

# ADVANCED CAKEPHP

---

## Membina Panel Kawalan Pengguna

Azril Nazli Alias [ [azril.nazli@gmail.com](mailto:azril.nazli@gmail.com) ]

8/21/2010

<http://azrilnazli.blogspot.com>

<http://facebook.com/azrilnazli>

Sistem anda menggunakan Authentiction, tetapi bagaimana anda menentukan Access Level setiap User ?

Di dalam tutorial ini, anda akan diajar untuk membina sistem kawalan pengguna yang lengkap

- Group based User Roles
- Group Based Security
- Security Access Control Panel
- Cake Shell ACL script
- User Authentication
- CakePHP Schema Migration
- Konsep Access Control List

**Language**

- PHP
- Ruby
- Python
- J2ME

**Framework**

- CakePHP
- Ruby On Rails
- Symfony

**System Administration**

- Redhat Enterprise
- CentOS
- Ubuntu

**Database**

- MySQL
- PostgreSQL

**Ajax**

- JQuery
- Prototype
- ExtJS

**Scripting**

- HTML
- Javascript
- CSS

**Training**

- CakePHP ( Beginner, Advanced, Expert )
- PHP/MySQL ( Beginner, Advanced )
- Linux System Administration

M **019 371 8462**

E azril.nazli [at] Gmail.com

F facebook.com/azrilnazli

Logged in as super [My Account](#) | [Logout](#)

[Home](#)
[Posts](#)
[Groups](#)
[Users](#)
[Securities](#)

Actions
 

[Group List](#)
[Access List](#)

**Security Access for Administrator**

Pages		Groups		Posts	
Action	Choice	Action	Choice	Action	Choice
display	<input type="radio"/> allow <input type="radio"/> deny	security	<input type="radio"/> allow <input type="radio"/> deny	index	<input type="radio"/> allow <input type="radio"/> deny
		index	<input type="radio"/> allow <input type="radio"/> deny	view	<input type="radio"/> allow <input type="radio"/> deny
		view	<input type="radio"/> allow <input type="radio"/> deny	add	<input type="radio"/> allow <input type="radio"/> deny
		add	<input type="radio"/> allow <input type="radio"/> deny	edit	<input type="radio"/> allow <input type="radio"/> deny
		edit	<input type="radio"/> allow <input type="radio"/> deny	delete	<input type="radio"/> allow <input type="radio"/> deny
		delete	<input type="radio"/> allow <input type="radio"/> deny		

## Pengenalan

ACL atau Access Control List amat penting untuk mengawal pergerakan pengguna setelah mereka login ke dalam system. Saya akan membina satu system blog berasaskan tutorial Blog dari website CakePHP tetapi ditambah dengan ACL dan Auth.

Model yang terlibat

1. Group
2. User
3. Post

Hubungan antara Model

1. Group hasMany User
2. User belongsTo Group
3. Post belongsTo Group

Komponen yang terlibat

1. Auth
2. Acl

Skillset yang diperlukan

1. Advanced PHP / Basic CakePHP
2. Pengalaman menggunakan Auth
3. Pengguna tegar Cake CLI
4. Sudah biasa dengan konsep MVC CakePHP
5. Tahu untuk edit baked files ( Model, Controller dan View )

Persiapan sebelum memulakan tutorial

Advanced CakePHP : Membina Panel Kawalan Pengguna

1. Xampp Lite dengan sokongan mod\_rewrite dalam Windows XP
2. CakePHP versi 1.3.3 stable
3. Notepad ++ Editor

# Ciri-Ciri Sistem Blog ACL

---

Sistem blog ini mempunyai ciri-ciri berikut

1. Public boleh membaca blog
2. User boleh register dan akan dimasukkan sebagai group User
3. Usergroup mempunyai group berikut
  - SuperUser
  - Admin
  - User
  - Suspended
4. Setiap Group akan mempunyai set kawalan tersendiri , contohnya :
  - SuperUser mempunyai kuasa mutlak dan boleh akses kesemua sistem
  - Admin boleh manage User kecuali delete dan boleh manage Post
  - User boleh manage Post sendiri
  - Suspended tidak boleh buat apa-apa kecuali redirect ke Home sahaja
5. Panel kawalan untuk manage ACO ( Controllers/Action )
6. Panel kawalan untuk kawal Permission setiap ARO ( resource ) dan ACO ( requester )

# Installation & Setup

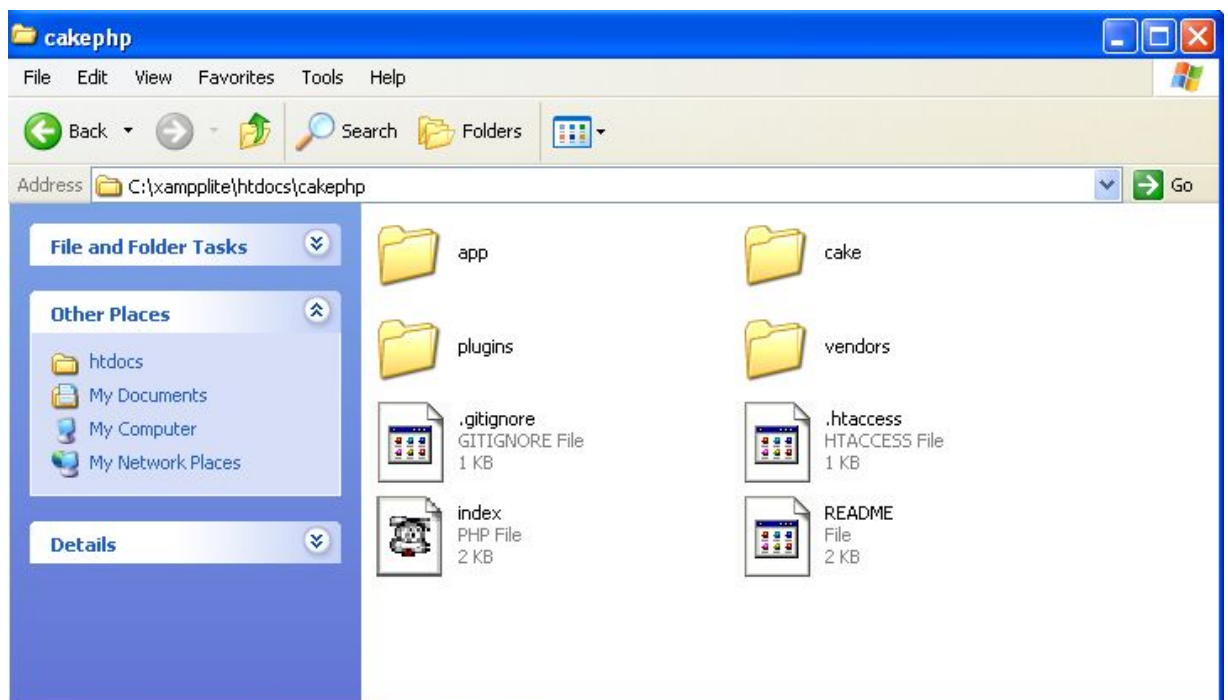
1. Install Xampplite di **c:\xampplite**
2. Bina database bernama **cake** menggunakan phpMyAdmin
3. Download dan Install CakePHP versi 1.3.3 Stable URL :  
<http://github.com/cakephp/cakephp/downloads>
4. Extract supaya fail cakephp terletak di **c:\xampplite\htdocs\cakephp\**
5. Tulis kod di bawah dan simpan di dalam **c:\Windows\cake.bat**

```

1 @echo.
2 @echo off
3 SET app=%0
4 SET lib=c:\xampplite\htdocs\cakephp\cake\console\
5 SET php=c:\xampplite\php\php.exe -q
6 %php% "%lib%cake.php" -working "%CD%" %*
7 echo.

```

Save sebagai c:\windows\cake.bat



Lokasi fail CakePHP 1.3

# Bina Projek Blog

1. Buka <http://localhost/phpMyAdmin>
2. Cipta database bernama **cake**
3. Buka MS-DOS dan sila ke **c:\xampplite\htdocs\**
4. Taip **cake bake**
5. Letakkan path project sebagai **c:\xampplite\htdocs\blog**
6. Ikut setting di bawah untuk konfigurasi database

Host: localhost

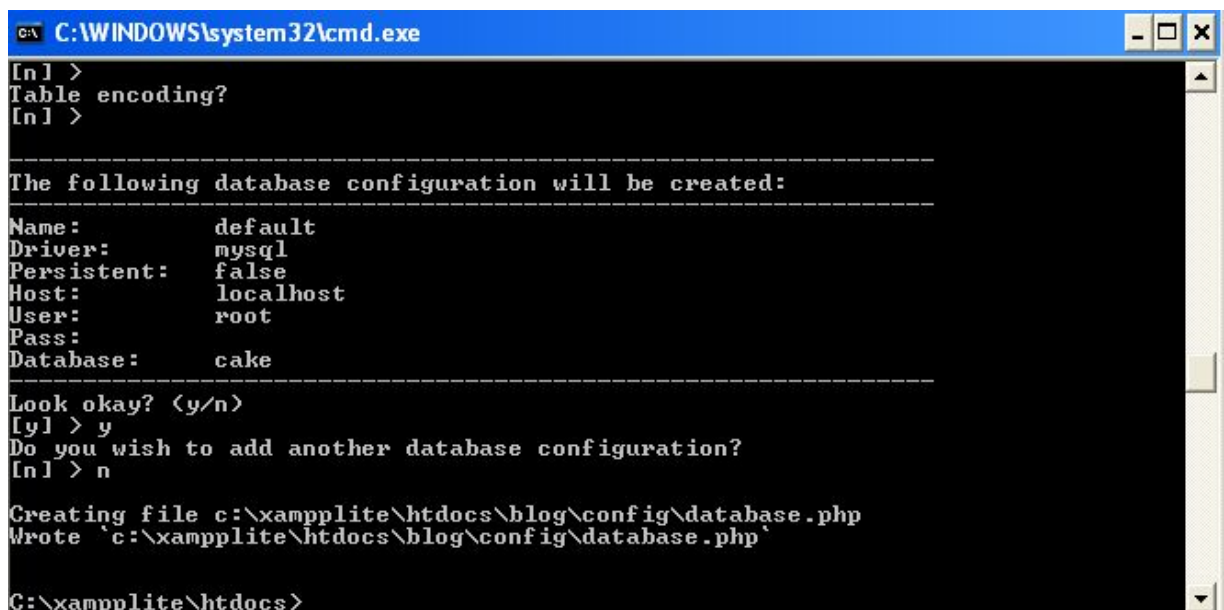
User: root

Password: <blank>

Database : cake

Driver : mysql

Blank password ? : Yes



```
C:\WINDOWS\system32\cmd.exe
[n] >
Table encoding?
[n] >

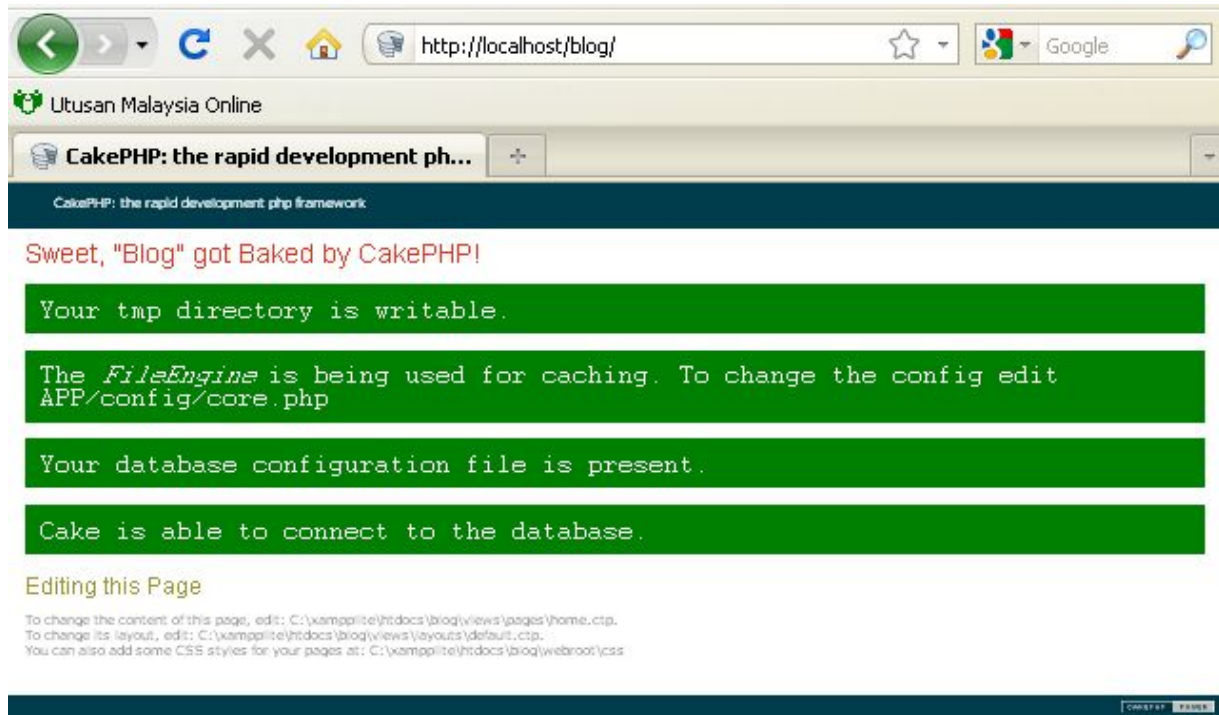
-----
The following database configuration will be created:
-----
Name:      default
Driver:    mysql
Persistent: false
Host:      localhost
User:      root
Pass:
Database:  cake
-----
Look okay? (y/n)
[y] > y
Do you wish to add another database configuration?
[n] > n

Creating file c:\xampplite\htdocs\blog\config\database.php
Wrote 'c:\xampplite\htdocs\blog\config\database.php'

C:\xampplite\htdocs>
```

Konfigurasi database untuk Default

# Sweet ... Blog got Baked



Untuk menguji sistem kita berjaya dicipta oleh Cake CLI, buka alamat <http://localhost/blog> menggunakan Mozilla Firefox / Google Chrome

Sepatutnya anda akan dapat 4 baris status berwarna hijau seperti gambar di atas. Seterusnya, kita akan membina Model Group, User dan Post menggunakan Cake Schema.



# Cake Schema

Saya tidak akan menggunakan phpMyAdmin untuk membina table dan column sistem Blog. Sebaliknya saya akan menggunakan khidmat Cake Schema.

## Skema Table untuk Group

id – integer  
name – string  
description – text

## Skema Table untuk User

id – integer  
group\_id – integer  
username – string  
password – string  
email – string  
created – datetime  
modified – datetime

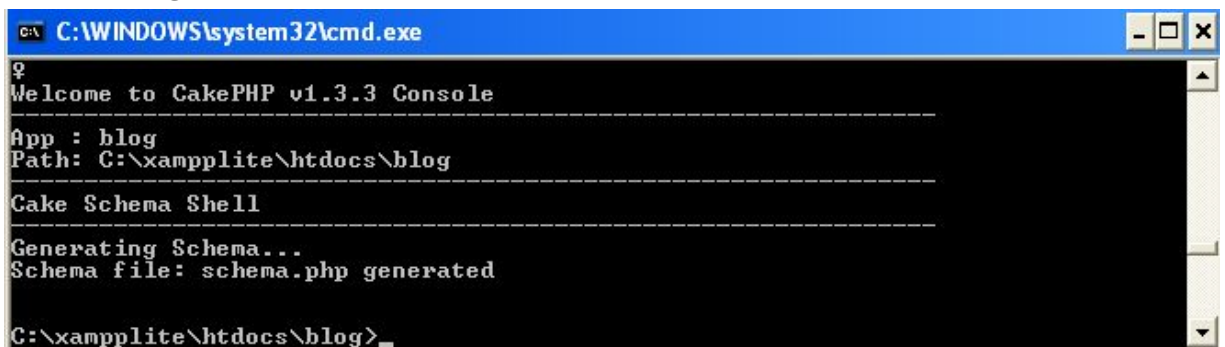
## Skema Table untuk Post

id – integer  
user\_id – integer  
title – string  
body – text  
published – Boolean  
hits – integer  
created – datetime  
modified – datetime

Agak mudah untuk menentukan jenis struktur table untuk database Blog. Untuk menterjemahkan skema di atas dalam bentuk kod, si ikut step di bawah :

Pastikan anda berada di dalam folder `c:\xampplite\htdocs\blog`

1. cake schema generate



```
C:\WINDOWS\system32\cmd.exe
Welcome to CakePHP v1.3.3 Console
-----
App : blog
Path: C:\xampplite\htdocs\blog
-----
Cake Schema Shell
-----
Generating Schema...
Schema file: schema.php generated
C:\xampplite\htdocs\blog>
```

2. Fail bernama schema.php akan tercipta di APP/config/schema/schema.php
3. Edit fail tersebut dan tambah kod berikut :

```

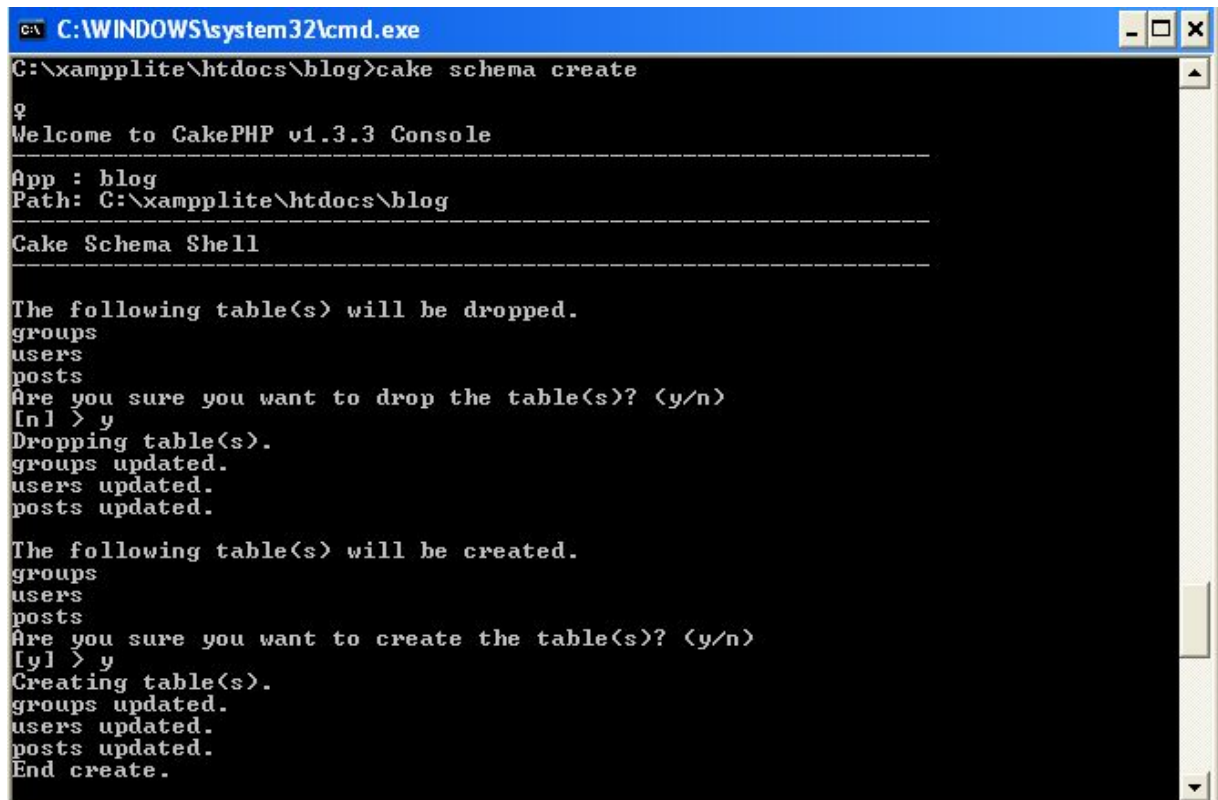
1  <?php
2  /* SVN FILE: $Id$ */
3  /* Blog schema generated on: 2010-08-18 14:08:08 : 1282140488*/
4  class BlogSchema extends CakeSchema {
5      var $name = 'Blog';
6
7      function before($event = array()) {
8          return true;
9      }
10
11     function after($event = array()) {
12     }
13
14     var $groups = array(
15         'id' => array('type' => 'integer', 'null' => false, 'default' => NULL, 'key' => 'primary'),
16         'parent_id' => array('type' => 'integer', 'null' => true, 'default' => NULL),
17         'name' => array('type' => 'string', 'null' => true, 'default' => NULL),
18         'description' => array('type' => 'text', 'null' => true, 'default' => NULL),
19         'created' => array('type' => 'datetime', 'null' => true, 'default' => NULL),
20         'modified' => array('type' => 'datetime', 'null' => true, 'default' => NULL),
21     );
22
23     var $users = array(
24         'id' => array('type' => 'integer', 'null' => false, 'default' => NULL, 'key' => 'primary'),
25         'group_id' => array('type' => 'integer', 'null' => true, 'default' => NULL),
26         'username' => array('type' => 'string', 'null' => true, 'default' => NULL),
27         'password' => array('type' => 'string', 'null' => true, 'default' => NULL),
28         'email' => array('type' => 'string', 'null' => true, 'default' => NULL),
29         'created' => array('type' => 'datetime', 'null' => true, 'default' => NULL),
30         'modified' => array('type' => 'datetime', 'null' => true, 'default' => NULL),
31     );
32
33     var $posts = array(
34         'id' => array('type' => 'integer', 'null' => false, 'default' => NULL, 'key' => 'primary'),
35         'user_id' => array('type' => 'integer', 'null' => true, 'default' => NULL),
36         'title' => array('type' => 'string', 'null' => true, 'default' => NULL),
37         'body' => array('type' => 'text', 'null' => true, 'default' => NULL),
38         'published' => array('type' => 'boolean', 'null' => true, 'default' => '0'),
39         'created' => array('type' => 'datetime', 'null' => true, 'default' => NULL),
40         'modified' => array('type' => 'datetime', 'null' => true, 'default' => NULL),
41     );
42 }
43 ?>

```

PasteBin : <http://pastebin.com/nmnYV5Lv>

Setelah fail **APP/config/schema/schema.php** disimpan, kita cipta table dalam MySQL menggunakan CakePHP CLI. Laksanakan arahan berikut :

**cake schema create**



```
C:\WINDOWS\system32\cmd.exe
C:\xampplite\htdocs\blog>cake schema create
?
Welcome to CakePHP v1.3.3 Console
-----
App : blog
Path: C:\xampplite\htdocs\blog
-----
Cake Schema Shell
-----

The following table(s) will be dropped.
groups
users
posts
Are you sure you want to drop the table(s)? <y/n>
[1] > y
Dropping table(s).
groups updated.
users updated.
posts updated.

The following table(s) will be created.
groups
users
posts
Are you sure you want to create the table(s)? <y/n>
[y] > y
Creating table(s).
groups updated.
users updated.
posts updated.
End create.
```

Hasil arahan. Table **groups**, **users** dan **posts** akan dicipta di dalam MySQL. Mari kita lihat table yang dicipta oleh Cake CLI di dalam konsol MySQL.

```

C:\WINDOWS\system32\cmd.exe - mysql -uroot

C:\xampplite\htdocs\blog>mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.1.33-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use cake;
Database changed
mysql> desc groups;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int(11) | NO   | PRI | NULL    | auto_increment |
| parent_id | int(11) | YES  |     | NULL    |                |
| name   | varchar(255) | YES  |     | NULL    |                |
| description | text | YES  |     | NULL    |                |
| created | datetime | YES  |     | NULL    |                |
| modified | datetime | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.09 sec)

mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int(11) | NO   | PRI | NULL    | auto_increment |
| group_id | int(11) | YES  |     | NULL    |                |
| username | varchar(255) | YES  |     | NULL    |                |
| password | varchar(255) | YES  |     | NULL    |                |
| email   | varchar(255) | YES  |     | NULL    |                |
| created | datetime | YES  |     | NULL    |                |
| modified | datetime | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> desc posts;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int(11) | NO   | PRI | NULL    | auto_increment |
| user_id | int(11) | YES  |     | NULL    |                |
| title  | varchar(255) | YES  |     | NULL    |                |
| body   | text | YES  |     | NULL    |                |
| published | tinyint(1) | YES  |     | 0       |                |
| created | datetime | YES  |     | NULL    |                |
| modified | datetime | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>

```

Berpanduan kepada skema yang diberikan, CakePHP CLI akan melaksanakan arahan khas untuk database MySQL untuk mencipta table dengan struktur di atas.

# Cake Bake All

Saya akan menggunakan arahan Cake Bake All Modelname untuk membina Model,Controller dan View dengan arahan berikut :

- cake bake all Post
- cake bake all Group
- cake bake all User

```
C:\xampplite\htdocs\blog>cake bake all Post
?
Welcome to CakePHP v1.3.3 Console
-----
App : blog
Path: C:\xampplite\htdocs\blog
-----
Bake All
-----

Baking model class for Post...
Creating file C:\xampplite\htdocs\blog\models\post.php
Wrote 'C:\xampplite\htdocs\blog\models\post.php'
Post Model was baked.
SimpleTest is not installed. Do you want to bake unit test files anyway? (y/n)
[y] > n

Creating file C:\xampplite\htdocs\blog\controllers\posts_controller.php
Wrote 'C:\xampplite\htdocs\blog\controllers\posts_controller.php'
Post Controller was baked.
SimpleTest is not installed. Do you want to bake unit test files anyway? (y/n)
[y] > n

Creating file C:\xampplite\htdocs\blog\views\posts\index.ctp
Wrote 'C:\xampplite\htdocs\blog\views\posts\index.ctp'

Creating file C:\xampplite\htdocs\blog\views\posts\view.ctp
Wrote 'C:\xampplite\htdocs\blog\views\posts\view.ctp'

Creating file C:\xampplite\htdocs\blog\views\posts\add.ctp
Wrote 'C:\xampplite\htdocs\blog\views\posts\add.ctp'

Creating file C:\xampplite\htdocs\blog\views\posts\edit.ctp
Wrote 'C:\xampplite\htdocs\blog\views\posts\edit.ctp'
Post Views were baked.
Bake All complete
```

Jika anda tiada SimpleTest, tidak perlu bake test files. Sila pilih No

Jika semuanya berjalan lancar, Cake CLI akan membina fail Model, Controller dan View tanpa masalah dan Model Relation akan dikesan secara automatik. Untuk menguji samaada Cake CLI menjalankan tugas dengan betul, mari kita lihat baked application untuk Post, sila ke <http://localhosts/blog/posts> dan paparan seperti di bawah akan tertera :

CakePHP: the rapid development php framework

**Actions**

- New Post
- List Users
- New User

**Posts**

Id	User	Title	Body	Published	Created	Modified	Actions
Page 0 of 1, showing 0 records out of 0 total, starting on record 0, ending on 0							
<< previous   next >>							

(default) 5 queries took 7 ms

Nr	Query	Error	Affected	Num. rows	Took (ms)
1	DESCRIBE `posts`		7	7	2
2	DESCRIBE `users`		7	7	2
3	DESCRIBE `groups`		6	6	2

Done

Sila uji paparan untuk User dan juga Group :

- <http://localhost/blog/users>
- <http://localhost/blog/groups>

# CakePHP Auth

Seterusnya, saya akan lindungi sistem Blog dengan hanya membenarkan user yang berdaftar sahaja yang boleh masuk. Strategi .

Buka alamat <http://localhost/blog/groups> dan cipta usergroup berikut :

1. SuperUser
2. Admin
3. User
4. Suspended

The screenshot shows the CakePHP Groups management interface. The browser window displays the URL <http://localhost/blog/groups>. The interface includes a sidebar with the following actions: New Group, List Groups, New Parent Group, List Users, and New User. The main content area, titled 'Groups', contains a table with the following data:

Id	Parent	Name	Description	Created	Modified	Actions
1		Super User	Super User Group	2010-08-18 16:26:53	2010-08-18 16:26:53	View Edit Delete
2		Admin	Administrator Group	2010-08-18 16:27:14	2010-08-18 16:27:14	View Edit Delete
3		User	Normal User	2010-08-18 16:27:33	2010-08-18 16:27:33	View Edit Delete
4		Suspended	Suspended User	2010-08-18 16:27:55	2010-08-18 16:27:55	View Edit Delete

Page 1 of 1, showing 4 records out of 4 total, starting on record 1, ending on 4

<< previous | next >>

At the bottom of the interface, a status bar indicates: (default) 2 queries took 1 ms.

Group yang telah ditambah. Fungsi Group ini ialah untuk membolehkan User yang berdaftar dikategorikan mengikut Group.

Strategi kita ialah melindungi semua aplikasi dalam Blog dari user yang tidak berdaftar secara automatic. Jadi dengan itu, kita akan menggunakan ibu kepada semua Controller, iaitu `app_controller.php`.

Fail ini terletak di `APP/app_controller.php`, dan bukannya terletak di dalam folder rasmi untuk controllers. Kita akan load Helpers dan Components yang akan digunakan di dalam `AppController` dan setiap Controller di dalam sistem akan load Components dan Helpers secara automatic.

Sebelum kita load Component Auth di dalam `AppController`, kita perlu cipta dahulu `action register()`, `login()`, dan `logout()` terlebih dahulu dan perlu benarkan user yang tidak login mempunyai kebenaran untuk `register()`.



### Convention untuk Auth

- Menggunakan table users dan model User
- Menggunakan field username dan password dan dalam bentuk varchar(255)
- Action login() dan logout() telah dibina di dalam UsersController
- View untuk login ( login.ctp ) telah dibina dalam APP/views/users/login.ctp

### Action dan View untuk register()

Buka fail APP/controllers/users\_controller.php dan tambah kod untuk membolehkan public user register dengan sistem blog. Seterusnya, linkkan user tersebut setelah berjaya daftar.

```

1  <?php
2  class UsersController extends AppController {
3
4      var $name = 'Users';
5
6      function register(){
7          if(!empty($this->data)){
8              if($this->User->validates()){
9                  // beri nilai group_id untuk group User
10                 $this->data['User']['group_id'] = 3;
11                 $this->User->save($this->data);
12
13                 // baca balik data untuk login secara automatik
14                 $data = $this->User->read();
15
16                 // autologin user
17                 $this->Auth->login($data);
18
19                 // redirect ke home
20                 $this->redirect('/');
21             }
22         }
23     }

```

### Action untuk register()

Sila perhatikan baris no 10. Kita akan beri nilai kepada group\_id = 3 ( User ) kepada pengguna yang berdaftar, sekaligus setiap user akan dimasukkan ke dalam UserGroup bernama User.



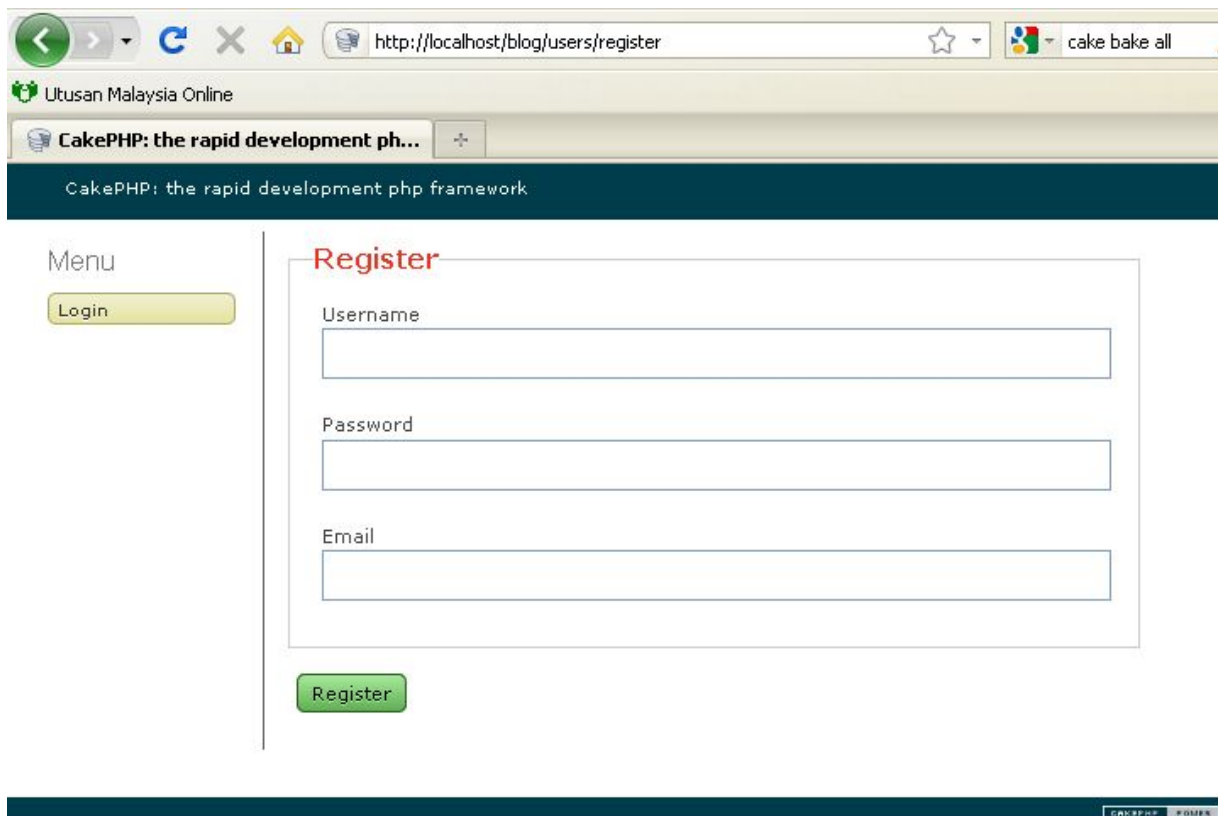
Kemudian, cipta fail **APP/views/users/register.ctp** dan letak kod seperti berikut :

```

1 <div class="users form">
2 <?php echo $this->Form->create('User');?>
3 <fieldset>
4 <legend>Register</legend>
5 <?php
6     echo $this->Form->input('username');
7     echo $this->Form->input('password');
8     echo $this->Form->input('email');
9 >?>
10 </fieldset>
11 <?php echo $this->Form->end('Register');?>
12 </div>
13 <div class="actions">
14 <h3>Menu</h3>
15 <ul>
16 <li><?php echo $this->Html->link('Login', array('action' => 'login'));?></li>
17 </ul>
18 </div>

```

Hasilnya adalah seperti berikut :



URL : <http://localhost/blog/users/register>

Advanced CakePHP : Membina Panel Kawalan Pengguna

Untuk tambahan keselamatan, anda boleh menambah sendiri Model Validation untuk User. Skop tutorial ini ialah ACL jadi saya akan skip proses tersebut.

Seterusnya, tambah action untuk login(), logout() dan beforeFilter().

```

1  <?php
2  class UsersController extends AppController {
3
4      var $name = 'Users';
5
6      function beforeFilter(){
7          parent::beforeFilter();
8          $this->Auth->allow(array('register','login','logout'));
9
10         if(!empty($this->data['User']['new_password'])){
11             $this->data['User']['password'] = $this->data['User']['new_password'];
12         }
13     }
14
15     function login(){
16         if(!empty($this->data)){
17             // Jika sah, loginkan mereka
18             if($this->Auth->login($this->data)){
19                 $this->redirect('/');
20             } else {
21                 // Jika salah, beri error
22                 $this->User->invalidate('username', 'Invalid user');
23             }
24         }
25     }
26
27     function logout(){
28         $this->Auth->logout();
29         $this->redirect('/');
30     }

```

Kita akan gunakan AppController untuk load Components Auth dan Acl. Di baris 7, parent::beforeFilter() bermaksud, kita akan inherit kod callback beforeFilter() yang terletak di dalam AppController.

Untuk fungsi login, saya akan cuba check data mereka menggunakan method \$this->Auth->login(\$this->data). Variable \$this->data datangnya dari User yang submit form.

Seterusnya, mari kita cipta fail View untuk action login() di APP/views/users/login.ctp

```

1 <div class="users form">
2 <?php echo $this->Form->create('User');?>
3 <fieldset>
4 <legend>Login</legend>
5 <?php
6     echo $this->Form->input('username');
7     echo $this->Form->input('password');
8 <?>
9 </fieldset>
10 <?php echo $this->Form->end('Login');?>
11 </div>
12 <div class="actions">
13 <h3>Menu</h3>
14 <ul>
15 <li><?php echo $this->Html->link('Register', array('action' => 'register'));?></li>
16 </ul>
17 </div>

```

Kod untuk login.ctp

Contoh paparan untuk Login.

Seterusnya, buka fail APP/app\_controller.php, dan letak kod seperti di bawah :

```
1 <?php
2 class AppController extends Controller {
3
4     var $helpers = array('Html', 'Form', 'Javascript', 'Session', 'Time');
5     var $components = array( 'Auth', 'Session' );
6
7     function beforeFilter() {
8     }
9
10    function isAuthorized() {
11        return true;
12    }
13 }
14
```

Kita akan load Component Auth dan Helpers yang akan digunakan kemudian. Component Auth memerlukan satu action bernama isAuthorized() di cipta dan return TRUE.

Sekarang sistem Blog akan menghalang User yang tidak berdaftar untuk akses ke dalam sistem.

Jika anda pergi ke semua URL di bawah

- <http://localhost/blog>
- <http://localhost/blog/posts>
- <http://localhost/blog/groups>
- <http://localhost/blog/users>

, kesemua URL di atas akan dihalang untuk akses dan sistem akan paparkan login page. Sebelum kita boleh login, daftar dahulu dengan klik button register di sebelah kiri.

Register dahulu

#### Editing this Page

To change the content of this page, edit: C:\xampplite\htdocs\blog\views\pages\home.ctp.  
To change its layout, edit: C:\xampplite\htdocs\blog\views\layouts\default.ctp.  
You can also add some CSS styles for your pages at: C:\xampplite\htdocs\blog\webroot\css

Selepas register, anda akan di bawa ke Home, dan anda kini telah berjaya login

Advanced CakePHP : Membina Panel Kawalan Pengguna

Utusan Malaysia Online

CakePHP: the rapid development php framework

**Actions**

- New User
- List Groups
- New Group
- List Posts
- New Post

**Users**

Id	Group	Username	Password	Email
1	User	bazet	86aaa9b5874c2e6c83ec90469dd3bcbc7df480de	azril.nazli@gmail.com
2	User	azril	1407b461e82a534f326bbc282095d259e083599f	azril.nazli@gmail.com
3	User	user	50319ebe5904ca950ab6cbaaca4a93805f2daecd	azril.nazli@gmail.com
4	User	cuba1	4801b9feff94e64cb6f9e01e62b6781472340b1f	cuba1@cuba.com
5	User	cuba2	dbd93dd96017fc3caf9f433f5c793257fdc29d90	cuba1@cuba.com
6	User	user4	aa1f9cd0249e148931e13740a80eddc49234d918	user4
7	User	user5	afc81aa7d5d8621d8d460155ee6eab65d76d953b	user5
8	User	blogger	18c2d1091c4044e18e000d7278c7e7e0767e6448	blogger@localhost

Page 1 of 1, showing 8 records out of 8 total, starting on record 1, ending on 8

Jika anda ke <http://localhost/blog/users>, anda akan dapat paparan seperti di atas. Password setiap user telah di'hashed'kan. Untuk logout, sila taip <http://localhost/users/logout>

Utusan Malaysia Online

CakePHP: the rapid development php framework

**Menu**

- Register

**Login**

Username

Password

Login

Dan anda akan di hantar ke login page.

Advanced CakePHP : Membina Panel Kawalan Pengguna

# Access Control List

Rujukan : <http://book.cakephp.org/view/1242/Access-Control-Lists>

Sekarang tiba masanya untuk memahami subjek yang ditunggu-tunggu, ACL.

Acl akan memulakan tugasnya sebaik sahaja kita load Acl sebagai Component, contoh

```
var $components = array('Acl', 'Auth');
```

Jika kita menggunakan Acl, di dalam callback beforeFilter(), kita perlu memberitahu Auth supaya menggunakan `$this->auth->authorize = 'actions'` dan function `isAuthorized()` tidak lagi diperlukan.

Lastly, you can use authorize with actions such as below

Plain Text View

```
1. <?php
2.     $this->Auth->authorize = 'actions';
3. ?>
```

By using actions, Auth will make use of ACL and check with `AdComponent::check()`. An `isAuthorized` function is not needed.

Rujukan : <http://book.cakephp.org/view/396/authorize>

## Bagaimana ACL berfungsi ?

Dalam ACL, ada 2 komponen yang terlibat

1. ACO – objek yang diperlukan
2. ARO – objek yang nak memerlukan

Paling mudah, bayangkan ACO ialah Controller/Action dan ARO ialah User/ID. Jadi jika user ingin masuk ke <http://localhost/posts/index> menggunakan browser setelah login,

Posts/index adalah ACO

User/3 adalah ARO ( dapatkan dari Auth Session )

ACL akan membuat pemeriksaan berdasarkan ACO dan ARO. Jika User/3 tidak mempunyai akses ke Posts/index, ACL tidak membenarkannya akses dengan redirect ke '/'.

Perlu diingat, ACL tidak mengawal user yang belum Authenticated. Tugas itu terletak kepada Component Auth.

Untuk tutorial ini, saya akan mulakan dengan contoh ACL yang mudah dahulu

# ACO

1. Install table untuk ACL, laksanakan arahan berikut di dalam Cake CLI

```

C:\WINDOWS\system32\cmd.exe
C:\xampplite\htdocs\blog>cake acl initdb
?
Welcome to CakePHP v1.3.3 Console
-----
App : blog
Path: C:\xampplite\htdocs\blog
-----
Cake Schema Shell
-----

The following table(s) will be dropped.
acos
aros
aros_acos
Are you sure you want to drop the table(s)? (y/n)
[n] > y
Dropping table(s).
acos updated.
aros updated.
aros_acos updated.

The following table(s) will be created.
acos
aros
aros_acos
Are you sure you want to create the table(s)? (y/n)
[y] > y
Creating table(s).
acos updated.
aros updated.
aros_acos updated.
End create.

C:\xampplite\htdocs\blog>

```

2. Seterusnya, kita akan bina Aco dalam bentuk berikut

Pages  _ display	Posts  _ index  _ view  _ add  _ edit  _ delete	Users  _ register  _ login  _ logout  _ index  _ view  _ add  _ edit  _ delete	Posts  _ index  _ view  _ add  _ edit  _ delete
---------------------	--	--	--

3. Pages, Posts, Users dan Posts sebenarnya adalah nama yang diberikan oleh Controller, dan kita akan gunakan sebagai Aco Root. Manakala Action untuk setiap Controller akan diletakkan di bawah setiap Aco Root yang bersesuaian,



4. Mari kita mulakan dengan membina Aco untuk Pages/display. Secara default route '/' akan pergi ke Pages/display. Jadi kita kena bina Aco yang merujuk kepada Pages/display

```
C:\xampplite\htdocs\blog>cake acl create aco root Pages
New Aco 'Pages' created.

C:\xampplite\htdocs\blog>cake acl create aco Pages display
New Aco 'display' created.

C:\xampplite\htdocs\blog>cake acl view aco
Aco tree:
-----
[1] Pages
    [2] display
-----

C:\xampplite\htdocs\blog>
```

Untuk Controller, gunakan arahan **cake acl create aco root Pages**

Untuk Action, gunakan arahan **cake acl create aco Pages Display**

Untuk melihat struktur Aco, laksanakan arahan **cake acl view aco**

```
C:\xampplite\htdocs\blog>cake acl create aco Pages cubapadam
New Aco 'cubapadam' created.

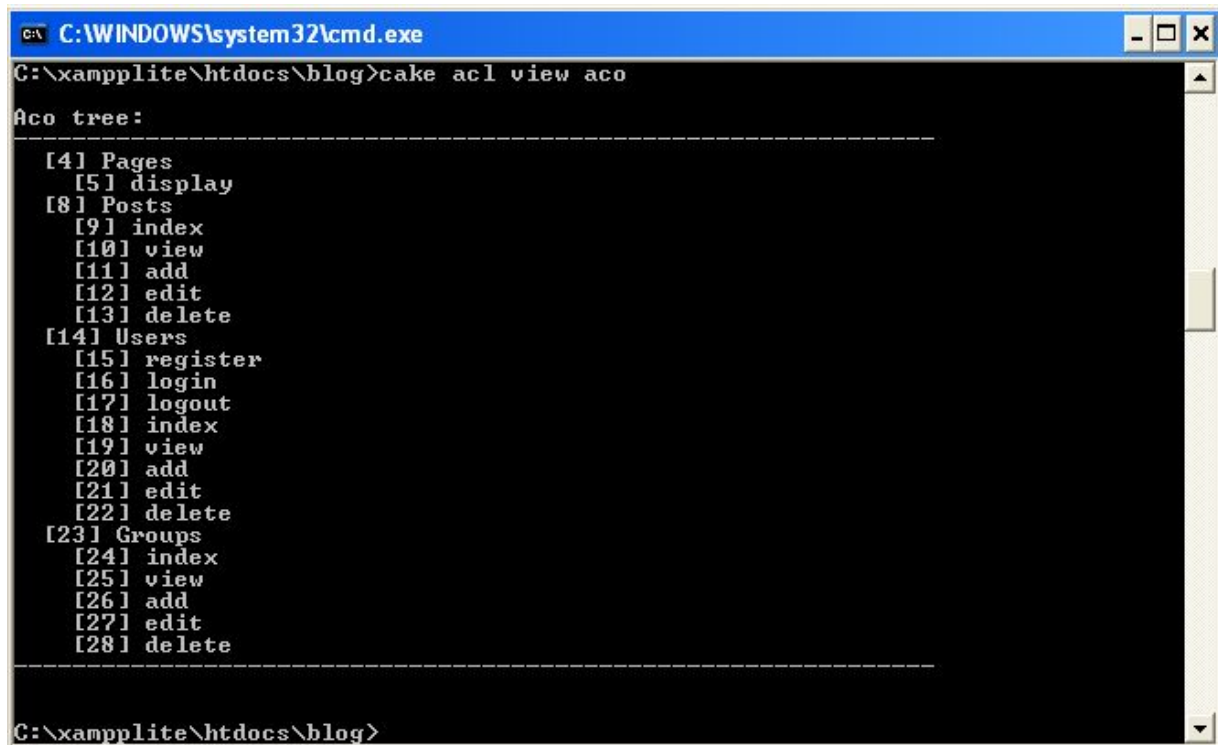
C:\xampplite\htdocs\blog>cake acl view aco
Aco tree:
-----
[4] Pages
    [5] display
    [7] cubapadam
-----

C:\xampplite\htdocs\blog>cake acl delete aco Pages/cubapadam
Aco deleted.

C:\xampplite\htdocs\blog>cake acl view aco
Aco tree:
-----
[4] Pages
    [5] display
-----

C:\xampplite\htdocs\blog>
```

Contoh untuk padam node yang silap. Mesti dalam format Root/parent



```
C:\WINDOWS\system32\cmd.exe
C:\xampplite\htdocs\blog>cake acl view aco

Aco tree:
-----
[4] Pages
[5] display
[8] Posts
[9] index
[10] view
[11] add
[12] edit
[13] delete
[14] Users
[15] register
[16] login
[17] logout
[18] index
[19] view
[20] add
[21] edit
[22] delete
[23] Groups
[24] index
[25] view
[26] add
[27] edit
[28] delete
-----

C:\xampplite\htdocs\blog>
```

Menggunakan contoh di atas, buat kesemua Aco yang diperlukan.

# ARO

Aro ialah objek yang ingin access ke objek Aco. Untuk create ARO, saya akan gunakan Cake Shell . Sila salin kod di bawah dan save sebagai **APP/vendors/shells/create\_aro.php**

```

1  <?php
2  class CreateAroShell extends Shell {
3
4      function main() {
5
6          // load Acl component
7          App::import('Component', 'Acl');
8          $this->Acl = new AclComponent(null);
9
10         // output to screen
11         $this->out('Creating Group');
12         $this->hr();
13
14         // this will create Aro::Root
15         $this->Acl->Aro->create();
16         $this->Acl->Aro->save(array(
17             'model' => 'Group',
18             'foreign_key' => 3, // get id for group 'User'
19             'parent_id' => null,
20             'alias' => 'Group:3',
21         ));
22
23         // this will create Aro::Parent
24         $this->out('Creating User');
25         $this->hr();
26         $this->Acl->Aro->create();
27         $parent_id = $this->Acl->Aro->getLastInsertId();
28         $this->Acl->Aro->save(array(
29             'model' => 'User',
30             'foreign_key' => 1, // get id for current user,
31             'parent_id' => $parent_id,
32             'alias' => 'User:1',
33         ));
34         $this->out('Done');
35     }
36 }

```

PasteBin : <http://pastebin.com/Fg2RyE6R>

Untuk Aro, kita akan gunakan alias **User:\$id** dan **Group:\$id** , contohnya :

User:1

User:2

Group:1

Group:2

**\$id** merujuk kepada nilai **id** setiap **User** dan **Group** kerana untuk memastikan nilai alias adalah unik.

Mari kita cipta **Aro** menggunakan **Cake CLI**. Jalankan arahan **cake create\_aro**

```
C:\xampplite\htdocs\blog>cake create_aro
?
Welcome to CakePHP v1.3.3 Console
-----
App : blog
Path: C:\xampplite\htdocs\blog
-----
Creating Group
-----
Creating User
-----
Done
C:\xampplite\htdocs\blog>
```

Jika melihat kepada data table aros di dalam database Cake, hasilnya seperti berikut

```
C:\WINDOWS\system32\cmd.exe - mysql -uroot cake
mysql> select * from aros;
+----+-----+-----+-----+-----+-----+-----+
| id | parent_id | model | foreign_key | alias | lft | rght |
+----+-----+-----+-----+-----+-----+-----+
| 16 | NULL      | Group | 3           | Group:3 | 1 | 4 |
| 17 | 16        | User  | 1           | User:1  | 2 | 3 |
+----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
mysql>
```

Dan jika melihat menggunakan arahan **cake acl view aro**

```
C:\xampplite\htdocs\blog>cake acl view aro
Aro tree:
-----
  [16] Group:3
    [17] User:1
-----
C:\xampplite\htdocs\blog>
```

User:1 adalah parent kepada Group:3. Nampaknya proses cipta Aro kita berjaya.

## ACL Setting Permission

Sekarang kita sudah berjaya set ACO dan ARO. Langkah terakhir ialah untuk set permission setiap ACO. Selepas User berjaya login, mereka akan di redirect ke Pages/display. Jadi kita harus set permission akses Pages/display kepada Group:3 dan User:1.

Laksanakan arahan : **cake acl grant Group:3 Pages \***

```
C:\xampplite\htdocs\blog>cake acl grant Group:3 Pages *
Permission granted.
```

Cuba periksa kebenaran ARO terhadap ACO ( Pages )

```
C:\xampplite\htdocs\blog>cake acl check Group:3 Pages *
Group:3 is allowed.

C:\xampplite\htdocs\blog>cake acl check User:1 Pages *
User:1 is allowed.
```

ARO/ACO	Posts	Posts/index	Posts/view	Posts/add	Posts/edit	Posts/delete
Group:3	Allow	Allow	Allow	Allow	Allow	Deny
User:1	Allow	Allow	Allow	Deny	Deny	Allow

Table di atas mapkan permission Group:3 dan User:1

```
C:\xampplite\htdocs\blog>cake acl grant Group:3 Posts/add *
Permission granted.

C:\xampplite\htdocs\blog>cake acl deny User:1 Posts/add *
Permission denied.

C:\xampplite\htdocs\blog>cake acl check User:1 Posts/add *
User:1 is not allowed.
```

Walaupun kebenaran untuk akses Posts/add telah diberi kepada Group:3, tetapi User:1 tetap tidak dapat akses ke Posts/add kerana sistem telah Deny akses hanya kepada User:1

## Setup ACL di Sistem Blog

Kini tiba masanya kita enablekan Component Acl di dalam sistem Blog.

1. Edit fail **APP/app\_controller.php** seperti kod di bawah :

```
1 <?php
2 class AppController extends Controller {
3
4     var $helpers = array('Html','Form','Javascript','Session','Time');
5     var $components = array( 'Acl', 'Auth', 'Session' );
6
7     function beforeFilter() {
8         // Acl
9         $this->Auth->authorize = 'actions';
10    }
11
12    #function isAuthorized() {
13    #    return true;
14    #}
15 }
16
```

Matikan fungsi isAuthorized() dan tambah callback beforeFilter(). Tambah Acl di dalam senarai Components

2. Edit fail `APP/models/user.php` dan tambah kod di bawah :

```

1  <?php
2  class User extends AppModel {
3      var $name = 'User';
4
5      // Acl, jadikan Group as ARO
6      var $actsAs = array('Acl' => 'requester');
7
8      // Acl memerlukan function parentNode
9      function parentNode() {
10         // used while registration
11         if (!$this->id && empty($this->data)) {
12             return null;
13         }
14         if (isset($this->data['User']['group_id'])) {
15             $groupId = $this->data['User']['group_id'];
16         } else {
17             $groupId = $this->field('group_id');
18         }
19         if (!$groupId) {
20             return null;
21         } else {
22             return array('Group' => array('id' => $groupId));
23         }
24     }
25 }

```

`parentNode()` diperlukan oleh Acl behavior untuk membolehkan Acl menyusun Aro mengikut Parent

PasteBin : <http://pastebin.com/2Q1G1FVT>

## What is the parentNode() function for?

It is used when you have cascading, or inherited permissions, for example a Group model. The `parentNode()` function must return the id of the parent Model (not the id of the parent Aro). Typically this will be the `parent_id` of your Model, if you are using the standard convention for trees in Cake.

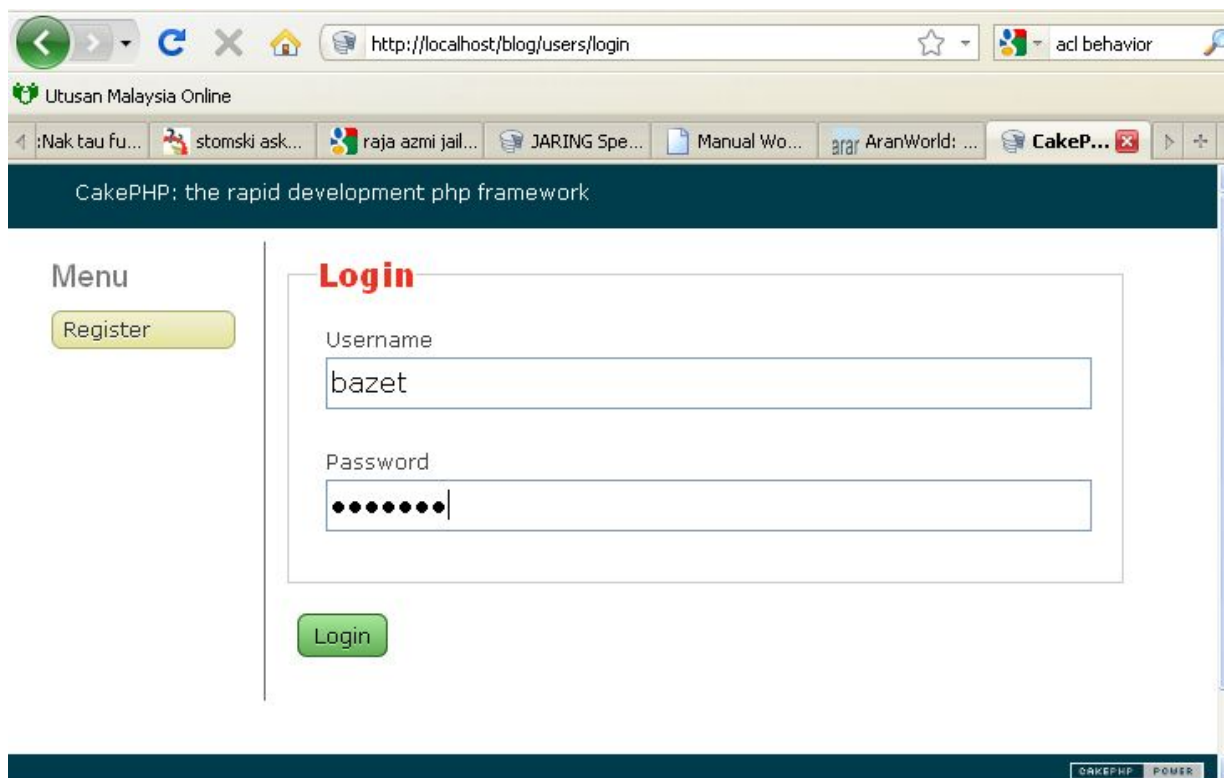
3. Edit fail **App/models/group.php** dan tambah kod di bawah :

```

1  <?php
2  class Group extends AppModel {
3      var $name = 'Group';
4
5      // Acl, jadikan Group as ARO
6      var $actsAs = array('Acl'=>'requester');
7
8      // Acl memerlukan function parentNode
9      function parentNode() {
10     }

```

4. Jadikan Model User dan Group untuk gunakan Behavior Acl sebagai requester
5. Seterusnya login menggunakan user id : 1 , dalam kes ini adalah user pertama.



Bazet mempunyai user\_id : 1 dan group\_id: 3 ( User )



CakePHP: the rapid development php framework

## Sweet, "Blog" got Baked by CakePHP!

Your tmp directory is writable.

The *FileEngine* is being used for caching. To change the config edit APP/config/core.php

Your database configuration file is present.

Cake is able to connect to the database.

Dapat login dan redirect ke home ( sebenarnya adalah Pages/display dan menggunakan Aco Pages/display )

Sekarang cuba ke <http://localhost/blog/posts>

CakePHP: the rapid development php framework

**Actions**

- New Post
- List Users
- New User

**Posts**

Id	User	Title	Body	Published	Created	Modified	Actions
Page 0 of 1, showing 0 records out of 0 total, starting on record 0, ending on 0							
<< previous     next >>							

CAKEPHP POWER

Jika anda klik button New Post, tentu sistem akan menghalang disebabkan Permission Deny yang diberikan kepada User:1.

- Logout dahulu dengan taip <http://localhost/users/logout>
- Register user baru eg cakephp:cakephp
- Cuba ke <http://localhost/posts/add>
- Anda boleh akses ke Posts/add dengan user baru disebabkan User:\$id tiada dalam senarai Deny.

## Seterusnya...

---

Kita berjaya untuk buat ACO/ARO tetapi secara manual...tetapi

1. Bagaimana kita hendak paparkan menu mengikut kebenaran ACL ?
2. Bagaimana kita hendak daftarkan ACO menggunakan web interface ?
3. Dapatkan list Controller/Action dari sistem ?
4. Bagaimana kita hendak memberi Permission mengikut Group dan User ?
5. Apa akan terjadi dengan nilai ACL jika user menukar Group ? Adakah User akan dipindahkan ke ParentGroup yang baru ?
6. CakePHP akan menggunakan alias Group.Id dan User.Id untuk new record, bagaimana kita nak ubah ke Group:Id dan User:Id jika ada rekod baru tercipta ?

# Access Helper

Seterusnya, kita akan bina menu navigasi. Menu kita akan berupa link ke Controller Index, seperti

Tajuk	URL	ACO
Home	/	Pages/display
Posts	/posts	Posts
Users	/users	Users
Groups	/groups	Groups

Kita perlukan Access Helper untuk check permission User yang daftar masuk supaya menu hanya dipaparkan jika User tersebut mempunyai Permission. Dan malangnya Acl ialah satu Component, dan hanya boleh diakses dalam Controller.

Untuk mengatasi masalah itu, kita akan buat AccessComponent dan AccessHelper akan menggunakan data dari AccessComponent.

Method dalam AccessHelper

- **isLoggedIn ()** ~ untuk check user dah login atau belum
- **check ()** untuk check loggedIn user mempunyai permission terhadap Aco
- **my ()** untuk dapatkan data user yang loggedIn, cth \$email = \$access->my('email');
- **link()** untuk bina link berdasarkan Permission

## Bina AccessComponent

Fungsi Component ini ialah untuk memudahkan checking permission di dalam controller, contohnya

```

If( $access->check($aco = '/Posts/index') ){
    Echo 'ada Permission';
} else {
    Echo 'Tiada Permission';
}

```

Dan juga untuk menjadi agen proxy kepada **AccessHelper** untuk periksa Permission di dalam View. **Access Component** hendaklah disimpan di dalam **APP/controllers/components/access.php** dan perlu dipanggil menggunakan **\$var components = array('Access');** dari dalam ApplicationController

## AccessComponent

Salin kod di bawah dan save sebagai APP/controllers/components/access.php

```

1  <?php
2  class AccessComponent extends Object{
3      var $components = array('Acl', 'Auth');
4
5      // Aro dalam format User:$id
6      function check($aco, $action='*'){
7          // check($Aro,$Aco,$Action)
8          if( $this->Acl->check('User:'.$this->Auth->user('id'), $aco, $action)){
9              return true;
10         } else {
11             return false;
12         }
13     }
14
15     function checkHelper($aro, $aco){
16         App::import('Component', 'Acl');
17         $acl = new AclComponent();
18         return $acl->check($aro, $aco, '*');
19     }
20 }

```

PasteBin : <http://pastebin.com/RgVnZtC8>

Component Access, function check() untuk kegunaan dalam Controller dan checkHelper() adalah agen kepada AccessHelper.

## AccessHelper

Salin kod di bawah dan save sebagai APP/views/helpers/access.php

```

1 <?php
2 class AccessHelper extends Helper{
3     var $helpers = array('Session');
4     var $Access;
5     var $Auth;
6     var $user;
7     function beforeRender(){
8         App::import('Component', 'Access');
9         $this->Access = new AccessComponent();
10
11         App::import('Component', 'Auth');
12         $this->Auth = new AuthComponent();
13         $this->Auth->Session = $this->Session;
14         $this->user = $this->Auth->user();
15     }
16
17     function check($aco, $action=''){
18         if(empty($this->user)) return false;
19         return $this->Access->checkHelper('User:'.$this->user['User']['id'], $aco);
20     }
21
22     function isLoggedIn(){
23         return !empty($this->user);
24     }
25
26     function my($column = null){
27         App::import('Component', 'Auth');
28         $auth = new AuthComponent();
29         $auth->Session = $this->Session;
30         if($auth->user($column)){
31             return $auth->user($column);
32         }else{
33             return FALSE;
34         }
35     }
36
37     function isSelected($controller = null, $action = 'index'){
38         $curr_controller = $this->params['controller'];
39         $curr_action = $this->params['action'];
40         if($curr_controller == $controller && $curr_action == $action){
41             return 'selected';
42         } else {
43             return null;
44         }
45     }
46 }

```

PasteBin : <http://pastebin.com/Nwm4tWhx>

Callback **beforeRender()** digunakan untuk import Component Access dan Auth. Baris ke 14 adalah salah satu cara untuk dapatkan Session user yang sedang logged in. Dengan cara ini kita boleh access data user yang sedang logged in menggunakan variable `$this->user`

Advanced CakePHP : Membina Panel Kawalan Pengguna

Function **check()** digunakan untuk periksa permission terhadap ACO. ARO ialah user yang sedang login dalam format **User : id**

Function **isLoggedIn()** digunakan untuk periksa samaada user telah login atau belum.

Function **my()** digunakan untuk mendapatkan data User dari table users.

Function **isSelected()** digunakan untuk kegunaan highlight menu.

## AppController

AccessComponent dan AccessHelper perlu dipanggil dari dalam AppController untuk membolehkan ia berfungsi di dalam Controller dan View

Dan Pages/display boleh diakses oleh Public dan juga LoggedIn User. Untuk itu kita akan benarkan Pages/display menggunakan `$this->Auth->allowedActions()`;

Tambah nilai Access dalam array `$helpers` dan `$components` ke dalam APP/app\_controller.php

```

1  <?php
2  class AppController extends Controller {
3
4      var $helpers = array('Html', 'Form', 'Javascript', 'Session', 'Time', 'Access');
5      var $components = array('Acl', 'Auth', 'Session', 'Access');
6
7      function beforeFilter() {
8          // Acl
9          $this->Auth->authorize = 'actions';
10         $this->Auth->allowedActions = array('display');
11     }
12 }
13

```

APP/app\_controller.php

# Strategi untuk Panel Navigasi



Menu adalah elemen penting dalam pembinaan laman web dan menu kita perlulah dinamik. Berupaya untuk melakukan perkara berikut :

1. Menu berbeza untuk Public dan LoggedIn User
2. Paparkan menu jika user mempunyai Access terhadap Aco

# CSS Panel Navigasi

Sila tambah kod CSS di bawah ke fail APP/webroot/css/cake.generic.css

```

539  /** Top Navigation **/
540  div.nav ul {
541      margin: 10px;
542      padding: 0;
543  }
544  div.nav li {
545      margin: 0 0 0.5em 0;
546      list-style-type: none;
547      white-space: nowrap;
548      padding: 0;
549      display: inline;
550  }
551  div.nav ul li a {
552      font-weight: bold;
553      padding: 8px 16px;
554      background: #e6e49f;
555      ;background: -webkit-gradient(linear, left top, left bottom, from(#f1f1d4), to(#e6e49f));
556      ;background-image: -moz-linear-gradient(top, #f1f1d4, #e6e49f);
557      background: -webkit-gradient(linear, left top, left bottom, from(#a8ea9c), to(#62af56));
558      background-image: -moz-linear-gradient(top, #a8ea9c, #62af56);
559      color: #333;
560      border: 1px solid #aaac62;
561      -webkit-border-radius: 8px;
562      -moz-border-radius: 8px;
563      border-radius: 8px;
564      text-decoration: none;
565      ;text-shadow: #fff 0px 1px 0px;
566      text-shadow: #8cee7c 0px 1px 0px;
567
568      #min-width: 0;
569  }
570  div.nav ul li a: hover {
571      text-decoration: underline;
572  }
573  div.nav .selected,
574  div.nav ul li a: hover {
575      background: #f0f09a;
576      background: -webkit-gradient(linear, left top, left bottom, from(#f7f7e1), to(#eeeeca9));
577  }

```

CSS di atas untuk menu di sebelah kanan.



```

579  /** Top Menu **/
580  div#top_menu {
581      #margin-top:15px;
582      #margin-left: 48px;
583      width: 100%;
584      margin:15px auto auto 30px;
585  }
586
587  #right{
588      float:right;
589      width:500px;
590  }
591
592  #left{
593      float:left;
594      width:auto;
595  }
596  }
597
598  .usercp{
599      font-weight:bold;
600      padding: 8px 16px;
601      #padding-left:15px;
602      #padding-right:15px;
603
604      background:#e6e49f;
605      ;background: -webkit-gradient(linear, left top, left bottom, from(#f1f1d4), to(#e6e49f));
606      ;background-image: -moz-linear-gradient(top, #f1f1d4, #e6e49f);
607      background: -webkit-gradient(linear, left top, left bottom, from(#a8ea9c), to(#62af56));
608      background-image: -moz-linear-gradient(top, #a8ea9c, #62af56);
609      color:#333;
610      border:1px solid #aaac62;
611      -webkit-border-radius:8px;
612      -moz-border-radius:8px;
613      border-radius:8px;
614      text-decoration:none;
615      ;text-shadow: #fff 0px 1px 0px;
616      text-shadow: #8cee7c 0px 1px 0px;
617      margin-bottom : 12px;;
618  }

```

Kod CSS untuk menu di sebelah kiri

PasteBin : <http://pastebin.com/NgfZPCF3>

## Elemen Top Menu

Seterusnya, buat satu fail bernama top\_menu.ctp dan fail ini akan berfungsi sebagai Elemen. Masukkan kod di bawah dan save di APP/views/elements/top\_menu.ctp

```

1 <div class = 'nav' >
2 <ul id = 'right'>
3     <li><a href = "">Home</a></li>
4     <?php if ( $access->check('Posts/index') ): ?>
5     <li><a href = "">Posts</a></li>
6     <?php endif; ?>
7     <li><a class = "selected" href = "">Groups</a></li>
8     <li><a href = "">Users</a></li>
9 </ul>
10 <span id = 'left' class = 'usercp'>
11     Logged in as <strong>Azril</strong>
12     <a href = "">My Account</a> |
13     <a href = "">Profile</a>
14 </span>
15 </div>

```

APP/views/elements/top\_menu.ctp

PasteBin : <http://pastebin.com/QEw2iLaA>

```

36 <body>
37     <div id = "container">
38         <div id = "top_menu">
39             <?php echo $this->element('top_menu'); ?>
40         </div>
41
42         <div id = "content">
43
44             <?php echo $this->Session->flash(); ?>
45
46             <?php echo $content_for_layout; ?>
47
48         </div>
49         <div id = "footer">
50             <?php echo $this->Html->link(
51                 $this->Html->image('cake.power.gif', array('alt' =>
52                     'http://www.cakephp.org/',
53                     array('target' => '_blank', 'escape' => false)
54                 ));
55             ?>
56         </div>
57     </div>
58     <?php echo $this->element('sql_dump'); ?>
59 </body>

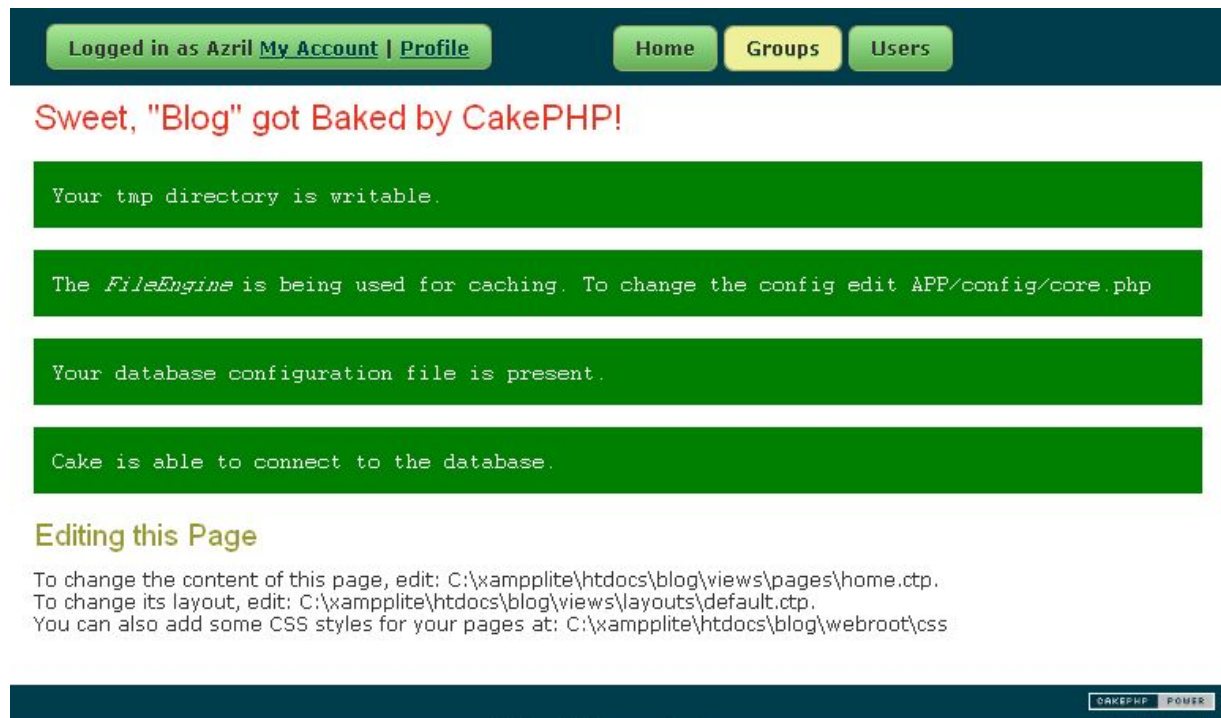
```

Di dalam fail default layout, buang div id = header dan ganti dengan kod dari baris 38 – 40

PasteBin : <http://pastebin.com/UTPX0H07>

APP/views/layouts/default.ctp

Advanced CakePHP : Membina Panel Kawalan Pengguna



Sila buka alamat <http://localhost/blog> dan anda akan dapat design seperti di atas.

## Menu untuk Public dan Logged In User

Untuk membolehkan sistem mengenalpasti samaada LoggedIn User atau Public, kita akan guna bantuan AccessHelper. Dengan menggunakan arahan `$access->isLoggedIn()`, kita dapat membuat checking dan mengasingkan menu untuk Public dan User

## Check samaada user ada Permission terhadap Menu ( Aco )

Untuk mengetahui User mempunyai akses terhadap menu ( Aco ), kita akan buat checking dengan arahan `$access->check('Posts/index')`, Posts/index adalah Aco yang kita ingin periksa.

Ubah fail `APP/elements/top_menu.ctp` seperti kod di bawah

```

1  <?php if( $access->isLoggedIn() ): // Menu untuk user ?>
2  <div class = 'nav' >
3  <ul id = 'right'>
4      <?php if( $access->check('Pages/display') ): ?>
5          <li><?php echo $this->Html->link('Home', '/',
6              array('class' => $access->isSelected('pages','display')) ); ?></li>
7      <?php endif; ?>
8
9      <?php if( $access->check('Posts/index') ): ?>
10         <li><?php echo $this->Html->link('Posts', '/posts/index',
11             array('class' => $access->isSelected('posts')) ); ?></li>
12     <?php endif; ?>
13
14     <?php if( $access->check('Groups/index') ): ?>
15         <li><?php echo $this->Html->link('Groups', '/groups/index',
16             array('class' => $access->isSelected('groups')) ); ?></li>
17     <?php endif; ?>
18
19     <?php if( $access->check('Users/index') ): ?>
20         <li><?php echo $this->Html->link('Users', '/users/index',
21             array('class' => $access->isSelected('users')) ); ?></li>
22     <?php endif; ?>
23 </ul>
24 <span id = 'left' class = 'usercp'>
25     Logged in as <strong><?php echo $access->my('username'); ?></strong>
26     <?php echo $this->Html->link('My Account', '/users/my_account'); ?> |
27     <?php echo $this->Html->link('Logout', '/users/logout'); ?>
28 </span>
29 </div>
30 <?php else: // menu untuk Public ?>
31 <div class = 'nav' >
32 <ul id = 'right'>
33     <li><?php echo $this->Html->link('Home', '/'); ?></li>
34     <li><?php echo $this->Html->link('Register', '/users/register'); ?></li>
35     <li><?php echo $this->Html->link('Login', '/users/login'); ?></li>
36 </ul>
37 <span id = 'left' class = 'usercp'>
38     Cake ACL Control Panel
39 </span>
40 </div>
41 <?php endif; ?>

```

PasteBin : <http://pastebin.com/srZsBqx7>

Advanced CakePHP : Membina Panel Kawalan Pengguna

## Uji Permission untuk Group:3

Jika kita lihat kod checking di atas, kita check berdasarkan

- Pages/display
- Posts/index
- Groups/index
- Users/index

Sekarang mari kita cuba halang Group:3 dari akses ke Posts/index, Groups /index dan Users/index

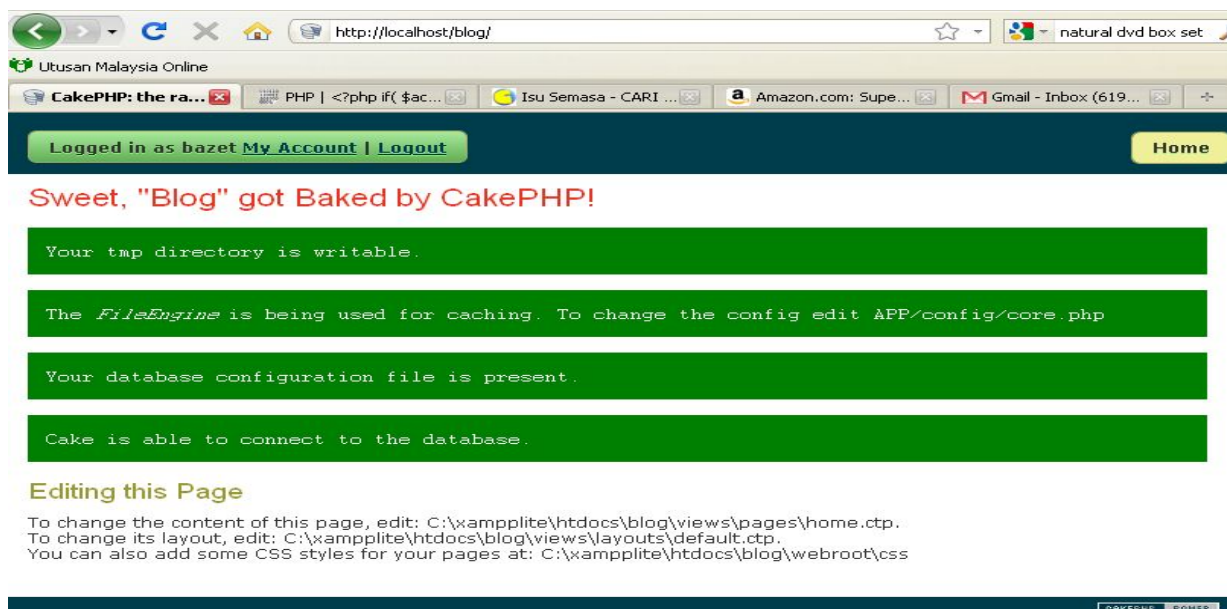
```
C:\xampplite\htdocs\blog>cake acl deny User:1 Posts/index *
Permission denied.

C:\xampplite\htdocs\blog>cake acl deny Group:3 Posts/index *
Permission denied.

C:\xampplite\htdocs\blog>cake acl deny Group:3 Groups/index *
Permission denied.

C:\xampplite\htdocs\blog>cake acl deny Group:3 Users/index *
Permission denied.
```

Laksanakan arahan DENY dalam CAKE CLI



Hasilnya menu di sebelah kanan disembunyikan. Nampaknya sistem menu kita berfungsi.

Advanced CakePHP : Membina Panel Kawalan Pengguna

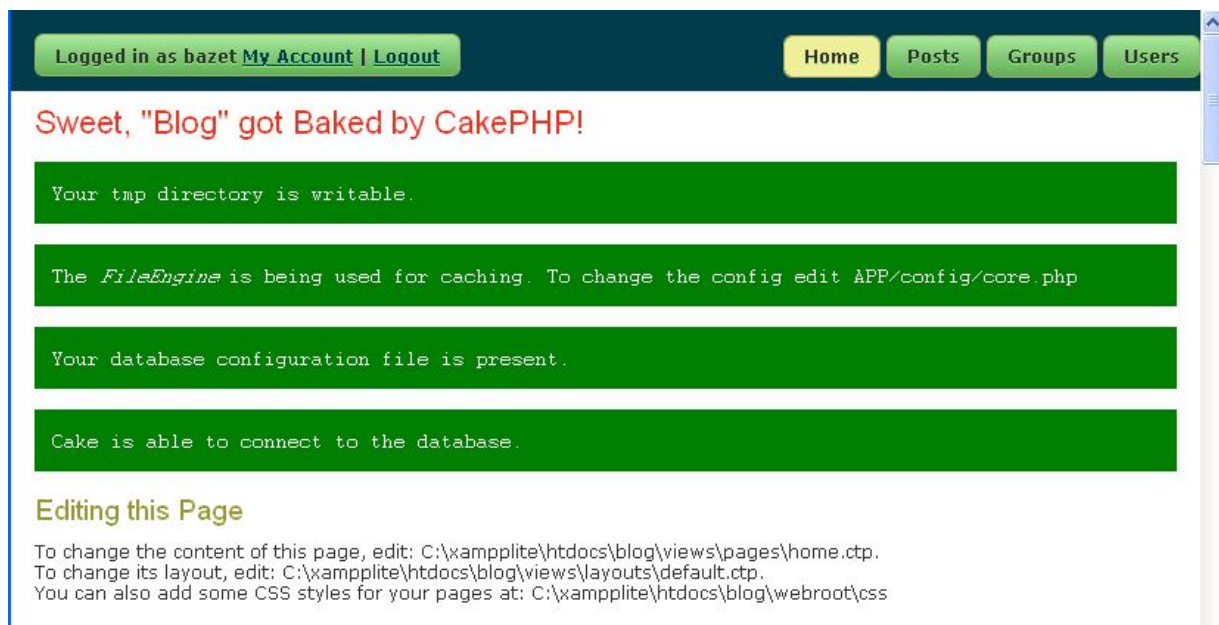
```
C:\xampplite\htdocs\blog>cake acl grant Group:3 Posts/index *
Permission granted.

C:\xampplite\htdocs\blog>cake acl grant Group:3 Groups/index *
Permission granted.

C:\xampplite\htdocs\blog>cake acl grant Group:3 Users/index *
Permission granted.

C:\xampplite\htdocs\blog>
```

Grant kembali akses ke Aco terhadap Group:3



Kini menu muncul kembali.

## Callback AfterSave() dalam UserModel

Nampaknya kita mempunyai masalah, jika User atau Group baru dicipta, CakePHP akan menggunakan alias seperti :

- Group:3
- User:1

Masalahnya, kita telah menetapkan untuk menggunakan format seperti berikut

- Group:3
- User:1

```
C:\xampplite\htdocs\blog>cake acl view aro
Aro tree:
-----
[16] Group:3
    [17] User:1
    [18] User:9
    [19] User:10
    [20] User:11
    [23] User:14
    [25] User:16
    [26] User:18
```

Lihat perbezaan format User:1 dan user yang lain. Jadi bagaimana nak memberitahu ACL supaya sentiasa save ARO alias dalam format yang ditentukan.

Kita akan menggunakan cara yang mudah, afterSave() callback di dalam Model User.

afterSave() call back akan melakukan perkara berikut :

1. Jika ada penambahan data baru, lakukan arahan updateAll untuk kemaskini nilai alias
2. Jika ada kemaskini data , kemaskini nilai Aro supaya merujuk kepada data terkini.  
Contohnya jika User:1 dikemaskini untuk berada di bawah Group:1, nilai parent\_id di dalam Aro juga perlu dikemaskini.

```
32 function afterSave($created=null) {
33     if($created){
34         // update alias format in Aro Model from User.id to User:id
35     } else {
36         // it's an update, maybe user changing user group
37     }
38     return TRUE;
39 }
```

Skeleton untuk function afterSave() callback di dalam APP/model/user.php

afterSave() callback di dalam APP/models/user.php

```

32 function afterSave($created) {
33
34     if($created) {
35
36         // its a creation
37         $aro = new Aro();
38         $aro->updateAll( array( 'alias'=>  "User:".$this->id  ."" ), // the field
39                         array(
40                             'Aro.model'=>'User', // condition 1
41                             'Aro.foreign_key'=> $this->id // condition 2
42                         )
43                     );
44     }
45     else {
46
47         // its an edit, we have to update the tree
48         $data = $this->read();
49         $parent_id = $data['User']['group_id'];
50
51         $aro = new Aro();
52
53         $aro_record = $aro->findByAlias( 'User:'.$this->id );
54         $parent_record = $aro->findByAlias( 'Group:'.$parent_id );
55
56         if ( !empty( $aro_record ) ) {
57             $parent_id = '0';
58             if ( !empty( $parent_record ) ) {
59                 $parent_id = $parent_record['Aro']['id']; // group_id
60             }
61
62             // just changing parents
63             $this->Aro->save( array(
64                 'parent_id' => $parent_id,
65                 'id'        => $aro_record['Aro']['id']
66             ) );
67         }
68     }
69     return true;
70 }

```

APP/models/user.php

PasteBin : <http://pastebin.com/Ru9Vz332>



afterSave() callback dalam APP/models/group.php

```

8      function afterSave($created) {
9          if($created) {
10
11              $aro = new Aro();
12              $aro->updateAll( array( 'alias'=>  '"Group:'.$this->id  .'",',
13                                     array(
14                                         'Aro.model'=>'Group', // condition 1
15                                         'Aro.foreign_key'=> $this->id // condit
16                                     )
17                                 );
18          }
19          else {
20              // its an edit, we have to update the tree
21              $data = $this->read();
22              $parent_id = $data['User']['group_id'];
23
24              $aro = new Aro();
25              $aro_record = $aro->findByForeignKey( $this->id );
26              $parent_record = $aro->findByForeignKey( $parent_id );
27              if ( !empty( $aro_record ) ) {
28                  // orphaned child
29                  $this->Aro->save( array(
30                      'model' => 'Group',
31                      'foreign_key' => $this->id,
32                      'alias' => 'Group:'.$this->id,
33                      'parent_id' => $parent_record['Aro']['id']
34                  ) );
35
36                  // just changing parents
37                  // just moving nodes
38                  $this->Aro->save( array(
39                      'model' => 'Group',
40                      'foreign_key' => $this->id,
41                      'alias' => 'Group:'.$this->id,
42                      'parent_id' => $parent_record['Aro']['id'],
43                      'id' => $aro_record['Aro']['id']
44                  ) );
45              }
46          }
47          return true;
48      }

```

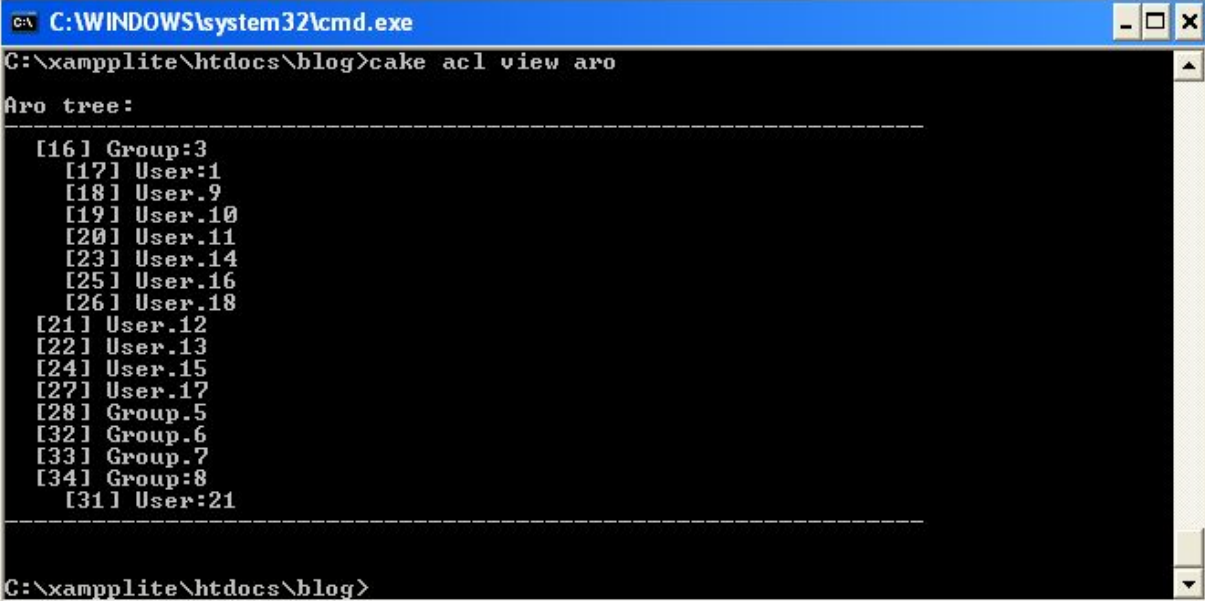
APP/models/group.php

PasteBin : <http://pastebin.com/TtwAp99h>

Mari kita coba :

- Tambah User baru
- Tambah Group baru
- Update Gser ke Group lain

Kemudian lihat nilai alias ARO



```
C:\WINDOWS\system32\cmd.exe
C:\xampplite\htdocs\blog>cake acl view aro

Aro tree:
-----
[16] Group:3
    [17] User:1
    [18] User:9
    [19] User:10
    [20] User:11
    [23] User:14
    [25] User:16
    [26] User:18
[21] User:12
[22] User:13
[24] User:15
[27] User:17
[28] Group:5
[32] Group:6
[33] Group:7
[34] Group:8
    [31] User:21
-----
C:\xampplite\htdocs\blog>
```

Perhatikan ARO id 34 dan 31. User dan Group kini selari dengan susunan di table Users dan Groups

## Panel Kawalan ACO , Security

Menggunakan Cake CLI untuk mengurus ACO mungkin mudah untuk 1-2 Controller. Tetapi bagaimana jika anda mempunyai sistem yang besar.

Jadi seterusnya kita akan bina panel kawalan Aco bernama SecuriyAccess. Panel ini boleh melakukan perkara berikut

- Senaraikan semua Controller dan Action
- Tambah/Buang Controller dan Action ke dalam rekod ACO

Jadi, bagaimana kita hendak dapatkan senarai Controller dan Action dalam setiap Controller ? Dalam CakePHP untuk dapatkan senarai semua Controller, gunakan arahan berikut

```
$controllers = App::objects('controller');
```

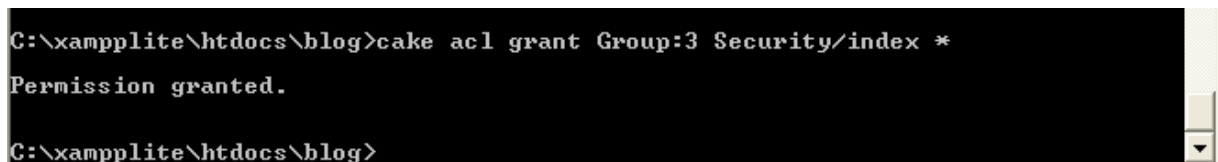
Dan untuk dapatkan senarai Action dalam setiap Controller

```
$actions = get_class_methods($className);
```

Saya telah menyiapkan kod lengkap untuk manage Controller dan Action

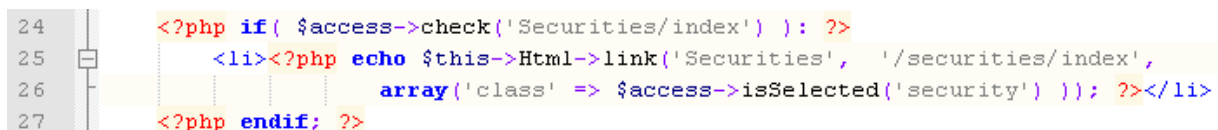
Download : <http://mamakspot.googlecode.com/files/security.zip>

1. salin securities\_controller.php ke dalam APP/controllers
2. salin index.ctp ke dalam APP/views/securities/index.ctp  
( bina dahulu folder APP/views/securities )
3. salin list\_actions ke dalam APP/views/elements/
4. Grant Group:3



```
C:\xampplite\htdocs\blog>cake acl grant Group:3 Security/index *
Permission granted.
C:\xampplite\htdocs\blog>
```

4. Tambah pautan ke Security di dalam APP/views/elements/top\_menu.ctp



```
24 <?php if( $access->check('Securities/index') ): ?>
25 <li><?php echo $this->Html->link('Securities', '/securities/index',
26     array('class' => $access->isSelected('security') ); ?></li>
27 <?php endif; ?>
```

5. Buka alamat <http://localhost/blog/securities>
6. Menu Security sepatutnya muncul dalam senarai.

Logged in as bazet My Account | Logout

Home Posts Groups Users

Pages	Groups	Posts
Display <input checked="" type="checkbox"/> Add	Index <input checked="" type="checkbox"/> Add	Index <input checked="" type="checkbox"/> Add
	View <input checked="" type="checkbox"/> Add	View <input checked="" type="checkbox"/> Add
	Add <input checked="" type="checkbox"/> Add	Add <input checked="" type="checkbox"/> Add
	Edit <input checked="" type="checkbox"/> Add	Edit <input checked="" type="checkbox"/> Add
	Delete <input checked="" type="checkbox"/> Add	Delete <input checked="" type="checkbox"/> Add

Security	Users
Index <input checked="" type="checkbox"/> Add	Login <input checked="" type="checkbox"/> Add
	Logout <input checked="" type="checkbox"/> Add
	Register <input checked="" type="checkbox"/> Add
	Index <input checked="" type="checkbox"/> Add
	View <input checked="" type="checkbox"/> Add
	Add <input checked="" type="checkbox"/> Add
	Edit <input checked="" type="checkbox"/> Add
	Delete <input checked="" type="checkbox"/> Add

Submit

Dengan menggunakan Panel Kawalan ACO, kita boleh menambah dan membuang senarai ACO dari dalam senarai database ACL

## Groups

Index	<input type="checkbox"/> Delete
View	<input type="checkbox"/> Delete
Add	<input type="checkbox"/> Delete
Edit	<input type="checkbox"/> Delete
Delete	<input type="checkbox"/> Delete

## Users

Login	<input checked="" type="checkbox"/> Add
Logout	<input checked="" type="checkbox"/> Add
Register	<input checked="" type="checkbox"/> Add
Index	<input checked="" type="checkbox"/> Add
View	<input checked="" type="checkbox"/> Add
Add	<input checked="" type="checkbox"/> Add
Edit	<input checked="" type="checkbox"/> Add
Delete	<input checked="" type="checkbox"/> Add

Warna merah bermaksud Action masih belum ditambah dan Warna hijau adalah sebaliknya

## Panel Kawalan ARO < > ACO

Kini tiba ke bahagian paling akhir, membina panel kawalan untuk mengawal akses Group (ARO) terhadap Controller/Action ( ACO ). Kita hanya menggunakan Group sahaja kerana setiap User di bawah Group akan mewarisi Permission yang disetkan kepada Group.

Sila download fail : <http://mamakspot.googlecode.com/files/group.zip>

1. Salin groups\_controller ke APP/controllers/groups\_controllers.php
2. Salin index.ctp dan security.ctp ke APP/views/\*
3. Buka Groups Index dan pilih link Security



4. Paparan seperti di bawah akan terpapar, kini kita boleh memberi akses berdasarkan group.

### Security Access for User

Pages		Groups		Posts	
Action	Choice	Action	Choice	Action	Choice
display	<input checked="" type="radio"/> allow <input type="radio"/> deny	security	<input checked="" type="radio"/> allow <input type="radio"/> deny	index	<input checked="" type="radio"/> allow <input type="radio"/> deny
		index	<input checked="" type="radio"/> allow <input type="radio"/> deny	view	<input checked="" type="radio"/> allow <input type="radio"/> deny
		view	<input checked="" type="radio"/> allow <input type="radio"/> deny	add	<input checked="" type="radio"/> allow <input type="radio"/> deny
		add	<input checked="" type="radio"/> allow <input type="radio"/> deny	edit	<input checked="" type="radio"/> allow <input type="radio"/> deny
		edit	<input checked="" type="radio"/> allow <input type="radio"/> deny	delete	<input checked="" type="radio"/> allow <input type="radio"/> deny
		delete	<input checked="" type="radio"/> allow <input type="radio"/> deny		

# Sesi Ujian

Untuk sesi ini kita perlu mengosongkan kesemua data dan bermula dari fresh.

1. Bina semua skema database dengan menaip **cake schema generate** ( pilih overwrite )

```
C:\xampplite\htdocs\blog>cake schema generate
Welcome to CakePHP v1.3.3 Console
-----
App : blog
Path: C:\xampplite\htdocs\blog
-----
Cake Schema Shell
-----
Generating Schema...
Schema file exists.
[O]verwrite
[S]napshot
[Q]uit
Would you like to do? (o/s/q)
[sl] > o
Schema file: schema.php generated
```

2. Bina semula table database, taip **cake schema create** ( akan overwrite semua data )

```
C:\xampplite\htdocs\blog>cake schema create
Welcome to CakePHP v1.3.3 Console
-----
App : blog
Path: C:\xampplite\htdocs\blog
-----
Cake Schema Shell
-----
The following table(s) will be dropped.
acos
aros
aros_acos
groups
posts
users
Are you sure you want to drop the table(s)? (y/n)
[ln] > y
Dropping table(s).
acos updated.
aros updated.
aros_acos updated.
groups updated.
posts updated.
users updated.

The following table(s) will be created.
acos
aros
aros_acos
groups
posts
users
Are you sure you want to create the table(s)? (y/n)
[y] > y
Creating table(s).
acos updated.
aros updated.
aros_acos updated.
groups updated.
posts updated.
users updated.
End create.
```

3. Benarkan akses tanpa login, ubah **APP/app\_controller** dengan letak kod untuk allow semua actions dalam fungsi `beforeFilter()`;

```

7  function beforeFilter() {
8      // Acl
9      $this->Auth->authorize = 'actions';
10     #$this->Auth->allowedActions = array('display');
11     // untuk tujuan setup sahaja
12     $this->Auth->allowedActions = array('*');
13 }
14

```

4. Buka <http://localhost/blog/securities>

**Pages**

Display	<input checked="" type="checkbox"/> Add
---------	---

**Groups**

Security	<input checked="" type="checkbox"/> Add
Index	<input checked="" type="checkbox"/> Add
View	<input checked="" type="checkbox"/> Add
Add	<input checked="" type="checkbox"/> Add
Edit	<input checked="" type="checkbox"/> Add
Delete	<input checked="" type="checkbox"/> Add

**Securities**

Index	<input checked="" type="checkbox"/> Add
-------	---

**Users**

Login	<input checked="" type="checkbox"/> Add
Logout	<input checked="" type="checkbox"/> Add
Register	<input checked="" type="checkbox"/> Add
Index	<input checked="" type="checkbox"/> Add
View	<input checked="" type="checkbox"/> Add
Add	<input checked="" type="checkbox"/> Add
Edit	<input checked="" type="checkbox"/> Add
Delete	<input checked="" type="checkbox"/> Add

Submit

Pilih semua Actions dan klik Submit

5. Buka <http://localhost/blog/groups> dan tambah Group seperti di bawah :

### Groups

Id	Name	Description	Created	Modified
1	Super User	Super User	2010-08-20 15:39:34	2010-08-20 15:39:34
2	Administrator	Administrator	2010-08-20 15:39:55	2010-08-20 15:39:55
3	Normal User	Normal User	2010-08-20 15:40:09	2010-08-20 15:40:09
4	Suspended	Suspended	2010-08-20 15:40:26	2010-08-20 15:40:26

Page 1 of 1, showing 4 records out of 4 total, starting on record 1, ending on 4

6. Pilih security untuk group Normal User ( Group:3 )

7. Allow kesemua Actions buat sementara

**Security Access for Normal User**

Pages		Groups		Posts	
Action	Choice	Action	Choice	Action	Choice
display	<input checked="" type="radio"/> allow <input type="radio"/> deny	security	<input checked="" type="radio"/> allow <input type="radio"/> deny	index	<input checked="" type="radio"/> allow <input type="radio"/> deny
		index	<input checked="" type="radio"/> allow <input type="radio"/> deny	view	<input checked="" type="radio"/> allow <input type="radio"/> deny
		view	<input checked="" type="radio"/> allow <input type="radio"/> deny	add	<input checked="" type="radio"/> allow <input type="radio"/> deny
		add	<input checked="" type="radio"/> allow <input type="radio"/> deny	edit	<input checked="" type="radio"/> allow <input type="radio"/> deny
		edit	<input checked="" type="radio"/> allow <input type="radio"/> deny	delete	<input checked="" type="radio"/> allow <input type="radio"/> deny
		delete	<input checked="" type="radio"/> allow <input type="radio"/> deny		

8. Ubah APP/app\_controller.php dan matikan kod untuk hanya Allow action 'display'

```

7      function beforeFilter() {
8          // Acl
9          $this->Auth->authorize = 'actions';
10         $this->Auth->allowedActions = array('display');
11         // untuk tujuan setup sahaja
12         #$this->Auth->allowedActions = array('*');
13     }

```

9. Pergi ke <http://localhost/blog>, register diri anda.

**Register**

Username  
bazet

Password  
••••••••

Email  
azril.nazli@gmail.com

10. Anda akan terus diloginkan ke dalam sistem dan mempunyai akses penuh
11. Pergi ke Groups, pilih Security untuk Super User. Kemudian update diri anda supaya berada di dalam group SuperUser di dalam menu Users.



## Edit User

Group

Normal User

Super User

Administrator

Normal User

Suspended

Password

••••••••

Email

azril.nazli@gmail.com

Submit

Pastikan anda taip semua password anda

12. Daftar satu user baru di bawah Group User, dan satu di bawah Suspended
13. Untuk group User, disablekan akses ke Users,Securities,Groups. Cuma Allow Pages dan Posts. Untuk group Suspended, disable semua akses kecuali Pages

## Security Access for Normal User

Pages		Groups		Posts	
Action	Choice	Action	Choice	Action	Choice
display	<input checked="" type="radio"/> allow <input type="radio"/> deny	security	<input type="radio"/> allow <input checked="" type="radio"/> deny	index	<input checked="" type="radio"/> allow <input type="radio"/> deny
		index	<input type="radio"/> allow <input checked="" type="radio"/> deny	view	<input checked="" type="radio"/> allow <input type="radio"/> deny
		view	<input type="radio"/> allow <input checked="" type="radio"/> deny	add	<input checked="" type="radio"/> allow <input type="radio"/> deny
		add	<input type="radio"/> allow <input checked="" type="radio"/> deny	edit	<input checked="" type="radio"/> allow <input type="radio"/> deny
		edit	<input type="radio"/> allow <input checked="" type="radio"/> deny	delete	<input checked="" type="radio"/> allow <input type="radio"/> deny
		delete	<input type="radio"/> allow <input checked="" type="radio"/> deny		

Hanya Allow Posts sahaja

14. Coba login sebagai Super User, Normal User dan Suspended. Menu di sebelah kanan akan berubah mengikut akses, dan jika anda cuba akses dengan menaip alamat URL, sistem ACL akan DENY akses user tersebut.
15. Di dalam APP/controllers/securities\_controller.php, matikan kod `$this->Auth->allow('index');`

```
1 <?php
2 class SecuritiesController extends AppController {
3
4     var $name = 'Securities';
5     var $uses = null;
6
7     function beforeFilter() {
8         // we need to temporarily allow access during the setup
9         //$this->Auth->allow( 'index' );
10    }
```

16. Ubah nilai Debug kepada 0 di dalam APP/Config/core.php

# Kesimpulan

---

Source code tutorial ini beserta Schema dan MySQL Dump boleh didapati di alamat berikut :

<http://mamakspot.googlecode.com/files/ACLControlPanel.zip>

Akhirnya kita berjaya membina sistem yang boleh mengawal access User berdasarkan Permission yang diberikan mengikut Group. Framework ini boleh dijadikan asas untuk membina sistem yang lagi besar dan kompleks. Tutorial ini menggunakan :

- CakePHP 1.3.3 Stable sebagai Framework
- Xampplite 1.7 versi Windows untuk solution PHP,MySQL dan Apache
- Notepad ++ Sebagai Editor
- Rujukan di CakePHP Google Group, Nettuts dan buku Practical CakePHP projects

Ebook ini adalah sebahagian dari bahan kursus **Advanced CakePHP** yang dikendalikan oleh penulis di <http://azrilnazli.blogspot.com/p/cakephp-training-modules.html>

Buku ini dihasilkan dengan tujuan untuk melahirkan seberapa banyak pembangun CakePHP di Malaysia.

Jika anda ingin menyumbang sekadar untuk galakan penulis untuk menghasilkan buku-buku CakePHP percuma di masa akan datang, sila gunakan akaun di bawah :

Nama : Azril Nazli Bin Alias  
Email : azril.nazli [at] Gmail.com  
Mobile : 019 371 8462  
Bank : Maybank  
No Akaun : **164 182 401 644**

Senarai buku-buku percuma CakePHP yang lain boleh didapati di <http://azrilnazli.blogspot.com/p/buku-panduan-cakephp.html>