

Pemrograman Berorientasi Objek

C#



#2 Mengimplementasikan Kelas
dan Method

Kompetensi Dasar

- Setelah mengikuti mata kuliah ini diharapkan mahasiswa memiliki kemampuan untuk:
 1. Menguraikan konsep dasar PBO
 2. Mengimplementasikan kelas (class) dan metode (method) untuk mewakili obyek-obyek dalam sistem
 3. Mengimplementasikan abstrak dan interface dalam suatu kelas
 4. Menggunakan initialization dan instance dalam class
 5. Menerapkan konsep orientasi objek : Inheritance dalam sistem
 6. Menerapkan konsep Polimorfisme
 7. Menguraikan dasar UML
 8. Mengidentifikasi Permasalahan menggunakan Use Case Diagram
 9. Mengidentifikasi Permasalahan menggunakan Activity Diagram
 10. Mengidentifikasi Permasalahan menggunakan Sequence Diagram
 11. Menyusun Class Diagram dari permasalahan
 12. Mewujudkan Object Oriented Design (OOD) dan Object Oriented Programming (OOP) menjadi sebuah aplikasi (Studi Kasus)

Overview

- Pemograman berorientasi Objek
 - Data menjadi property
 - Prosedur untuk data menjadi method
 - Data + Prosedur menjadi kelas

Deklarasi Kelas

- Deklarasi kelas memiliki sintaks seperti berikut:

```
class NamaKelas  
{  
    /* nol atau lebih property */  
    /* nol atau lebih method */  
}
```

Property

- Deklarasi/definisi property:

Tipe data namaproperty;

- Tipe data adalah tipe data primitif atau kelas

int x;

point p;

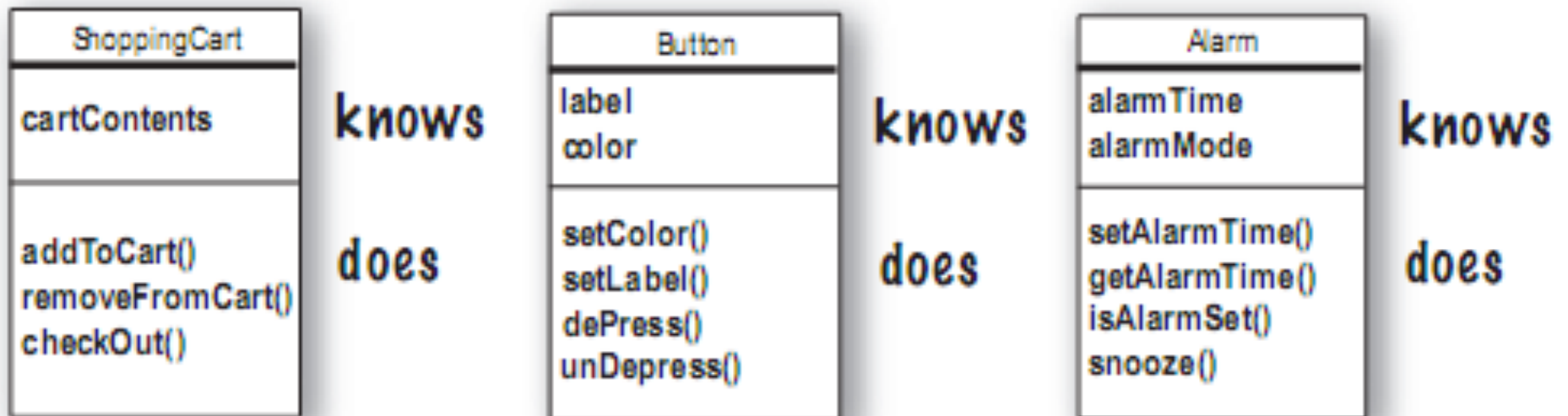
Method

- Deklarasi method sama seperti prosedur di C:

```
public class ClassXXX  
{  
    tipekembalian method (tipe parameter)  
    {  
        /* isi method */  
        return hasil; /* jika nilai kembali tidak void*/  
    }  
}
```

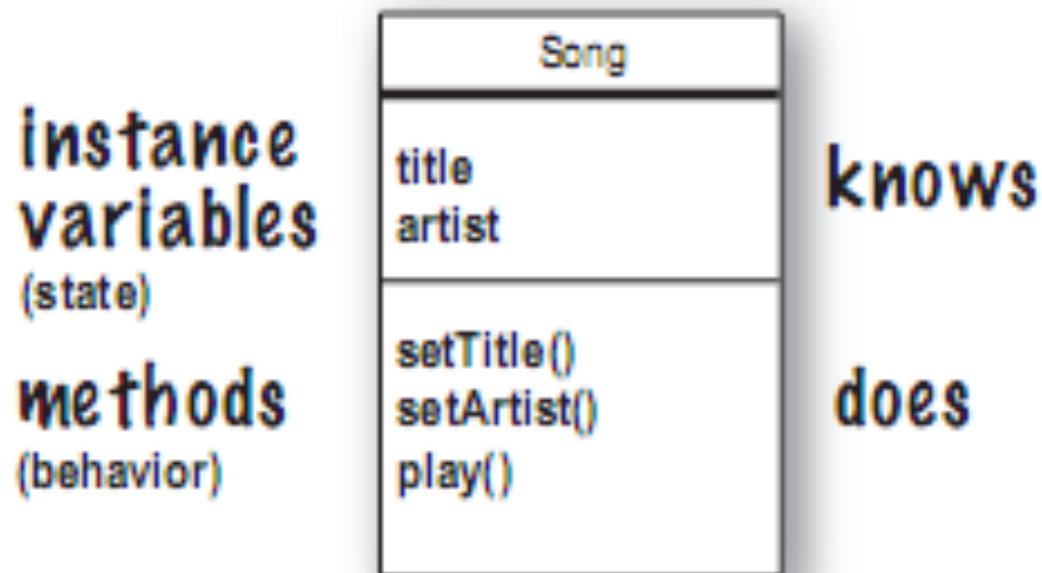
Class Design

- Ketika akan mendesain suatu kelas, pikirkan:
 - Hal yang diketahui objek
 - Hal yang dikerjakan objek

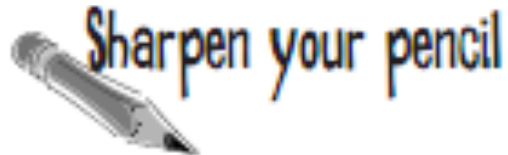


Class Design (2)

- Hal yang diketahui objek disebut dengan ***instance variable***
- Hal yang dilakukan objek disebut dengan ***method***



Latihan



Fill in what a television object might need to know and do.

Television

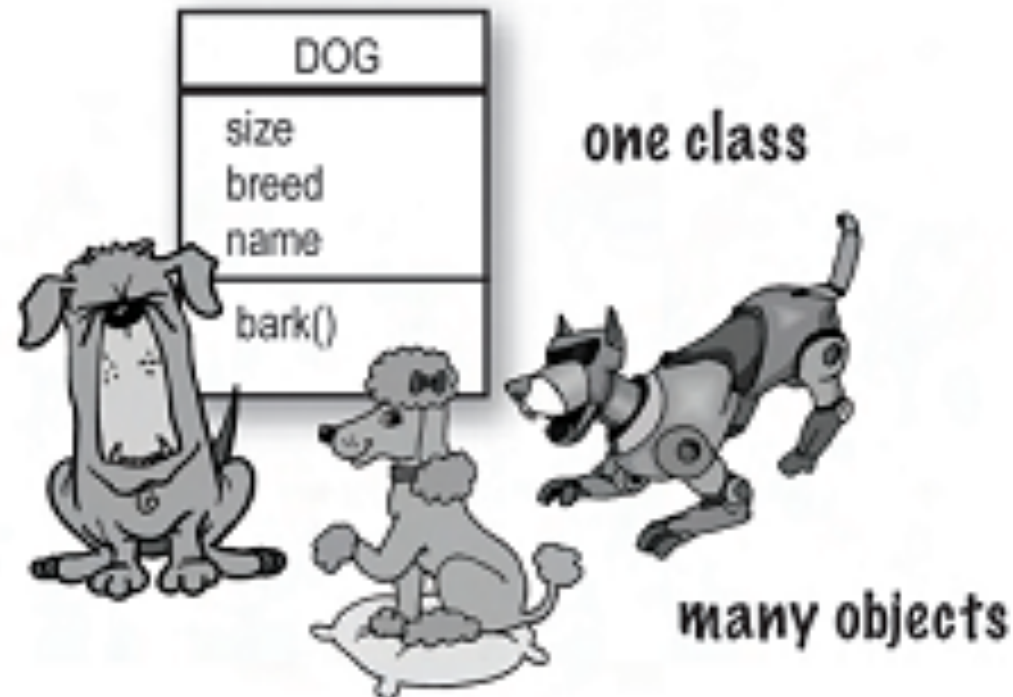


**instance
variables**

methods

Class Design

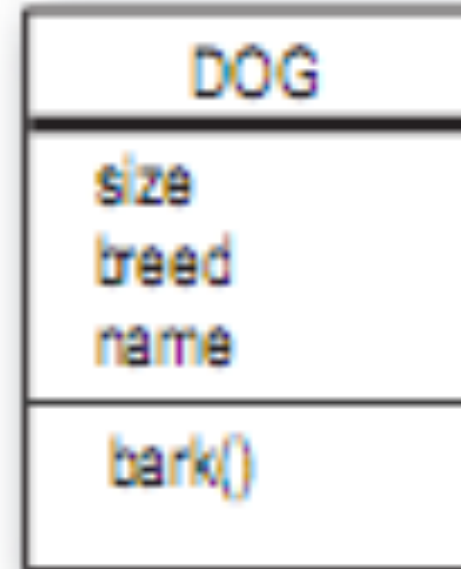
- Apa perbedaan antara class dan objek?
 - Class bukan objek, tetapi dipergunakan untuk membangun objek



Class Design

- Menulis Class

```
class Dog  
{  
    public int size;  
    public string breed;  
    public string name;  
  
    public void bark()  
    {  
        Console.WriteLine("Woof! Woof!", name);  
    }  
}
```



Class Design

- Class Tester

Class DogTestDrive

```
{  
    static void Main(String[] args)  
    {  
        Dog d = new Dog();  
        d.size = 40;  
        d.bark();  
    }  
}
```

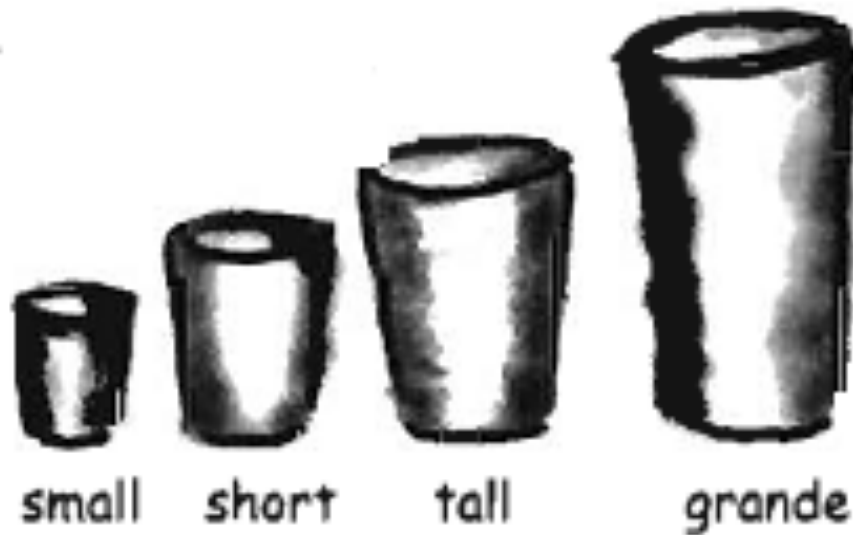
Variables

- Ada dua jenis variabel:
 - Primitive
 - Reference
- Deklarasi variabel
 - Variabel harus punya type
 - Variable harus punya nama
 - contoh:

```
int count;
```

Variables

- Variabel bisa dianalogikan seperti suatu gelas/kontainer, yang berfungsi menyimpan sesuatu



Variables



Variables

- Jangan mengisi variabel melebihi dari kapasitasnya!

```
int x = 24;
```

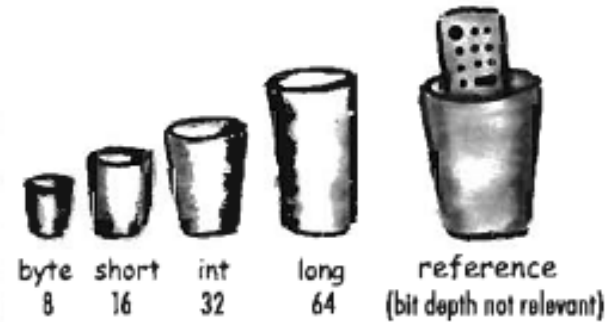
```
byte b = x;
```

```
//won't work!
```



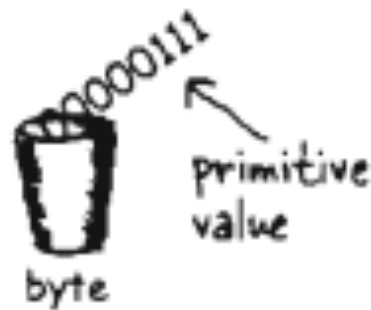
Reference Variable

- Tidak ada yang namanya object variable, yang ada hanya object reference variable
- **Object reference variable** berisi bit yang merepresentasikan suatu cara untuk mengakses suatu object
- Untuk lebih mudahnya, reference variable bisa dianalogikan dengan suatu remote control



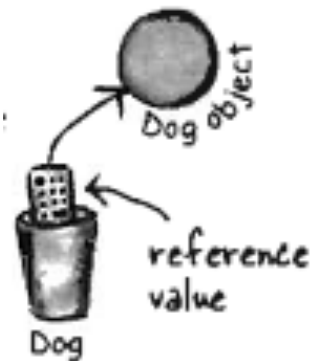
- Primitif variable

```
byte b = 7;
```



- Reference Variable

```
Dog d = new Dog ();
```



3 tahap dalam object declaration, creation & assignment

```

    1      2
    Dog myDog = new Dog();
  
```

1. Deklarasi reference variable

```

Dog d = new Dog();
  
```

2. Create Object

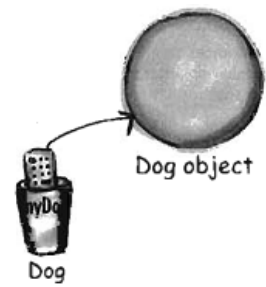
```

Dog d = new Dog();
  
```

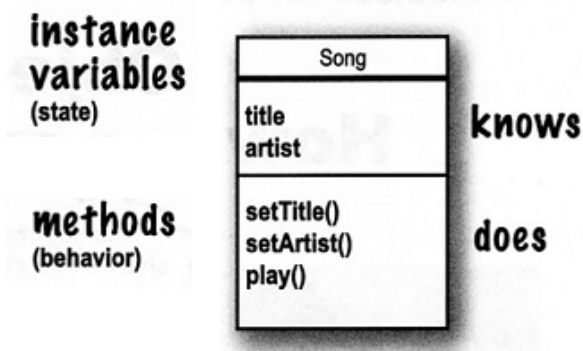
3. Link Object & reference

```

Dog d = new Dog();
  
```



- Ingat! : suatu class mendeskripsikan apa yang object ketahui dan apa yang object lakukan



- Objek dapat berperilaku berbeda walaupun menjalankan method yang sama kalau memiliki state yang berbeda

Class Sapi

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace SapiBaruSaya
```

```
{
```

```
    public class Sapi
```

```
    {
```

```
        public String nama;
        public int jumlahKaki;
        public int umur;
```

Variables

```
        public void Makan()
```

```
        {
```

```
            Console.WriteLine(" {0}sedang makan.", nama);
```

```
        }
```

```
        public void Jalan()
```

```
        {
```

```
            Console.WriteLine(" {0}sedang jalan.", nama);
```

```
        }
```

```
        public void Bobo()
```

```
        {
```

```
            Console.WriteLine(" {0}sedang bobo.", nama);
```

```
        }
```

```
    }
```

```
}
```

Method

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace SapiBaruSaya
{
    class Program
    {
        static void Main(string[] args)
        {
            Sapi sapiSaya;

            sapiSaya = new Sapi();
            sapiSaya.nama="Unyil";
            sapiSaya.umur=10;
            sapiSaya.jumlahKaki=4;

            console.WriteLine("Nama sapi saya adalah {0}.", sapiSaya.nama);
            console.WriteLine("Ia memiliki kaki sebanyak{0}.", sapiSaya.jumlahKaki);
            console.WriteLine("Saat ini ia berumur {0}.", sapiSaya.umur);

            sapiSaya.Jalan();
            sapiSaya.Makan();
            sapiSaya.Bobo();
        }
    }
}
```

```
Nama sapi saya adalah Unyil  
Ia memiliki kaki sebanyak 4  
Saat ini ia berumur 10  
Unyil sedang jalan  
Unyil sedang makan  
Unyil sedang bobo
```