

Pengenalan Pemrograman Berbasis Objek dengan Dart

Fadjar Fathurrahman

1 Tujuan

- Mengetahui konsep dasar pemrograman berbasis objek dan implementasinya dalam Dart.
- Mampu membuat program sederhana berdasarkan pemrograman berbasis objek

2 Pemrograman berbasis objek

Pemrograman berbasis dan berorientasi objek (*object oriented programming*, sering disingkat OOP) adalah suatu paradigma pemrograman yang berpusat pada konsep "objek" yang secara garis besar terdiri dari:

- data, *fields*, atribut, atau *properties*
- prosedur, fungsi, atau metode

"Objek" dapat berupa abstraksi dari suatu objek fisik (benda) atau nonfisik yang dapat ditemukan pada dunia nyata, seperti manusia, mahasiswa, hewan, tumbuhan, bangunan, universitas, sekolah, sensor, pabrik, dan sebagainya.

Terdapat banyak bahasa pemrograman yang mendukung OOP seperti Java, C++, Python, C#, dan Dart. Sebagian besar bahasa pemrograman tersebut mengimplementasikan OOP dengan menggunakan *class*, yang dapat dianggap sebagai cetak biru (blue print) dari sebuah objek. Dalam hal ini, objek dikatakan sebagai *instance* dari sebuah *class*. Kumpulan dari data dan prosedur dari sebuah objek yang didefinisikan pada suatu *class* sering juga disebut dengan *class members*.

2.1 Contoh sederhana

Sebagai contoh, kita ingin mendeskripsikan suatu objek kucing yang akan diimplementasikan dalam suatu kelas dengan nama Kucing. Kelas Kucing akan memiliki field berupa nama, warna, dan umur.

```
class Kucing {  
    String nama;  
    String warna;  
    int umur;  
}
```

Untuk membuat suatu instance dari Kucing, kita dapat menggunakan kode sebagai berikut:

```
Kucing kucingku = Kucing();
```

Kemudian, kita dapat mengakses member dari kucingku sebagai berikut:

```
kucingku.nama = 'Ronaldo';
kucingku.warna = 'putih';
kucingku.umur = 2;
```

Sekarang kucingku dapat dianggap suatu variabel dan dapat digunakan dalam kode Dart yang lain. Misalnya kita dapat mendeklarasikan suatu fungsi dengan argumen bertipe Kucing.

```
void sayHello(Kucing k) {
  print('Miaw, namaku adalah ${k.nama}');
}
```

2.2 Konstruktor dan member function

Selain dengan cara manual, kita dapat menginisialisasi suatu objek dengan menggunakan metode atau fungsi khusus yang dinamai dengan konstruktor. Dart menyediakan sintaks khusus untuk konstruktor. Program diatas dapat ditulis sebagai berikut.

```
class Kucing {
  String nama;
  String warna;
  int umur;

  Kucing(this.nama, this.warna, this.umur); // konstruktor
}

void sayHello(Kucing k) {
  print('Miaw, namaku adalah ${k.nama}');
}

void main() {
  Kucing kucingku = Kucing('Ronaldo', 'hitam', 1);
  sayHello(kucingku);
}
```

Sintaks yang digunakan pada konstruktor di atas merupakan versi ringkas dari konstruktor berikut.

```
Kucing(String nama, String warna, int umur) {
  this.nama = nama;
  this.warna = warna;
  this.umur = umur;
}
```

Kita juga dapat membuat fungsi sayHello menjadi fungsi member.

```
class Kucing {
  String nama;
  String warna;
  int umur;

  Kucing(this.nama, this.warna, this.umur);

  void sayHello() {
    print('Miaw, namaku adalah ${this.nama}');
  }
}
```

```

}

void main() {
    Kucing kucingku = Kucing('Ronaldo', 'hitam', 1);
    kucingku.sayHello();
}

```

Kelas Kucing dapat digunakan untuk membuat objek atau instance yang lain. Misalnya:

```

void main() {
    Kucing kucingku = Kucing('Ronaldo', 'hitam', 1);
    kucingku.sayHello();

    Kucing kucingmu = Kucing('Messi', 'putih', 2);
    kucingmu.sayHello();
}

```

2.3 Contoh lain

```

class Ayam {
    String nama;
    String warna;
    int umur;
    String gender;

    Ayam(String nama, String warna, int umur, String gender) {
        this.nama = nama;
        this.warna = warna;
        this.umur = umur;
        this.gender = gender;
    }

    void sayHello() {
        if(this.gender == 'jantan') {
            print('Kukuruyuk, namaku adalah ${this.nama}');
        }
        else if(this.gender == 'betina') {
            print('Petok petok, namaku adalah ${this.nama}');
        }
        else {
            print('Kukuruyuk petok, namaku adalah ${this.nama}');
        }
    }
}

void main() {
    Ayam ayam1 = Ayam('Andre', 'hitam', 1, 'jantan');
    ayam1.sayHello();

    Ayam ayam2 = Ayam('Nunung', 'hitam', 1, 'betina');
    ayam2.sayHello();
}

```

```
Ayam ayam3 = Ayam('XXX', 'hitam', 1, 'lainnya');
 ayam3.sayHello();
}
```

2.4 Parameter fungsi opsional dengan kata kunci

```
void myFunc({int a: 2, int b: 3, String name: 'Sule'}) {
    print('a = $a');
    print('b = $b');
    print('name = $name');
}

void main() {
    myFunc();
    myFunc(a: 123);
    myFunc(name: 'Andre');
}
```

2.5 Inheritance

Pada banyak kasus kita dapat menemukan bahwa terdapat suatu hierarki pada banyak *class* yang kita buat. Misalnya, kita dapat menggabungkan kelas Cat dan Chicken ke dalam suatu *super-class* atau *parent class* Animal. Dalam kasus ini dikatakan bahwa Cat dan Chicken mewarisi atau *inherited from* Animal

```
class Animal {
    String name;
    int legCount;
}

class Cat extends Animal {
    String makeNoise() {
        print('Miaw');
    }
}

class Chicken extends Animal {
    String makeNoise() {
        print('Kukuruyuk');
    }
}

void main() {
    Cat cat = Cat();
    cat.name = 'Nora';
    cat.legCount = 4;
    print(cat);
    cat.makeNoise();

    Chicken chicken = Chicken();
    chicken.name = 'Kukuk';
    chicken.legCount = 2;
    print(chicken);
    chicken.makeNoise();
}
```

3 Tugas

Buatlah sebuah program untuk mengolah nilai mahasiswa. Nilai mahasiswa terdiri dari komponen nilai UTS, nilai UAS, dan nilai praktikum. Nilai akhir dihitung dari hasil pembobotan komponen tiga komponen nilai tersebut (Anda dapat menentukan sendiri). Nilai akhir juga dapat dikonversi menjadi dalam skala huruf seperti A, AB, B, BC, C, dan E (Anda juga dapat menentukan aturan konversi yang diperlukan). Data nilai mahasiswa dibaca melalui suatu file CSV (comma-separated values):

```
Andre;A19001;90;80;85  
Brewok;A19002;60;50;70  
Cucun;A19003;70;80;80  
Dodo;A19004;70;80;90  
.... dan seterusnya
```

Program ini dapat dibuat dengan menggunakan paradigma non-OOP, meskipun demikian kita akan menggunakan OOP untuk membuat program ini. Anda perlu mendefinisikan kelas Mahasiswa beserta membernya untuk keperluan ini.