

# Object Oriented Programming

**Taufiq Tirtajiwangga**

# Object Oriented Programming

**Object Oriented Programming** adalah paradigma pemrograman yang berfokus pada data atau objek, berbeda dengan **Procedure Oriented Programming** yang berfokus pada logika fungsi.

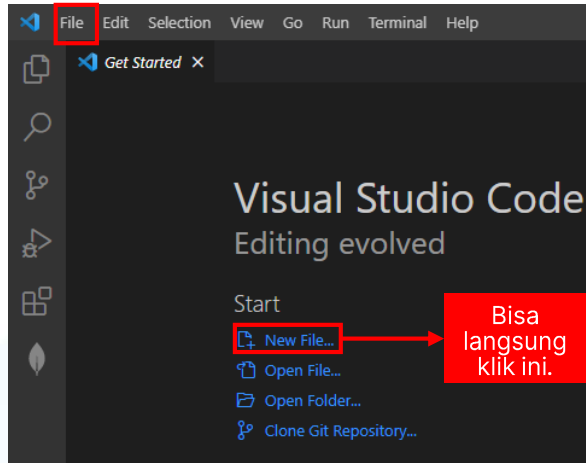
# Istilah Class-based OOP

1. Class
2. Object
3. Class Attribute dan Method
4. Class Method
5. Parent Class / Superclass dan Child Class / Subclass
6. Access Modifier
7. Constructor dan Destructor

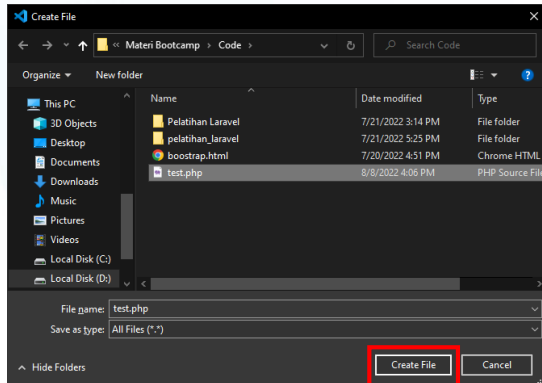
# Class

**Class** adalah tipe data yang ditentukan oleh user.

Hewan



Membuka VS Code kemudian membuat file **php** baru dengan klik menu **File** kemudian pilih **New File**. Ketikkan nama file yang akan dibuat sebagai contoh **test.php** kemudian enter.



Memilih direktori untuk menyimpan file **php** kemudian klik **Create File**.

test.php X

D: > INOSOFT > Bootcamp > Materi Bootcamp > Code > test.php > Kalkulator > tambah

```
1  <?php
2
3  class Kalkulator{
4      public function tambah(int $a, int $b){
5          return $a + $b;
6      }
7  }
8
9  $kalkulator = new Kalkulator();
10 echo $kalkulator->tambah(5, 19);
11
```

Class

# Object

**Object** adalah data nyata yang merujuk ke class.

Hewan  
(class)



**Sumber gambar:**

<http://keswan.ditjenpkh.pertanian.go.id/?p=825>

<https://www.harapahrakyat.com/2020/12/hewan-jarang-berwarna-biru-alami-ternyata-ini-penyebabnya/>

test.php

D: > INOSOFT > Bootcamp > Materi Bootcamp > Code > test.php > Kalkulator > tambah

```
1  <?php
2
3  class Kalkulator{
4      public function tambah(int $a, int $b){
5          return $a + $b;
6      }
7  }
8
9  $kalkulator = new Kalkulator();
10 echo $kalkulator->tambah(5, 19);
11
```

Object



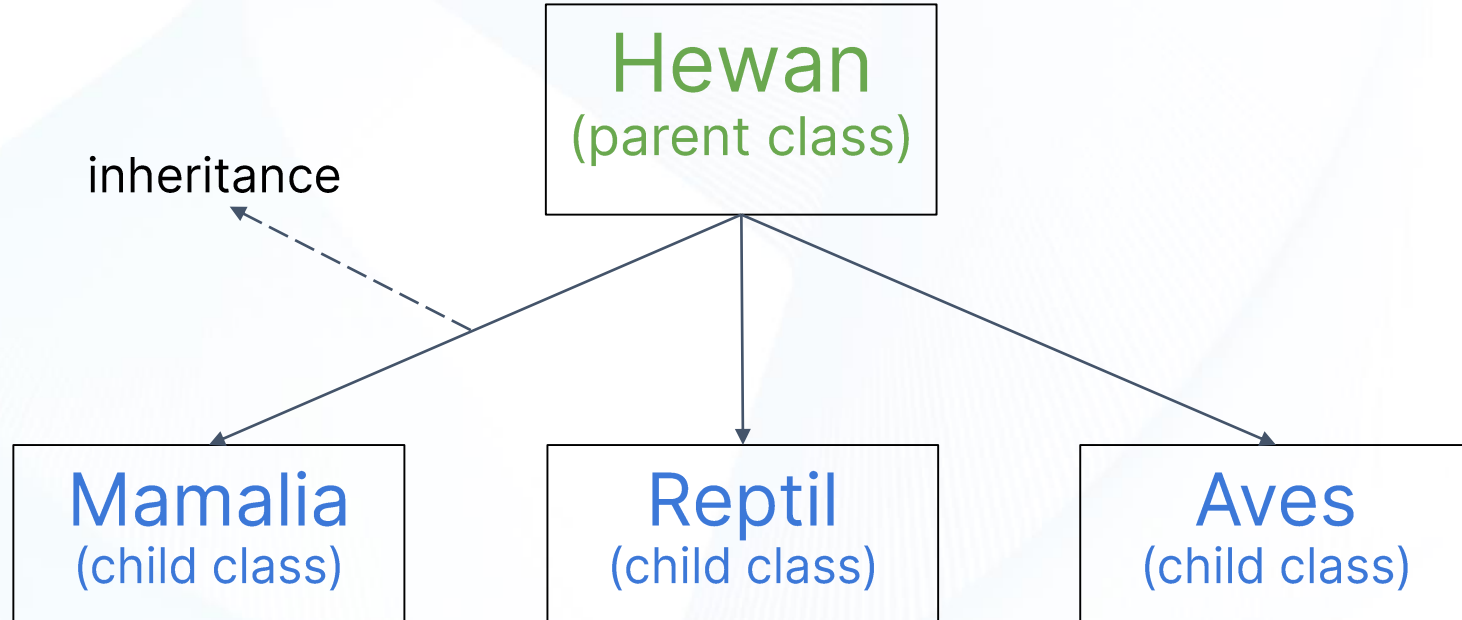
# Class Attribute dan Method



```
1  <?php
2
3  class Kalkulator{
4
5      public float $dayaBaterai;
6
7      public function __construct(float $dayaBaterai){
8          $this->dayaBaterai = $dayaBaterai;
9      }
10
11     public function tambah(int $a, int $b){
12         return $a + $b;
13     }
14 }
15
16 class KalkulatorMewah extends Kalkulator{
17
18 }
19
20 $kalkulator = new Kalkulator(77);
21 // echo $kalkulator -> dayaBaterai;
22 echo $kalkulator->tambah(5, 19);
```

Attribute

# Parent Class / Superclass dan Child Class / Subclass



```
1  <?php
2
3  class Kalkulator{
4
5      public float $dayaBaterai;
6
7      public function __construct(float $dayaBaterai){
8          $this->dayaBaterai = $dayaBaterai;
9      }
10
11     public function tambah(int $a, int $b){
12         return $a + $b;
13     }
14 }
15
16 class KalkulatorMewah extends Kalkulator{
17
18 }
19
20 $kalkulator = new Kalkulator(77);
21 // echo $kalkulator -> dayaBaterai;
22 echo $kalkulator->tambah(5, 19);
```

Parent Class

Child Class

# Access Modifier

**Access Modifier** adalah aksesibilitas class attribute / class method.

- **public** — attribute / method bisa diakses dari manapun (default access modifier di PHP)
- **protected** — attribute / method bisa diakses dari class yang bersangkutan dan child class nya
- **private** — attribute / method **hanya** bisa diakses dari class yang bersangkutan.

test.php

D: > INOSOFT > Bootcamp > Materi Bootcamp > Code > test.php > Kalkulator > tambah

```
1  <?php
2
3  class Kalkulator{
4      public function tambah(int $a, int $b){
5          return $a + $b;
6      }
7  }
8
9  $kalkulator = new Kalkulator();
10 echo $kalkulator->tambah(5, 19);
11
```

Access  
modifier

# Constructor dan Destructor

Dalam OOP dikenal 2 fungsi khusus dalam class, yaitu constructor dan destructor.

- Fungsi constructor dipanggil ketika objek dibuat (instansiasi)
- Fungsi destructor dipanggil ketika objek dihapus dari memory

```
1 <?php
2
3 class Test
4 {
5     public function __construct()
6     {
7         echo "construct\n";
8     }
9
10    public function __destruct()
11    {
12        echo "destruct\n";
13    }
14 }
15
16 new Test();
```

construct  
destruct

```
1  <?php
2
3  class Kalkulator{
4
5      public float $dayaBaterai;
6
7      public function __construct(float $dayaBaterai){
8          $this->dayaBaterai = $dayaBaterai;
9      }
10
11     public function tambah(int $a, int $b){
12         return $a + $b;
13     }
14 }
15
16 class KalkulatorMewah extends Kalkulator{
17
18 }
19
20 $kalkulator = new Kalkulator(77);
21 // echo $kalkulator -> dayaBaterai;
22 echo $kalkulator->tambah(5, 19);
```

Constructor



# Terima **Kasih**