

**TUGAS MATA KULIAH**  
**PEMROGRAMAN BERORIENTASI OBJEK**

**LAPORAN PRAKTIKUM PERTEMUAN 4**



**Disusun Oleh:**

**221511004**

**Aulia Aziizah Fauziyyah**

**KELAS 2 – A**

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

**2023**

## DAFTAR ISI

KASUS 1 .....	1
KASUS 2 .....	3
KASUS 3 .....	5

## KASUS 1

### 1.1. Hasil Akhir Program

Code Program :

```
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;
        this.stok = stk;
    }

    public int getStok() {
        return stok;
    }

    public void tambahStok(int jumlah) {
        if (jumlah > 0) {
            stok += jumlah;
        } else {
            System.out.println("Jumlah penambahan stok harus lebih dari 0.");
        }
    }
}
```

```
public class inventori {
    private barang[] barangs;

    public void initBarang() {
        barangs = new barang[2];
        barangs[0] = new barang(kode: "001", nama: "Baju", stk:10);
        barangs[1] = new barang(kode: "002", nama: "Celana", stk:20);
    }

    public void showBarang() {
        System.out.println(barangs[0].nama_barang + "(" + barangs[0].getStok() + ")");
        System.out.println(barangs[1].nama_barang + "(" + barangs[1].getStok() + ")");
    }

    public void pengadaan() {
        initBarang();
        barangs[0].tambahStok(jumlah: 20);
        showBarang();
    }

    public static void main(String[] args) {
        inventori beli = new inventori();
        beli.pengadaan();
    }
}
```

### 1.2. Jawaban Soal

Output :

---

```
run:
Baju(30)
Celana(20)
BUILD SUCCESSFUL (total time: 0 seconds)
```

Variable stok adalah variable yang dienkapsulasi atau dilindungi dengan modifier private sehingga hanya dapat diakses didalam kelas itu sendiri. Untuk dapat mengakses atribut kelas tertentu agar tidak terjadi ambiguitas maka diperlukan kata kunci “this” untuk dapat membantu mengklarifikasi akses atribut kelas tertentu.

Metode get adalah metode yang digunakan untuk mengakses nilai dari atribut atau properti suatu objek atau kelas. Metode ini sering digunakan untuk mengakses baca ke atribut – atribut yang bersifat private atau protected. Dengan menggunakan metode ini dapat melindungi atribut dari akses secara langsung dari luar kelas dan memastikan bahwa akses ke atribut hanya dapat dilakukan dari luar kelas dan memastikan bahwa akses ke atribut hanya dapat dilakukan melalui metode yang telah ditentukan.

Metode get biasanya diikuti pola yang sesuai dengan nama atribut yang ingin diaksesnya. Metode ini biasanya tidak memiliki parameter dan mengembalikan nilai dari atribut tersebut.

### **1.3. Permasalahan**

### **1.4. Solusi**

### **1.5. Referensi dan Nama Teman yang Membantu Memecahkan Permasalahan di Permasalahan**

## KASUS 2

### 2.1. Hasil Akhir Program

Code Program :

```
public class item {  
    private String name;  
    private item() {  
        name = "ipin";  
    }  
  
    public item(String name) {  
        this();  
        System.out.println(x: this.name);  
    }  
}  
  
public class upinipin {  
    public static void main(String[] args) {  
        item name = new item(name: "upin");  
    }  
}
```

### 2.2. Jawaban Soal

Output :

```
run:  
ipin  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pada konstruktor kelas item, terdapat atribut nama dengan inisialisasi nilai “ipin” yang bersifat private. Untuk dapat memastikan bahwa konstruktor ini dapat dieksekusi ketika item objek dibuat maka dapat menggunakan pemanggilan “this()”. “this()” adalah pemanggilan konstruktor lain dalam kelas yang sama dari konstruktor lain yang sedang berjalan. Terdapat beberapa aturan dalam menggunakan metode “this()” diantaranya adalah harus menjadi instruksi pertama dalam konstruktor yang memanggil konstruktor lain, tidak boleh digunakan dalam metode non konstruktor atau diluar konteks konstruktor.

**2.3. Permasalahan**

**2.4. Solusi**

**2.5. Referensi dan Nama Teman yang Membantu Memecahkan Permasalahan di Permasalahan**

## KASUS 3

### 3.1. Hasil Akhir Program

Code Program :

```
public class KelasSatu {
    {
        System.out.println(x: 11);
    }
    static
    {
        System.out.println(x: 2);
    }
    public KelasSatu(int i)
    {
        System.out.println(x: 3);
    }
    public KelasSatu()
    {
        System.out.println(x: 4);
    }
}

public class KelasDua {
    {
        System.out.println(x: 5);
    }
    public static void main(String[] args)
    {
        System.out.println(x: 6);
        KelasSatu satu = new KelasSatu();
        KelasSatu dua = new KelasSatu(i: 10);
    }
}
```

### 3.2. Jawaban Soal

Output :

```
run:
6
2
11
4
11
3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Terdapat blok inisialisasi main utama yang akan dijalankan ketika program dimulai dan akan mengeluarkan output angka 6 ke layar. Selanjutnya terdapat static yang akan di eksekusi saat “KelasSatu” pertama kali dibuat dan akan mengeluarkan output angka 2 ke layar. Selanjutnya untuk blok non-static, akan dieksekusi setiap objek kelas “KelasSatu” dibuat dan akan mengeluarkan output 11 ke layar. Selanjutnya untuk konstruktor “KelasSatu()” berguna untuk menciptakan objek dari “KelasSatu” tanpa parameter dan akan mengeluarkan output angka 4 ke layar. Selanjutnya untuk konstruktor “KelasDua()” berguna untuk menciptakan objek dari “KelasDua” tanpa parameter dan akan mengeluarkan output angka 3 ke layar.

### **3.3. Permasalahan**

### **3.4. Solusi**

### **3.5. Referensi dan Nama Teman yang Membantu Memecahkan Permasalahan di Permasalahan**