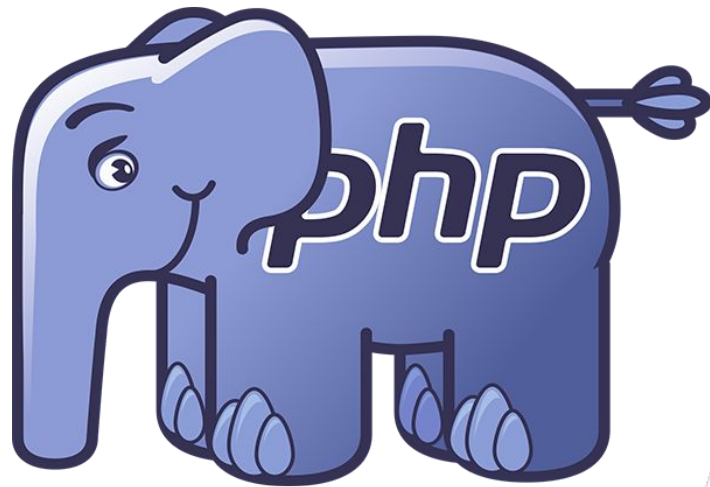


# PHP Pertemuan ke-3

Object Oriented Programming



# Apa yang akan kita pelajari hari ini?

- Object Oriented Programming
- Class and Object
- Inheritance
- Static properties and methods





# "Object Oriented" . " Programming"



# Apa itu Object Oriented Programming?

Object-Oriented Programming (OOP) adalah paradigma pemrograman yang memungkinkan Kita untuk mengorganisir kode ke dalam objek-objek yang terdiri dari data dan fungsi-fungsi terkait.

Pemrograman berorientasi objek memiliki beberapa keuntungan dibandingkan dengan pemrograman prosedural:

1. OOP lebih cepat dan lebih mudah untuk dieksekusi.
2. OOP menyediakan struktur yang jelas untuk program-program.
3. OOP membantu menjaga kode PHP tetap DRY "Don't Repeat Yourself", dan membuat kode lebih mudah untuk dipelihara, dimodifikasi, dan didebug.
4. OOP memungkinkan pembuatan aplikasi yang dapat digunakan kembali secara penuh dengan kode yang lebih sedikit dan waktu pengembangan yang lebih singkat.



**"Class and" .  
" Object"**



# Class dan Object

Kelas adalah sebuah template atau blueprint yang digunakan untuk menciptakan objek. Dalam kelas, Kita mendefinisikan properties (variabel) dan methods (fungsi) yang akan dimiliki oleh objek yang dibuat berdasarkan kelas tersebut.

Objek adalah instance konkret dari suatu kelas. Ketika Kita menciptakan objek, Kita membuat salinan dari kelas dengan properties dan methods yang telah ditentukan. Dalam hal ini, objek dapat dianggap sebagai realisasi nyata dari kelas. Kita dapat membuat banyak objek dari satu kelas yang sama, dan setiap objek dapat memiliki keadaan yang berbeda.

# Properties and methods

Properties adalah variabel yang terkait dengan objek. Mereka mewakili data yang dimiliki oleh objek. Misalnya, properti "merk" pada objek "Mobil" dapat berisi nilai seperti "Mercedes" atau "Toyota".

Methods adalah fungsi-fungsi yang terkait dengan objek. Mereka digunakan untuk melakukan operasi tertentu pada objek atau memanipulasi data yang dimiliki oleh objek. Misalnya, metode "hidupkanMesin" pada objek "Mobil" akan menghidupkan mesin mobil.

# Access Modifier

Access modifier (hak akses) digunakan untuk mengatur tingkat visibilitas (keterlihatan) properti dan metode dalam sebuah class. Dalam PHP, terdapat tiga jenis access modifier yang dapat digunakan:

1. **Public:** Properti atau metode dengan access modifier public dapat diakses dari mana saja, baik dari dalam class itu sendiri, class turunan, atau dari luar class melalui objek.
2. **Protected:** Properti atau metode dengan access modifier protected hanya dapat diakses dari dalam class itu sendiri dan class turunannya. Properti atau metode yang bersifat protected tidak dapat diakses langsung dari luar class.
3. **Private:** Properti atau metode dengan access modifier private hanya dapat diakses dari dalam class itu sendiri. Properti atau metode yang bersifat private tidak dapat diakses dari class turunannya atau dari luar class melalui objek.



# Class and Object

## Class Kucing

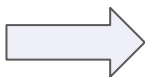


### Properties:

- warna
- corak
- warna mata

### methods:

- meow()
- makan()
- tidur()



## Oyen



### Properties:

- warna = "oren"
- corak = "belang"
- warna\_mata = "kuning"

### methods:

- meow()
- makan()
- tidur()

## Bleki



### Properties:

- warna = "hitam"
- corak = "polos"
- warna\_mata = "kuning"

### methods:

- meow()
- makan()
- tidur()

# Class and Object

## ●●● class\_and\_object.php

```
<?php
class Kucing {
    public $warna;
    public $corak;
    public $warna_mata;

    public function meow() {
        echo "Meow...";
    }

    public function makan() {
        echo "Kucing sedang makan.";
    }

    public function tidur() {
        echo "Kucing sedang tidur.";
    }
}
```

## ●●● class\_and\_object.php

```
$soyen = new Kucing();
$soyen->warna = "oren";
$soyen->corak = "belang";
$soyen->warna_mata = "kuning";

$soyen->meow();
$soyen->makan();
$soyen->tidur();

$bleki = new Kucing();
$bleki->warna = "hitam";
$bleki->corak = "polos";
$bleki->warna_mata = "kuning";

$bleki->meow();
$bleki->makan();
$bleki->tidur();
```

# Constructor

Constructor / Konstruktor adalah metode khusus dalam PHP yang secara otomatis dipanggil saat sebuah objek baru dibuat dari sebuah class. Metode konstruktor biasanya digunakan untuk melakukan inisialisasi awal terhadap properti objek.

Dalam PHP, konstruktor dibuat menggunakan `__construct()`.

# Constructor

## ●●● class\_and\_object.php

```
<?php
class Kucing {
    public $warna;
    public $corak;
    public $warna_mata;

    public function __construct($warna, $corak, $warna_mata){
        $this->warna = $warna;
        $this->corak = $corak;
        $this->warna_mata = $warna_mata;
    }
    public function meow() { echo "Meow..."; }

    public function makan() { echo "Kucing sedang makan."; }

    public function tidur() { echo "Kucing sedang tidur."; }
}
```

## ●●● class\_and\_object.php

```
$soyen = new Kucing("oren", "belang",
    "kuning");

$soyen->meow();
$soyen->makan();
$soyen->tidur();

$bleki = new Kucing("hitam", "polos",
    "kuning");

$bleki->meow();
$bleki->makan();
$bleki->tidur();
```



# "Inheritance"



# Inheritance

Inheritance (pewarisan) adalah salah satu konsep penting dalam pemrograman berorientasi objek (OOP) yang memungkinkan Kita untuk membuat class baru (child class) berdasarkan class yang sudah ada (parent atau superclass).

Child class mewarisi properti dan metode dari parent class, dan juga dapat menambahkan properti dan metode baru, serta mengubah perilaku yang sudah ada.

# Inheritance

● ● ● inheritance.php

```
<?php
class Hewan{
    protected $nama;

    public function __construct($nama){
        $this->nama = $nama;
    }

    public function makan() { echo "Hewan sedang makan."; }

    public function tidur() { echo "Hewan sedang tidur."; }
}
```

# Inheritance

## ● ● ● inheritance.php

```
<?php
class Kucing extends Hewan{
    public $warna;
    public $corak;
    public $warna_mata;

    public function __construct($nama, $warna, $corak,
$warna_mata){
        parent::__construct($nama);
        $this->warna = $warna;
        $this->corak = $corak;
        $this->warna_mata = $warna_mata;
    }
    public function meow() { echo "Meow..."; }

    public function makan() { echo "Kucing sedang makan."; }

    public function tidur() { echo "Kucing sedang tidur."; }
}
```

## ● ● ● inheritance.php

```
$oyen = new Kucing("oyen", "oren", "belang",
"kuning");

$oyen->meow();
$oyen->makan();
$oyen->tidur();

$bleki = new Kucing("bleki", "hitam",
"polos", "kuning");

$bleki->meow();
$bleki->makan();
$bleki->tidur();
```





# "Static Properties". "and Methods"



# Static Properties and Methods

Static properties and methods adalah properti dan metode yang terkait dengan class itu sendiri, bukan dengan instance objek yang dibuat dari class. Ini berarti, properti static dan metode static dapat diakses dan digunakan tanpa harus menciptakan objek dari class tersebut.

Nilai dari static properties bersifat bersama (shared) di antara semua instance objek yang dibuat dari class tersebut.

Untuk mendeklarasikan properti atau metode static, kita harus menggunakan keyword 'static'

# Static Properties and Methods

## ● ● ● static\_properties\_and\_methods.php

```
<?php
class Kucing {
    public $warna;
    public $corak;
    public $warna_mata;
    public static $jumlah_kucing = 0;

    public function __construct($warna, $corak, $warna_mata){
        self::$jumlah_kucing++;
        $this->warna = $warna;
        $this->corak = $corak;
        $this->warna_mata = $warna_mata;
    }
    public function meow() { echo "Meow..."; }

    public function makan() { echo "Kucing sedang makan."; }

    public function tidur() { echo "Kucing sedang tidur."; }
}
```

## ● ● ● static\_properties\_and\_meth..

```
$soyen = new Kucing("oren", "belang",
    "kuning");

$soyen->meow();
$soyen->makan();
$soyen->tidur();

$bleki = new Kucing("hitam", "polos",
    "kuning");

$bleki->meow();
$bleki->makan();
$bleki->tidur();

echo Kucing::$jumlah_kucing;
```



# "Ada Pertanyaan?"



# Jadi, Apa saja yang sudah kita pelajari?

**Object Oriented  
Programming**

**Class and Object**

**Inheritance**

**Static Properties and  
Methods**

# Tugas?

1. Buat file php baru di folder pertemuan kali ini “tugas.php”
2. di dalam tugas.php, buat class Kendaraan minimal memiliki 2 properti, 1 konstruktor, dan 1 method
3. masih di dalam tugas.php, buat class baru yang **mewarisi class Kendaraan** dengan nama yang kalian tentukan sendiri, minimal memiliki 1 properti baru, 1 konstruktor, dan 1 method
4. masih di dalam tugas.php, buat object berdasarkan class baru yang kalian buat

Push semua codenya ke GitHub



# "Thank You"

