



# Pemrograman Framework

OOP Dalam Pemrograman Web

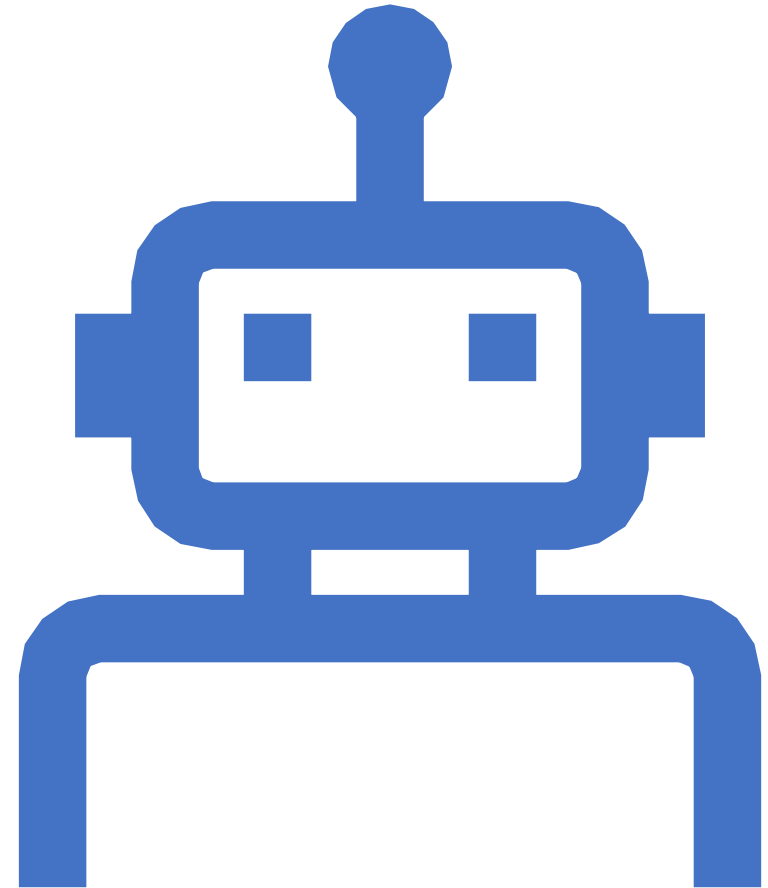
Galet Guntoro Setiaji

# Sekilas tentang Object Oriented Programing (OOP)

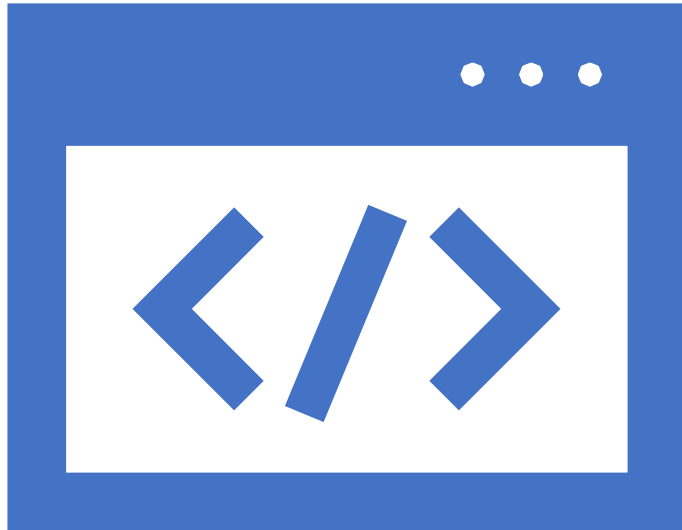
Pemograman Berorientasi Object atau dalam bahasa inggris lebih dikenal dengan Object Oriented Programming (OOP) adalah sebuah paradigma dalam pemograman yang menyelesaikan masalah program dengan menyediakan objek-objek(terdiri dari beberapa attribute dan method) yang saling berkaitan dan disusun kedalam satu kelompok atau yang disebut dengan class. Nantinya objek-objek tersebut akan saling berinteraksi untuk menyelesaikan masalah program yang rumit

# Alasan menggunakan konsep object oriented:

- Pengembangan perangkat lunak yang sulit.
- Kompleksitas pengembangan perangkat lunak yang terus bertumbuh membutuhkan dukungan konsep yang lebih handal, guna ulang (reusable) dan natural.
- OO menawarkan tipe data abstrak, modularitas, pemodelan informasi, proses software untuk mengatasinya.
- Walaupun demikian, OO bukan jaminan sukses pengembangan perangkat lunak



# Bahasa pemrograman yang mendukung OOP :



- Java
- C++ ( Dikembangkan dari bahasa C )
- Visual Basic.NET
- SIMULA
- Smalltalk
- Ruby
- Python
- PHP
- Delphi ( Dikembangkan dari bahasa PASCAL )
- Eiffel
- Perl

# Mengenal Konsep Object Oriented Programming

Objek dan  
Class

Encapsulation

Inheritance

Polymorphism

# Apa itu class dan objek?

Misalkan saat anda duduk dibangku sekolah, anda pasti ditempatkan dalam satu kelas yang berisikan siswa lainnya yang sama dengan anda. Kita dapat menyimpulkan bahwa kelas itu sama dengan class yang kita gunakan dalam OOP dan siswa-siswi yang berada dalam kelas tersebut sama dengan objek-objek yang ada didalam class itu juga.

# Encapsulation

Encapsulation adalah proses dalam menciptakan sebuah objek dimana terdapat beberapa bagian atau attribute yang terbagi berdasarkan sifat yakni public(umum) dan private (khusus). Sebagaimana yang kita ketahui dalam definisi OOP, objek akan selalu berinteraksi dengan objek lain untuk menyelesaikan masalah dengan mengakses proses dari objek tersebut. Untuk menghindari redundansi atau ketidakakuratan program, maka diberikan batasan akses setiap objek terhadap objek lain. Saat objek bisa mengakses method dari objek lain, maka kita menyebutnya method bersifat public. Sedangkan saat suatu attribute hanya mampu diakses oleh objek itu sendiri, maka kita menyebutnya sebagai attribute bersifat private

# Contoh Encapsulation

tubuh manusia yang terdiri dari beberapa class seperti tangan(hand), kaki(leg) dan sebagainya. Class hand terdiri dari beberapa objek seperti lengan, siku, dan jari. Kita lihat satu objek jari yang memiliki beberapa atribut private yaitu seperti tulang, pembuluh darah, dan kulit. Selain itu, jari juga memiliki aktivitas yang umum yaitu public method menggenggam(), menyentuh(), dan menulis (). Pemisahan tersebut merupakan tujuan utama dari encapsulation.



# Inheritance

Saat membuat objek-objek dalam sebuah class, mungkin kita sering menemukan sifat yang sama antar objek dan menulisnya berulang kali sebanyak objek yang memiliki sifat tersebut. Hal itu tentu terlalu merepotkan dan memerlukan waktu yang lebih lama. Oleh karena itu terdapat suatu konsep dalam OOP yang mampu membantu kita dalam mengatasi masalah tersebut. Inheritance merupakan hubungan antara dua objek atau lebih dimana akan terdapat sebuah objek utama yang mewariskan attribute atau method yang dimilikinya kepada objek lain, baik itu keseluruhan atau sebagian.

# Contoh Inheritance

Misalkan saat kita membuat sebuah sistem informasi kampus yang akan diakses oleh Dosen dan Mahasiswa. Kedua objek tersebut pasti memiliki atribut yang sama seperti nama, umur, alamat, dan lainnya. Untuk menghindari penulisan ulang ketiga atribut tersebut disetiap objek, ada baiknya kita membuat objek baru yang mampu menyimpan ketiga attribute tersebut dan nantinya akan diwariskan kepada objek Dosen dan Mahasiswa seperti gambar disamping. Lalu bagaimana objek Person mengetahui apakah objek Dosen atau objek Mahasiswa yang mengaksesnya? Untuk menghindari kebingungan saat diakses maka Objek Dosen dan Mahasiswa memiliki attribute pembeda yang didefinisikan dalam masing-masing objek yaitu IdDosen dan NIM mahasiswa. Pengaksesan ini juga erat kaitannya dengan Polymorphism dimana sebuah objek yang diakses menggunakan konsep inheritance juga memiliki attribute-attribute pembeda untuk mengedintifikasi setiap objek tersebut.

# Polymorphism

Prinsip terakhir dalam OOP adalah polymorphism. Pada dasarnya polymorphism adalah kemampuan suatu pesan atau data untuk diproses lebih dari satu bentuk. Salah satu ciri utama dari OOP adalah adanya polymorphism. Tanpa hal ini, suatu pemrograman tidak bisa dikatakan sebagai OOP. Polymorphism sendiri adalah konsep di mana suatu objek yang berbeda-beda dapat diakses melalui interface yang sama

## Contoh Polymorphism

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 ke tiga benda tersebut, dengan tiap benda memiliki metode perhitungannya sendiri. Ini tentu akan mempermudah perintah yang sama untuk beberapa class atau subclass tertentu.

# Contoh sederhana OOP dengan PHP

```
<?php
class mahasiswa{
    //property dari class mahasiswa
    var $nim;
    var $nama;
    //method dari class mahasiswa
    function mhs_progdi(){
        //isi method data_mhs
        return "Mahasiswa tersebut dari Progdi Teknik Informatika";
    }
}

//buat object class mahasiswa
$mhs = new mahasiswa();
//set property
$mhs->nim="G.231.20.0009";
$mhs->nama="Alexander Joseph Luthor";
//menampilkan property
echo "Nim : ".$mhs->nim."<br>";
echo "Atas nama : ".$mhs->nama."<br>";

//menampilkan method mhs_progdi()
echo $mhs->mhs_progdi();
?>
```