Course-Net
Empowering IT People

Object Oriented Programming

Course-Net
Empowering IT People

EmpoweringITPeople

Agenda

1. Pengenalan OOP
2. Perbedaan class dan object
3. Teori- teori OOP

Course-Net
Empowering IT People

#EmpoweringITPeople

Object Oriented Programming

OOP adalah paradigma pemrograman dengan menggunakan object.

Istilah dalam OOP:

1. Class: Blue print dari object
2. Key/Attribute dan Method: Berfungsi sama seperti key dan method dalam object
3. Instance: Merupakan hasil dari object yang sudah jadi dari class

Course-Net
Empowering IT People

#EmpoweringITPeople

Apa itu class?

Class adalah blueprint dari object.

Contoh: Denah rumah, tetapi belum jadi rumah secara fisik.

Class merupakan bagian dari ES6.

Dalam class terdapat attribute/key dan juga method sama seperti object.

Course-Net
Empowering IT People

#EmpoweringITPeople

Perbedaan Class dan object

console.log(hello);
Var hello;

```
let vehicle = {
    name : "Civic",
    type : "Sedan",
    price : 1000000,

    startEngine : function(){
        console.log("Start engine");
    },

    stopEngine : function(){
        console.log("Stop engine");
    }
}
```

```
class Vehicle{
    constructor(name, type, price){
        this.name = name;
        this.type = type;
        this.price = price;
    }

    startEngine(){
        console.log("Start engine")
    }
    stopEngine(){
        console.log("Stop engine")
    }
}
```

Instance
merupakan hasil jadi dari sebuah class

```
class Vehicle{
    constructor(name, type, price){
        this.name = name;
        this.type = type;
        this.price = price;
    }

    startEngine(){
        console.log("Start engine")
    }
    stopEngine(){
        console.log("Stop engine")
    }
}

let vehicle = new Vehicle("Civic","Sedan",1000000)
```

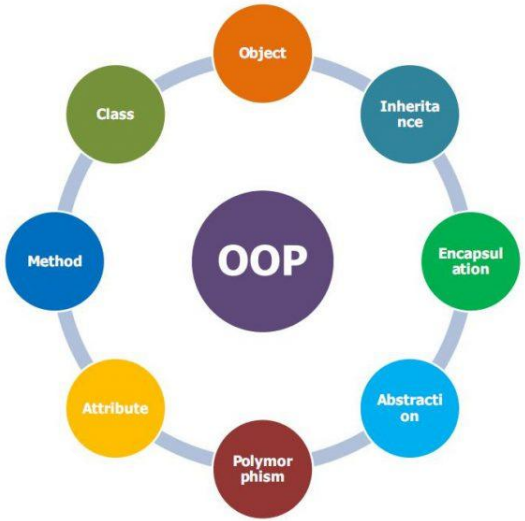
```
// hasil jadi
Vehicle {
 name: “Civic”
 type: “Sedan”,
 price: 1000
}

Typeof vehicle
// obj
```

Kelebihan OOP:

1. Parallel Development
2. Reusable
3. Scalability

Kekurangan OOP:

1. Tidak efisien
2. Kemungkinan duplikasi



Inheritance

Dapat menurunkan sifat class

Encapsulation

Class memiliki access: public, protected, private

Polymorphism

Akan menghasilkan array baru dari hasil proses

Abstract

Tak perlu mengetahui proses, cukup menggunakan

Inheritance

- *Inheritance* dalam konsep OOP adalah kemampuan untuk membentuk *class* baru yang memiliki fungsi turunan atau mirip dengan fungsi yang ada sebelumnya.
- Konsep ini menggunakan sistem hierarki atau bertingkat.
- Maksudnya, semakin jauh turunan atau *subclass*-nya, maka semakin sedikit kemiripan fungsinya.

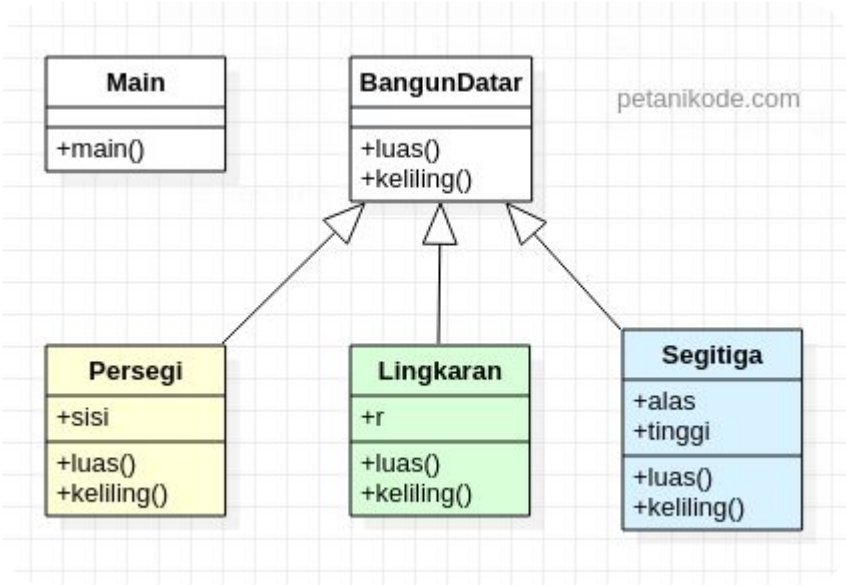
```
class Person {
    constructor(name, age) {
        this.name = name;
        this.age = age;
    }
}

class Programmer extends Person {
    constructor(name, age) {
        super(name, age)
    }
}
```

Course-Net
Empowering IT People

#EmpoweringITPeople

Polymorphism

- *Polymorphism* adalah konsep di mana suatu objek yang berbeda-beda dapat diakses melalui *interface* yang sama.
- Sebagai contoh, kamu memiliki fungsi untuk menghitung luas suatu benda, sementara benda tersebut berbentuk segitiga, lingkaran, dan persegi. **Tentu, ketiga benda tersebut memiliki rumus perhitungan tersendiri.**
- Dengan *polymorphism*, kamu dapat memasukkan fungsi perhitungan luas ketiga benda tersebut, dengan tiap benda memiliki metode perhitungan berbeda.
- Ini tentu akan mempermudah perintah yang sama untuk beberapa *class.*



Course-Net
Empowering IT People

#EmpoweringITPeople

Encapsulation

Encapsulation atau pengkapsulan adalah konsep tentang pengikatan data atau metode berbeda yang disatukan atau “dikapsulkan” menjadi satu unit data.

Terdapat 3 jenis akses :

- Public
- Protected
- Private

```
class Student {
    constructor(name, gpa) {
        this._name = name;
        this._gpa = gpa;
    }
    get name() {
        return this._name;
    }
    get gpa() {
        return this._gpa;
    }
    set setName(name) {
        this._name = name;
    }
    set setGpa(gpa) {
        this._gpa = gpa;
    }
}
```

Course-Net
Empowering IT People

#EmpoweringITPeople

Abstract

Abstraction adalah menghilangkan informasi yang tidak perlu untuk user.