**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJECT**

Logo, icon, company name

Description automatically generated

Disusun oleh:

Jihan Afifah Yanriyani

201511046

D32B Teknik Informatika

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI D3 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2021**

Kasus 1

1. Output sebelum :

Graphical user interface, text, application

Description automatically generated

Output sesudah :

Graphical user interface, text, application, email

Description automatically generated

Masalah dan Solusi :

Data stok masih bisa di manipulasi dengan operasi artimatika. Solusi merubah public int menjadi private, dengan kata lain melakukan proses enkapsulasi menambahkan method getter dan addStok dan merubah showBarang() getstok dan menambah addStok(20) sehingga hasil akhir dari stok baju ialah 30.

Source Code :

**Barang.java**

public class Barang {

String kode\_barang;

String nama\_barang;

private int stok;

public Barang(String kode, String nama, int stk) {

kode\_barang = kode;

nama\_barang = nama;

stok = stk;

}

public int getStok() {

return stok;

}

public void addStok(int stok){

this.stok = this.stok + stok;

}

}

**Invetori.Java**

public class Invetori {

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].getStok() + ")");

System.out.println(barangs[1].nama\_barang + "(" + barangs[1].getStok() + ")");

}

void pengadaan() {

initBarang();

barangs[0].addStok(20);

// 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) {

Invetori beli = new Invetori();

beli.pengadaan();

}

}

KASUS 2

1. Output :

Graphical user interface, text, application, email

Description automatically generated

Masalah dan Solusi :

Modifikasi code class item sehingga output yang dihasilkan hanya ipin. Solusi menambahkan this(); karena digunakan untuk memanggil constructor item yang identik.

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) {

this.name = "Ipin";

System.out.println(this.name);

}

}

KASUS 3

1. Output :

Graphical user interface, text, application, email

Description automatically generated

Analisis urutan konstruksi objek :

* 6 ada diurutan pertama karena dia terdapat di main sehingga dia berada urutan paling awal
* 2, secara hierarki 2 dan 11 memiliki kedudukan yang sama, tetapi static{} akan otomatis terpanggil sebelum class-nya di akses. “Static initialiser block that will also be called automatically before the class is first used”
* 11, berada di kurung kurawal jadi ketika kelasSatu diakses, akan melakukan default dan memberi value dari initializer block.
* Urutan antara 4 dan 3 mengapa 4 lebih dahulu karena dalam print(3) itu terdapat parameter int I sedangkan urutan prioritas dimulai dari non-parameter.