

Pemrograman Berorientasi Objek Dengan PHP5

Ditulis oleh: Gerry Sabar

Pendahuluan

Tutorial ini berawal dari pengalaman saya mengamati teman-teman di universitas yang belajar pemrograman berorientasi objek. Tetapi mereka tidak memahami seperti apa & bagaimana pemrograman berorientasi objek itu. Hal ini terjadi menurut saya mungkin dikarenakan terlalu banyak teori tetapi kurang atau bahkan tidak ada contoh konkrit sehingga mereka kebingungan.

Maka pada tutorial ini saya mencoba untuk memberi penjelasan ringkas mungkin dan lebih banyak dengan contoh konkrit sehingga mudah untuk dipahami. Tutorial ini mengenalkan anda pada pemrograman berorientasi objek (Object Oriented Programming) dengan menggunakan PHP5.

Tutorial ini memerlukan banyak sekali perbaikan, oleh karena itu jika anda ada pertanyaan atau masukan untuk perbaikan tutorial ini, silakan kirim email ke penulis di [gerrysabar\(at\)gmail.com](mailto:gerrysabar(at)gmail.com). Masukan-masukan yang anda berikan sangat bermanfaat untuk pengembangan tutorial ini supaya lebih baik & berguna.

Apa itu OOP?

Pemrograman berorientasi objek atau dalam bahasa Inggrisnya disebut dengan Object Oriented Programming (selanjutnya disingkat OOP) adalah teknik pemrograman yang menggunakan objek. Pemrograman berorientasi objek ini memiliki kemampuan untuk menyembunyikan apa yang tidak penting bagi penggunanya. Pada tutorial singkat ini saya menggunakan PHP5 karena PHP5 telah memiliki fitur pemrograman berorientasi objek yang lebih lengkap daripada versi sebelumnya.

Mengapa harus menggunakan OOP?

Jika anda membuat program skala kecil tentu saja manfaat yang jelas dari OOP ini kurang terasa. Tetapi ketika membuat projek program skala menengah sampai besar maka akan terlihat betapa pentingnya penggunaan OOP.

Mungkin anda berfikir toh sekarang ini baru pemula buat apa belajar OOP kalau OOP lebih ditunjukkan dalam projek pembuatan program skala menengah sampai besar. Mungkin untuk sekarang bagi anda yang masih pemula dalam pemrograman belum terasa, tetapi jika anda dari sekarang sudah memiliki pemahaman dasar yang baik mengenai OOP maka ketika tiba waktunya bagi anda untuk memperdalam OOP anda sudah memiliki persiapan yang matang.

Oleh karena itu pada tutorial ini pun saya tidak membuat anda menjadi ahli dalam OOP. Tutorial ini lebih ditunjukkan supaya anda memiliki pemahaman yang baik mengenai OOP.

Membuat Objek

Langkah awal dalam OOP adalah pembuatan objek di mana objek itu sendiri berasal dari kelas. Oleh karena itu pertama mari kita membuat kelas. Untuk membuat kelas di dalam PHP anda menggunakan kata kunci **class**. Pada contoh berikut kita akan membuat sebuah kelas `Emailer`.

```
<?php
class Emailer
{
?>
```

Pada contoh di atas kita membuat sebuah kelas bernama `Emailer`. Dalam pembuatan kelas, pertama kita menggunakan kata kunci `class` yang kemudian diikuti oleh nama kelas. Kemudian diakhiri dengan kurung kurawal. Di dalam kurung kurawal kita menuliskan kode-kode supaya kelas tersebut bekerja seperti yang kita inginkan. Seperti yang bisa anda lihat, pada contoh kode di atas kita belum memasukan kode apa pun sehingga kelas `Emailer` tidak melakukan pekerjaan apa-apa.

Kode yang ada di dalam kelas terbagi menjadi dua kelompok, yaitu `property` dan `method`. `Property` adalah suatu wadah penyimpanan di dalam kelas yang bisa menampung informasi. Sederhananya `property` itu bisa disebut sebagai variabel di dalam kelas. Sedangkan `method` adalah fungsi yang ada di dalam kelas. Sekarang mari kita berikan sebuah `property` dan dua buah `method` di dalam kelas `Emailer`. Sehingga kode akan terlihat menjadi seperti berikut:

```
<?php
class Emailer
{
    private $EmailAddress; //property

    public function getEmailAddress() //method
    {
        echo $this->$EmailAddress
    }

    public function setEmailAddress()
    {
        $this->$EmailAddress;
    }
}
?>
```

Untuk sekarang abaikan dulu kata kunci **public**, **private**, serta **\$this** yang terlihat pada kode di atas. Kita akan membahasnya nanti. Pada contoh di atas kita menambahkan sebuah `property` bernama `EmailAddress` yang nantinya digunakan untuk menampung alamat email. Kemudian kita juga menambahkan dua buah `method`, yang pertama bernama `getEmailAddress` yang berfungsi untuk menampilkan

isi property EmailAddress. Yang kedua adalah setEmailAddress yang berfungsi untuk memasukan alamat email ke property EmailAddress.

Menggunakan Kelas

Kode-kode kelas harus berada di dalam script PHP yang menggunakan kelas tersebut. Seringkali kelas yang dibuat disimpan dalam file yang berbeda dan kemudian disisipkan dengan menggunakan kata kunci include ketika diperlukan di dalam script PHP.

Untuk menggunakan objek, pertama-tama anda harus membuat objek dari suatu kelas. Pada konteks ini kelasnya adalah EMailer. Membuat objek dari sebuah kelas dalam bahasa Inggrisnya disebut *instantiating*. Untuk membuat objek di dalam PHP, kita menggunakan kata kunci **new**. Tata kalimatnya adalah sebagai berikut:

```
$NamaObjek = new NamaKelas( );
```

Pada kode berikut kita akan membuat objek dari kelas EMailer

```
<?php
class EMailer
{
    private $EmailAddress; //property

    public function getEmailAddress() //method
    {
        $this->$EmailAddress
    }

    public function setEmailAddress()
    {
        return $this->$EmailAddress;
    }
}

//contoh membuat objek di PHP
$emailerObject = new EMailer();
?>
```

Menggunakan Objek

Kita telah membuat sebuah kelas dan membuat objek dari kelas tersebut. Sekarang mari kita bereksperimen sedikit dengan objek yang telah kita buat. Pada kode yang baru saja dibuat, kita telah membuat sebuah property dan dua buah method pada kelas EMailer. Untuk menggunakan method yang ada pada suatu objek maka harus digunakan operator `->` yang kemudian diikuti dengan nama method yang hendak digunakan. Berikut adalah contoh untuk menggunakan method:

```
<?php
class EMailer
{
```

```

private $EmailAddress; //property

public function getEmailAddress() //method
{
    return $this->$EmailAddress;
}
public function setEmailAddress($emailName)
{
    $this->$EmailAddress = $emailName;
}
}

//contoh membuat objek di PHP
$mailerObject = new Mailer();

//menggunakan method
$mailerObject->setEmailAddress("username@example.com");
echo $mailerObject->getEmailAddresses();
?>

```

Ketika kode dijalankan maka akan memunculkan teks `username@example.com` pada layar web browser anda. Ini adalah salah satu contoh pemrograman berorientasi objek yang paling sederhana. Di mana anda menggunakan sebuah objek pada script PHP anda.

Public & Private

Mungkin dari sebelumnya anda sudah bertanya-tanya apa itu public & private yang ada pada kelas Mailer. Property dan method yang ada pada kelas dapat memiliki sifat public atau private (masih terdapat satu lagi yaitu protected, tetapi akan dibahas pada tutorial berikutnya).

Apa perbedaan di antara keduanya? Private berarti method atau property yang ada di dalam suatu kelas hanya bisa diakses di dalam kelasnya. Sedangkan pada method atau property yang bersifat public berarti method atau property tersebut bisa diakses di dalam dan di luar kelas. Pada contoh kode sebelumnya, kita lihat property `EmailAddresses` bersifat private. Sekarang kita coba property tersebut diakses dari luar objek `mailerObject`, sehingga kodenya akan terlihat seperti berikut:

```

<?php
class Mailer
{
    private $EmailAddress; //property

    public function getEmailAddresses() //method
    {
        return $this->$EmailAddress;
    }

    public function setEmailAddresses($emailName)
    {
        $this->$EmailAddress = $emailName;
    }
}

```

```
//contoh membuat objek di PHP
$mailerObject = new Mailer();

//mengakses property dari luar objek
$mailerObject->EmailAddress = "username@example.com";
?>
```

Ketika kode di atas dijalankan maka akan muncul pesan kesalahan sebagai berikut:

Fatal error: Cannot access private property Mailer::\$EmailAddress in C:\xampp\htdocs\gerry\latihan\test.php on line 21

Coba ubah property EmailAddress menjadi public, kemudian jalankan kembali kode, maka pesan kesalahan tidak akan muncul.

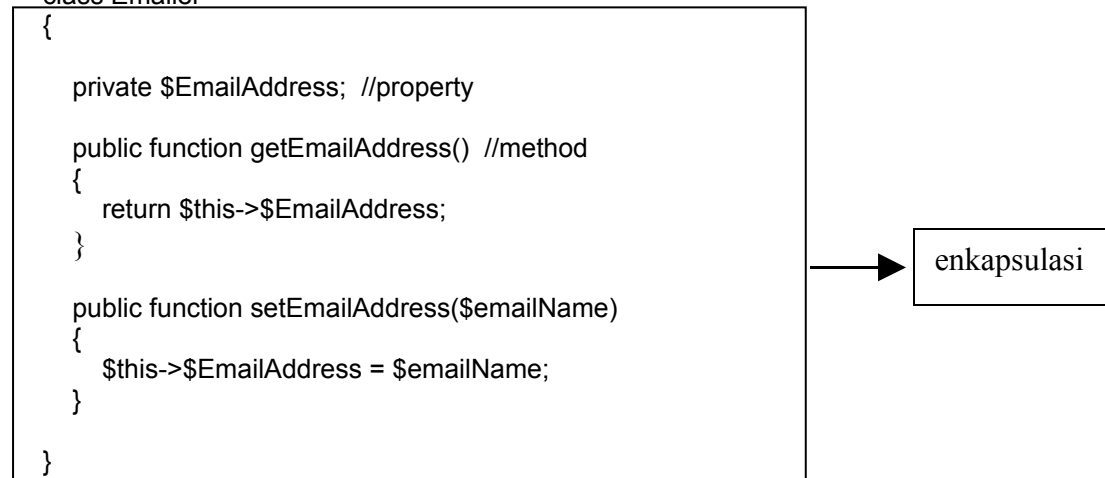
Enkapsulasi

Enkapsulasi atau dalam bahasa Inggrisnya disebut sebagai encapsulation adalah mekanisme untuk mengikat kode dengan data yang dimanipulasinya sehingga kode dan data yang ada di dalamnya aman dari gangguan luar. Menggabungkan data dan method ke dalam sebuah kelas disebut enkapsulasi. Pada contoh kode sebelumnya anda baru saja melakukan enkapsulasi:

```
<?php
class Mailer
{
    private $EmailAddress; //property

    public function getEmailAddress() //method
    {
        return $this->$EmailAddress;
    }

    public function setEmailAddress($emailName)
    {
        $this->$EmailAddress = $emailName;
    }
}
```



```
//contoh membuat objek di PHP
$mailerObject = new Mailer();
?>
```

\$this

Sekarang kita berada pada bagian akhir dari tutorial singkat ini. Pembahasan terakhir adalah operator \$this. Di dalam sebuah kelas, \$this adalah sebuah variabel istimewa untuk mengakses property yang ada di dalam kelas yang sedang digunakan. \$this tidak bisa digunakan di luar kelas. Format penggunaan \$this adalah sebagai berikut:

```
$this->namavariabel
```

Pada kode contoh sebelumnya, kelas `Mailer` memiliki property `EmailAddress`. Anda dapat mengakses property `EmailAddress` seperti berikut:

```
$this->EmailAddress
```

Dengan menggunakan `$this` untuk mengakses property, anda bisa melakukan berbagai operasi terhadap `EmailAddress` seperti berikut:

```
$this->EmailAddress = "example@example.com";  
$this->EmailAddress = $UserEmail;  
ArrayEmail[$this->EmailAddress] = $UserEmail;
```

Perhatikan tanda dolar (`$`), ketika menggunakan variabel anda mengawali nama variabel dengan tanda dolar. Pada contoh di atas `$UserEmail`. Tetapi ketika menggunakan `$this->` maka variabel atau property tidak boleh menggunakan tanda dolar. Kesalahan seperti ini seringkali terjadi pada programmer PHP yang masih pemula. Tetapi dengan latihan secara konsisten, kesalahan dasar seperti ini akan hilang dengan sendirinya.

Method

Method mendefinisikan apa yang bisa dikerjakan oleh objek. Method ini dibuat di dalam kelas. Mudah-mudahan Method itu adalah fungsi (function) di dalam kelas atau objek. Pada contoh kode sebelumnya anda telah membuat dua buah method yaitu `getEmailAddress` dan `setEmailAddress`.

Untuk menggunakan method, sama seperti property anda harus menggunakan operator `->` sehingga pada contoh kode sebelumnya ketika anda menggunakan method `getEmailAddress` dan `setEmailAddress`, kode yang anda ketikkan adalah sebagai berikut:

```
$mailerObject->setEmailAddress(username@example.com);  
echo $mailerObject->getEmailAddress();
```

Latihan

1. Apa itu pemrograman berorientasi objek?
2. Sebutkan perbedaan antara kelas dan objek!
3. Buatlah sebuah kelas, kemudian buat objek serta buat juga sebuah property dan dua buah method di dalam kelas tersebut.
4. Apa perbedaan antara `public` dan `private` di dalam property atau method?
5. Sebutkan kegunaan `$this`!

Penutup

Akhirnya tutorial pengenalan pemrograman berorientasi objek ini telah selesai. Setelah anda selesai mempelajari tutorial ini, diharapkan anda mulai mengerti:

1. Apa itu pemrograman berorientasi objek.
2. Apa itu objek dan kelas.
3. Pembuatan objek & kelas di dalam PHP.
4. Bagaimana membuat sifat suatu objek.

Pada tutorial berikutnya saya berniat membahas:

1. Pewarisan (inheritance).
2. Polimorphisme (polymorphism).
3. Constructor
4. Destructor
5. dan masih banyak lagi.

Jika anda ada masukan apa saja yang sebaiknya dimasukan pada tutorial berikutnya, silakan kirim email ke penulis. Semoga tutorial ini bermanfaat bagi anda! ☺

Sekilas Tentang Penulis

Penulis adalah mahasiswa di salah satu perguruan tinggi di Bandung (pusing mikirin kuliah yang lama lulusnya nih hehehe 😊). Jika tidak sedang duduk di depan komputer, maka penulis biasa berlatih alat musik. Penulis bisa dihubungi di [gerrysabar\(at\)gmail.com](mailto:gerrysabar(at)gmail.com).