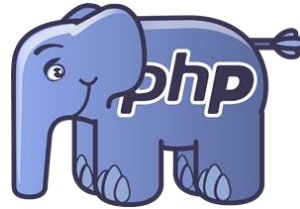


Pemrograman PHP

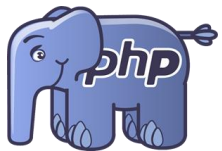


Pesantren PeTIK II YBM PLN

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



Pertemuan Ke-10





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 Sesion User



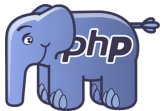
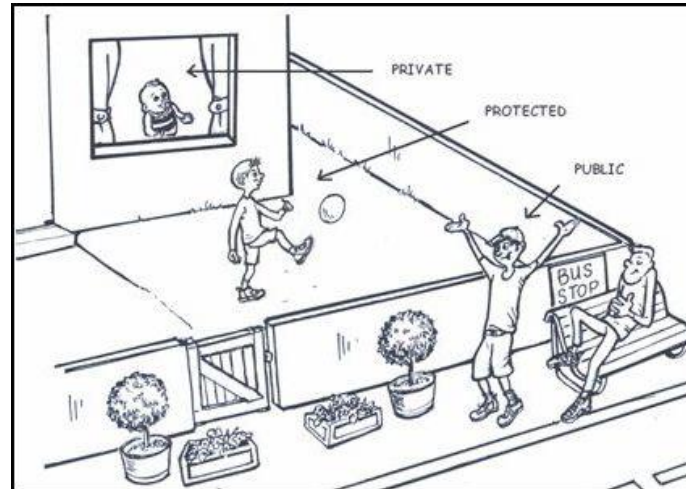
10. Object Oriented Programming 2





PHP OOP – Enkapsulasi

- Enkapsulasi adalah proses melindungi member class dari akses user (access modifier)
- Dalam enkapsulasi ditentukan hak akses terhadap member class : property dan method
- Hak akses terhadap member class, yaitu:
 1. Public
 2. Private
 3. Protected





Class – Problem di dunia nyata



- Sebuah wadah penyimpan cairan minuman memiliki kapasitas (volume) untuk menyimpan, terdapat sejumlah gelas dengan volume tampung 250ml yang akan digunakan, setiap gelas dapat saja dikenakan biaya. Ketika wadah habis dapat di isi ulang, dan tentu saja berkurang ketika pengguna mengambil segelas minuman. Dalam beberapa kasus bisa saja wadah ini menyimpan uang pembayaran minumannya.
- Dapatkah permasalahan dunia nyata ini diterjemahkan menjadi program OOP ?



Class Diagram - Visibility

+ <i>public</i>
- <i>private</i>
<i>protected</i>

Class Name
- attribute
- attribute
+ operation
+ operation
+ operation

class_mesinkopi.php

```
1  <?php
2  class Dispenser{
3      protected $volume;
4      protected $hargaSegelas;
5      private $volumeGelas;
6      public $namaMinuman;
7
8      public isi($vol){
9          $this->volume = $vol;
10     }
11 }
12 ?>
```



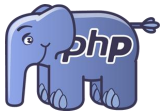


Access Modifier :: member class variabel

```
<?php
class Fruit {
    public $name;
    protected $color;
    private $weight;
}

$mango = new Fruit();
$mango->name = 'Mango'; // OK
$mango->color = 'Yellow'; // ERROR
$mango->weight = '300'; // ERROR
?>
```

- ❑ **public** : dapat diakses dari dalam maupun luar class
- ❑ **private** : hanya dapat diakses dari dalam class
- ❑ **protected** : dapat diakses dari dalam class dan class-class turunannya





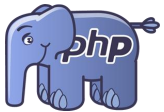
Access Modifier :: member class fungsi

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

    function set_name($n) { // a public function (default)
        $this->name = $n;
    }
    protected function set_color($n) { // a protected function
        $this->color = $n;
    }
    private function set_weight($n) { // a private function
        $this->weight = $n;
    }
}

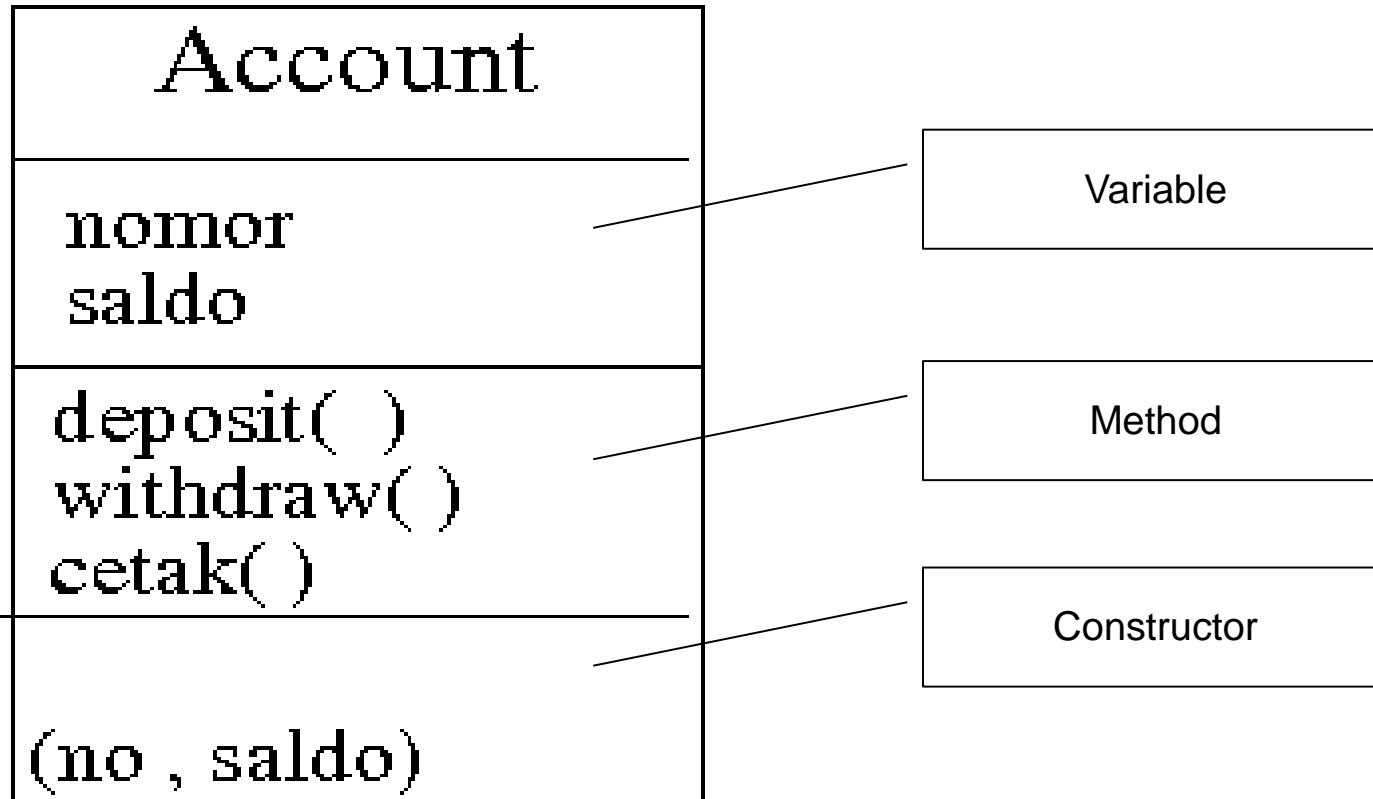
$mango = new Fruit();
$mango->set_name('Mango'); // OK
$mango->set_color('Yellow'); // ERROR
$mango->set_weight('300'); // ERROR
?>
```

- ❑ **public** : dapat diakses dari dalam maupun luar class
- ❑ **private** : hanya dapat diakses dari dalam class
- ❑ **protected** : dapat diakses dari dalam class dan class-class turunannya





Class Diagram



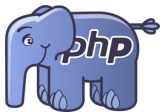


Class Account (1)

class_account.php

```
1 <?php
2 class Account {
3     public $nomor;
4     private $saldo;
5
6     function __construct($no, $saldo)
7     {
8         $this->nomor = $no;
9         $this->saldo = $saldo;
10    }
11    public function deposit($uang)
12    {
13        $this->saldo = $this->saldo + $uang;
14    }
```

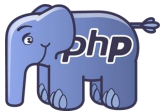
```
15
16    public function withdraw($uang)
17    {
18        $this->saldo = $this->saldo - $uang;
19    }
20
21    public function cetak() {
22        echo 'Nomor ' . $this->nomor .
23            ', Saldonya : ' . $this->saldo;
24    }
25 }
26 ?>
```





Class Account (1)

```
class_account.php
1  <?php
2      class Account{
3          public $nomor;
4          protected $saldo;
5
6          function __construct($nomor,$saldo){
7              $this->nomor = $nomor;
8              $this->saldo = $saldo;
9          }
10
11         public function deposit($uang){
12             $this->saldo = $this->saldo + $uang;
13         }
14
15         public function withdrawl($uang){
16             $this->saldo = $this->saldo - $uang;
17         }
18
19         public function cetak(){
20             echo 'Nomor Account : '.$this->nomor;
21             echo '<br/>Saldonya : '.$this->saldo;
22         }
23     }
24  ?>
```

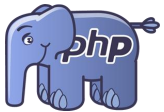




Class Account (2)

```
$ac1 = new Account("020", 4000);  
$ac1->deposit(200);  
$ac1->cetak();
```

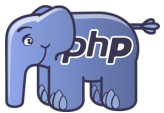
```
$ac2 = new Account("045", 2000);  
$ac2->withdraw(40);  
$ac2->cetak();
```



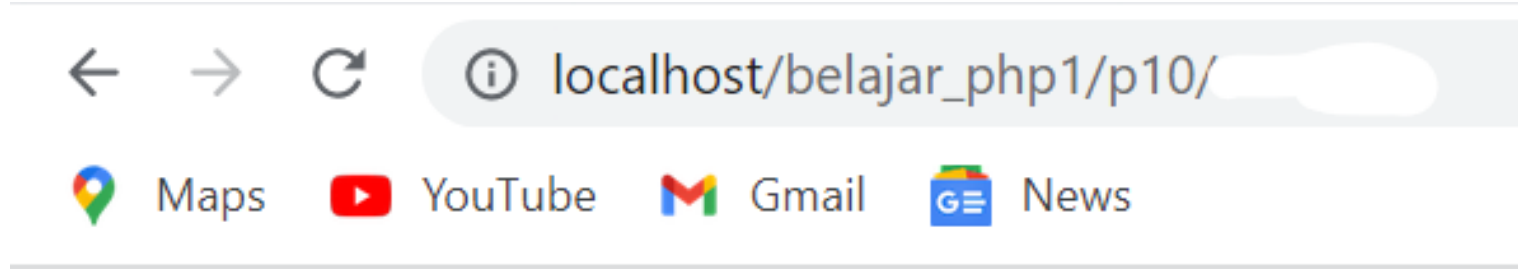


Class Account (1)

```
atm.php
1  <?php
2      require_once 'class_account.php';
3
4      $ac1 = new Account("03201",5000000);
5      $ac2 = new Account("01010",4500000);
6
7      $ac1->cetak();
8  ?>
9
```

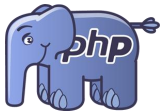


Class Account (1)



Nomor Account : 03201

Saldonya : 5000000

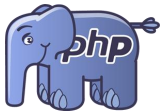




Class Account (1)

atm.php

```
1  <?php
2      require_once 'class_account.php';
3
4      $ac1 = new Account("03201",5000000);
5      $ac2 = new Account("01010",4500000);
6
7      $ac1->cetak();
8      echo '<br/>';
9      $ac1->deposit(75000);
10     echo '<br/>Nabung 75000 <br/>';
11     $ac1->cetak();
12
13  ?>
```





Class Account (1)



localhost/belajar_php1/p10,



Maps



YouTube



Gmail



News

Nomor Account : 03201

Saldonya : 5000000

Nabung 75000

Nomor Account : 03201

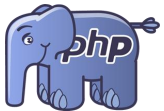
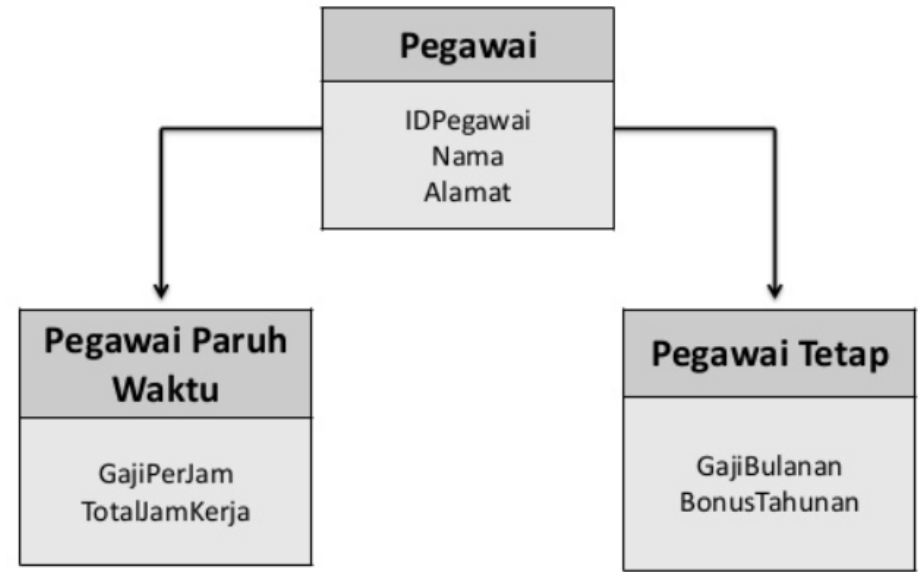
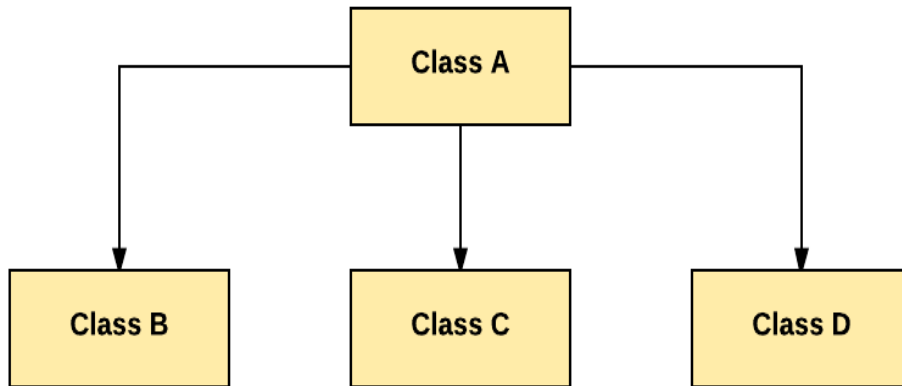
Saldonya : 5075000





PHP OOP - Inheritance

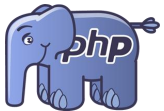
- Inheritance adalah konsep OOP dimana sebuah class dapat menurunkan property dan method yang dimilikinya





PHP OOP - Inheritance

- Inheritance – Pewarisan : mekanisme dalam OOP untuk menurunkan fitur (property dan method) yang dimiliki oleh suatu class (parent class) ke class turunannya (sub class)
- Inheritance memungkinkan programmer mengembangkannya (menambah fitur) dari kode program tanpa merubah kode program sebelumnya
- Tidak semua fitur dari class dapat diturunkan, property dan method dengan hak akses private tidak dapat diturunkan kepada sub class
- PHP menggunakan konsep single – inheritance : suatu class hanya dapat memiliki satu parent class

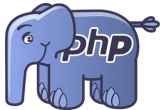
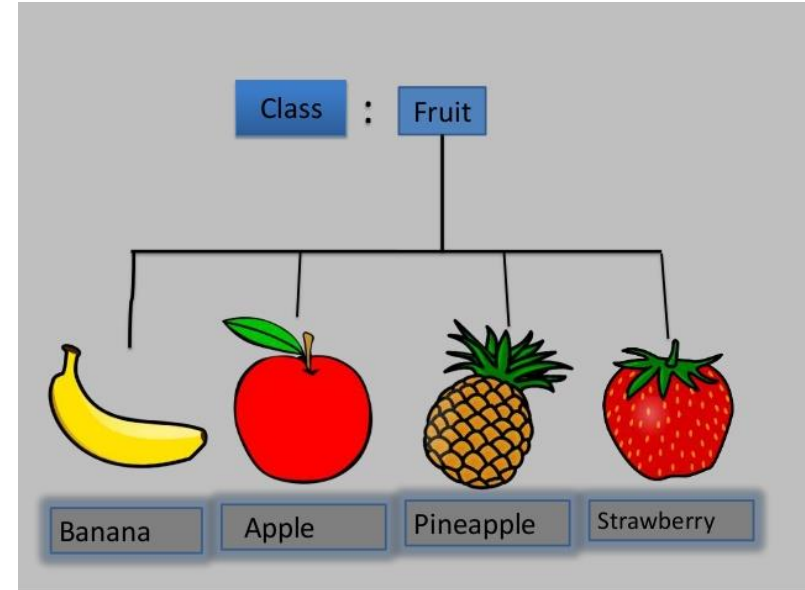




Parent Class - Fruit

```
<?php
```

```
class Fruit {  
    public $name;  
    public $color;  
    public function __construct($name, $color) {  
        $this->name = $name;  
        $this->color = $color;  
    }  
    public function intro() {  
        echo "The fruit is {$this->name} and the color is {$this->color}.";  
    }  
}
```



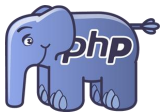
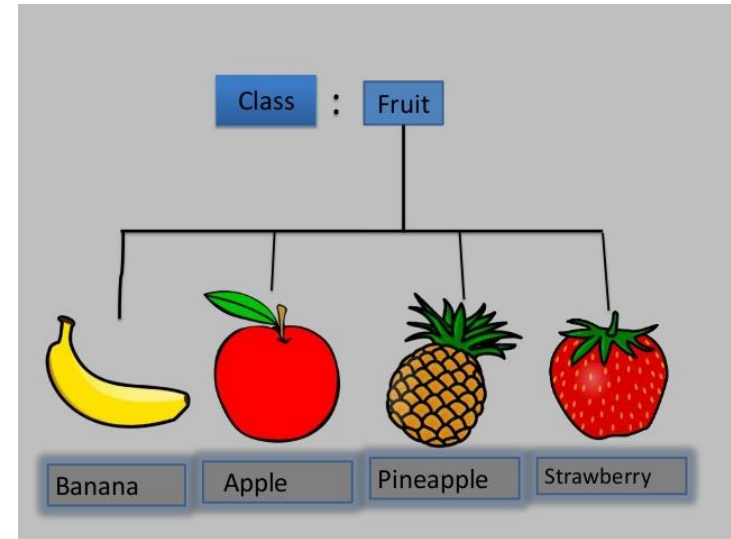


Sub Class - Strawberry

Gunakan keyword **extends**

```
// Strawberry is inherited from Fruit
class Strawberry extends Fruit {
    public function message() {
        echo "Am I a fruit or a berry? ";
    }
}
```

```
$strawberry = new Strawberry("Strawberry", "red");
$strawberry->message();
$strawberry->intro();
?>
```





Class Account (1)





```
fruit.php
1  <?php
2      class Fruit{
3          public $name;
4          public $color;
5          public function __construct($name,$color){
6              $this->name = $name;
7              $this->color = $color;
8          }
9          public function intro(){
10             echo "The fruit is {$this->name} and the color is {$this->color}.";
11         }
12     }
13
14     class Strawberry extends Fruit{
15         public function message(){
16             echo "Am I a fruit or a berry? ";
17         }
18     }
19     $strawberry = new Strawberry("Strawberry","Red");
20     $strawberry->message();
21     $strawberry->intro();
22
23     ?>
```



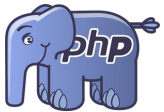


Class Account (1)

← → ↺ ⓘ localhost/belajar_php1/p10/

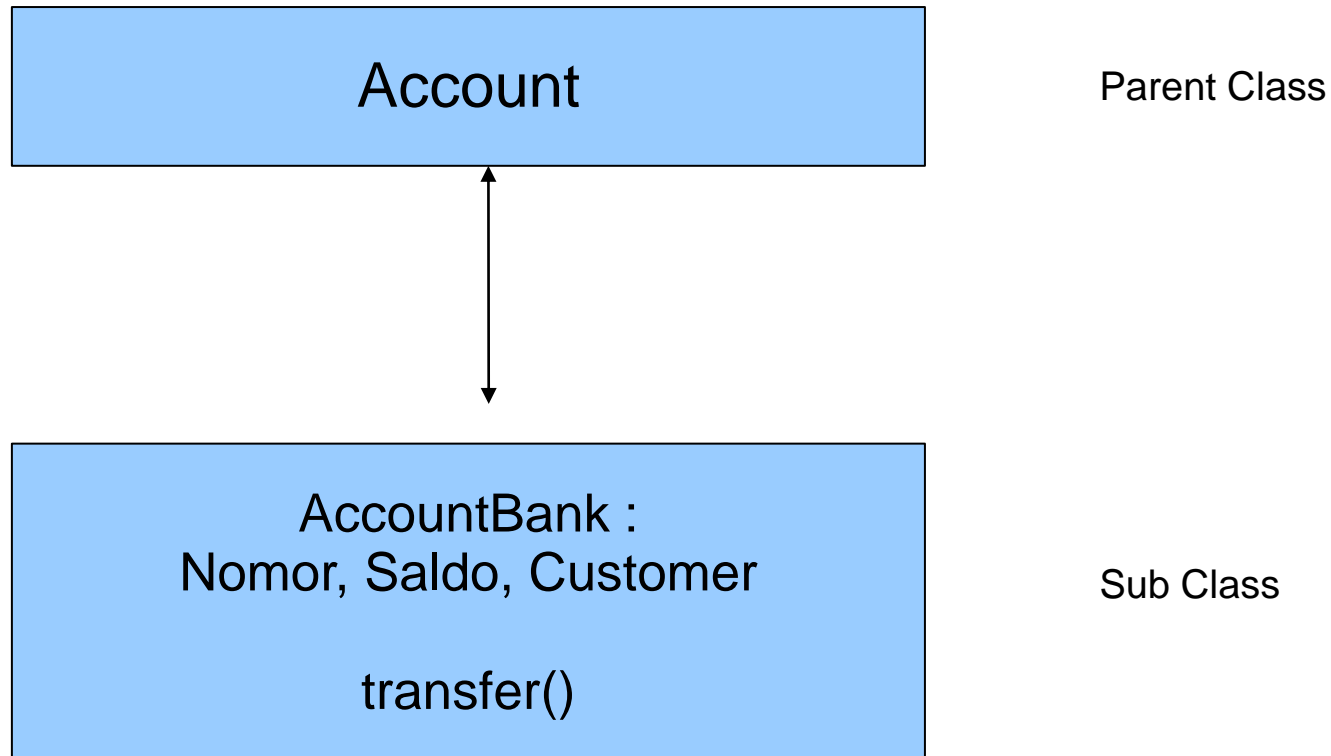
 Maps  YouTube  Gmail  News

Am I a fruit or a berry? The fruit is Strawberry and the color is Red.





Inheritance → :: extends





Inheritance → :: extends

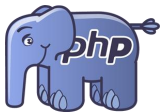
```
class_account.php
1  <?php
2      class Account{
3          public $nomor;
4          protected $saldo;
5
6          function __construct($nomor,$saldo){
7              $this->nomor = $nomor;
8              $this->saldo = $saldo;
9          }
10
11         public function deposit($uang){
12             $this->saldo = $this->saldo + $uang;
13         }
14
15         public function withdrawl($uang){
16             $this->saldo = $this->saldo - $uang;
17         }
18
19         public function cetak(){
20             echo 'Nomor Account : '.$this->nomor;
21             echo '<br/>Saldonya : '.$this->saldo;
22         }
23     }
24  ?>
```





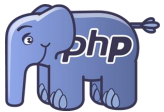
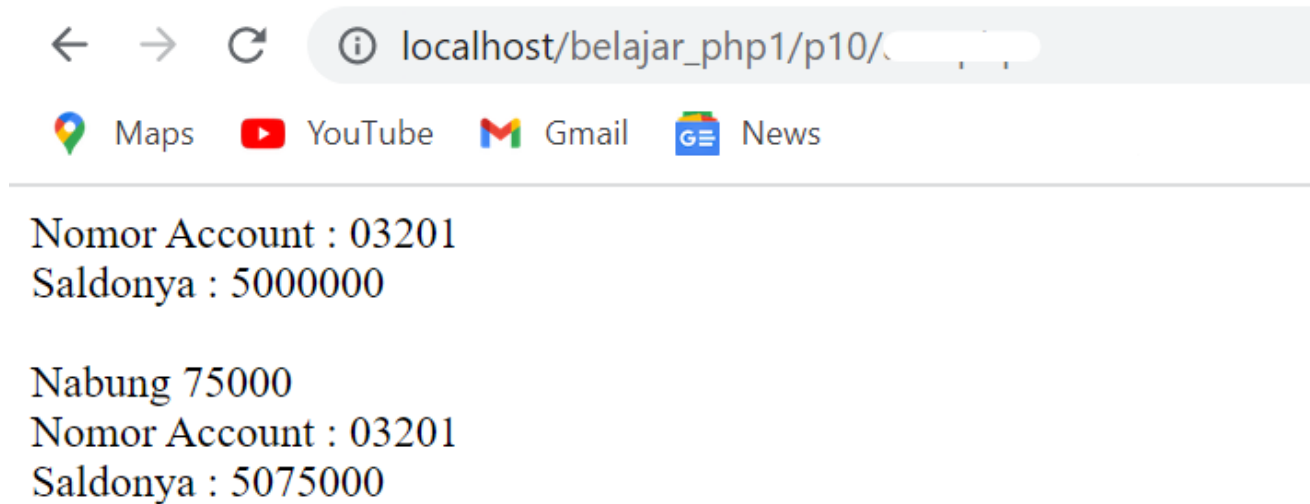
Inheritance → :: extends

```
atm.php
1  <?php
2      require_once 'class_account.php';
3
4      $ac1 = new Account("03201",5000000);
5      $ac2 = new Account("01010",4500000);
6
7      $ac1->cetak();
8      echo '<br/>';
9      $ac1->deposit(75000);
10     echo '<br/>Nabung 75000 <br/>';
11     $ac1->cetak();
12
13  ?>
```





Inheritance → :: extends





Inheritance → :: extends

class_bankaccount.php

```
1 <?php
2 require_once 'class_account.php';
3 class BankAccount extends Account {
4     // tambahkan data baru customer
5     public $customer;
6
7     function __construct($no, $saldo_awal, $cust) {
8         // panggil constructor parent class
9         parent::__construct($no, $saldo_awal);
10        $this->customer = $cust;
11    }
12    // tulis ulang method cetak
13    function cetak() {
14        parent::cetak();
15        echo ', Customer : ' . $this->customer;
16    }
```

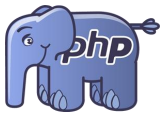
```
17
18 function transfer($obj_account,$uang) {
19     $obj_account->deposit($uang);
20     $this->withdraw($uang);
21 }
22
23
```





Inheritance → :: extends

```
atm2.php
1  <?php
2      require_once 'class_accountBank.php';
3
4      $ab1 = new AccountBank("010",5000000,"Messi");
5      $ab2 = new AccountBank("070",10000000,"Ronaldo");
6
7      $ab1->cetak();
8      echo '<hr/>';
9      $ab2->cetak();
10     echo '<br/>Ronaldo transfer uang ke Messi 1250000<br/>';
11     echo 'Biaya Admin : '.AccountBank::$biaya_admin.'<br/>';
12     $ab2->transfer($ab1,1250000);
13     $ab1->cetak();
14     echo '<hr/>';
15     $ab2->cetak();
16
17     $ar_ab = [$ab1,$ab2];
18
19     ?>
```





Inheritance → :: extends



localhost/belajar_php1/p10/



Maps



YouTube



Gmail



News

Nomor Account : 010

Saldonya : 5000000

Customer : Messi

Nomor Account : 070

Saldonya : 10000000

Customer : Ronaldo

Ronaldo transfer uang ke Messi 1250000

Biaya Admin : 6500

Nomor Account : 010

Saldonya : 6250000

Customer : Messi

Nomor Account : 070

Saldonya : 8743500

Customer : Ronaldo





PHP OOP - Trait

- Trait mekanisme penulisan kode yang dapat dipakai ulang (reuse code)
- Trait memberikan solusi dari keterbatasan single-inheritance, dengan developer dapat menulis fungsi yang dapat digunakan secara bebas oleh file program lainnya
- Trait mirip dengan class, hanya Trait berisi kumpulan fungsi-fungsi yang langsung dapat dipakai tanpa membuat object seperti class.
- Trait mulai ada pada PHP versi 5.4 keatas





PHP OOP - Trait

trait01.php

```
1 <?php
2 trait ToolsWeb{
3     public function cetakJudul($str){
4         echo "<h2>$str</h2>";
5     }
6
7     public function cetakFormatUang($uang){
8         echo number_format($uang,0,"",".");
9     }
10 }
11
12 class Kasir {
13     use ToolsWeb;
14     public function cetak($customer,$uangbayar){
15         $this->cetakJudul("Pelanggan ".$customer);
16         $this->cetakFormatUang($uangbayar);
17     }
18 }
19
20 $kasir = new Kasir();
21 $_customer = "Danira Lusi";
22 $_uangbayar = 500000;
23 $kasir->cetak($_customer,$_uangbayar);
24
25 ?>
```





Package – Namespace (1)

- Fitur PHP > 5.4
- Pustaka program (Class – Class) tersimpan dalam direktori yang terstruktur sesuai dengan fungsi / kegunaannya
- Digunakan pada beberapa framework PHP
- Pustaka program dapat didistribusikan dalam GITHUB library oleh vendor / programmer yang membuatnya
- Keyword:

```
namespace keu\models; // direktori file keu\models  
class Account{
```

....





Package -- Namespace

- Cara penggunaan:

```
<?php
```

```
// to be handle by framework on bootstrap file
```

```
include_once 'keu\models\Account.php';
```

```
use keu\models\Account;
```

```
$ac1 = new Account('A001', 5000);
```

```
$ac1->cetak();
```

```
?>
```

