

OBJECT ORIENTED PROGRAMMING
QUIZ



By :

Sri Kresna Maha Dewa

2241720244

STUDY PROGRAM D-IV INFORMATIC ENGINEERING
INFORMATION TECHNOLOGY DEPARTMENT
MALANG STATE POLYTECHNIC

Soekarno Hatta Street No.9, Jatimulyo, Lowokwaru District, Malang City, East Java

65141

Class and Object

1. Class adalah sebuah blueprint atau prototype dari sebuah objek, dalam membuat sebuah objek kita perlu membuat class terlebih dahulu
2. Mendefinisikan class dalam Bahasa dapat menggunakan struktur modifier – class – nama class, contoh *public class Sepeda*. Pemberian nama class juga sebaiknya diawali dengan huruf besar, kata benda, dan jika lebih dari 1 maka format penamaan yang digunakan adalah Camel Case.
3. Untuk membuat laptop maka saya perlu melakukan instansiasi dari class Barang lalu memberinya nama laptop

Encapsulation

1. Encapsulation adalah sebuah konsep yang melindungi data dan method dari intervensi luar, karena encapsulation akan membatasi akses langsung terhadap attribute dan method. Hal ini penting untuk program inventaris karena setiap class pada program yang akan dibuat pasti akan memiliki sebuah attribute atau method yang memiliki akses terbatas untuk berinteraksi dengan class lainnya.
2. Misal kode barang yang merupakan kode unik untuk mengidentifikasi barang dalam sistem informasi inventaris. Attribute ini harus diencapsulate karena merupakan informasi penting yang digunakan untuk mencari, menyimpan, dan mengelola barang. Jika attribute ini diakses atau diubah secara tidak sah, dapat menyebabkan kesalahan data atau sistem error.

Relasi Kelas

1. Relasi antar class adalah hubungan antara dua class atau lebih yang dimana sebuah class memerlukan attribute atau method dari class lain agar dapat menjadi sebuah class yang utuh dan berfungsi.
2. Barang dan Kategori memiliki relasi Aggregation dimana barang yang sama dan berkumpul akan membentuk sebuah kategori dengan nama tertentu.

PBL

1. Class Barang

```
1 public class Barang {
2     private String nama;
3     private int kodeBarang;
4     private int jumlah;
5     private double harga;
6
7     public Barang(String nama, int kodeBarang, int jumlah, double harga) {
8         this.nama = nama;
9         this.kodeBarang = kodeBarang;
10        this.jumlah = jumlah;
11        this.harga = harga;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public void setNama(String nama) {
19        this.nama = nama;
20    }
21
22    public int getKodeBarang() {
23        return kodeBarang;
24    }
25
26    public void setKodeBarang(int kodeBarang) {
27        this.kodeBarang = kodeBarang;
28    }
29
30    public int getJumlah() {
31        return jumlah;
32    }
33
34    public void setJumlah(int jumlah) {
35        this.jumlah = jumlah;
36    }
37
38    public double getHarga() {
39        return harga;
40    }
41
42    public void setHarga(double harga) {
43        this.harga = harga;
44    }
45
46    public double hitungTotalHarga() {
47        return jumlah * harga;
48    }
49
50    public void tampilkanInfoBarang() {
51        System.out.println("Nama Barang: " + nama);
52        System.out.println("Kode Barang: " + kodeBarang);
53        System.out.println("Jumlah: " + jumlah);
54        System.out.println("Harga Satuan: Rp " + harga);
55        System.out.println("Total Harga: Rp " + hitungTotalHarga());
56    }
57 }
58
```

2. Dengan menggunakan modifier private

3. Gambaran hierarki :

