

## PERTEMUAN VI

### OBJECT ORIENTED PROGRAMMING (Cont)

#### VI.1. Menggunakan class inheritance

Sebuah class dapat mewariskan method dan property dari kelas yang lain, dengan menggunakan keyword extends.

Contoh:

```
1.<?php
2.class oop1
3.{
4.public function penjumlahan($a,$b)
5. {
6.  $c=$a+$b;
7.  return $c;
8. }
9.}
10.
11.class MyanotherClass extends oop1
12. {
13.  public function perkalian($a,$b)
14.  {
15.    $c=$a*$b;
16.    return $c;
17.  }
18. }
19.
20. $obj = new oop1;
21. echo $obj->penjumlahan(2,6)."<br/>";
22.
23. $obj1=new MyanotherClass;
24. echo $obj1->penjumlahan(4,10)."<br/>";
25. echo $obj1->perkalian(4,10)."<br/>";
26. ?>
```

Output :

```
8
14
40
```

#### VI.2. Memberikan hak akses dari property dan method

Untuk mengakses property dan method terdapat beberapa keyword yang berbeda-beda, yakni public, protected dan private. Terdapat juga keyword static, yang mana property dan method dengan hak akses ini dapat di akses langsung tanpa menginstansiasi class.

##### - Public property dan Method

Public : Arti nya property dan method dengan keyword ini dapat di akses di mana saja, di dalam maupun di luar class nya.Protected property dan Method

Protected : Arti nya property dan method dengan keyword ini dapat di akses hanya di dalam class atau class yang meng extends class yang memiliki property dan method ini.

```
1.<?php
2. class oop2
3. {
4.  protected function penjumlahan($a,$b)
```

```

5.     {
6.         $c=$a+$b;
7.         return $c;
8.     }
9. }
10.
11.class MyanotherClass extends oop2
12. {
13.     public function perkalian($a,$b)
14.     {
15.         $c=$a*$b;
16.         return $c;
17.     }
18.
19.public function callprotected()
20. {
21.     return $this->penjumlahan(4,5);
22. }
23. }
24.
25. $obj = new oop2;
26.
27. $obj1=new MyanotherClass;
28. echo $obj1->callprotected()."<br/>";
29. echo $obj1->perkalian(4,10)."<br/>";
30. ?>

```

Output :

9  
40

Pada contoh source code di atas, function penjumlahan(\$a,\$b) adalah protected (baris : 4) dan tidak dapat di panggil langsung oleh objek \$obj1. Function penjumlahan(\$a,\$b) dapat di panggil oleh method callprotected yang bersifat public pada class MyanotherClass (baris : 19 - 22). MyanotherClass sendiri mewarisi class oop2. Baru, method callprotected yang public dapat di panggil oleh \$obj1 (baris : 28).

### - Private properties dan methods

Private : Method dan property hanya dapat di akses oleh class itu sendiri. Class lain dan class yang mengextends nya tidak dapat mengakses. Pada contoh source code di atas, jika baris 4 di ganti menjadi: private function penjumlahan(\$a,\$b), maka saat program di eksekusi, output nya akan keluar pesan error, sbb :

Fatal error: Call to private method oop2::penjumlahan() from context 'MyanotherClass' in /opt/lampp/htdocs/belajar-raharja/oop2.php on line 21 karena method penjumlahan di panggil oleh class MyanotherClass, yang merupakan class eksternal, walaupun class tersebut mewarisi class MyClass, class nya method penjumlahan.

### - Static properties dan method

Sebuah method atau property yang di deklarasikan static, dapat langsung di akses tanpa melakukan instantiate class. Anda hanya memberikan nama class, scope operator resolusi dan nama property atau method.

Contoh:

```

1.<?php
2.class oop2
3. {
4.     public static function penjumlahan($a,$b)
5.     {
6.         $c=$a+$b;

```

```

7.     return $c;
8.     }
9.}
10.
11. echo oop2::penjumlahan(10,7);
12. ?>

```

Output :  
17

### VI.3. Memberikan komentar dengan DocBlocks

Seringkali kita menjumpai komentar yang ada di atas sebuah class. Komentar di perlukan untuk memberikan informasi mengenai sejumlah informasi terhadap class atau file tersebut, seperti : author, copyright dan lisensi. Komentar di dalam PHP diawali dengan tag `/*` dan diakhiri dengan `*/` atau jika perbaris, dapat diawali dengan `//`. Komentar tidak di proses oleh komputer.

Cara memberikan komentar dengan DocBlocks, sudah di terima secara meluas.

Tag yang paling umum di gunakan oleh Docblock adalah :

`@author` : Pencipta dari class atau file . Format nya adalah nama pencipta dan e-mail.

Contoh :

`@author` Akhmad Sofwan (sofwan@sofwan.net)

`@Copyright` : Berisi tahun copyright dan nama pemegang copyright.

Contoh :

`@Copyright` 2015 Akhmad Sofwan

`@license` : Link license ke file tersebut dan nama license.

Contoh :

`@license` <https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html> GPL 2

### VI.4. Membandingkan Procedural dengan Object Oriented Programming (OOP)

Procedural bagus diterapkan di dalam pemrograman, namun di sisi lain OOP menawarkan kelebihan mudahan dalam berhubungan dengan data. Sebagai contoh, anda memiliki informasi mengenai orang yang perlu diproses, informasi tersebut adalah nama, pekerjaan dan usia. Terdapat beberapa

- **keunggulan OOP dibanding procedural, yaitu :**

#### 1. Kemudahan implementasi

Object di dalam OOP dapat menyimpan data secara internal, variabel tidak perlu dilewatkan dari function ke function. Contoh source code procedural dan OOP dibawah ini, menjelaskan hal tersebut.

```

1.<?php
2. function changeJob($person, $newjob)
3.{
4.     $person['job'] = $newjob; // Change the person's job
5.     return $person;
6.}
7.
8. function happyBirthday($person)
9.{
10.     ++$person['age']; // Add 1 to the person's age
11.     return $person;
12. }
13.
14. $person1 = array(
15.     'name' => 'Tom',
16.     'job' => 'Button-Pusher',

```

```

17.'age' => 34
18.);
19.
20.$person2 = array(
21.'name' => 'John',
22.'job' => 'Lever-Puller',
23.'age' => 41
24.);
25.
26.// Output the starting values for the people
27.echo "<pre>Person 1: ", print_r($person1, TRUE), "</pre>";
28.echo "<pre>Person 2: ", print_r($person2, TRUE), "</pre>";
29.
30.// Tom got a promotion and had a birthday
31.$person1 = changeJob($person1, 'Box-Mover');
32.$person1 = happyBirthday($person1);
33.
34.// John just had a birthday
35.$person2 = happyBirthday($person2);
36.
37.// Output the new values for the people
38.echo "<pre>Person 1: ", print_r($person1, TRUE), "</pre>";
39.echo "<pre>Person 2: ", print_r($person2, TRUE), "</pre>";
40. ?>

```

Output :

```

Person 1: Array
(
    [name] => Tom
    [job] => Button-Pusher
    [age] => 34
Person 2: Array
(
    [name] => John
    [job] => Lever-Puller
    [age] => 41
)
Person 1: Array
(
    [name] => Tom
    [job] => Box-Mover
    [age] => 35
)
Person 2: Array
(
    [name] => John
    [job] => Lever-Puller
    [age] => 42
)

```

Pada kode procedural di atas, array yang menyimpan atribut person harus di lewatkan dan dikembalikan dari masing-masing pemanggilan function yang dapat meninggalkan margin untuk error.

Untuk mengatasi hal ini, hendaknya hanya informasi yang penting saja untuk operasi saat ini yang di lewatkan ke functions atau metode. Hal ini di akomodir oleh OOP, dengan kode sebagai berikut :

```

1.<?php
2.class Person
3. {
4.     private $_name;
5.     private $_job;
6.     private $_age;
7.     public function __construct($name, $job, $age)
8.     {
9.         $this->_name = $name;
10.        $this->_job = $job;
11.        $this->_age = $age;
12.    }
13.
14.    public function changeJob($newjob)
15.    {
16.        $this->_job = $newjob;
17.    }
18.
19.    public function happyBirthday()
20.    {
21.        ++$this->_age;}
22.    }
23.
24. // Create two new people
25. $person1 = new Person("Tom", "Button-Pusher", 34);
26. $person2 = new Person("John", "Lever Puller", 41);
27.
28. // Output their starting point
29. echo "<pre>Person 1: ", print_r($person1, TRUE), "</pre>";
30. echo "<pre>Person 2: ", print_r($person2, TRUE), "</pre>";
31.
32. // Give Tom a promotion and a birthday
33. $person1->changeJob("Box-Mover");
34. $person1->happyBirthday();
35.
36. // John just gets a year older
37. $person2->happyBirthday();
38.
39. // Output the ending values
40. echo "<pre>Person 1: ", print_r($person1, TRUE), "</pre>";
41. echo "<pre>Person 2: ", print_r($person2, TRUE), "</pre>";
42. ?>

```

Output :

Person 1: Person Object

```

(
  [_name:Person:private] => Tom
  [_job:Person:private] => Button-Pusher
  [_age:Person:private] => 34
)

```

Person 2: Person Object

```

(
  [_name:Person:private] => John
  [_job:Person:private] => Lever Puller
  [_age:Person:private] => 41
)

```

Person 1: Person Object

```
(  
[_name:Person:private] => Tom  
[_job:Person:private] => Box-Mover  
[_age:Person:private] => 35  
)
```

Person 2: Person Object

```
(  
[_name:Person:private] => John  
[_job:Person:private] => Lever Puller  
[_age:Person:private] => 42 )
```

OOP membutuhkan waktu untuk memahami dan menerapkannya, namun saat sudah implementasi, kemudahan dalam pengorganisasian kode dan data akan kita dapatkan, terutama di dalam aplikasi yang cukup besar. Walau begitu, tidak semua hal di dalam pemrograman ideal untuk diterapkannya OOP, hal hal kecil di dalam program, seringkali justru lebih baik diterapkan di dalam procedural.

## **2. Organisasi struktur program yang lebih baik**

Class-class di dalam OOP dapat dibuat menjadi sebuah file yang di dalam file tersebut dapat dibuat menjadi 1 fitur atau lebih namun yang berdekatan. Hal itu memudahkan jika kita ingin menggunakan kembali fitur-fitur tersebut, di tempat lain dalam sebuah program, yakni hanya tinggal memanggilnya kembali.

## **3. Mudah dalam pemeliharaan**

Bayangkan, jika di tengah-tengah pemrograman, kita ingin menambahkan parameter di dalam sebuah kode. Di dalam procedural, kita harus menambahkan parameter tersebut secara manual dari function ke function. Di dalam OOP, kita tinggal membuat parameter yang bersifat publik dan menambahkan metode untuk menangani fitur baru yang ingin ditambahkan [3].

## BAB VII

### PEMBUATAN PROGRAM SEDERHANA BERBASIS PHP

Pada Pertemuan ke VII akan di bahas pembuatan program sederhana Berbasis PHP. Pembuatan program dalam bentuk soal dan jawab.

#### 1. Program 1.

Buatlah program dengan PHP untuk mencari sisi miring dan keliling pada segitiga siku-siku. Entri alas (a) dan tinggi (b) dengan Keyboard.

Rumus sisi miring :  $c = \sqrt{a^2 + b^2}$

Rumus Keliling :  $k = a + b + c$

Jika nilai keliling  $> 100$  , maka keterangan adalah “Segitiga Besar”

Jika nilai keliling  $\geq 50$  dan  $< 100$ , maka keterangan adalah “Segitiga Sedang”

Jika nilai keliling  $> 0$  dan  $< 50$ , maka keterangan adalah “Segitiga Kecil”

Contoh :

Alas (a) : 4

Tinggi (b) : 6

(Enter)

Sisi Miring : 8.94

Keliling : 18.94

Keterangan : Segitiga kecil

#### Nama file : keliling.php

```
1.<!DOCTYPE html>
2.<html>
3.<head>
4. <title>Segitiga</title>
5.</head>
6.<body>
7. <table>
8. <form method="post" action="hasil.php">
9. <tr>
10. <td>Alas(a)</td><td>:</td><td><input type="text" name="valas" size="5" /></td>
11.</tr>
12.<tr>
13. <td>Tinggi(b)</td><td>:</td><td><input type="text" name="vtinggi" size="5" /></td>
14. </tr>
15. <tr>
16. <td colspan="3"><input type="submit" value="OK" /></td>
17.</tr>
18.</form>
19.</table>
20.</body>
21.</html>
```

**Nama File : hasil.php**

```
1.<?php
2. $xalas=$_POST["valas"];
3. $xtinggi=$_POST["vtinggi"];
4. $xsisi_miring=sqrt(($xalas*$xalas)+($xtinggi*$xtinggi));
5. $xkeliling=$xalas+$xtinggi+$xsisi_miring;
6.
7.      // Cari keterangan //
8. if ($xkeliling > 100)
9. {
10.     $keterangan="Segitiga Besar";
11. }
12. else if ($xkeliling >=50 and $xkeliling <100)
13. {
14.     $keterangan="Segitiga Sedang";
15. }
16.else if ($xkeliling >0 and $xkeliling < 50)
17. {
18.     $keterangan="Segitiga kecil";
19. }
20.
21.     echo "Nilai Alas : $xalas <br/>";
22.     echo "Nilai Tinggi: $xtinggi <br/>";
23. echo "----- <br/>";
24. echo "Sisi Miring : ".round($xsisi_miring,2)."<br/>";
25. echo "Keliling   : ".round($xkeliling,2)."<br/>";
26. echo "Keterangan  : $keterangan <br/>";
27. echo "<input type='button' value='Kembali' onClick='history.back(1)' />";
28.??>
```

**2. Program 2.**

Buatlah program untuk mencetak deret dari X ke max Y dengan lompatan Z. Jika X lebih kecil dari Y, maka merupakan deret naik, dengan pencetakan di awali karakter : ">". Jika X lebih besar dari Y, maka merupakan deret turun, dengan pencetakan di awali karakter : "<". Dan jika tidak ada deret yang dibuat, maka tidak mencetak apapun. X, Y dan Z diinput melalui keyboard

**Contoh 1 :**

X : 4  
Y : 11  
Z : 2

Output :

> 4 6 8 10

**Contoh 2 :**

X : 13  
Y : 3  
Z : 4

Output :  
< 13 9 5

**Jawab :**

```
1.<!DOCTYPE html>
2.<html>
3.<head>
```



```
4. <title>Pertemuan 7-2</title>
5.</head>
6.<body>
7.<table>
8.<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>" >
9.<tr>
10. <td>X</td><td>:</td><td><input type="text" name="vx" size="5" /></td>
11.</tr>
12.<tr>
13. <td>Y</td><td>:</td><td><input type="text" name="vy" size="5" /></td>
14.</tr>
15.<tr>
16. <td>Z</td><td>:</td><td><input type="text" name="vz" size="5" /></td>
17.</tr>
18.<tr>
19. <td><input type="submit" value="Proses" /></td>
20.</tr>
21.</form>
22.</table>
23.<?php
24. if (isset($_POST["vx"]) && isset($_POST["vy"]) && isset($_POST["vz"]))
25. {
26.     $zx=$_POST["vx"];
27.     $zy=$_POST["vy"];
28.     $zz=$_POST["vz"];
29.
30.     if ($zx<$zy)
31.     {
32.         echo "> ";
33.         for ($i=$zx;$i<=$zy;$i=$i+$zz)
34.         {
35.             echo $i." ";
36.         }
37.     }
38.     else if ($zx>$zy)
39.     {
40.         echo "< ";
41.         for ($j=$zx;$j>=$zy;$j=$j-$zz)
42.         {
43.             echo $j." ";
44.         }
45.     }
46. }
47. ?>
48.</body>
49.</html>
```

## Bab IX

### Operasi String dan File

#### IX.I. Operasi String

Berikut akan di uraikan beberapa fungsi yang berkaitan dengan operasi String.

##### 1. substr

Syntax : string substr (string \$string, int \$start [,int \$length])

Fungsi : Untuk mengambil bagian dari sebuah string.

Contoh :

1. <?php
2. \$nama = "Akhmad Sofwan";
3. \$ambil1 = substr (\$nama,6); //Mengambil 6 karakter dari kanan
4. \$ambil2 = substr (\$nama,0,6); // Mengambil 6 karakter dari posisi 0 (awal)
5. echo "Part 1 : \$ambil1 <br/>";s
6. echo "Part 2 : \$ambil2";
7. ?>

Output :

Part 1 : Sofwan

Part2 : Akhmad

##### 2. strtolower

Syntax : string strtolower (string \$str)

Fungsi : Merubah string menjadi huruf kecil

Contoh:

1. <?php
2. \$kata = "Bandung Lautan Api";
3. echo strtolower(\$kata);
4. ?>

Output :

bandung lautan api

##### 3. strtoupper

Syntax : string strtoupper (string \$str)

Fungsi : Merubah string menjadi huruf besar

Contoh :

1. <?php
2. \$kata = "Bandung Lautan Api";
3. echo strtoupper(\$kata);
4. ?>

Output :

BANDUNG LAUTAN API

##### 4.strlen

Syntax : int strlen (string \$string)

Fungsi : Menghitung jumlah karakter pada sebuah string

Contoh :

1. <?php
2. \$kata = "Bandung Lautan Api";
3. echo strlen(\$kata);
4. ?>

Output :

18

## 5. **strpos**

Syntax : int strpos (string \$string, mixed \$needle[, int \$offset = 0])

Keterangan

\$string : String untuk pencarian

mixed \$needle : Elemen yang di cari

\$offset : Awal pencarian. Namun hasil pencarian tetap dari posisi awal.

Fungsi : Mencari posisi awal dari string yang di cari.

Contoh :

```
1. <?php
2.     $kata = "Bandung Lautan Api";
3.     $cari = "u";
4.     $dicari=strpos($kata,$cari);
5.     echo "Hasil : $dicari";
6. ?>
```

Output :

Hasil : 4

## 6. **implode**

Syntax : string implode (string \$glue, array \$pieces)

Keterangan

\$glue : Berisi string yang akan di gabungkan dengan elemen array

array \$pieces : Array , yang elemen nya akan di gabung kan.

Fungsi : Untuk menggabungkan string dengan elemen array

Contoh :

```
1. <?php
2.     $array = array('lastname','email','phone');
3.     $comma_separated = implode(" ", $array);
4.     echo $comma_separated; // lastname,email,phone
5.     // Empty string when using an empty array:
6.     var_dump(implode('hello', array())); // string(0) ""
7. ?>
```

Output :

lastname\*email\*phone

string(0) ""

## 7. **explode**

Syntax : array explode (string \$delimiter, string \$string [,int \$limit])

Keterangan :

\$delimiter : string yang akan menjadi pemisah

\$string : inputan string

\$limit : batas

Fungsi : Untuk memisahkan string dengan string

```
1. <?php
2.     //Example 1
3.     $pizza="piece1 piece2 piece3 piece4 piece5 piece6";
4.     $pieces=explode(" ", $pizza);
5.     echo $pieces[0]; //piece1
6.     echo $pieces[1]; //piece2
7.
```

```

8. //Example 2
9. $data="foo.*:1023:1000:/home/foo:/bin/sh";
10. list($user,$pass,$uid,$gid,$gecos,$home,$shell)=explode(":",$data);
11. echo $user; //foo
12. echo $pass; //*
13. ?>

```

Output :

```

piece1 piece2
foo*

```

## **Operasi file**

Berikut akan diuraikan beberapa syntax di dalam operasi file .

### **1. fopen**

Syntax : resource fopen (string \$filename, string \$mode);

Keterangan :

\$filename : Berupa nama file

\$mode : Tipe akses file

Beberapa mode yang dapat di gunakan :

'r' : Hanya membuka, meletakkan file pointer di awal file

'r+' : Membuka untuk membaca dan menulis. Meletakkan pointer di awal file

'w' : Di buka untuk di tulis. Meletakkan pointer file di awal file dan memotong panjang file menjadi 0.

Jika file tidak ada, maka file akan di buat.

'w+' : Membuka untuk di baca dan di tulis. Meletakkan pointer file di awal file dan memotong panjang file

menjadi 0. Jika file tidak ada, maka file akan di buat.

Contoh :

```

1. <?php
2. $buka = fopen ("/home/sofwan/test.txt","r");
3. ?>

```

### **2. fread**

Syntax : string fread (resource \$handle, int \$length);

Keterangan :

\$handle : Pointer file , biasa nya di buat dengan menggunakan fopen

\$length : Sampai panjang jumlah byte yang di baca

Contoh :

```

1. <?php
2. //get contents of a file into a string
3. $filename="/usr/local/something.txt";
4. $handle=fopen($filename,"r");
5. $contents=fread($handle,filesize($filename));
6. fclose($handle);
7. ?>

```

### **3. fwrite**

Syntax : int fwrite (resource \$handle, string \$string [, int \$length])

Keterangan :

\$handle : File system pointer, biasa nya di bentuk oleh fopen.

\$string : String yang akan di tulis.

Length : panjang byte max yang akan di tulis.

Fungsi : Untuk menulis string ke dalam sebuah file.

```
1. <?php
2. $fp=fopen('/home/sofwan/test.txt','w+');
3. fwrite($fp,'1');
4. fwrite($fp,'23');
5. fclose($fp);
6.
7. // the content of 'data.txt' is now 123 and not 23!
8. ?>
```

## **BAB X**

### **Koneksi PHP dengan Database menggunakan PDO**

PHP Data Object (PDO) extension adalah interface yang ringan dan konsisten untuk mengakses berbagai macam database di PHP. Setiap Driver database yang menerapkan interface PDO dapat mengekspose fitur khusus database sebagai fungsi ekstensi regular. Dengan menggunakan PDO, anda dapat membuat code yang portabel/sama untuk Database dan Platform yang berbeda.

Perbedaan hanya pada driver dan koneksi saja. Untuk retrieve, delete, update dan insert, anda dapat menggunakan script yang sama atau hampir sama untuk database dan platform yang berbeda.

PDO sudah tersedia sejak PHP 5.1 dan di kenal sebagai PECL Extension untuk PHP 5.0. PDO membutuhkan fitur OO yang ada di PHP 5.x, sehingga PDO tidak kompatibel pada PHP 4.x dan versi di bawah nya.

PDO mendukung Database sbb :

- DBLIB : FreeTDS / Microsoft SQL Server / Sybase
- Firebird : Firebird / Interbase 6
- IBM (IBM DB2)
- Informix – IBM Informix Dynamic Server
- Mysql : Mysql 3.x / 4.0
- OCI : Oracle Call Interface
- ODBC : ODBC v3 (IBM DB2 dan UnixODBC)
- PGSQL : PotgreSql
- SqlLite : SQLite 3.x

Untuk mengakses setiap Database di atas, di butuhkan driver yang di kenal oleh PHP. Anda dapat melihat di phpinfo driver-driver apa saja yang sudah terinstall dan di kenal oleh PHP di system anda, atau anda dapat menjalankan kode berikut [1]:

```
1.<?php
2. foreach(PDO::getAvailableDrivers() as $driver)
3. {
4.     echo $driver.'<br />';
5. }
6. ?>
```

Untuk dapat menjalankan driver database pada PHP, anda dapat menggunakan parameter sbb (Untuk Mysql, PostgreSQL dan Sqlite) :

- Mysql : --with-pdo-mysql=mysqlnd
- PostgreSQL : --with-pdo-pgsql=/opt/lampp/postgresql
- Sqlite : --with-pdo-sqlite

Untuk mengaktifkan driver-driver database lain, pada pengguna Windows, dapat meng enable dll file database-database tsb pada php.ini, seperti :

```
extension=php_pdo.dll
extension=php_pdo_firebird.dll
extension=php_pdo_informix.dll
extension=php_pdo_mssql.dll
```

## X.1 Menghubungkan ke Database

### Mysql

Mysql adalah database yang paling banyak di gunakan oleh Programmer PHP dan mendapat dukungan penuh dari PHP. Contoh menghubungkan Mysql dari PHP dengan PDO adalah sbb :

```
1.<?php
2.$dsn = 'mysql:host=localhost;dbname=test';
3.$username = 'opan';
4.$password = 'opansan';
5.
6. try {
7. $dbh = new PDO($dsn, $username, $password);
8. echo "Database connected";
9. }
10.catch (PDOException $e)
11. {
12. echo $e->getMessage();
13. }
14. ?>
```

### PostgreSQL

```
1.<?php
2. try
3. {
4. $db = new PDO("pgsql:dbname=pdo:host=localhost","username","password");
5. echo "PDO Connection object created";
6. }
7. catch (PDOException $e)
8. {
9. echo $e->getMessage();
10. }
11. ?>
```

## X.2 Insert, Select dan Update

Seperti yang sudah di sebutkan di atas, bahwa keunggulan PDO adalah, bahwa sekali database terkoneksi, maka untuk masalah query, itu sama, antara 1 database dengan database lain, sehingga anda tidak perlu di repotkan saat ingin mengganti database. Tentu saja selama database tersebut di akses dengan bahasa SQL. Walaupun ada perbedaan, namun hanya sedikit, menyesuaikan gaya SQL pada masing-masing database. Untuk penggunaan query, kita menggunakan database Mysql.

### Insert

Sebelum anda mempraktekan query, buatlah tabel sbb :

Nama kolom	Tipe	Keterangan
id	int (4)	Primary Key, Auto Increment
nim	varchar (30)	
nama	varchar (30)	
alamat	varchar (30)	

Pada script di atas, tambahkanlah script untuk menginsert, yakni :

```
$dbh->exec ("insert into mahasiswa (nim,nama,alamat) values ('1211500100','Budiman','Jl.Mawar No.14
```