**TUGAS MATA KULIAH**

**PEMROGRAMAN BERORIENTASI OBJEK**

**LAPORAN TUGAS KE - 4**

**TOPIK : Object, Class, and Encapsulation**

A logo with a blue and orange design

Description automatically generated

**Disusun Oleh:**

**221511046 Danendra Gafrila**

**KELAS 2 – B**

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA PROGRAM STUDI DIPLOMA III TEKNIK INFORMATIKA POLITEKNIK NEGERI BANDUNG**

**2023**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc145857570)

[CONTOH SOAL 3](#_Toc145857571)

[1. Latihan 1 3](#_Toc145857572)

[2. Latihan 2 4](#_Toc145857573)

[3. Latihan 3 5](#_Toc145857574)

# CONTOH SOAL

## Latihan 1

Code :

A screenshot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

Masalah dan Penjelasan :

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

Dalam hal ini saya mengatasi dengan mengubah stok yang asalnya public menjadi private dan dimana ada addStok itu hanya ada operasi penjumlahan sehingga tidak akan bisa diubah apabila dikurangi, dikali, maupun dibagi.

Dalam menyelesaikan kasus ini saya dibantu oleh :

* Fadel Mohammad Fadillah.

## Latihan 2

A computer code with text

Description automatically generated

Masalah dan Solusi :

* Cari solusi agar ouput yang keluar adalah Ipin bukan Upin !

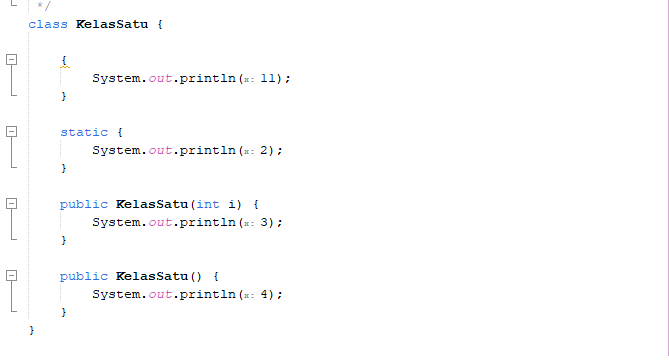
Solusi dari permasalahan kali ini adalah pemanggilan constructor dengan code this(), this() ini dapat memanggil constructor di class yang sama sehingga output yang dipanggil dengan this.name adalah ipin dan bukan upin Kembali.

Dalam menyelesaikan kasus ini saya dibantu oleh :

* Rizki Gunawan.

## Latihan 3

Code :



A computer screen shot of a code

Description automatically generated

Masalah dan Penjelasan :

* Analisis outputnya mengapa seperti itu ?

penjelasan mengapa hasilnya seperti itu:

1. Blok Static:
   * Blok **static** dalam sebuah kelas dieksekusi saat kelas itu sendiri dimuat ke dalam memori. Ini hanya terjadi satu kali sebelum Anda membuat objek dari kelas tersebut. Dalam kasus ini, blok **static** dalam **KelasSatu** mencetak **2** pertama kali.
2. Blok Inisialisasi Objek (Non-Static):
   * Blok inisialisasi objek (non-static) dalam kelas dieksekusi setiap kali objek dari kelas itu dibuat. Dalam kasus ini, blok inisialisasi objek dalam **KelasSatu** mencetak **11** setiap kali objek dari **KelasSatu** dibuat.
3. Method **main**:
   * Method **main** adalah entry point program dan akan dieksekusi pertama kali. Dalam kasus ini, **System.out.println(6);** akan mencetak **6** saat **main** method dari **KelasDua** dieksekusi.
4. Pembuatan Objek **KelasSatu**:
   * Ketika Anda membuat objek **KelasSatu satu = new KelasSatu();**, konstruktor tanpa argumen dari **KelasSatu** dipanggil, yang mencetak **4**. Setelah itu, blok inisialisasi objek non-static dalam **KelasSatu** dieksekusi lagi dan mencetak **11** untuk kedua kalinya.
5. Pembuatan Objek **KelasSatu** dengan Argumen:
   * Ketika Anda membuat objek **KelasSatu dua = new KelasSatu(10);**, konstruktor dengan argumen integer dari **KelasSatu** dipanggil, yang mencetak **3**.