

8. Object Oriented Programming (OOP)

Pemrograman PHP

Nasrul, S.Pd.I, S.Kom, M.Kom



- Dosen STT
- Nurul Fikri

Instruktur IT
NF Computer

Instruktur IT
Sekolah
Programmer YBM
PLN

Instruktur IT
Fast Com

 nasrul99@gmail.com

 [nasrul tutorial](#)

Tujuan

Setelah mempelajari materi ini peserta diharapkan memahami dan mengerti tentang Object Oriented Programming di dalam bahasa pemrograman PHP serta dapat mengimplementasikannya pada aplikasi

Class & Object

Seorang arsitek ketika membangun rumah, ia akan menggambar atau menuangkan ide bentuk dan arsitektur rumah dalam sebuah blue print. Pada terminologi OOP, blue print dari rumah disebut dengan class dan rumah yang dibangun berdasarkan blue print disebut object.



Member Class

- **Variable : Data**

- Contoh : norek, saldo, nasabah

- **Method : Behavior**

- Void
- Return value
- * Special Method : Constructor

Modifier Class

Akses Modifiers : adalah akses kontrol yang mungkin dipakai pada class, variabel, fungsi dan konstruktor. Berikut ini jenis-jenis modifier class:

- public : akses kontrol yang bisa diakses di dalam dan luar class
- protected: akses kontrol yang bisa diakses di dalam class dan class turunannya (inheritance)
- private: akses kontrol yang hanya bisa diakses di dalam class nya saja

Membuat Class#1

```
<?php
class Bank{
    //member1 variabel
    protected $norek;
    public $nama;
    private $saldo;
    //variabel static dan konstanta di dlm class
    static $jml = 0;
    const BANK = 'Bank Syariah Nurul Fikri';

    //member2 konstruktor
    public function __construct($no, $nasabah, $saldo)
    {
        $this->norek = $no;
        $this->nama = $nasabah;
        $this->saldo = $saldo;
        self::$jml++;
    }
}
```

Membuat Class#2

```
//member3 method
public function setor($uang){
    //$this->saldo = $this->saldo + $uang;
    $this->saldo += $uang;
}

public function ambil($uang){
    //$this->saldo = $this->saldo - $uang;
    $this->saldo -= $uang;
}

public function cetak(){
    echo '<b><u>'.self::BANK.'</u></b>';
    echo '<br/>No. Rekening :'. $this->norek;
    echo '<br/>Nama Nasabah :'. $this->nama;
    echo '<br/>Saldo : Rp. '.number_format($this->saldo,0, ',', '.');
    echo '<br/>';
}
```


Menciptakan Object#1

```
<?php  
require 'Bank.php';  
  
//ciptakan object  
$n1 = new Bank('001', 'Siti', 5000000);  
$n2 = new Bank('002', 'Andi', 7000000);  
$n3 = new Bank('003', 'Dede', 3000000);  
$n4 = new Bank('004', 'Didin', 1000000);
```

Menciptakan Object#2

```
//panggil fungsi2
```

```
$n1->setor(4000000);
```

```
$n3->setor(1000000);
```

```
$n1->ambil(6000000);
```

```
$n2->ambil(2000000);
```

```
echo '<h3 align="center">' . Bank::BANK . '</h3>';
```

```
$n1->cetak();
```

```
$n2->cetak();
```

```
$n3->cetak();
```

```
echo 'Jumlah Nasabah : ' . Bank::$jml . ' orang';
```

Mencetak Data dari Object

← → ↻ ⓘ localhost:8484/belajarphp/transaksi.php	
Bank Syariah Nurul Fikri	
<u>Bank Syariah Nurul Fikri</u> No. Rekening :001 Nama Nasabah :Siti Saldo : Rp. 3.000.000	
<u>Bank Syariah Nurul Fikri</u> No. Rekening :002 Nama Nasabah :Andi Saldo : Rp. 5.000.000	
<u>Bank Syariah Nurul Fikri</u> No. Rekening :003 Nama Nasabah :Dede Saldo : Rp. 4.000.000	
<u>Bank Syariah Nurul Fikri</u> No. Rekening :004 Nama Nasabah :Didin Saldo : Rp. 1.000.000	
Jumlah Nasabah :4 orang	

Latihan Membuat Class Gempa#1

```
<?php
class Gempa{
    //member1 variabel
    private $lokasi;
    private $skala;

    //member2 konstruktor
    public function __construct($lokasi, $skala)
    {
        $this->lokasi = $lokasi;
        $this->skala = $skala;
    }
}
```

Latihan Membuat Class Gempa#2

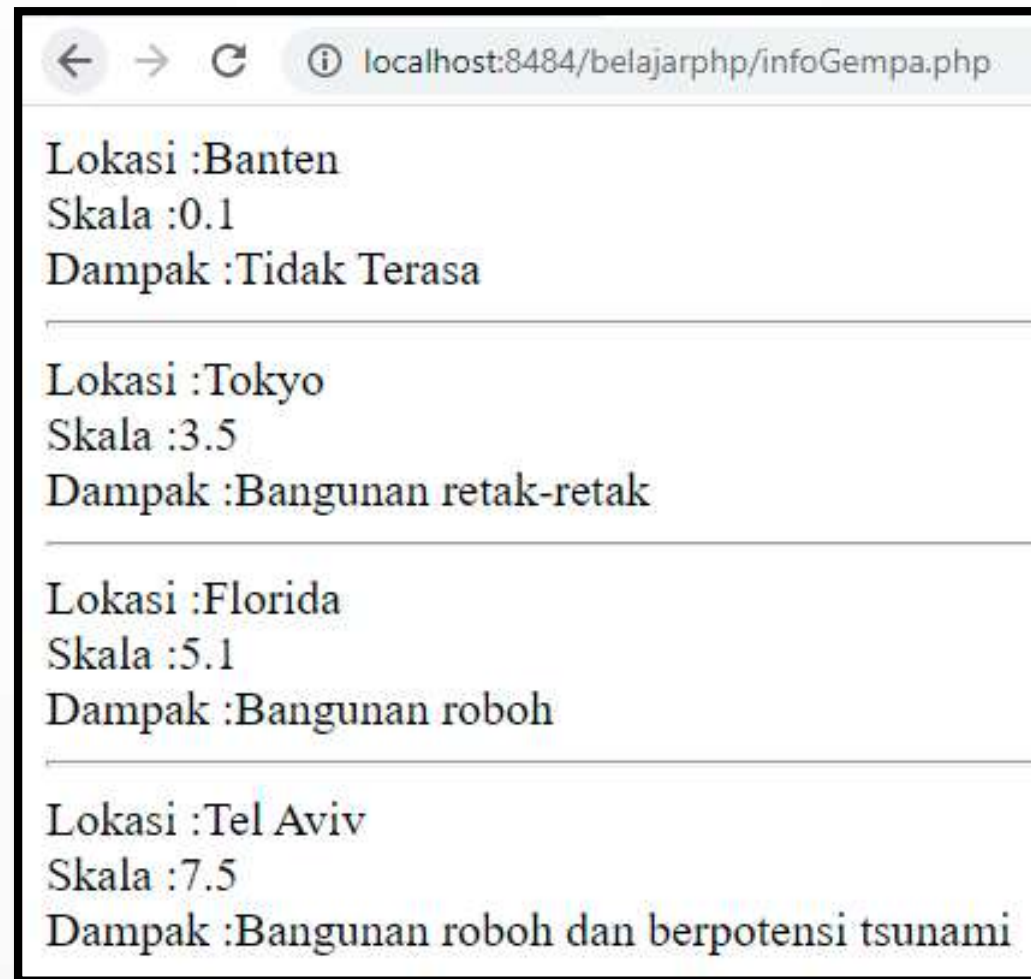
```
//member3 method
private function dampak($skala){
    $this->skala = $skala;
    if($skala >= 0 && $skala <= 2)
        $dampak = 'Tidak Terasa';
    else if($skala > 2 && $skala <= 4)
        $dampak = 'Bangunan retak-retak';
    else if($skala > 4 && $skala <= 6)
        $dampak = 'Bangunan roboh';
    else $dampak = 'Bangunan roboh dan berpotensi tsunami';
    return $dampak;
}
public function cetak(){
    echo 'Lokasi :'. $this->lokasi;
    echo '<br/>Skala :'. $this->skala;
    echo '<br/>Dampak :'. $this->dampak($this->skala);
    echo '<hr/>';
}
```

Menciptakan Object Class Gempa

```
<?php
require 'Gempa.php';
//ciptakan object
$g1 = new Gempa('Banten',0.1);
$g2 = new Gempa('Tokyo',3.5);
$g3 = new Gempa('Florida',5.1);
$g4 = new Gempa('Tel Aviv',9.5);

//panggil fungsi
$g1->cetak(); $g2->cetak();
$g3->cetak(); $g4->cetak();
```

Mencetak Info Gempa dari Object



A screenshot of a web browser window displaying the output of a PHP script. The browser's address bar shows the URL 'localhost:8484/belajarphp/infoGempa.php'. The page content is organized into four horizontal sections, each separated by a thin grey line. Each section contains three lines of text: 'Lokasi :', 'Skala :', and 'Dampak :', followed by their respective values. The data represents four different earthquakes: one in Banten (scale 0.1), one in Tokyo (scale 3.5), one in Florida (scale 5.1), and one in Tel Aviv (scale 7.5).

Lokasi :Banten
Skala :0.1
Dampak :Tidak Terasa
Lokasi :Tokyo
Skala :3.5
Dampak :Bangunan retak-retak
Lokasi :Florida
Skala :5.1
Dampak :Bangunan roboh
Lokasi :Tel Aviv
Skala :7.5
Dampak :Bangunan roboh dan berpotensi tsunami

Referensi

- <https://www.php.net/>
- <https://www.javapoint.com/>
- <https://www.tutorialspoint.com/>