**BAB 3**

**TRIK JITU YII FRAMEWORK**

Pada bab 3 ini kita akan membahas trik-trik yang ada di yiiframework. Sebut saja seperti widget, renderpartial, extensions, components, dll. Semua akan kita bahas dengan menggunakan teknik yang ada di yii framework, sehingga akan memperdalam tentang pengetahuan kita mengenai cara-cara bekerja dengan yii framework.

1. **Widget dan render partial**

Apa itu widget dalam yii framework? Dikutip dari website resmi yii bahwa widget adalah sebuah contoh dari CWidget atau anak kelas dari CWidget. Ini adalah komponen yang terutama untuk tujuan penyajian. Widget biasanya tertanam dalam tampilan script untuk menghasilkan data yang komplek, namun ditampilkan untuk halaman itu sendiri. Sebagai contoh, widget kalender bisa dipakai untuk menyiapkan antar muka kalender. Widget memfasilitasi usabilitas baik dalam kode antarmuka pengguna. Sedangkan render partial merupakan suatu teknik yang digunakan untuk menampilkan file view yang akan di ambil dari folder view lainnya atau file view memanggil file view lainnya. Jadi kalau di controller biasa disebut dengan render, dan sedangkan view memanggil view disebut render partial. Sebagai contoh kita akan menampilkan form di bagian interface A, dan form tersebut terdapat di file view lain. Maka kita dapat memanggilnya dengan teknik render partial.

* + 1. **Tab**

Pada subbab ini kita akan belajar membuat tab dengan menggunakan widget. Baiklah langsung saja kita buat tab nya. Silahkan anda buat Controller dengan nama **WidrenpartController.php** kemudian anda ketikkan kode berikut:

**protected/controllers/WidrenpartController.php**

<?php

class WidrenpartController extends Controller{

*/\*set layout menjadi null*

*sehingga akan tanpa layout (putih)\*/*

public $layout='null';

*//action tab*

public function actionTab(){

*/\*render file view*

*protected/views/widrenpart/tab.php\*/*

$this->render('tab');

}

}

?>

Setelah kita membuat WidrenpartController beserta kodenya, selanjutnya kita membuat view nya. Oke, jika kita membuat controller pastinya kita akan membuat folder baru di protected/views dengan nama sama dengan nama controller yang kita buat. Silahkan buat folder dengan nama widrenpart di protected/views, kemudian buatlah file baru dengan nama **tab.php**. berikut kode untuk file tab.php.

**Skrip protected/views/widrenpart/tab.php :**

<?php

$this->widget('zii.widgets.jui.CJuiTabs',array(

//untuk membuat tab beserta isi nya

'tabs'=>array(

'TAB 1'=>'Ini adalah tab dengan data statis',

'TAB 2'=>'Ini adalah tab dengan data statis juga',

),

*// option tambahan*

'options'=>array(

'collapsible'=>true,

//'event'=>'mouseover',

),

));

?>

Penjelasan kode :

* $this->widget('zii.widgets.jui.CJuiTabs' ini kita gunakan untuk menggunakan widget dengan format $this->widget(zii.widget.jui.nama\_widget
* 'tabs'=>array(

'TAB 1'=>'Ini adalah tab dengan data statis',

'TAB 2'=>'Ini adalah tab dengan data statis juga',

), ini digunakan untuk membuat tab beserta isinya dengan format

‘judul tab’=>’isi tab’

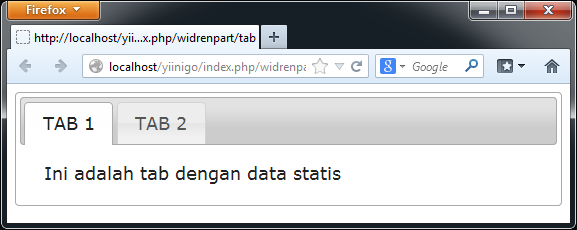
* 'options'=>array(

'collapsible'=>true,

), ini digunakan untuk menambahkan option pada tab, misal kita

ingin membuka ta pada saat mouse over, maka kita tambahkan event.

Untuk melihat hasilnya silahkan anda ketikkan url pada broswer anda <http://localhost/yiinigo/index.php/widrenpart/tab>, maka akan tampil seperti gambar 3.1.



Gambar 3.1 Tampilan tab

* + 1. **Tab dengan render partial**

Pada subbab ini kita akan membuat tab dengan render partial. Dimana isi dari tab akan kita ambil dari file view lain. Kita disini bisa juga parsing variabel untuk di render ke file view tersebut. Untuk langkahnya silahkan anda buat file baru dan simpan di dalam folder protected/views/widrenpart/ dengan nama **\_tabRender.php**. dan ketikkan kode ini:

<?php

*//digunakan untuk menampilkan nilai $data*

*//yang di parsing*

echo $data;

?>

Langkah selanjutnya silahkan buka kembali file tab.php dan silahkan modifikasi kodenya seperti text yang ditebalkan berikut:

**Skrip protected/views/widrenpart/tab.php :**

<?php

$this->widget('zii.widgets.jui.CJuiTabs',array(

//untuk membuat tab beserta isi nya

'tabs'=>array(

'TAB 1'=>'Ini adalah tab dengan data statis',

'TAB 2'=>'Ini adalah tab dengan data statis juga',

**'Render Partial'=>$this->renderPartial(**

**'\_tabRender',array('data'=>'isi data'),TRUE**

**),**

),

*// option tambahan*

'options'=>array(

'collapsible'=>true,

//'event'=>'mouseover',

),

));

?>

Penjelasan kode :

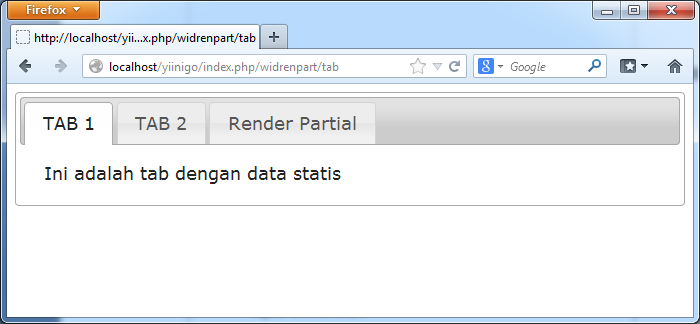
* **'Render Partial'=>$this->renderPartial(**

**'\_tabRender',array('data'=>'isi data'),TRUE**

**),** inilah kode yang digunakan untuk memanggil file view lainnya atau yang disebut dengan render partial.

* \_tabRender merupakan nama file viewnya. Disini penamaan filenya menggunakan underscore (\_), tidak diwajibkan menggunakan underscore. Anda bisa menamainya dengan tanpa underscore.
* Dan terdapat **array**('data'=>'isi data')dimana data mengandung nilai dan akan di parsing ke view \_tabRender sehingga dapat di panggil dengan perintahecho $data**.**

Sekarang silahkan refresh browser anda, dan hasilnya akan tampak seperti gambar 3.2



Gambar 3.2 Tab dengan render partial

* + 1. **Accordion**

Kali ini kita akan membuat accordion dengan widget. Langsung saja kita menuju langkah-langkahnya. Silahkan anda buka kembali WidrenpartController.php dan silahkan modifikasi kodenya seperti tampak pada text yang ditebalkan berikut.

**Skrip WidrenpartController.php**

<?php

class WidrenpartController extends Controller{

*//set layout menjadi null sehingga akan tampil putih*

public $layout='null';

*//action tab*

public function actionTab(){

*//render file view*

*//protected/views/widrenpart/tab.php*

$this->render('tab');

}

*//action accordion*

**public function actionAccordion(){**

**$this->render('accordion');**

**}**

}

?>

Untuk penjelasan kode diatas, nampaknya sudah jelas yah.. dan sekarang kita lanjut ke langkah berikutnya. Silahkan anda buat filew view baru dan save as ke folder protected/views/widrenderpart dengan nama accordion.php, dan silahkan ketikkan kode berikut.

<?php

$this->widget('zii.widgets.jui.CJuiAccordion',array(

'panels'=>array(

'AC 1'=>'ini adalah isi accordion 1',

'AC 2'=>'accordion 2 besert isinya',

),

*// option tambahan*

'options'=>array(

'animated'=>'slide',

),

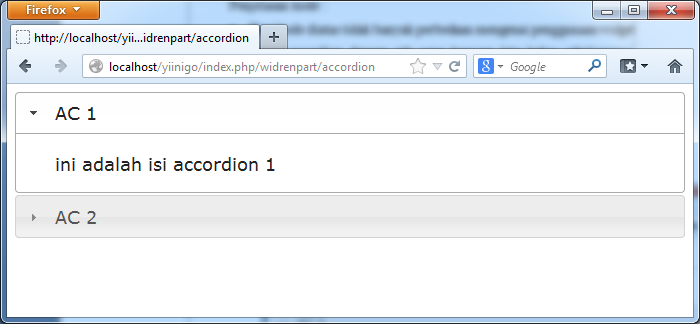
));

?>

Penjelasan kode :

* Dari kode diatas tidak banyak perbedaan mengenai penggunaan widget untuk accordion dengan tab yang barusan kita bahas sebelumnya. Hanya saja untuk membuat accordion menggunakan panels untuk bagian accordionnya.

Silahkan ketikkan <http://localhost/yiinigo/index.php/widrenpart/accordion> pada browser anda untuk melihat hasilnya. Jika benar akan tampak seperti gambar 3.3.

****

**Gambar 3.3 Tampilan accordion**

* + 1. **Accordion dengan render partial**

Kali ini kita akan membuat accordion dengan render partial. Langsung saja kita ke langkah-langhnya. Silahkan anda buat file view baru dengan dan save as di folder protected/views/widrenpart dengan nama **\_accordionRender.php**. lalu ketikkan kode berikut:

**Skrip \_accordionRender.php**

<?php

echo $data;

?>

Lalu kemudian, silahkan buka kembali file view accordion.php, lalu silahkan modifikasi kodenya seperti text yang ditebalkan berikut:

<?php

$this->widget('zii.widgets.jui.CJuiAccordion',array(

'panels'=>array(

'AC 1'=>'ini adalah isi accordion 1',

'AC 2'=>'accordion 2 besert isinya',

//render partial

**'AC 3'=>$this->renderPartial(**

**'\_accordionRender',array('data'=>'isi data render**

**acc render'),TRUE),**

**),**

*//option tambahan*

'options'=>array(

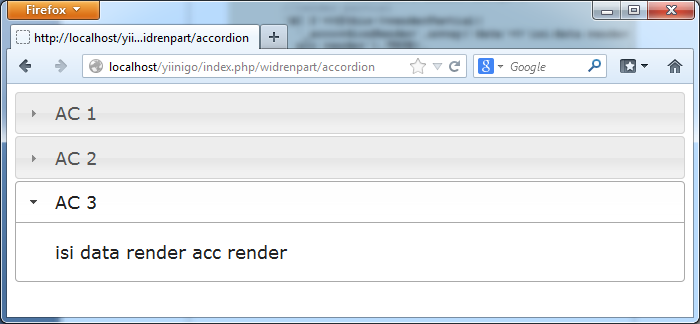
'animated'=>'slide',

),

));

?>

Untuk render partial tampaknya udah jelas. Karena kode untuk render partial nya sama dengan yang kita bahas di subbab 3.1.2. untuk melihat hasilnya silahkan refresh browser anda, lalu jika benar tampilannya seperti gambar 3.4.



Gambar 3.4 Accordion dengan render partial

1. **Arti extensions dalam yiiframework**

Extensions dalam yii framework merupakan sebuah library external atau group file yang dapat digunakan kembali untuk project sekarang atau yang akan datang. Dan extensions biasanya disediakan oleh pihak ke tiga. Contoh, sebuah facebook menghungkan library anda, sebuah library yang dikembangkan untuk menghubungkan ke API ebay, sebuah library untuk menggunakan google map API, dll.

Dalam subbab ini kita akan belajar bagaimana menggunakan extensions yang ada.

1. **Yii extensions dan widget**
   * 1. **Dropdown menu**
     2. **Tree menu**
     3. **Autocomplete**
     4. **Thumb image dan crop image**
     5. **Masked image**
     6. **FB-Comment**
     7. **Livegridview**
     8. **Yii smart captcha**
     9. **JwPlayer**
     10. **EjwPlayer**
     11. **Back to top**
     12. **Tile slide menu**
     13. **Pinterest-feed**
2. **Yii components**

Kalau kita lihat, di dalam folder yii yang telah kita install terdapat folder protected\components. Apa guna folder tersebut? Dan Apa itu components dalam yii??. Components adalah suatu class yang dapat membantu kita untuk melakukan sebuah proses tertentu. Dan components merupakan turunan dari Ccomponent. Jadi, component merupakan class yang kita buat sendiri atau class yang kita adopsi dari class lain yang berguna untuk menangai proses-proses tertentu dalam program. Sebagai contoh, saya membuat class untuk keperluan convert mata uang, tanggal, dan untuk keperluan proses lainnya. jadi class tersebut dapat kita letakkan di dalam folder components.

* + 1. **Default Component**

Pada saat kita telah menginstall yii framework terdapat default components yang tersedia yaitu Controller dan component ini selalu kita extends pada saat kita membuat controller baru pada pelajaran-pelajaran sebelumnya. Silahkan anda buka file protected\components\Controller.php, dan berikut kodenya.

<?php

*/\*\**

*\* Controller is the customized base controller class.*

*\* All controller classes for this application should extend from this base class.*

*\*/*

class Controller extends CController {

*/\*\**

*\* @var string the default layout for the controller view. Defaults to '//layouts/column1',*

*\* meaning using a single column layout. See 'protected/views/layouts/column1.php'.*

*\*/*

public $layout='//layouts/column1';

*/\*\**

*\* @var array context menu items. This property will be assigned to {@link CMenu::items}.*

*\*/*

public $menu = array();

*/\*\**

*\* @var array the breadcrumbs of the current page. The value of this property will*

*\* be assigned to {@link CBreadcrumbs::links}. Please refer to {@link CBreadcrumbs::links}*

*\* for more details on how to specify this property.*

*\*/*

public $breadcrumbs = array();

}

Penjelasan kode:

* Kalau kita lihat terdapat kode Class Controller extends Ccontroller. Jadi component ini meng-extends Ccontroller yang merupakan induk dari controller.
* Public $layout=’//layout/column1’ ini digunakan untuk menentukan default layout yang kita gunakan yaitu column1.php yang terdapat di dalam folder views\layouts. Dan juga variabel inilah yang membuat layout menjadi default pada saat kita tidak mendeklarasikan apa-apa untuk menggunakan layout yang mana saat kita membuat controller. Jadi layout terpasang pada controller yang kita extends.
* Public $menu = array() ini digunakan untuk menampilkan menu yang disajikan dalam bentuk array yang disajikan di column2.php yang ada di folder views\layouts.
* Public $breadcrumbs = array(); ini digunakan untuk menampilkan breadcrumbs yang menginformasikan kepada pengguna, sedang berada di bagian mana dari website yang dia kunjungi.
  + 1. **Cara membuat dan menggunakan component**
       1. **Membuat component**

Pada subbab ini kita akan belajar bagaimana membuat components, yang mana nantinya akan selalu kita extends pada saat membuat controller selanjutnya. Langsung saja, silahkan anda buat file baru dan simpan di dalam folder protected\components dengan nama YiinigoController.php. Dan silahkan ketikkan kode berikut:

**Skrip protected\components\YiinigoController.php**

<?php

*//kita membuat class dengan nama YiinigoController extends Controller*

class YiinigoController extends Controller{

*//..... disinilah tempat kita membuat function*

}

?>

**Penjelasan kode:**

* Dari kode diatas kita membuat Class dengan nama YiinigoController yang meng-extends Controller. Jadi component ini nantinya tetat terhubung dengan component Controller dan variabel-variabel yang ada didalamnya tetap dapat digunakan tanpa kurang sedikitpun. ☺
  + - 1. **Membuat function sederhana**

Baiklah kita akan membuat function-function sederhana pada componen YiinigoController yang barusan kita buat tadi. Silahkan anda buka YiinigoController.php dan silahkan tambahkan kode seperti yang ditebalkan berikut:

Skrip protected\components\YiinigoController.php

<?php

/\*kita membuat class dengan nama YiinigoController extends Controller\*/

class YiinigoController extends Controller {

//..... disinilah tempat kita membuat function

**//function untuk convert mata uang IDR**

**public function idrCurency($val) {**

**return number\_format($val, 0, '', '.');**

**}**

}

?>

**Penjelasan kode :**

* **Public function idrCurency($val){…}** kita buat untuk mengkonvert angka ke mata uang IDR. Dan terdapat kode **return number\_format($val, 0, '', '.');** merupakan untuk mengembalikan nilai dari suatu function. Dan number\_format(..) digunakan untuk conversi angka.
  + - 1. **Cara-cara menggunakan components**

Ada beberapa cara dalam menggunakan components yang telah kita buat. Yaitu dengan cara menggunakan **New**, di **Extends**, **dan bisa menggunakan tanda ::**. Baiklah saya akan menjelaskan satu persatu cara penggunaannya. Silahkan perhatikan potongan kode-kode berikut:

1. **Menggunakan component dengan teknik extends:**

<?php

class LessoncomponentController extends YiinigoController {

public function actionMoney() {

/\*

\* panggil function idrCurency

\* oleh karena sudah di extends YiinigoController

\* maka pemanggilannya cukup menggunakan $this->...

\*/

echo $this->idrCurency(7500000);

}

}

?>

Penjelasan :

* Teknik ini menggunakan extends dimana Class LessoncomponentController memanggil components YiinigoController dengan extends.
* Setelah di extends kita dapat menggunakan function-function yang ada di dalam component dengan cara

$this->nama\_function($param). Contoh diatas menggunakan $this->idrCurency(7500000).

* Yang harus diperhatikan dalam pemanggilan component pada teknik ini bahwa kompoent yang dipanggil harus meng-extends ke component default yang digunakkan yii. Lihat lagi skrip YiinigoController.php. kalau tidak meng-extends ke components default akan terjadi error.

1. **Menggunakan component denggan teknik New dan :: :**

Silahkan buat component baru dengan nama Mycomponent.php dan berikut kode nya:

**Skrip protected\components\Mycomponent.php**

<?php

class Mycomponent{

//membuat function untuk format tanggal

public function formatDate($param){

//format tanggal ke tgl-nmbln-thn

return date('m-F-Y', strtotime($param));

}

//membuat function untuk format jam

public function formatTime($param){

//format jam ke format AM/PM

return date("g:i:s A", strtotime($param));

}

}

?>

Cara penggunaannya: silahkan modifikasi file controller Lessoncomponent.php seperti teks yang ditebalkan berikut:

<?php

class LessoncomponentController extends YiinigoController {

public function actionMoney() {

/\*

\* panggil function idrCurency

\* oleh karena sudah di extends YiinigoController

\* maka pemanggilannya cukup menggunakan $this->...

\*/

echo $this->idrCurency(7500000);

}

**//format tanggal**

**public function actionDate(){**

**//deklarasi tanggal**

**$date = date('Y-m-d');**

**/\*memanggil component Mycomponent**

**function formatDate\*/**

**echo Mycomponent::formatDate($date);**

**}**

**//format jam**

**public function actionWhattime(){**

**//memanggil componen Mycomponent**

**$callTime = new Mycomponent;**

**$time = '14:34:00';**

**/\*tampilkan jam dengan format yang**

**telah ditentukan\*/**

**echo $callTime->formatTime($time);**

**}**

}

?>

* + 1. **Cara menambahkan component dari library lain**

1. **DAO (Data Access Object) Custom QUERY**

DAO kepanjangan dari Data Access Objects, dimana dengan fitur ini kita dapat menggunakan custom query atau tanpa harus mengubah kode SQL. Dalam pengunaan DAO terdiri dari empat kelas berikut yang akan kita bahas pada subbab ini:

* CDbConnection: mewakili koneksi ke database.

Yii::app()->db;

* CDbCommand: mewakili pernyataan SQL untuk mengeksekusi terhadap database.

Yii::app()->db->createCommand(…)

* CDbDataReader: untuk mengeksekusi query yang dibuat createCommand(…).
* Yii::app()->db->execute();

Digunakan untuk excute non query seperti delete, update,insert.

* Yii::app()->db->queryAll();

Digunakan untuk menghasilkan semua data.

* Yii::app()->db->queryRow();

Digunakan untuk menghasilkan 1 baris data saja.

* Yii::app()->db->queryColumn ();

Digunakan untuk menghasilkan data dari kolom pertama dari hasil data yang ada. Contoh ada field suatu tabel: nama,alama,telp. Maka yang dapat ditampilkan adalah kolom pertama saja yaitu nama

* Yii::app()->db->queryScalar();

Digunakan untuk menghasilkan data dari baris pertama dan kolom pertama. Contoh dari field tabel : nama, alamat, telp. Maka jika kita ingin menampilkan satu baris data dan satu kolom pertama yaitu nama, maka kita dapat menggunakan queryScalar().

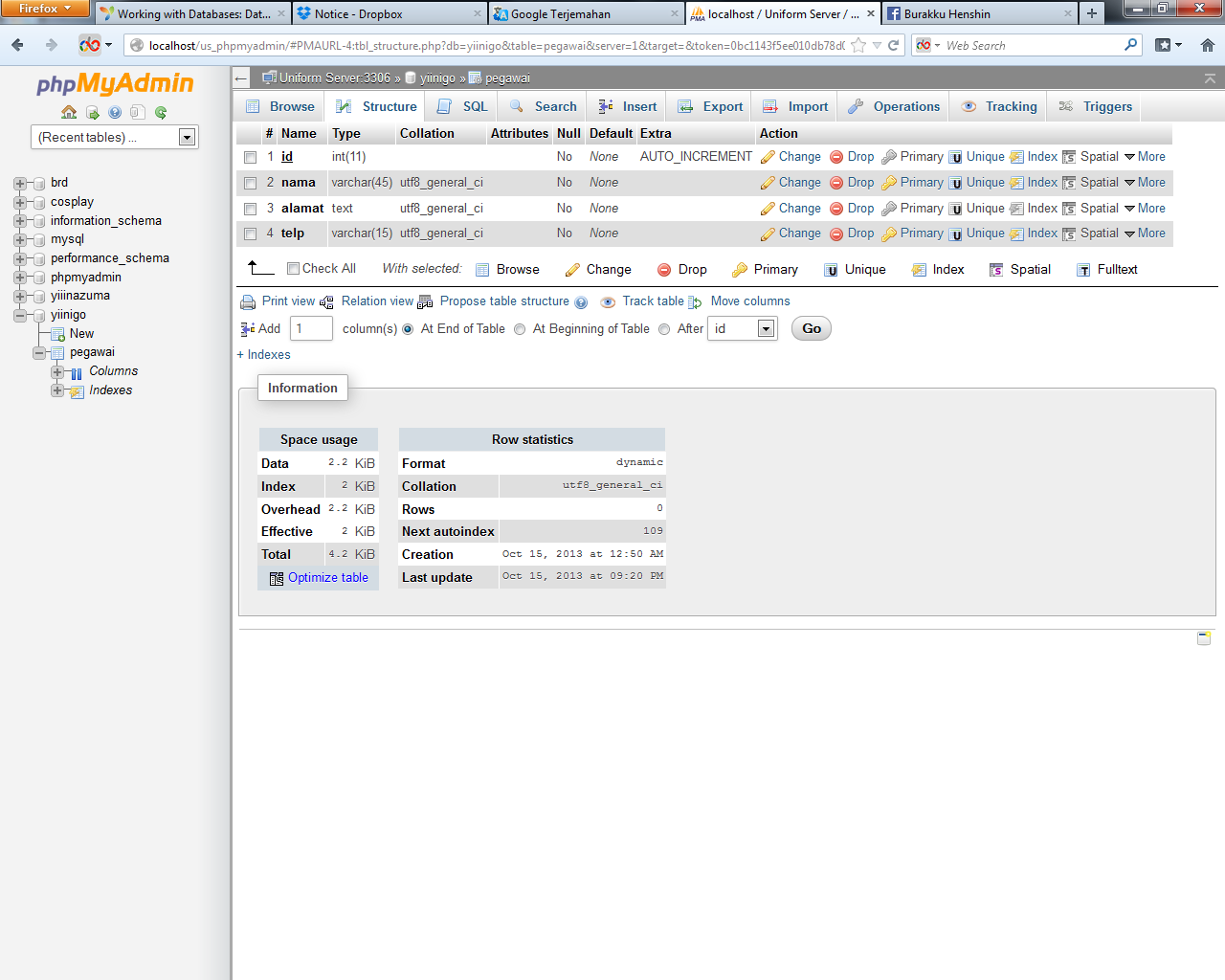
Untuk mengenai CDbDataReader diatas, saya selalu menggunakan queryAll() karena eksekusi yang dilakukan CDbReader lainnya bisa dilakukan dengan queryAll().

* CDbTransaction: mewakili transaksi DB.

CdbTransaction akan kita bahas pada subbab selanjutnya.

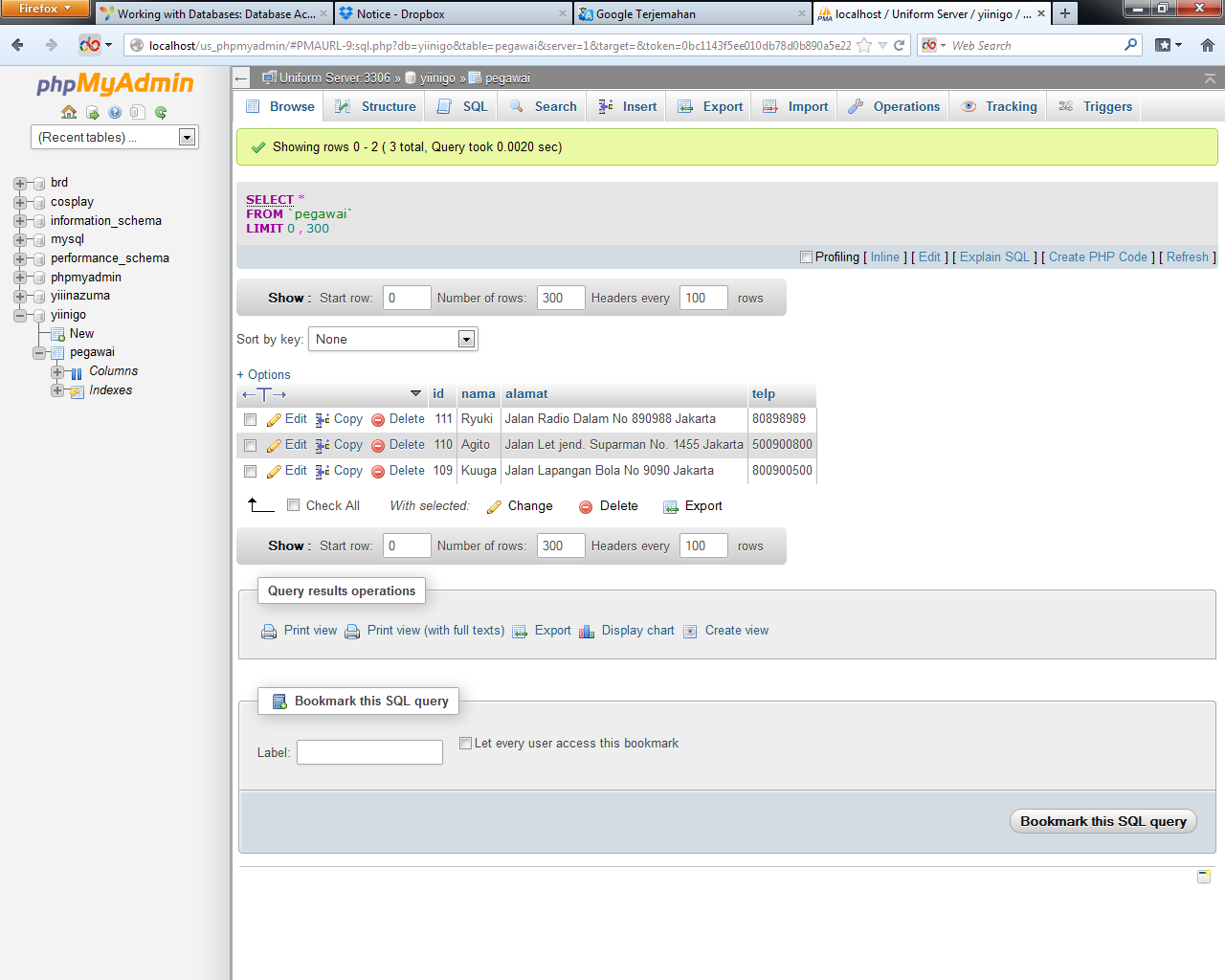
* 1. **Persiapan database dan tabel**

Silahkan anda buat database dengan nama **yiinigo** dan tabel dengan nama **pegawai**. Dan berikut stuktur tabelnya:



**Gambar 3.5 struktur tabel pegawai**

Setelah anda membuat tabel diatas. Silahkan isikan beberapa data pada tabel pegawai tersebut. Silahkan lihat contoh yang terlihat pada gambar 3.6.



**Gambar 3.6 Isi tabel pegawai**

* 1. **Yii::app()->db->queryAll()**

Kita akan belajar bagai mana menampilkan data yang ada pada tabel pegawai dengan menggunakan kode SQL biasa. Oke, silahkan anda buat Controller baru dengan nama LessondaoController.php dan seperti biasa simpan di folder protected\controller. Dan ketikkan kode berikut:

<?php

class LessondaoController extends YiinigoController {

public $layout ='null';

public function actionIndexpegawai() {

/\*buat/declare sql untuk select semua data pegawai

Dan ditampung oleh $sql

\*/

$sql = "SELECT \* FROM pegawai ORDER BY id ASC";

/\*digunakan untuk mengkoneksikan ke databse\*/

$conncection = Yii::app()->db;

/\*membuat command/query\*/

$command = $conncection->createCommand($sql);

/\*membaca query dan akan dapat di tampilkan semua

datanya\*/

$results = $command->queryAll();

/\*render ke file view nya indexpegawai.php\*/

$this->render('indexpegawai', array(

/\*dataProvider menampung data dari $results dan

akan di foreach di file view\*/

'dataProvider' => $results)

);

}

}

?>

Penjelasan tambahan:

* Yii::app()->db fungsi untuk memanggil component yang yang ada di main.php yang berfungsi untuk mengkoneksikan ke databse. Perhatikan potongan kode berikut:

'db' => array(

'connectionString' =>

'mysql:host=localhost;dbname=yiinigo',

),

Pada saat kita mengkoneksikan database kita men-setting component ini yang ada di main.php. jadi component main inilah yang di panggil Yii::app()->db.

* $conncection->createCommand($sql); digunakan untuk membuat command sql dimana didalamnya terdapat query sql layaknya mysql\_query() pada php biasa.
* $results = $command->queryAll(); digunakan untuk membaca semua query dan data akan dapat ditampilkan semuanya, dari kolom pertama sampai terakhir, record pertama sampai terakhir semua dapat di tampilkan.

Kemudian silahkan buat file view, dan seperti biasa untuk sebuah controller akan memiliki satu folder view dengan nama sesuai dengan nama controllernya. Silahkan anda buat folder pada views dengan nama lessondao. Dan buat file dengan nama indexpegawai.php dan ketikkan kode berikut:

**Skrip : views\lessondao\indexpegawai.php**

<table border="1">

<tr>

<th>Nama</th>

<th>Alamat</th>

<th>No. Telp</th>

</tr>

<?php foreach ($dataProvider as $pegawai):?>

<tr>

<td><?php echo $pegawai['nama'];?></td>

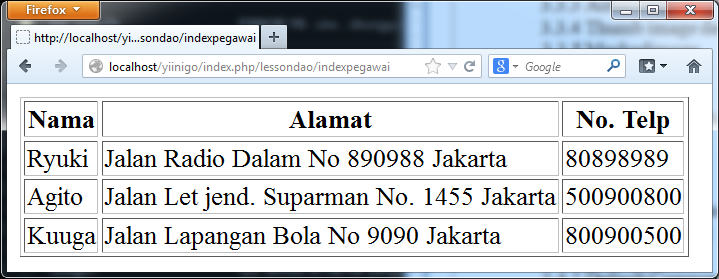
<td><?php echo $pegawai['alamat'];?></td>

<td><?php echo $pegawai['telp'];?></td>

</tr>

<?php endforeach;?>

</table>

Lalu silahkan jalankan di browser anda dengan mengetik alamat url <http://localhost/yiinigo/index.php/lessondao/indexpegawai>, dan akan terlihat seperti gambar 3.7.

**Gambar 3.7 Tampil data pegawai**

Penjelasan kode:

* Kalau kita lihat untuk menampilkan data tidak jauh berbeda dengan cara menggunakan Cactiverecord. Hanya saja untuk menampilkan data berdasarkan nama fieldnya disini harus menggunakan $data[‘nama\_field’]. Dan jika menggunakan CactiveRecor harus menggunakan $data->nama\_field.
  1. **Yii::app()->db->queryRow()**

Seperti penjelasan sebelumnya queryRow() digunakan untuk menampilkan satu baris data saja. Seperti query select \* from where id=@id.

Oke, untuk prakteknya silahkan anda modifikasi file indexpegawai.php yang ada di folder views\lessondao seperti text yang ditebalkan padakode berikut:

Skrip protected\views\lessondao\indexpegawai.php

**<a href="<?php echo $this->createUrl('add');?>">Tambah pegawai</a>**

<table border="1">

<tr>

<th>Nama</th>

<th>Alamat</th>

<th>No. Telp</th>

**<th>&nbsp;</th>**

</tr>

<?php foreach ($dataProvider as $pegawai):?>

<tr>

<td><?php echo $pegawai['nama'];?></td>

<td><?php echo $pegawai['alamat'];?></td>

<td><?php echo $pegawai['telp'];?></td>

**<td><a href="<?php echo $this->createUrl('view',array('id'=>$pegawai['id']));?>">View</a> |**

**<a href="<?php echo $this->createUrl('delete',array('id'=>$pegawai['id']));?>">Delete</a> |**

**<a href="<?php echo $this->createUrl('update',array('id'=>$pegawai['id']));?>">Update</a>**

**</td>**

</tr>

<?php endforeach;?>

</table>

**Penjelasan kode:**

* **$this->createUrl('view',array('id'=>$pegawai['id']))** digunakan untuk membuat url yang mengarah ke action view.
* **$this->createUrl('delete',array('id'=>$pegawai['id']))** digunakan untuk membuat url yang mengarah ke action delete.
* **$this->createUrl('update',array('id'=>$pegawai['id']))** digunakan untuk membuat url yang mengarah ke action update.

Kemudian silahkan tambahkan action baru pada LessondaoController seperti text yang ditebalkan berikut:

Skrip protected\controllers\LessondaoController.php

<?php

class LessondaoController extends YiinigoController {

public $layout = 'null';

public function actionIndexpegawai() {

/\* buat sql untuk select semua data pegawai \*/

$sql = "SELECT \* FROM pegawai ORDER BY id ASC";

/\* digunakan untuk mengkoneksikan ke databse \*/

$conncection = Yii::app()->db;

/\* membuat command/query \*/

$command = $conncection->createCommand($sql);

/\* membaca query dan akan dapat di tampilkan semua

datanya \*/

$results = $command->queryAll();

/\* render ke file view nya indexpegawai.php \*/

$this->render('indexpegawai', array(

'dataProvider' => $results)

);

}

**//action untuk view data**

**public function actionView($id) {**

**$sql = "SELECT \* FROM pegawai WHERE id='$id'";**

**$connection = Yii::app()->db;**

**$command = $connection->createCommand($sql);**

**/\*untuk read data hanya satu baris\*/**

**$results = $command->queryRow();**

**$this->render('view', array(**

**'dataProvider' => $results)**

**);**

**}**

}

?>

Kemudian silahkan buat file view baru pada folder protected\views\lessondao dengan nama view.php, dan ketikkan kode berikut:

Skrip protected\views\lessondao\view.php

<table>

<tr>

<td>#ID</td>

<td>:</td>

<td><?php echo $pegawai['id'];?></td>

</tr>

<tr>

<td>Nama</td>

<td>:</td>

<td><?php echo $pegawai['nama'];?></td>

</tr>

<tr>

<td>Alamat</td>

<td>:</td>

<td><?php echo $pegawai['alamat'];?></td>

</tr>

<tr>

<td>No. Telp</td>

<td>:</td>

<td><?php echo $pegawai['telp'];?></td>

</tr>

<tr>

<td>

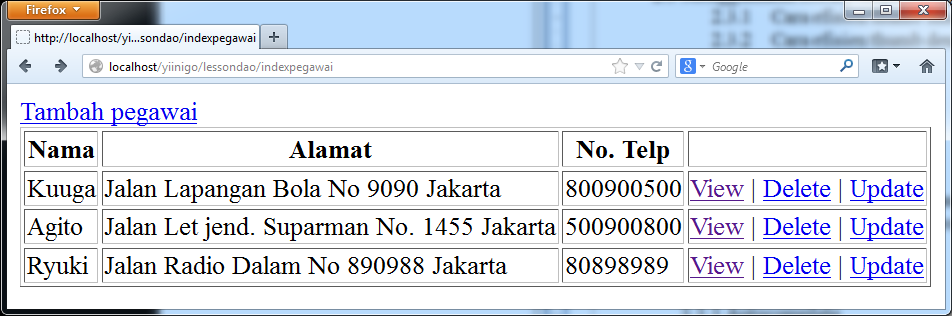
<a href="<?php echo $this->createUrl('indexpegawai');?>">Kembali ke index</a>

</td>

</tr>

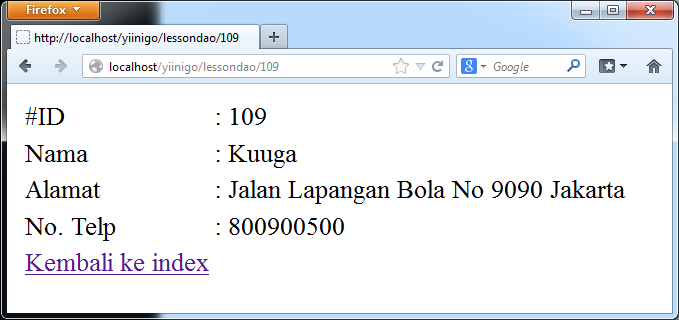
</table>

Kemudian untuk melihat hasilnya silahkan anda jalankan kembali url <http://localhost/yiinigo/lessondao/indexpegawai> lalu akan tampil seperti gambar 3.8.

****

**Gambar 3.8 Tampil data pegawai beserta link action**

Seperti yang terlihat pada gambar 3.8 silahkan anda klik link view nya yang terdapat pada kolom terakhir dan akan tampil seperti gambar 3.9.

****

**Gambar 3.9 Tampil data pegawai secara spesifik**

* 1. **Yii::app()->db->execute()**

Yii::app()->db->execute() atau yang biasa disebut dengan execute non query ini digunakan untuk mengeksekusi query yang tidak perlu mengembalikan data yang di eksekusinya untuk ditampilkan. Seperti insert data, update data, delete data. Untuk prakteknya kita akan membuat kode untuk delete data dan mari kita buka kembali controller LessondaoController lalu silahkan tambahkan actionDelete seperti kode berikut.

**Skrip protected\controller\LessondaoController.php**

//kode sebelumnya . . .

//action untuk delete data

public function actionDelete($id){

$sql ="DELETE FROM pegawai WHERE id='$id'";

$connection = Yii::app()->db;

$command = $connection->createCommand($sql);

/\*eksekusi query atau execute non query\*/

$command->execute();

/\*redirect ke indexpegawai\*/

$this->redirect(array('indexpegawai'));

}

Dan untuk melihat hasilnya, silahkan anda buka kembali url <http://localhost/yiinigo/lessondao/indexpegawai>. lalu silahkan klik link delete nya maka data akan ter-delete dan akan otomatis di redirect kembali ke halaman indexpegawainya.

* 1. **Insert data (DAO)**

Untuk membuat insert data kita masih menggunakan Yii::app()->db->execute() silahkan tambahkan actionAdd() pada LessondaoController dan berikut kodenya:

public function actionAdd() {

if ($\_POST) {

//ambil data dari masing - masing attributes

$nama = $\_POST['nama'];

$alamat = $\_POST['alamat'];

$telp = $\_POST['telp'];

//perintah simpan data

$sql = "INSERT INTO pegawai(nama,alamat,telp) VALUES('$nama','$alamat','$telp')";

$connection = Yii::app()->db;

//execute query simpan data

$command = $connection->createCommand($sql);

/\* eksekusi query atau execute non query \*/

$command->execute();

$this->redirect(array(

'indexpegawai')

);

}

$this->render('add');

}

Kemudian silahkan buat file view nya dengan nama **add.php** dan berikut adalah kodenya:

**Skrip protected\views\lessondao\add.php**

<form action="" method="post">

<table>

<tr>

<td>Nama</td>

<td>:</td>

<td><input type="text" name="nama"></td>

</tr>

<tr>

<td>Alamat</td>

<td>:</td>

<td><textarea name="alamat"></textarea></td>

</tr>

<tr>

<td>Telp</td>

<td>:</td>

<td><input type="text" name="telp"></td>

</tr>

<tr>

<td>**&nbsp;**</td>

<td>**&nbsp;**</td>

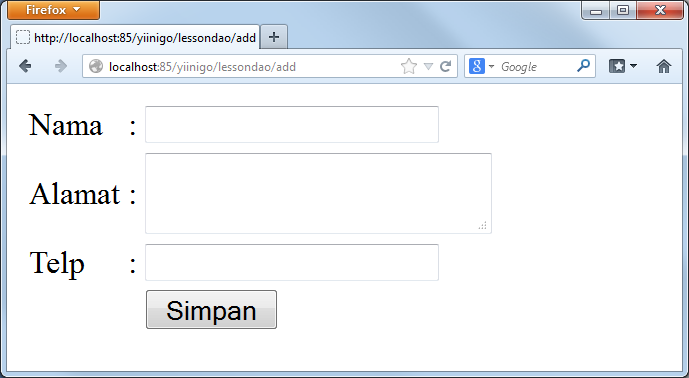
<td><input type="submit" value="Simpan"></td>

</tr>

</table>

</form>

Kemudian, untuk melihat hasilnya silahkan anda ketikkan kembali url <http://localhost/yiinigo/lessondao/indexpegawai> pada browser anda dan silahkan klik tombol Tambah pegawai. Lalu akan tampil seperti gambar 3.10.



* 1. **Update data (DAO)**
  2. **Paging data (DAO)**
  3. **Searching data (DAO)**

**Gambar 3.10 form add data**

* 1. **Update Data (DAO)**

Untuk update data dengan dao kita tetap menggunakan Yii::app()->db->execute(). Silahkan tambahkan action baru pada LessondaoController dengan nama actionUpdate(). Dan berikut kodenya.

Skrip protected\controllers\LessondaoController.php

public function actionUpdate($id) {

$connection = Yii::app()->db;

if ($\_POST) {

$nama = $\_POST['nama'];

$alamat = $\_POST['alamat'];

$telp = $\_POST['telp'];

$sql = "UPDATE pegawai SET

nama='$nama',alamat='$alamat',telp='$telp' WHERE

id='$id'";

$command = $connection->createCommand($sql);

/\* eksekusi query atau execute non query \*/

$command->execute();

$this->redirect(array(

'indexpegawai')

);

}

$results = $connection->createCommand("SELECT \* FROM pegawai

WHERE id='$id'")->queryRow();

$this->render('update',array(

'pegawai'=>$results,

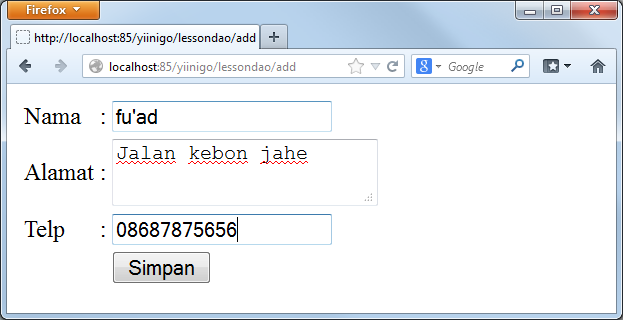
));

}

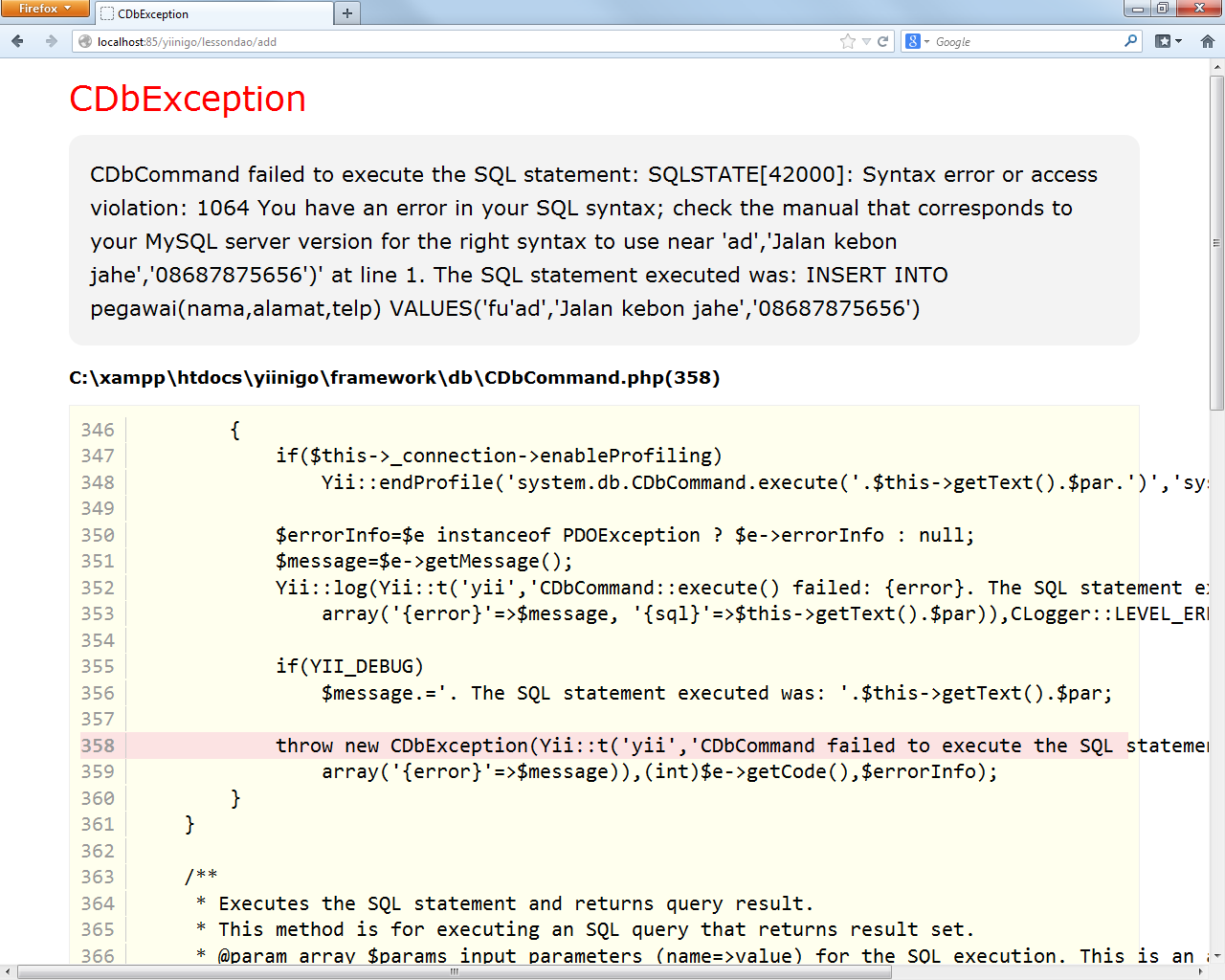
1. **SQL Injection**

SQL Injection adalah sebuah teknik yang digunakan untuk menghancurkan lapisan basis data yang terdapat dalam aplikasi. Cara ini biasanya menggunakan karakter-karater tertentu yang mengakibatkan terjadinya hal-hal yang tidak diharapkan.

Pada fitur CactiveRecord yang ada pada yii framework telah ddipasang secara otomatis untuk menangkal serangan ini. namun pada fitur DAO kita harus memasangnya sendiri. Sebagai contoh, coba anda simpan data dengan text yang mengandung tanda petik seperti yang terlihat pada gambar 3.11. lalu klik tombol simpan. Dan hasilnya akan tampak seperti gambar 3.12.

****

**Gambar 3.12 Simpan data dengan SQL Injection**

****

**Gambar 3.13 Error SQL Injection**

Dari gambar 3.11 dan 12 terlihat kita memasukkan satu karakter untuk SQL Injection. Dan tampak error, sehingga terlihatlah nama tabel beserta fielnya.

1. **Solusi SQL Injection**

Yii framework telah menyediakan fitur untuk menangani serangan sql injection dengan menggunakan fitur bindingParam(). Untuk menggunakannya kita bagi menjadi 2 subbab pada buku ini yaitu bindingaParam() dan bindingValue().

* + 1. **Binding Parameters.**

Mari kita terapkan pada saat menyimpan data. Silahkan modifikasi actionAdd yang ada pada LessondaoController, dan perhatikkan text yang ditebalkan seperti berikut:

Skrip Protected\controllers\LessondaoController.php

public function actionAdd() {

if ($\_POST) {

//ambil data dari masing - masing attributes

$nama = $\_POST['nama'];

$alamat = $\_POST['alamat'];

$telp = $\_POST['telp'];

//perintah simpan data

**$sql = "INSERT INTO pegawai(nama,alamat,telp) VALUES(:nama,:alamat,:telp)";**

$connection = Yii::app()->db;

//execute query simpan data

$command = $connection->createCommand($sql);

**//bindingParam untuk menangani karakter SQL injection**

**$command->bindParam(":nama",$nama,PDO::PARAM\_STR);**

**$command->bindParam(":alamat",$alamat,PDO::PARAM\_STR);**

**$command->bindParam(":telp",$telp,PDO::PARAM\_STR);**

/\* eksekusi query atau execute non query \*/

$command->execute();

$this->redirect(array(

'indexpegawai')

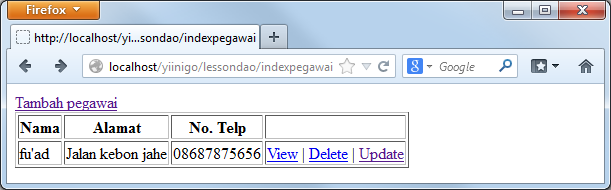
);

}

$this->render('add');

}

Setelah itu silahkan anda coba lagi menyimpan data nya seperti yang tampak pada gambar 3.12. jika berhasil akan tampak seperti gambar 3.14.



**Gambar 3.14 Karakter SQL injection berhasil disimpan**

Penjelasan kode:

* Terdapat perbedaan pada query penyimpanan $sql = "INSERT INTO pegawai(nama,alamat,telp) VALUES(:nama,:alamat,:telp)"; pada bagian VALUES ditandai dengan :nama,:alamat,:telp ini biasa disebut dengan placeholder yang akan dirujuk ke bindingParam.
* $command->bindParam(":nama",$nama,PDO::PARAM\_STR); digunakan untuk menggantikan placeholder :nama dengan nilai yang benar sehingga karakter sql injection bisa di tangani.
  + 1. **Binding Values**
    2. **sad**

1. **Membuat laporan data ke format excel**
2. **Membuat laporan data ke format pdf**