

# Pertemuan 06

## Pengenalan Penggunaan Database di PHP

### Pendahuluan

Nah di pertemuan kemarin kita sudah bahas nih cara membuat database, table, nambah data, hapus data lewat PhpMyAdmin yang sudah disediakan oleh XAMPP, sekarang kita akan menggunakan Database tersebut di PHP.

### Membuka Koneksi

Sebelum kita memakai Database Mysql kita harus membuka koneksi antara server kita dengan server database, di PHP terdapat macam-macam cara untuk meng-akses Database, tetapi kita fokus di fitur PHP PDO karena fitur ini dapat digunakan di berbagai macam Database (seperti Postgree atau Sqlite).

Untuk membuka koneksi kita mendeklarasikan variabel untuk menyimpan class object PDO yang berisikan data-data koneksi kita, pada argummen pertama saat mendeklarasikan PDO dengan keyword new kita harus mengisi DNS Database, terus argummen kedua kita mengisikan username yang akan digunakan di Database tersebut tetapi dapat dikosongkan, lalu di argummen ketiga kita mengisi password username tersebut tetapi dapat dikosongkan, yang terakhir tetapi hanya opsional ini berisikan konfigurasi tambahan.

Di <https://www.php.net/manual/en/class.pdo>

```
class PDO {  
  
    /* Methods */  
    public __construct(  
        string $dsn,  
        ?string $username = null,  
        ?string $password = null,  
        ?array $options = null  
    )  
}
```

Setiap Database memiliki DNS yang berbeda, kita hanya fokus di Mysql.

```
1 <?php  
2  
3 $pdo = new PDO('mysql:host=localhost;dbname=nama_database', 'root', '');  
4
```

DNS-nya Mysql seperti di-atas, karena Mysql memiliki sistem akun kita harus memberikan username dan password-nya tetapi biasanya memiliki username root, karena kita menggunakan XAMPP password-nya username root ini kosong.

Setelah kita definisikan seperti ini kita coba cek di browser kita apakah error atau tidak, kalau tidak berarti koneksi-nya sudah jalan dengan baik.

## Raw Query

Nah setelah kita melakukan koneksi kita dapat melakukan query, query yang kita lakukan ini Raw Query dikarenakan keamanan-nya sangat jelek ketika query ini dapat dimanipulasikan oleh user input (seperti query search).

Untuk melakukan query tersebut kita tinggal menggunakan method query yang sudah disediakan oleh class PDO tadi.

untuk contoh kali ini kita menggunakan database *try* dan menggunakan table *users*.

Untuk mendapatkan data.

```
3 $pdo = new PDO('mysql:host=localhost;dbname=try', 'root', '');
4 $query = $pdo->query('SELECT * FROM users');
5 $query->execute();
6 var_dump($query->fetchAll());
7
```

Hasil var\_dump():

```
array (size=3)
  0 =>
    array (size=6)
      'id' => int 1
      0 => int 1
      'name' => string 'Test Users' (length=10)
      1 => string 'Test Users' (length=10)
      'address' => string 'Test' (length=4)
      2 => string 'Test' (length=4)
  1 =>
    array (size=6)
      'id' => int 2
      0 => int 2
      'name' => string 'Test Users 2' (length=12)
      1 => string 'Test Users 2' (length=12)
      'address' => string 'users' (length=5)
      2 => string 'users' (length=5)
  2 =>
    array (size=6)
      'id' => int 3
      0 => int 3
      'name' => string 'Test User 3' (length=11)
      1 => string 'Test User 3' (length=11)
      'address' => string 'User' (length=4)
      2 => string 'User' (length=4)
```

nah ini kita melakukan query lalu menggunakan bahasa Sql untuk melakukan komunikasi dengan database (karena database sql bukan nosql), lalu kita *execute*, lalu kita ambil data yang dihasilkan oleh query tersebut dengan *fetchAll*.

Kita dapat menggunakan method *query* ini untuk berbagai hal, seperti menambah data, menghapus data, edit data, kita tinggal isi Sql statement-nya di dalam argumen *query*.

Note : method `fetchAll()` itu digunakan untuk mengambil data lebih dari 1, dan method `fetch()` hanya digunakan untuk mengambil 1 data, jika query kita menghasilkan lebih dari satu tetapi kita ambil menggunakan method `fetch()` kita hanya mendapatkan 1 data.

### Prepared Statement

Prepared statement mirip seperti raw query tetapi lebih aman, karena sebelum di kirimkan ke database, sesuai standar kita harus memberikan parameter query-nya lewat method `bindParam`.

Contoh mengambil user yang ber-id 1 :

```
1 <?php
2
3 $userId = 1;
4
5 $pdo = new PDO('mysql:host=localhost;dbname=try', 'root', '');
6 $query = $pdo->prepare('SELECT * FROM users WHERE id=:id');
7 $query->bindParam('id', $userId, PDO::PARAM_INT);
8 $query->execute();
9 var_dump($query->fetchAll());
10
```

Hampir sama dengan method query tetapi beda-nya ada method `bindParam`, method ini digunakan untuk keamanan ketika handle data yang diberikan oleh user, argummen pertama merupakan nama yang akan di bind (kalau `id` berarti di query-nya `:id`), lalu variabel-nya, lalu yang terakhir tipe data-nya.

Tipe datanya ada :

- |                             |                       |
|-----------------------------|-----------------------|
| 1. String                   | : PDO::PARAM_STR      |
| 2. Integer / Float / Double | : PDO::PARAM_INT      |
| 3. Boolean (bool)           | : PDO::PARAM_BOOL     |
| 4. Char                     | : PDO::PARAM_STR_CHAR |

hasil `var_dump`

```
array (size=1)
  0 =>
    array (size=6)
      'id' => int 1
      0 => int 1
      'name' => string 'Test Users' (length=10)
      1 => string 'Test Users' (length=10)
      'address' => string 'Test' (length=4)
      2 => string 'Test' (length=4)
```

Note : Kalau ingin in-depth tentang SQL bisa check di <https://www.w3schools.com/sql/>

Note : Bahasa Pemrograman SQL itu strict.

Contoh :

Kita ingin membuat website sederhana untuk registrasi user, lalu kita juga menampilkan list user yang sudah registrasi dibawahnya

Misal kita punya database belajar-ngoding  
terus memiliki tabel user yang memiliki schema

ID	int	Auto Increment Primary
nama	varchar(255)	
email	varchar(255)	
password	varchar(255)	

lalu kita buat index.php

pertama kita buat form-nya dulu

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Document</title>
9 </head>
10
11 <body>
12     <h2>Registrasi</h2>
13     <form action="" method="POST">
14         <input type="text" name="nama">
15         <input type="email" name="email">
16         <input type="password" name="password">
17         <input type="submit" value="Registrasi" name="register">
18     </form>
19     <h2>List user yang sudah daftar</h2>
20     <ul>
21         <li>Nama User</li>
22     </ul>
23 </body>
24
25 </html>
26
```

Setelah kita buat html kita bisa membuat phpnya

kita buat instansi PDO untuk menyalakan koneksinya

```
$pdo = new PDO('mysql:host=localhost;dbname=belajar-ngoding', 'root', '');
```

```
1 <?php
2
3 // Buat instansi PDO Mysql
4 $pdo = new PDO('mysql:host=localhost;dbname=belajar-ngoding', 'root', '');
```

lalu ketika user submit form-nya kita bisa gunakan global variabel `$_POST`

```

6 // Ketika key 'register' di global variabel $_POST ada~
7 if (isset($_POST['register'])) {~
8     ....// buat query tambah user~
9     ....$queryTambahUser = $pdo->prepare('INSERT INTO users (nama, email, password) VALUES (:nama, :email, :password)');~
10    ~
11    ....// bind parameternya~
12    ....$queryTambahUser->bindParam('nama', $_POST['nama'], PDO::PARAM_STR);~
13    ....$queryTambahUser->bindParam('email', $_POST['email'], PDO::PARAM_STR);~
14    ....$queryTambahUser->bindParam('password', $_POST['password'], PDO::PARAM_STR);~
15    ~
16    ....// eksekusi query tersebut~
17    ....echo $queryTambahUser->execute();~
18 }~

```

isset itu bermaksud apakah variabel ini ada, kalau konteks diatas apakah key register di dalam variabel \$\_POST ada

Lalu kita bisa buat query seluruh user

```

// buat query untuk ambil semua user~
$query = $pdo->prepare('SELECT * FROM users');~
~
// eksekusi query tersebut~
$query->execute();~
~
// ambil semua hasil~
$users = $query->fetchAll();~
~
?>~

```

gambar keseluruhan kode

```
1 <?php
2
3 // Buat instansi PDO Mysql
4 $pdo = new PDO('mysql:host=localhost;dbname=belajar-ngoding', 'root', '');
5
6 // Ketika key 'register' di global variabel $_POST ada
7 if (isset($_POST['register'])) {
8     // buat query tambah user
9     $queryTambahUser = $pdo->prepare('INSERT INTO users (nama, email, password) VALUES (:nama, :email, :password)');
10
11     // bind parameternya
12     $queryTambahUser->bindParam('nama', $_POST['nama'], PDO::PARAM_STR);
13     $queryTambahUser->bindParam('email', $_POST['email'], PDO::PARAM_STR);
14     $queryTambahUser->bindParam('password', $_POST['password'], PDO::PARAM_STR);
15
16     // eksekusi query tersebut
17     echo $queryTambahUser->execute();
18 }
19 // buat query untuk ambil semua user
20 $query = $pdo->prepare('SELECT * FROM users');
21
22 // eksekusi query tersebut
23 $query->execute();
24
25 // ambil semua hasil
26 $users = $query->fetchAll();
27
28 ?>
29 <!DOCTYPE html>
30 <html lang="en">
31
32 <head>
33     <meta charset="UTF-8">
34     <meta http-equiv="X-UA-Compatible" content="IE=edge">
35     <meta name="viewport" content="width=device-width, initial-scale=1.0">
36     <title>Document</title>
37 </head>
38
39 <body>
40     <h2>Registrasi</h2>
41     <form action="" method="POST">
42         <input type="text" name="nama" placeholder="nama">
43         <input type="email" name="email" placeholder="email">
44         <input type="password" name="password" placeholder="password">
45         <input type="submit" value="Registrasi" name="register">
46     </form>
47     <h2>List user yang sudah daftar</h2>
48     <ul>
49         <?php foreach ($users as $user) : ?>
50             <li><?=$user['nama'] ?></li>
51         <?php endforeach; ?>
52     </ul>
53 </body>
54
55 </html>
```