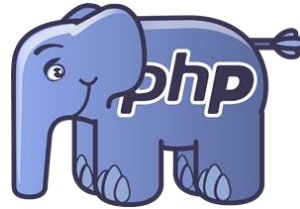


Pemrograman PHP

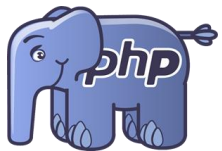


Pesantren PeTIK II YBM PLN

Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,
Kec. Jombang, Kabupaten Jombang, Jawa Timur



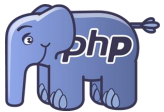
Pertemuan Ke-9





Materi

1. Pengantar Pemrograman PHP
2. Variabel dan Tipe Data
3. Array dan Struktur Data
4. PHP Form Processing
5. PHP Control Structure
6. PHP Function
7. Implementasi Template Web
9. PHP Object Oriented Programming
10. PHP Object Oriented Programming 2
11. PHP Database Connection
12. CRUD Database Operation 1
13. CRUD Database Operation 2
14. PHP Operasi file
15. Manajemen Session User



9. PHP Object Oriented Programming





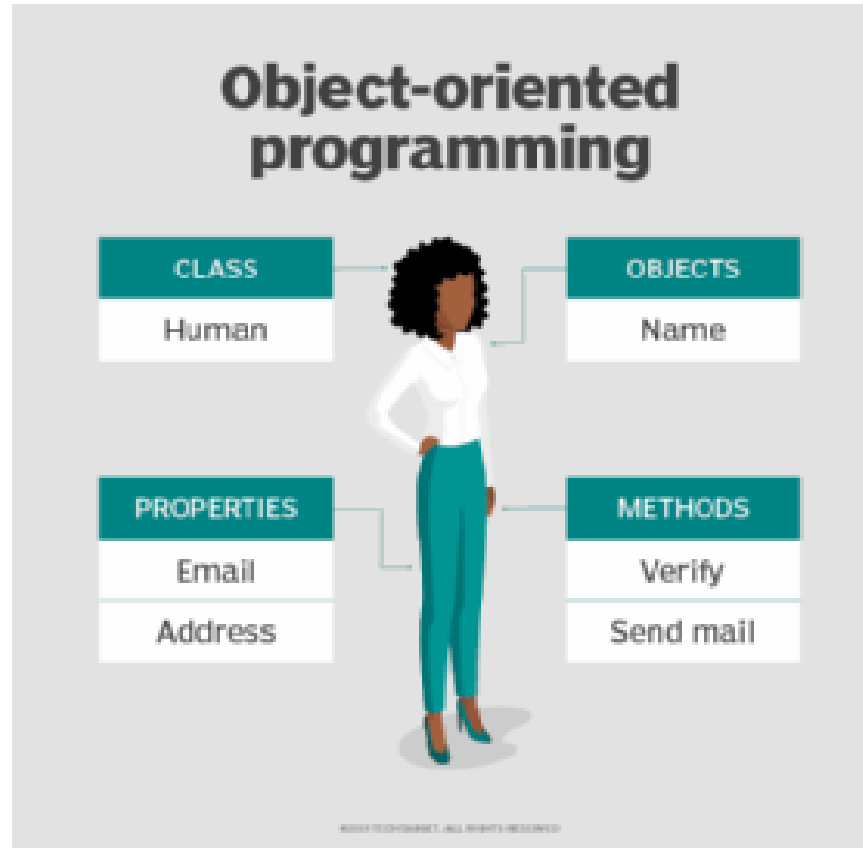
Object Oriented Programming

- ❑ Metode pemrograman yang berorientasi pada obyek, layaknya obyek yang ada dalam dunia nyata
- ❑ Setiap permasalahan diterjemahkan kedalam suatu obyek yang memiliki karakteristik : properti dan perilaku (behaviour)
- ❑ Suatu obyek memiliki (1) property untuk menyimpan data dan behaviour berupa (2) fungsi yang melakukan aksi atau proses terhadap data

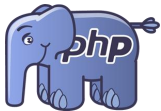




Object Oriented Programming



- Class
- Objects
- Properties
- Methods/Function



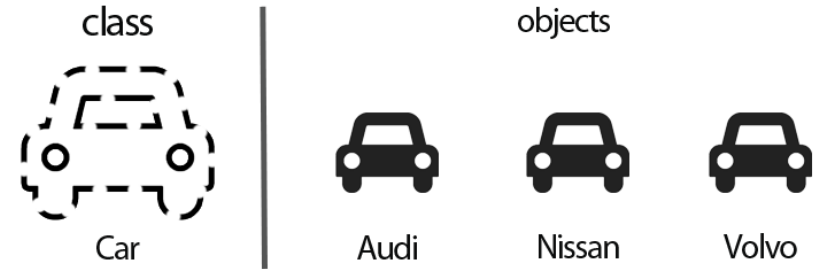
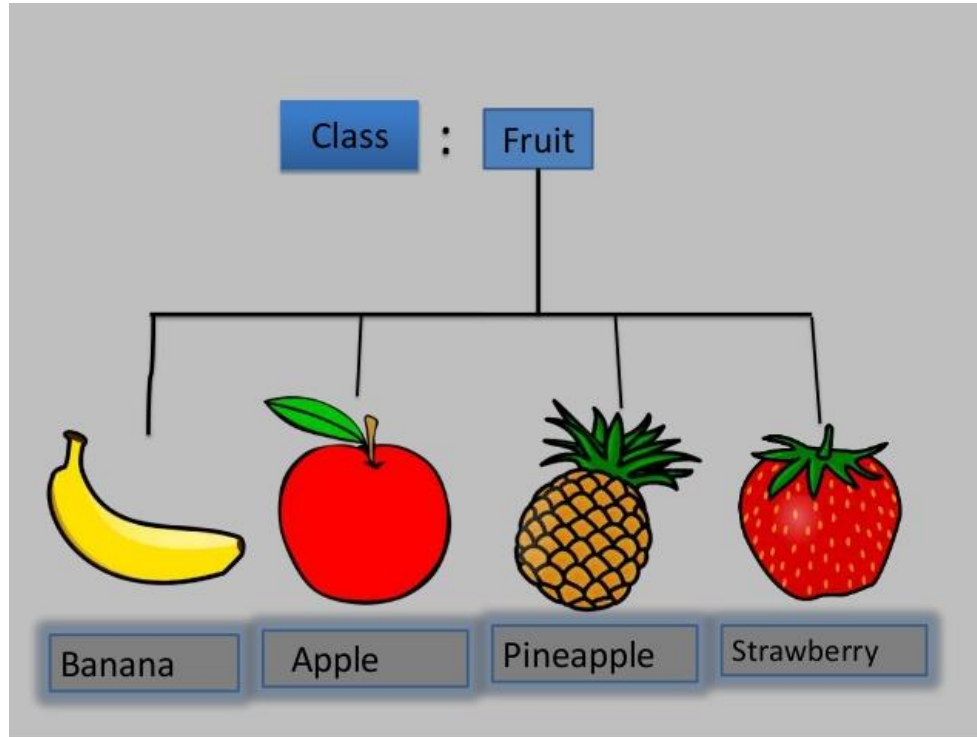


Keuntungan OOP

1. OOP lebih cepat dan lebih mudah dijalankan (faster)
2. OOP memberikan struktur yang jelas untuk program (good structure)
3. OOP membantu menjaga kode PHP lebih mudah untuk dimaintenance, dimodifikasi, dan di-debug (easy to maintenance)
4. OOP memungkinkan untuk membuat aplikasi yang dapat digunakan kembali secara penuh dengan kode yang lebih sedikit dan waktu pengembangan yang lebih singkat (reusable)



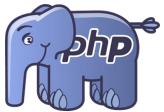
Class vs Object





Class & Object

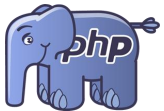
- Class : **Blue Print** dari Object
 - Gambar/Desain Rumah, Resep Masakan
- Object : **Instance** dari Class
 - Rumah , Hidangan di meja makan





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**.





PHP :: Class Declarations

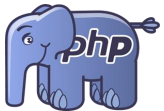
```
class Foo  
{  
  
}
```

```
$foo = new Foo();
```

```
$fii = new Foo();
```

```
$buah1 = new Fruit();  
$buah2 = new Fruit();
```

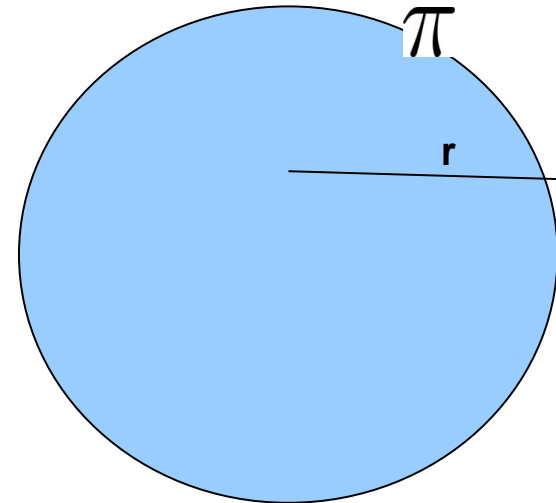
```
<?php  
class Fruit {  
    // code goes here...  
}  
?>
```





Member Class

- Variable : **Properti & Konstanta** class
 - Jari , Konstanta PHI
- Method : **Behavior** Class
 - getLuas , getKeliling



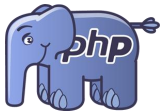


Static Member Class → **ClassName::**

```
class Matematika
{
    public static $counter = 100;

    public static function tambahkan($a , $b)
    {
        return $a + $b;
    }
}

Matematika::$counter++;
echo 'Counter Sekarang : '. Matematika::$counter;
$x = Matematika::tambahkan(5,4);
echo "5 + 4 = " . $x;
```



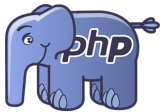


Static Member Class → **self::**

```
class Matematika
{
    public static $counter = 100;

    public static function naikanCounter()
    {
        self::$counter++;
    }
}
```

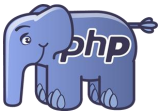
```
Matematika::naikanCounter();
echo 'Counter Sekarang : '. Matematika::$counter;
```





Static Member Class → **self::**

```
class_matematika.php
1  <?php
2      class Matematika {
3          // Konstanta class
4          const PHI = 3.14;
5
6          // static member variable
7          public static $counter = 100;
8
9          // static member function
10         public static function tambahkan($a,$b){
11             return $a + $b;
12         }
13
14         // akses member variable dari dalam class
15         public static function naikanCounter(){
16             self::$counter;
17         }
18
19         // akses konstanta class
20         public static function luaslingkaran($jari){
21             $luas = self::PHI * $jari * $jari;
22             return $luas;
23         }
24     }
25
26  ?>
```



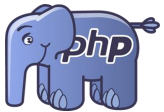


Konstanta Class → **self::**

```
class Matematika
{
    const PHI = 3.14;

    public static function luasLingkaran($jari)
    {
        $luas = self::PHI * $jari * $jari;
        return $luas;
    }
}

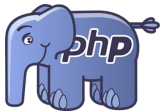
echo 'NILAI PHI '. Matematika::PHI ;
$luasling = Matematika::luasLingkaran(8);
echo 'Luas Lingkaran Jari2nya 8 = ' . $luasling;
```





Konstanta Class → **self::**

```
use_matematika.php
1  <?php
2      // tangkap request class_matematika.php
3      require_once 'kelas_matematika.php';
4
5      // akses static member variable class Matematika
6      Matematika::$counter++;
7      Matematika::$counter++;
8      Matematika::naikanCounter();
9      echo 'Counter Sekarang : '. Matematika::$counter;
10     echo '<hr/>';
11
12     // akses static member function class Matematika
13     $x = Matematika::tambahkan(4,5);
14     echo "4 + 5 = $x";
15     echo '<hr/>';
16
17     // akses constanta class Matematika
18     echo 'Nilai PHI : '. Matematika::PHI;
19     $luas_lingkaran = Matematika::luasLingkaran(8);
20     echo '<br/>Luas Lingkaran dengan Jari-jari 8 adalah : '. $luas_lingkaran;
21
22
23  ?>
```



Tampilan - Browser



localhost/belajar_php1/p9,

Counter Sekarang : 103

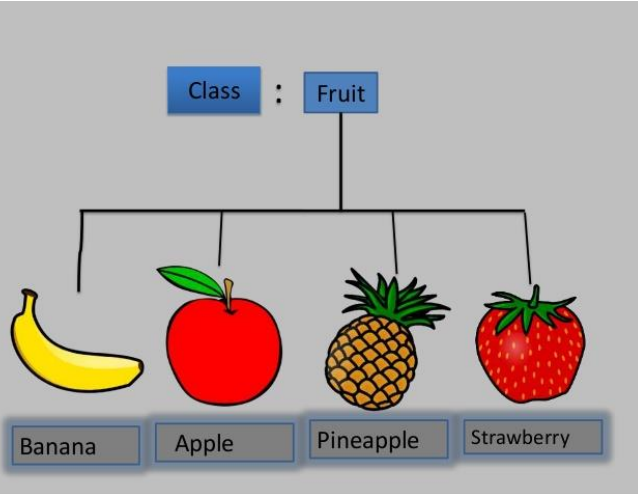
$4 + 5 = 9$

Nilai PHI : 3.14

Luas Lingkaran dengan Jari-jari 8 adalah : 200.96



PHP :: Class Non Static



```
<?php
class Fruit {
    // Properties
    public $name;
    public $color;

    // Methods
    function set_name($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
}
?>
```

```
$apple = new Fruit();
$banana = new Fruit();
$apple->set_name('Apple');
$banana->set_name('Banana');

echo $apple->get_name();
echo "<br>";
echo $banana->get_name();
?>
```

<https://www.w3schools.com/php/>





PHP :: Class Non Static

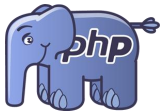
```
class_fruit.php
1  <?php
2      class fruit{
3          // properties atau variable
4          public $name;
5          public $color;
6
7          function set_name($name){
8              $this->name = $name;
9          }
10
11         function set_color($color){
12             $this->color = $color;
13         }
14
15         function get_name(){
16             return $this->name;
17         }
18
19         function get_color(){
20             $this->color;
21         }
22     }
23  ?>
```





PHP :: Class Non Static

```
keranjang_buah.php
1  <?php
2      // tangkap request class_fruit
3      require_once 'class_fruit.php';
4
5      // create instan objek fruit : $apple and $banana
6      $apple = new Fruit();
7      $banana = new Fruit();
8
9      // call member class
10     $apple->set_name('Apple');
11     $apple->set_color('Red');
12     $banana->set_name('Banana');
13     $banana->set_color('Yellow');
14
15     echo 'Nama Buah ' . $apple->get_name() . ' Warnanya ' . $apple->get_color();
16     echo '<br/>Nama Buah ' . $banana->get_name() . ' Warnanya ' . $banana->get_color();
17  ?>
```





Tampilan - Browser

← → ↻ ⓘ localhost/belajar_php1/p9,



Maps



YouTube

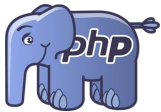


Gmail



News

Nama Buah Apple Warnanya Red
Nama Buah Banana Warnanya Yellow

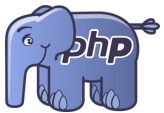




Member Class Variable → **:: non static**

```
class NilaiSiswa
{
    public $nama;
    public $nilai;
    public $sekolah='SDIT NF';
}
```

```
$nsl = new NilaiSiswa();
$nsl->nama = 'Faiz Fikri';
$nsl->nilai = 80;
echo $nsl->nama .' Sekolah di ' . $nsl->sekolah;
```

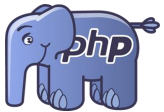




Member Class Variable → **:: non static**

```
class_nilaiSantri.php

1  <?php
2      class NilaiSantri{
3          public $nama;
4          public $nilai;
5          public $sekolah = 'PeTIK';
6
7          public function getHasil(){
8              if($this->nilai > 70) return 'Lulus';
9              else return 'Tidak Lulus';
10         }
11     }
12
13  ?>
```

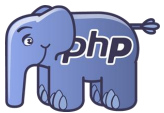




Member Class Variable → **:: non static**

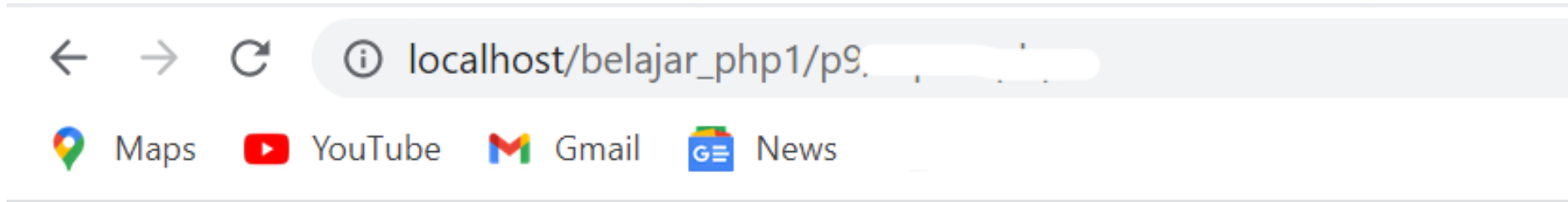
🐘 raport.php

```
1  <?php
2      //tangkap request class_nilaiSantri.php
3      require_once 'class_nilaiSantri.php';
4
5      $ns1 = new NilaiSiswa();
6      $ns1->nama = 'Fulan';
7      $ns1->nilai = 70;
8      echo $ns1->nama . ' Kuliah di ' . $ns1->sekolah;
9      echo '<br/>Hasil Ujian : ' . $ns1->nilai . ' dinyatakan ' . $ns1->getHasil();
10
11  ?>
```





Tampilan - Browser



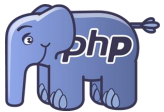
Fulan Kuliah di PeTIK
Hasil Ujian : 70 dinyatakan Lulus





Member Class Method → **:: non static**

```
class NilaiSiswa {  
    public $nama;  
    public $nilai;  
    public $sekolah='SDIT NF';  
  
    public function getHasil()  
    {  
        If ($this->nilai > 55 ) return 'Lulus';  
        else return 'Ngak Lulus';  
    }  
}  
  
$ns1 = new NilaiSiswa();  
$ns1->nama = 'Faiz Fikri';  
$ns1->nilai = 83;  
echo $ns1->nama .' Dinyatakan ' . $ns1->getHasil();
```





Constructor Class (1)

- ❑ method bernama `__construct()`
- ❑ baris kode program pada konstruktor akan dieksekusi ketika objek diciptakan
- ❑ Digunakan untuk inisialisasi variable class
- ❑ Contoh:
 - ❑ *Untuk membuat object Fruit harus set name dan color*
 - ❑ *Untuk membuat objek NilaiSiswa harus ada nama dan nilainya*



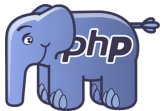


Constructor Class (2)

```
<?php
class Fruit {
    public $name;
    public $color;

    function __construct($name, $color) {
        $this->name = $name;
        $this->color = $color;
    }
    function get_name() {
        return $this->name;
    }
    function get_color() {
        return $this->color;
    }
}
```

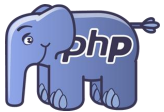
```
$apple = new Fruit("Apple", "red");
echo $apple->get_name();
echo "<br>";
echo $apple->get_color();
```





Constructor Class (3)

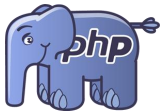
```
class NilaiSiswa {  
    public $nama;  
    public $nilai;  
    public $sekolah='SDIT NF';  
  
    function __construct($nama, $nilai)  
    {  
        $this->nama = $nama;  
        $this->nilai = $nilai;  
    }  
    public function getHasil()  
    {  
        if ($this->nilai > 55 ) return 'Lulus';  
        else return 'Ngak Lulus';  
    }  
}
```





Constructor Class (3)

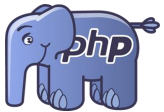
```
class_nilai.php
1  <?php
2      class NilaiSantri{
3          public $nama;
4          public $nilai;
5          public $kuliah = 'PeTIK Jombang';
6
7          function __construct($nama,$nilai){
8              $this->nama = $nama;
9              $this->nilai = $nilai;
10         }
11
12         public function getHasil(){
13             if($this->nilai >= 70) return 'Lulus';
14             else return 'Tidak Lulus';
15         }
16     }
17  ?>
```





Constructor Class (3)

```
$ns1 = new NilaiSiswa('Rosa',90);  
$ns2 = new NilaiSiswa('Badu',30);  
  
echo $ns1->nama .' : ' . $ns1->getHasil();  
  
echo $ns2->nama .' : ' . $ns2->getHasil();
```



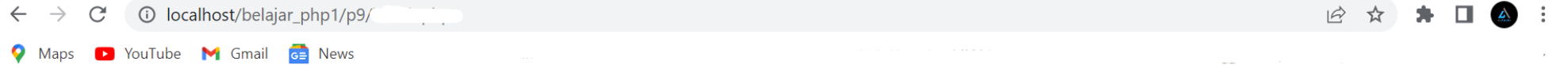


Constructor Class (3)

```
kuliah.php
1  <?php
2      //tangkap request class_nilaiSantri.php
3      require_once 'class_nilai.php';
4
5      $ns1 = new NilaiSantri('Fulan',70);
6      $ns2 = new NilaiSantri('Badu',69);
7      $ns3 = new NilaiSantri('Usro',85);
8      $ns4 = new NilaiSantri('Jarwo',40);
9      $ar_santri = [$ns1,$ns2,$ns3,$ns4];
10  ?>
11  <table>
12      <thead>
13          <tr>
14              <th>No</th>
15              <th>Nama</th>
16              <th>Nilai</th>
17              <th>Hasil</th>
18          </tr>
19      </thead>
20      <tbody>
21          <?php
22              $nomor = 1;
23              foreach($ar_santri as $san){
24                  echo '<tr><td>'.$nomor.'</td><td>'.$san->nama.'</td>';
25                  echo '<td>'.$san->nilai.'</td><td>'.$san->gethasil().'</td></tr>';
26                  $nomor++;
27              }
28          ?>
29      </tbody>
30  </table>
31  </table>
```



Tampilan - Browser



No	Nama	Nilai	Hasil
1	Fulan	70	Lulus
2	Badu	69	Tidak Lulus
3	Usro	85	Lulus
4	Jarwo	40	Tidak Lulus

**TERIMA KASIH
ATAS SEGALA PERHATIAN
SEMOGA BERMANFAAT...**

