

Hari-8-Object Oriented 2

Inheritance

Inheritance adalah konsep OOP dimana sebuah class dapat menurunkan **property** dan **method** yang dimilikinya kepada **class** lain. Konsep **inheritance** dipakai untuk memanfaatkan fitur *code reuse*, yakni menghindari terjadinya duplikasi kode program.

Dalam bahasa Indonesia, inheritance ini disebut juga sebagai pewarisan atau penurunan.

Konsep *inheritance* membuat sebuah struktur atau hierarchy **class** dalam kode program. Class yang akan diturunkan bisa disebut sebagai **class induk (parent class)**, **super class**, atau **base class**.

Sedangkan class yang menerima penurunan bisa disebut sebagai **class anak (child class)**, **sub class**, **derived class** atau **heir class**.

Tidak semua property dan method class induk akan diturunkan. Property dan method dengan hak akses **private** tidak akan diturunkan kepada class anak. Hanya property dan method dengan hak akses **protected** dan **public** saja yang bisa diakses dari class anak.

contoh:

main.dart

```
import 'godzilla.dart';
import 'ultramen.dart';

Run | Debug
void main(List<String> args) {
  Ultramen u = Ultramen();
  Godzilla g = Godzilla();

  u.levelPoint = 8;
  g.levelPoint = 10;

  print("level point ultramen: ${u.levelPoint}");
  print("level point godzilla: ${g.levelPoint}");

  print("sifat dari ultramen: "+ u.defance());
  print("sifat dari godzilla: "+g.attack());
}
```

character.dart

```
class Character {
  int _levelPoint;

  int get levelPoint => _levelPoint;
  set levelPoint(int value){
    if(value <=5){
      value = 5;
    }
    _levelPoint = value;
  }
}
```

godzilla.dart

```
class Godzilla extends Character {  
  String attack()=>"puch punch punch";  
}
```

ultramen.dart

```
import 'character.dart';  
  
class Ultramen extends Character {  
  String defance() => "dush dush dush";  
}
```

Jdi yang dimaksud dengan inheritance dari class ultramen dan godzilla adalah extends, dimana class ultramen dan godzilla mewarisi sifat dari character yang dimana character memiliki object berupa level point sehingga saat dikumpulkan di main.dart secara default godzilla dan ultramen dari mengisi dari levelPoint yang ada pada parentnya yaitu character.

Polymorism

Polymorphism adalah konsep di mana suatu objek yang berbeda-beda dapat diakses melalui interface yang sama. Sebuah objek yang polymorphic dapat beradaptasi dengan metode apapun yang diimplementasikan pada objek tersebut, dan setiap class memiliki interpretasinya tersendiri terhadap interfacenya.

contoh :

main.dart

```
import 'operasi_math.dart';  
import 'pembagian.dart';  
import 'perkalian.dart';  
  
Run | Debug  
void main(List<String> args) {  
  OperasiMath operasiMath = new OperasiMath();  
  Perkalian perkalian = new Perkalian(4, 7);  
  Pembagian pembagian = new Pembagian(10, 2);  
  
  operasiMath.convert();  
  operasiMath.convert();  
  
  print("Perkalian: ${perkalian.convert()}");  
  print("Pembagian: ${pembagian.convert()}");  
}
```

operasi_math.dart

```
class OperasiMath {  
  double convert(){  
    print("menghitung operasi matematika");  
    return 0;  
  }  
}
```

perkalian.dart

```

class Perkalian extends OperasiMath {
  double a;
  double b;

  Perkalian (double a, double b){
    this.a = a;
    this.b = b;
  }

  @override
  double convert(){
    return a * b;
  }
}

```

pembagian.dart

```

import 'operasi_math.dart';

class Pembagian extends OperasiMath {
  double aa;
  double bb;

  Pembagian (double aa, double bb){
    this.aa = aa;
    this.bb = bb;
  }

  @override
  double convert(){
    return aa / bb;
  }
}

```

Constructor

Constructor adalah method khusus yang akan dijalankan secara otomatis pada saat sebuah objek dibuat (*instansias*), yakni ketika perintah “**new**” dijalankan.

Constructor biasa digunakan untuk membuat proses awal dalam mempersiapkan objek, seperti memberi nilai awal kepada property, memanggil method internal dan beberapa proses lain yang digunakan untuk mempersiapkan objek.

main.dart

```

import 'mahasiswa.dart';

Run | Debug
main(List<String> args) {
  var myData01 = new Mahasiswa.nim(12);
  var myData02 = new Mahasiswa.name("Achmad Hilmy Firdaus");
  var myData03 = new Mahasiswa.jurusan("Sistem Komputer");

  print(myData01.nim);
  print(myData02.name);
  print(myData03.jurusan);
}

```

mahasiswa.dart

```
String name;  
String jurusan;  
  
Mahasiswa.nim(this.nim);  
Mahasiswa.jurusan(this.jurusan);  
Mahasiswa.name(this.name);  
}
```

Rating - Feedback

Berikan Rating pada posting ini:



Berikan kritik dan saran..

Submit