

# MATAKULIAH

## Pemrograman Web II

M. Khairul Anam, M.Kom



Pertemuan 12 – Pemrograman Berorientasi Objek

# Pengertian

- Secara sederhana, pemrograman berorientasi objek adalah sebuah cara penulisan kode program dengan menggunakan object untuk memecahkan masalah. Object ini bisa dianggap sebagai bagian dari kode program utama yang bisa berfungsi secara mandiri.
- Jika didefinisikan seperti ini, object berfungsi layaknya sebuah function. Seperti yang mungkin sudah anda pahami, function (atau fungsi) juga bagian dari kode program utama dan dipakai untuk memecahkan masalah tertentu.
- Tapi object jauh lebih powerful daripada function. Sebuah object nantinya bisa diturunkan kepada object lain, bisa di-setting *hak aksesnya*, memiliki *property* dan *method*, serta berbagai fitur lain.

# Fungsi

- pembuatan program dalam PHP tanpa memerlukan objek disebut dengan pemrograman prosedural atau pemrograman fungsional. Disebut pemrograman prosedural, sebab memecah kode program menjadi bagian-bagian atau fungsi kecil dan menyatukan guna mencapai nilai akhir.
- Dengan membuat program prosedural, maka aplikasi dapat dibuat dengan mudah untuk dipelajari dibandingkan dengan pemrograman yang berbasis objek. Namun, keuntungan OOP akan terasa jika program tersebut sudah besar atau bekerja dengan tim dalam membagi tugas.
- Konsep objek untuk memisahkan program menjadi bagian-bagian yang dapat berdiri sendiri akan memudahkan dalam pembuatan program.

## Kelebihan

- Kelebihan dari metode pemrograman berbasis objek diantaranya, maintenance program lebih mudah. Program yang dibuat akan mudah dibaca dan mudah dipahami. Selain itu, dapat mengontrol kesulitan dalam program hanya dengan cara mengizinkan rincian yang dibutuhkan programmer.
- Kelebihannya lainnya, mudah dalam pengubahan program, berupa penambahan atau penghapusan fitur objek tertentu. Perubahan yang dapat dilakukan, misalnya penambahan dan penghapusan data pada suatu database.



# Basic OOP PHP

## Class dan Object

- Class dan object merupakan fondasi paling dasar dari *object oriented programming*, keduanya serupa tapi tak sama. Class adalah *blueprint* atau "*cetakan*" untuk *object*. Bisa disebut juga bahwa object adalah implementasi konkret dari sebuah class.

## Property dan Method

- Property dan method tidak lain adalah sebutan untuk *variabel dan function yang berada di* dalam class. Cara penulisannya pun sama seperti variabel dan function, tapi dengan tambahan *access modifier di awal penulisan*

## Basic yang lainnya

- Pseudo-variable \$this
- Argument Method
- Constructor dan Destructor
- Inheritance
- Visibility
- Getter dan setter
- Static property dan static method
- Class constant
- Constructor property promotion

# Advanced OOP PHP

- Mayoritas materi yang akan kita pelajari mungkin tidak akan sering dipakai, tapi tetap saja ini merupakan bagian dari object oriented programming di PHP.
- Sebagai contoh, kecil kemungkinan anda butuh abstract class ketika membuat website system informasi sekolah. Materi seperti ini baru akan terpakai jika membuat website sekelas traveloka (dengan tim tentunya).
- Materi *advanced OOP* ini bukanlah sebuah keharusan (*tidak harus dipakai*), tapi lebih ke perancangan sistem. Dalam konsep OOP ada yang dinamakan "design pattern", yakni pola penyusunan class untuk membuat efek tertentu. Teknik *design pattern* ini juga butuh pemahaman mendalam tentang object oriented programming.



## Abstract Class

- Abstract class adalah class khusus yang berfungsi sebagai class dasar (*base class*) untuk class turunan. Menggunakan abstract class, kita bisa "memaksa" class turunan agar mengimplementasikan method tertentu.
- Tujuan akhir dari abstract class adalah konsep OOP yang dikenal sebagai polymorphism (*polimorfisme*). Secara sederhana, *polymorphism dipakai agar semua class turunan memiliki nama method yang sama meskipun method tersebut melakukan hal yang berbeda.*

# Interface

- Interface adalah sebuah mekanisme di dalam OOP PHP untuk membuat *'kontrak' atau perjanjian implementasi method. Interface bisa juga disebut sebagai bentuk yang lebih khusus dari abstract class.*
- Secara garis besar, interface berfungsi sama seperti abstract class, yakni berisi kumpulan *signature method yang harus di implementasikan. Setiap class yang menggunakan interface akan "dipaksa" membuat ulang method tersebut.*

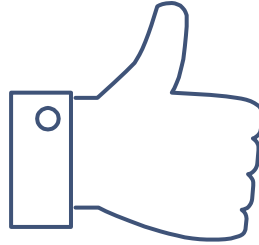


## Advance lainnya

Magic constant

- Magic method
- Perbandingan object
- Object clone
- Methone chaining
- Type hinting
- Typed properties
- Nullsafe operator
- Late static binding





# THANKS!

Ada Pertanyaan?  
Boleh juga ke  
[khairulanam@sar.ac.id](mailto:khairulanam@sar.ac.id)