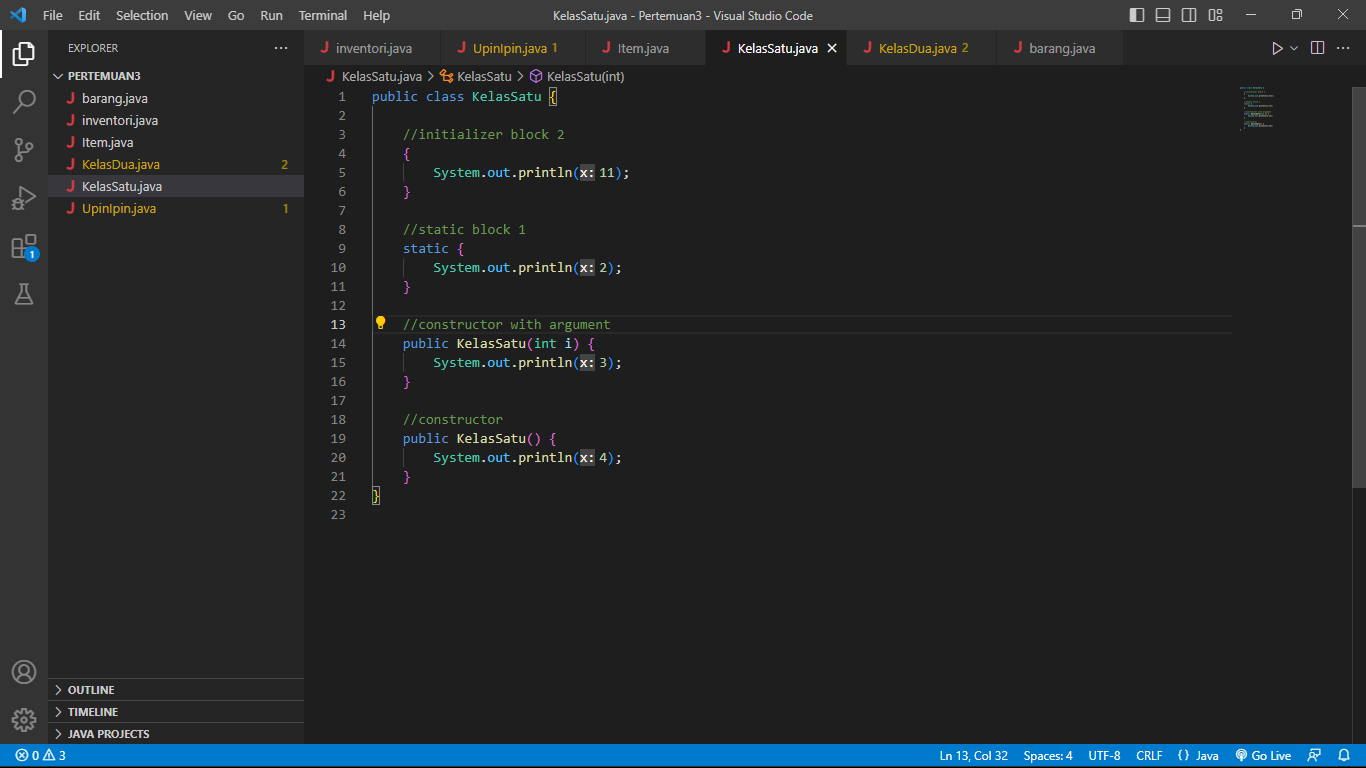
Hasil Akhir Program :





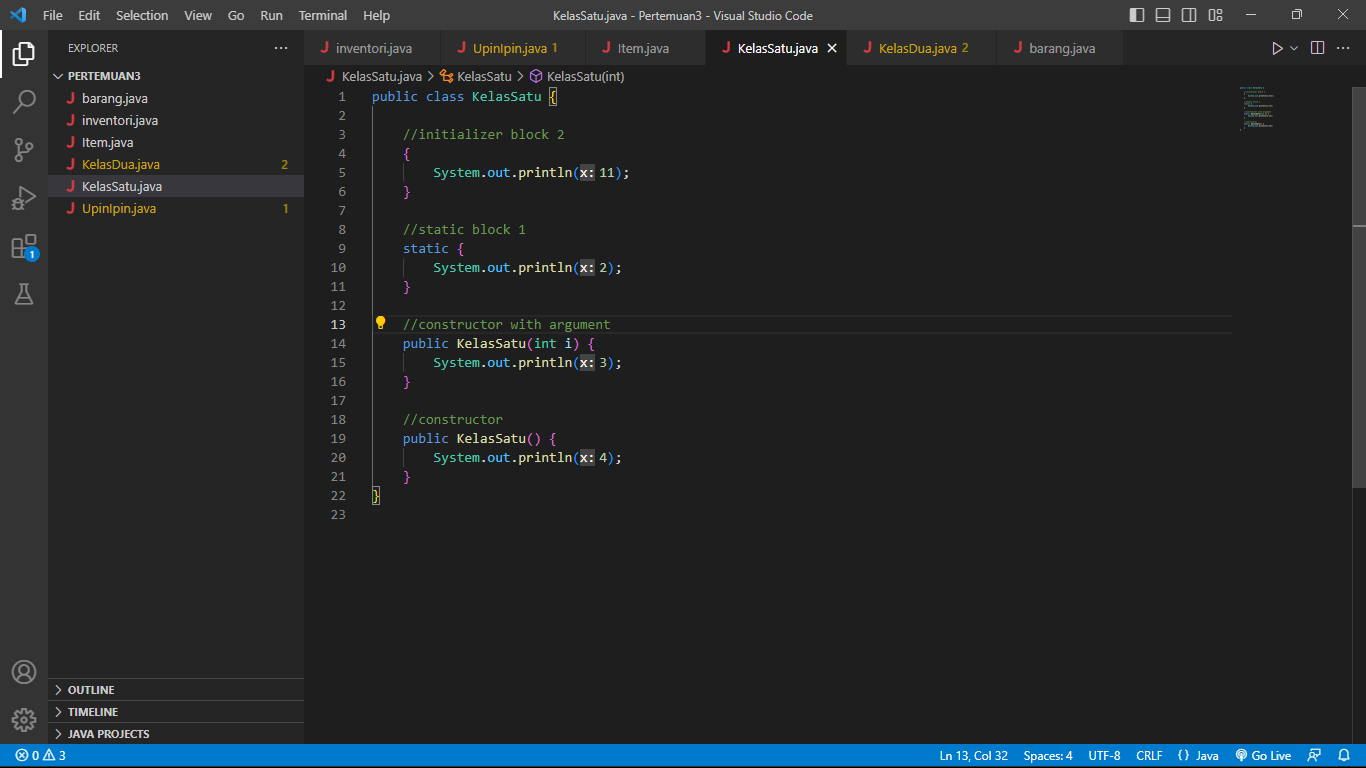
**Penjelasan :**

***Static Initialization Block***atau *Static Block ,* contoh :



Merupakan block yang dijalankan/dieksekusi paling awal bahkan sebelum main() method pada java. Dan hanya dijalankan 1x, sehingga ketika dipanggil 2x static block tetap hanya mengeksekusi perintah didalamnya 1x.

***Instance Initialization Block*,** contoh :



Merupakan block yang dijalankan ke-2 jika ada *Static Block* di class yang sama. Dan *Instance Initialization Block* akan berjalan setiap kali ada memanggil block ini. Tidak seperti *Static Block* yang hanya 1x.

***Constructor Without an Argument*,** contoh :

A screenshot of a computer

Description automatically generated with medium confidence

***Constructor With an Argument,*** contoh :

A screenshot of a computer

Description automatically generated with medium confidence

*Constructor* adalah merupakan suatu method yang biasa digunakan untuk menginisialisasi sebuah objek yang kemudian nanti dipanggil di main().

Apabila *constructor* tidak memiliki argument, maka ketika dipanggil (misal: KelasSatu s = new KelasSatu()) maka *constructor* KelasSatu didalam *class* KelasSatu yang tidak memiliki argument akan terpanggil dan sebaliknya jika cara memanggilnya diganti menjadi “KelasSatu s = new KelasSatu(10)” maka *constructor* KelasSatu(int i) akan terpanggil dikarenakan kita memasukkan sebuah argument ke dalam parameter, Sehingga *constructor* KelasSatu yang tidak memiliki argument tidak akan terpanggil.