

FREE

# Menjelajah Yii Framework

Belajar Membuat Aplikasi dengan Yii Framework



yii framework

With  
Source Code



Sabit Huraira

2012

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat-Nya sehingga penulis dapat menyelesaikan ebook ini. Ebook ini saya tulis dengan tujuan untuk membantu teman-teman komunitas dan penulis sendiri dalam menggali lebih jauh ilmu pengetahuan terutama Yii Framework.

Ebook ini merupakan kompilasi dari artikel-artikel blog saya <http://sabitlabscodewordpress.com/> yang saya kemas ulang dengan tujuan mempermudah teman-teman dalam belajar. Ebook ini penulis tuju untuk para developer Yii maupun teman-teman yang baru mendalami Yii. Developer dari framework PHP yang berbeda maupun bahasa pemrograman yang berbeda juga tidak mungkin saya larang untuk membaca ebook ini, bahkan saya anjurkan agar anda mencicipi kelezatan Yii...

Walau sudah berusaha keras, penulis yakin buku ini masih jauh sangat... dari sempurna. Untuk itu feedback dari pembaca sangat saya harapkan baik itu dalam bentuk kata-kata ataupun uang tunai.. hehe.. Ga kok, saran, kritik dan masukan yang membangun amat sangat saya harapkan dari para pembaca..

Rasa terima kasih akhirnya harus penulis sampaikan kepada berbagai pihak yang telah membantu dalam menyelesaikan ebook ini: orang tua, keluarga, pacar, teman kampus Sekolah Tinggi Ilmu Statistik (STIS), rekan kerja di Badan Pusat Statistik (BPS) dan pihak lain yang ga bisa saya sebutkan satu persatu. Juga kepada penghuni programmer forum KASKUS yang sering debat sakti (asp.net vs php lah, java vs .net lah, hingga aksi “stop using CI”), walau disana penuh caci maki tapi sangat banyak ilmu yang saya dapat dari debat-debat kalian. Dan tentunya kepada Qiang sang founder Yii Framework dan semua komunitas Yii Framework terima kasih sudah bersedia berbagi ilmu pengetahuan bersama...

Akhir kata, semoga ebook ini dapat membantu dan berguna bagi kita semua...

Jakarta, Januari 2012

Sabit Huraira

## DAFTAR ISI

KATA PENGANTAR .....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR .....	vi
PART I.....	1
OVERVIEW .....	1
Apa itu Yii?? .....	2
Apa yang akan dipelajari dari ebook ini??.....	2
Target Pembaca.....	3
Aplikasi seperti apa yang akan kita bangun dalam ebook ini?? .....	4
Apa yang anda butuhkan untuk bermain bersama Yii Framework?? .....	11
PART II.....	13
MENJAMAH YII FRAMEWORK .....	13
Instalasi .....	14
Membuat Aplikasi Pertama dengan Yii Framework.....	15
Model-View-Controller.....	17
Membuat Model .....	18
Membuat View (Tampilan).....	23
Membuat Controller .....	25
Menjelajah Database dengan Yii Framework .....	30

Extension.....	38
PART III .....	39
YII IN ACTION .....	39
Membuat Database.....	40
Menghubungkan Database dengan Aplikasi Yii Framework .....	40
Make It Easy With Gii .....	42
Membuat Operasi CRUD dengan Gii .....	44
How About Gii Code??.....	50
PART IV .....	69
MEMBUAT APLIKASI FORUM DENGAN YII FRAMEWORK .....	69
Mau Aplikasi Seperti Apa?? .....	71
Merancang Database .....	72
Membuat Aplikasi, Menghubungkan Database, dan Membuat CRUD Operation .....	76
Membuat Operasi Register, Login dan Logout.....	76
Membuat User Level Akses .....	85
Memanfaatkan Extension Untuk Tampilan .....	87
Beresin Aplikasi.....	93
DAFTAR PUSTAKA .....	113
TENTANG PENULIS .....	114
Masa Depan Ebook Ini.....	116
Anda berminat bergabung untuk menulis ebook ini ? .....	116





## DAFTAR GAMBAR

<b>No.Gambar</b>	<b>Judul Gambar</b>	<b>Halaman</b>
Gambar 1.	Halaman Utama Aplikasi .....	5
Gambar 2.	Halaman Utama Aplikasi Ketika Login Sebagai Admin .....	6
Gambar 3.	Tampilan Halaman List Forum .....	6
Gambar 4.	Halaman Kelola Kategori .....	7
Gambar 5.	Halaman Pada Kategori .....	7
Gambar 6.	Halaman Membuat Thread Baru .....	8
Gambar 7.	Halaman Suatu Thread .....	8
Gambar 8.	Halaman Membuat Komentar .....	9
Gambar 9.	Halaman Profile User .....	9
Gambar 10.	Halaman Memberi Reputasi Kepada User .....	10
Gambar 11.	Tampilan Halaman Requirement Yii Framework .....	14
Gambar 12.	Perintah Membuat Aplikasi Yii Framework .....	15
Gambar 13.	Pesan Aplikasi Yii Berhasil Dibuat .....	16
Gambar 14.	Tampilan Web Yii .....	17
Gambar 15.	Tampilan CGridView .....	25
Gambar 16.	Tampilan Halaman Gii .....	43
Gambar 17.	Tampilan Halaman Utama Gii .....	44
Gambar 18.	Tampilan Halaman Model Generator Pada Gii .....	45
Gambar 19.	Tampilan Halaman CRUD Generator Pada Gii .....	46
Gambar 20.	Tampilan Halaman Tabel Mahasiswa .....	47
Gambar 21.	Halaman Untuk Membuat Data Baru .....	47
Gambar 22.	Halaman Untuk Melihat Data Dengan ID Tertentu .....	48

Gambar 23. Halaman Untuk Memperbaharui Data Dengan ID Tertentu .....	48
Gambar 24. Halaman Untuk Melihat Data Semua Mahasiswa.....	49
Gambar 25. Halaman Admin .....	49
Gambar 26. Halaman Admin Dengan Advanced Search.....	50
Gambar 27. Tampilan Extension CLEditor .....	88
Gambar 28. Tampilan Extension MbMenu.....	90
Gambar 29. Tampilan Halaman Kategori .....	95
Gambar 30. Tampilan Suatu Kategori Thread .....	96
Gambar 31. Halaman Utama Aplikasi .....	104
Gambar 32. Tampilan Halaman Suatu Thread.....	107
Gambar 33. Tampilan Halaman Membuat Komentar Baru .....	108





# **PART I**

# **OVERVIEW**

## **Apa itu Yii??**

Apa sih itu Yii?? Hmm, berikut penjelasan Yii yang saya kutip dari website resmi Yii di <http://www.yiiframework.com/> : “Yii adalah framework (kerangka kerja) PHP berbasis-komponen, berkinerja tinggi untuk pengembangan aplikasi Web berskala-besar. Yii menyediakan reusability maksimum dalam pemrograman Web dan mampu meningkatkan kecepatan pengembangan secara signifikan.”. Okeh-okeh, silahkan dicerna ya.. ☺ .Yii sendiri merupakan kepanjangan dari “Yes It Is”. Yes It Is sendiri memiliki makna kalo Yii mampu dan tepat anda pilih untuk mengerjakan project anda. Jadi kalo sewaktu di warteg ada yang nanya : “Is It Fast??.. Is It Secure??.. Is It Professional??.. Is it Right for my next project?? ” Jawablah “Yes It Is”.

Project ini sendiri pertama kali dikembangkan oleh seorang master bernama Qiang Xue pada Januari 2008 dan pada Desember 2008 Yii 1.0 dirilis untuk publik. Dan sungguh menakjubkan, baru berselang beberapa tahun Yii sekarang sudah menjadi PHP framework yang diminati banyak orang termasuk anda yang sedang membaca ebook ini. Yii sendiri tentunya ga kalah bagus dengan PHP Framework yang lainnya, hal tersebut dapat dilihat dari segudang fitur yang dimiliki oleh Yii sendiri.

## **Apa yang akan dipelajari dari ebook ini??**

Ebook ini akan mempelajari penggunaan Yii Framework untuk membangun sebuah aplikasi web, selain itu akan dibahas juga contoh

pembangunan sebuah aplikasi real yaitu aplikasi forum. Berikut bahasan yang akan dipaparkan dalam ebook ini :

- Sekilas tentang Yii Framework
- Model-View-Controller pada Yii Framework
- Form dengan Yii Framework
- Database dengan Yii Framework
- Menggunakan Extension pada Yii Framework
- Membuat operasi CRUD(Create-Read-Update-Delete) dengan Yii Framework.
- Membangun Aplikasi Forum dengan Yii Framework
- Dan lain-lain...

Yah, kurang lebih itulah yang akan kita bahas pada ebook ini. Semoga anda tertarik dengan bahasannya.. ☺

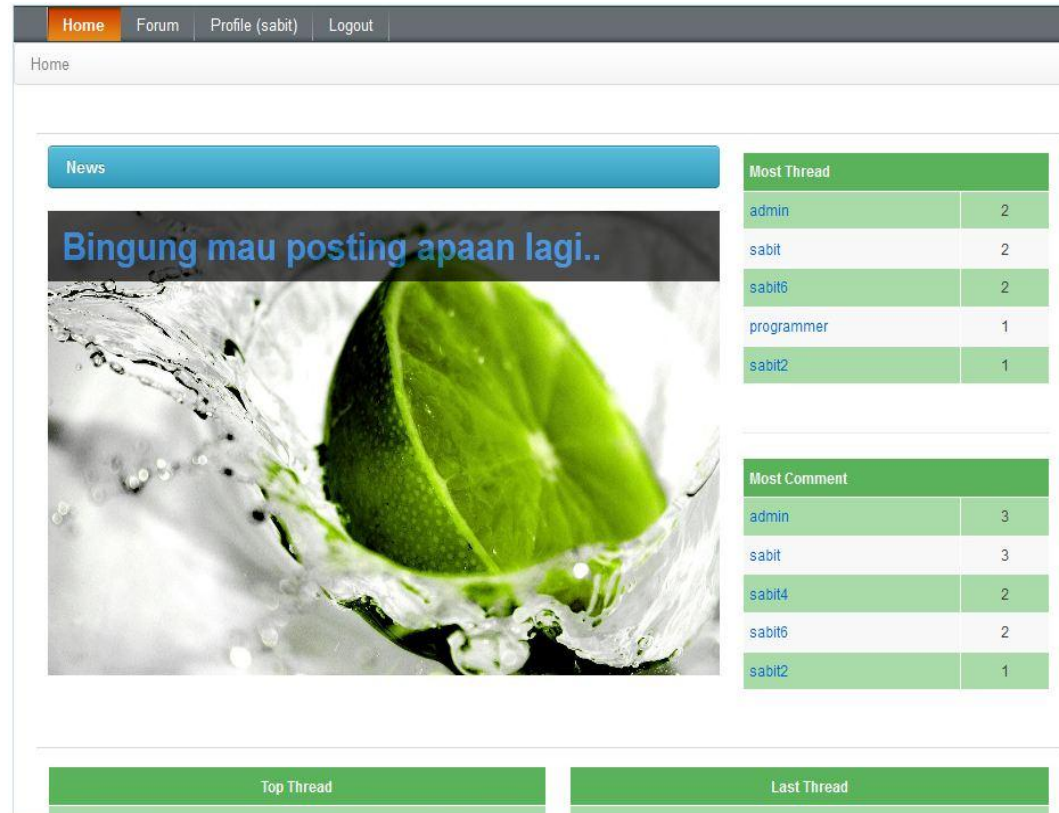
### **Target Pembaca**

Ebook ini saya peruntukkan untuk semua kalangan yang tertarik untuk mengenal, mempelajari dan mengembangkan aplikasi berbasis web dengan menggunakan Yii Framework. Ebook ini tidaklah membahas Yii Framework secara luas, lengkap dan mendalam tetapi saya harap buku ini dapat menjadi fondasi awal agar dapat mempermudah anda mempelajari Yii Framework. Pembaca ebook ini saya asumsikan paling tidak mengerti akan bahasa pemrograman PHP, HTML dan pengoperasian database (dalam buku ini saya gunakan MySql). Jika anda belum banyak memahami hal tersebut, saya harap

anda dapat mulai mempelajarinya terlebih dahulu.. Karena akan sangat sulit untuk memahami Yii Framework jika anda tidak memahami dasar pemrograman PHP, HTML dan pengoperasian database.. Okeh, Selamat menikmati.. ☺

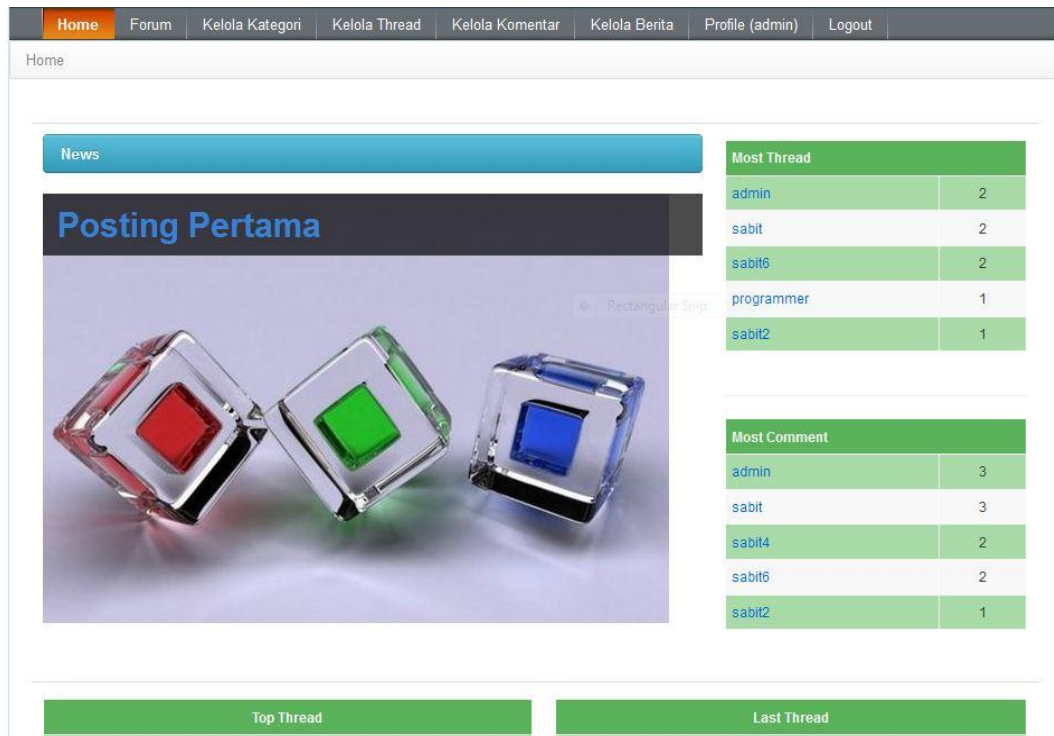
### **Aplikasi seperti apa yang akan kita bangun dalam ebook ini??**

Selain mengupas Yii Framework, ebook ini juga akan memaparkan sebuah contoh pembangunan aplikasi web secara real. Sebelum kita melangkah lebih jauh, saya akan jelaskan aplikasi seperti apa yang akan kita pelajari dalam ebook ini. Yah, kali-kali aja bisa menambah minat anda sebagai pembaca untuk mengupas tuntas isi dari ebook ini setelah tau sedikit gambaran aplikasi yang akan kita bangun dari ebook ini... Aplikasi yang akan kita pelajari dari ebook ini adalah sebuah aplikasi forum yang di dalamnya akan terjadi interaksi antar user. User dapat membuat sebuah postingan dan user lain dapat melihat postingan tersebut dan memberikan komentar atau penilaian pada postingan tersebut.. Berikut sedikit gambaran screenshot dari aplikasi yang akan kita bahas pada buku ini :



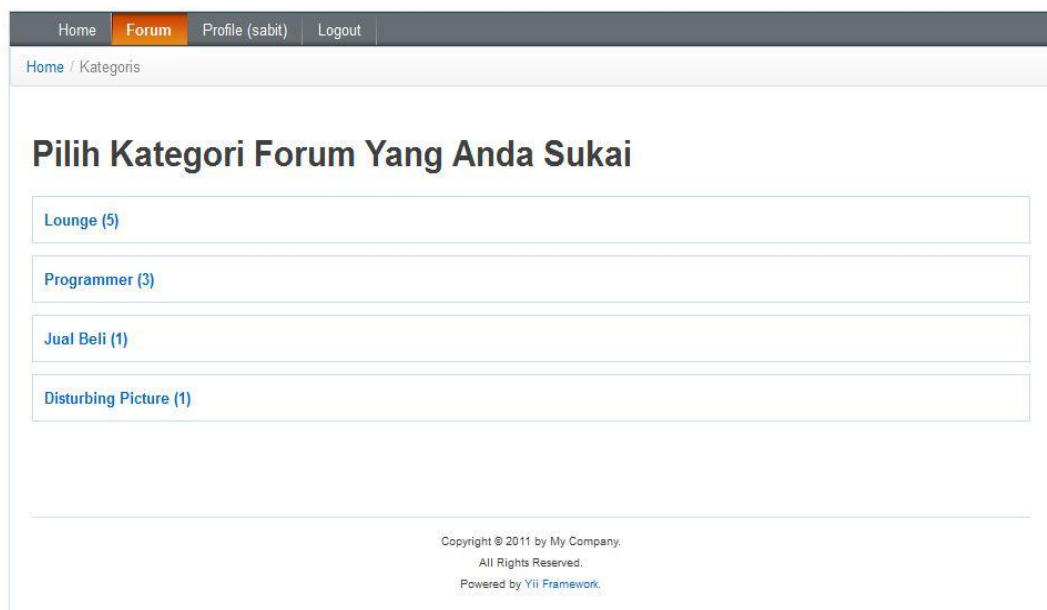
Gambar 1. Halaman Utama Aplikasi

Halaman utama aplikasi yang akan menampilkan berita terkini dalam bentuk slide, list nama user yang paling banyak membuat thread dan paling banyak membuat komentar, list thread yang paling banyak dikomentari dan thread terbaru yang telah di posting oleh user. Tampilan di atas adalah tampilan halaman utama aplikasi ketika diakses dengan member biasa. Apabila aplikasi diakses oleh user dengan level admin, maka akan tampil halaman seperti gambar berikut :



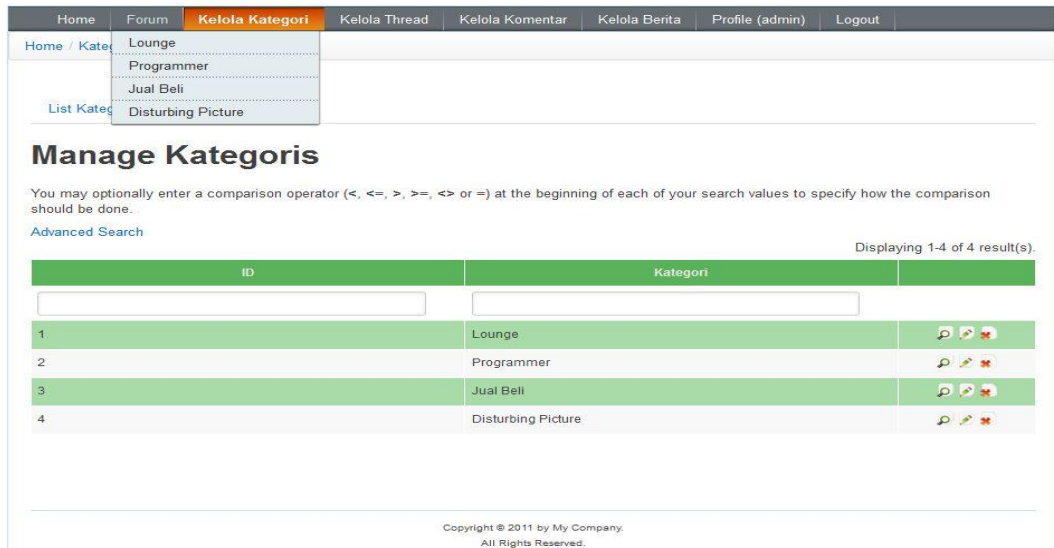
Gambar 2. Halaman Utama Aplikasi Ketika Login Sebagai Admin

Yang berbeda dengan halaman yang aplikasi diakses oleh user adalah menu yang disediakan. Jika kita login sebagai admin, maka kita diizinkan melakukan pengelolaan kategori forum, thread, komentar, dan berita.



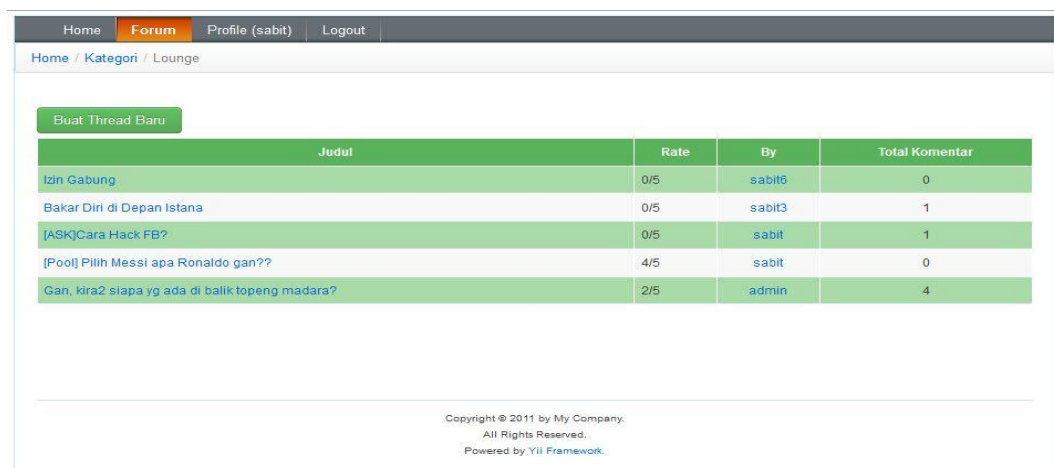
Gambar 3. Tampilan Halaman List Forum

Tampilan ini menampilkan semua kategori forum yang ada pada aplikasi beserta total thread yang ada pada kategori tersebut.



Gambar 4. Halaman Kelola Kategori

Halaman kelola kategori hanya dapat diakses oleh user dengan level admin. Selain melakukan pengelolaan kategori, admin juga dapat disediakan menu untuk melakukan pengelolaan thread, komentar, dan berita.

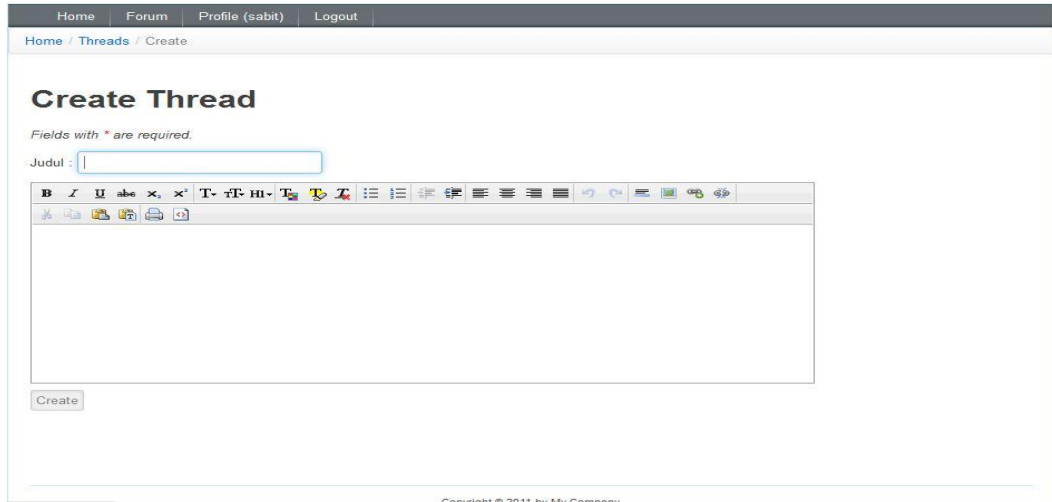


Gambar 5. Halaman Pada Kategori

Ketika user membuka salah satu halaman kategori, maka akan ditampilkan list semua thread yang ada pada kategori tersebut. User akan melihat informasi judul



thread, rating, pembuat thread, dan total komentar yang ada pada thread tersebut. Jika user ingin melihat thread secara detail, user cukup melakukan klik pada judul thread tersebut. User juga bisa membuat thread baru pada forum tersebut.



Gambar 6. Halaman Membuat Thread Baru

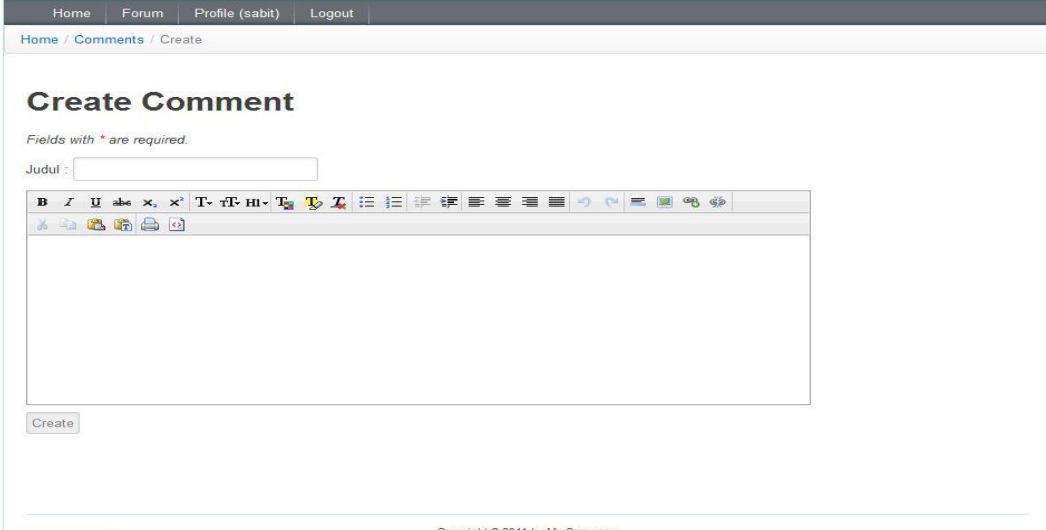
Halaman di atas merupakan halaman yang akan muncul jika user ingin membuat thread baru.



Gambar 7. Halaman Suatu Thread

Halaman ini adalah halaman yang muncul ketika kita akan melihat suatu thread. Kita akan diperlihatkan isi dari thread tersebut beserta komentar-komentar yang

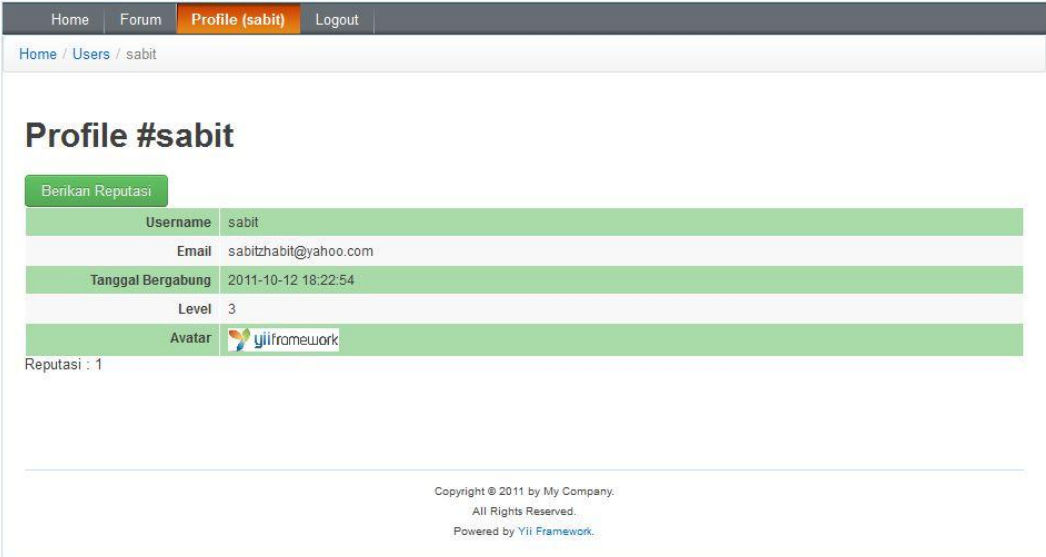
ada pada thread tersebut. User juga dapat memberikan komentar pada thread tersebut dan memberikan penilaian pada thread tersebut dengan memilih bintang dan menekan tombol “vote” yang terdapat halaman tersebut.




The screenshot shows a web interface for creating a comment. At the top, there is a navigation bar with links for Home, Forum, Profile (sabit), and Logout. Below the navigation bar, the breadcrumb trail reads 'Home / Comments / Create'. The main heading is 'Create Comment'. A note states 'Fields with \* are required.' Below this, there is a text input field labeled 'Judul :'. Underneath the input field is a rich text editor toolbar with various icons for bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, and other formatting options. A large text area is provided for the comment content. At the bottom left of the form is a 'Create' button. At the bottom center of the page, there is a small copyright notice: 'Copyright © 2011 by My Company.'

Gambar 8. Halaman Membuat Komentar

Halaman ini akan muncul ketika user memilih untuk mengomentari suatu thread.



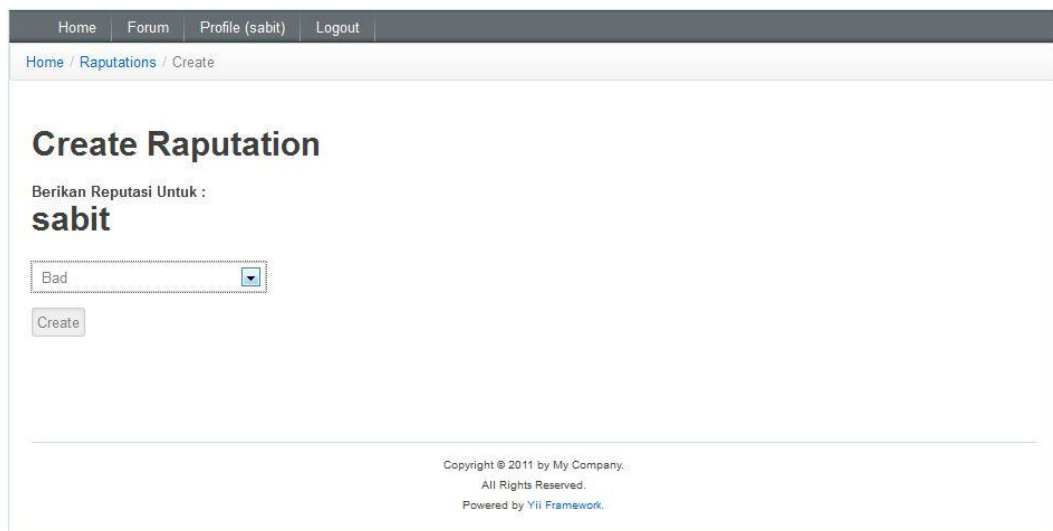
The screenshot shows a user profile page. At the top, there is a navigation bar with links for Home, Forum, Profile (sabit), and Logout. Below the navigation bar, the breadcrumb trail reads 'Home / Users / sabit'. The main heading is 'Profile #sabit'. Below the heading is a green button labeled 'Berikan Reputasi'. The profile information is displayed in a table-like format with alternating green and white rows:

Username	sabit
Email	sabithabit@yahoo.com
Tanggal Bergabung	2011-10-12 18:22:54
Level	3
Avatar	

Below the profile information, it says 'Reputasi : 1'. At the bottom center of the page, there is a small copyright notice: 'Copyright © 2011 by My Company. All Rights Reserved. Powered by Yii Framework.'

Gambar 9. Halaman Profile User

Halaman ini adalah halaman profile seorang user. Pada halaman ini akan diperlihatkan informasi user. Tiap user dapat memberikan penilaian pada seorang user dengan memilih menu “Berikan Reputasi” yang tersedia pada halaman profile user.



Gambar 10. Halaman Memberi Reputasi Kepada User

Halaman ini adalah halaman untuk memberikan reputasi kepada user. Kita dapat memberikan reputasi “Bad” atau “Good”, dimana setiap pemberian reputasi akan mempengaruhi penilaian kepada user.

Itulah sedikit mengenai gambaran aplikasi yang akan kita buat. Aplikasi itu sendiri merupakan aplikasi sederhana yang masih jauh jika ingin dibilang kompleks, jadi peluang untuk anda berkreasi dan mengembangkannya sangat terbuka lebar..

## **Apa yang anda butuhkan untuk bermain bersama Yii Framework??**

Yang jelas anda butuh niat dan kemauan, komputer/laptop juga jangan dilupakan, mp3 dan kopi optional.. kalo bener-bener udah suntuk, bolehlah siapin game untuk selingan. Dan jelas anda harus siapkan mata yang udah siap untuk begadang biar lebih produktif menggali Yii Framework ☺. Tapi apalah artinya persiapan di atas kalo kita ga menyiapkan alat perang kita untuk menaklukan Yii Framework. Yak berikut adalah beberapa tools yang anda perlukan untuk melanjutkan perjuangan anda menundukkan Yii Framework :

- Yii Framework, bisa anda download langsung di : <http://www.yiiframework.com/download/>. Dalam ebook ini, saya menggunakan Yii Framework versi 1.1.7. Kalo anda mau menggunakan fitur terbaru ga masalah, tapi saya sarankan jangan pake fitur yang lebih jadul dari versi 1.1.7 , takutnya ada fitur yang ga di-support fungsi pada versi 1.1.7 ...
- Web Server dan database. Saya sendiri menggunakan apache server dan mysql database. Kalo anda ga mau repot, anda bisa download packet server yang di dalamnya sudah terdapat apache server dan mysql database. Jadi anda cukup satu kali install dan selesai. Beberapa contoh packet server yang dapat anda gunakan adalah Wamp Server, Xampp, dll.. Tanpa maksud promosi, saya sendiri menggunakan wamp server.. Untuk link download, silahkan googling yak..
- Browser buat testing, running, dan googling kalo lagi mentok.. Terserah mau pake browser apa aja..

- [Optional] Editor untuk ngetik code. Untuk mempermudah proses coding, ga ada salahnya anda pake editor/ IDE. Editor/IDE adalah sebuah alat yang dapat mempermudah dan mempercepat kita melakukan proses coding. Beberapa contoh yang terkenal adalah Netbeans, Eclipse, hingga Notepad++..

# **PART II**

## **MENJAMAH YII FRAMEWORK**

Setelah sudah menyiapkan alat-alat perang, selanjutnya kita harus berlatih dan mempersiapkan segala sesuatu untuk berperang terlebih dahulu...

## Instalasi

Instalasi Yii Framework sangatlah mudah. Setelah anda mendownload yii framework dari website Yii, silahkan anda unpack file tersebut dan pindahkan pada direktori yang dapat diakses di web komputer anda. Lalu akses framework anda dengan memasukkan alamat berikut pada browser anda : <http://localhost/namaFolderYiiFrameworkAnda/requirements/>. Maka pada browser anda akan tampil halaman seperti berikut :

### Description

This script checks if your server configuration meets the requirements for running [Yii](#) Web applications. It checks if the server is running the right version of PHP, if appropriate PHP extensions have been loaded, and if php.ini file settings are correct.

### Conclusion

Your server configuration satisfies the minimum requirements by [Yii](#). Please pay attention to the warnings listed below if your application will use the corresponding features.

### Details

Name	Result	Required By	Memo
PHP version	Passed	<a href="#">Yii Framework</a>	PHP 5.1.0 or higher is required.
\$_SERVER variable	Passed	<a href="#">Yii Framework</a>	
Reflection extension	Passed	<a href="#">Yii Framework</a>	
PCRE extension	Passed	<a href="#">Yii Framework</a>	
SPL extension	Passed	<a href="#">Yii Framework</a>	
DOM extension	Passed	<a href="#">CHtmlPurifier</a> , <a href="#">CWsdGenerator</a>	
PDO extension	Passed	All <a href="#">DB-related classes</a>	
PDO SQLite extension	Passed	All <a href="#">DB-related classes</a>	This is required if you are using SQLite database.
PDO MySQL extension	Passed	All <a href="#">DB-related classes</a>	This is required if you are using MySQL database.
PDO PostgreSQL extension	Passed	All <a href="#">DB-related classes</a>	This is required if you are using PostgreSQL database.
Memcache extension	Warning	<a href="#">CMemCache</a>	
APC extension	Warning	<a href="#">CApcCache</a>	
Mcrypt extension	Passed	<a href="#">CSecurityManager</a>	This is required by encrypt and decrypt methods.
SOAP extension	Passed	<a href="#">CWebService</a> , <a href="#">CWebServiceAction</a>	
GD extension with FreeType support	Passed	<a href="#">CCaptchaAction</a>	

■ passed
 ■ failed
 ■ warning

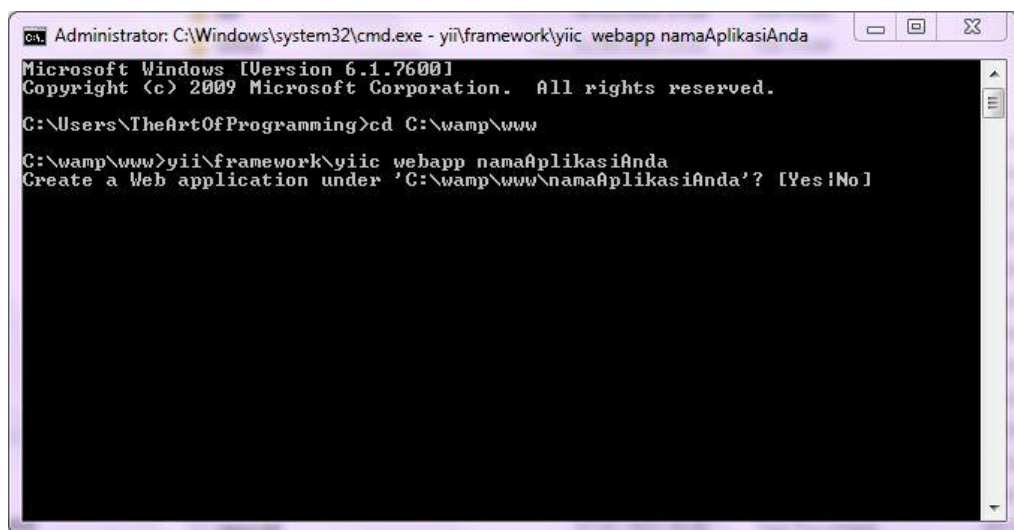
Gambar 11. Tampilan Halaman Requirement Yii Framework

Silahkan anda cek requirement anda masing-masing dan lengkapi requirement tersebut sesuai kebutuhan anda. Selain bisa melihat requirement, anda juga bisa melihat beberapa contoh aplikasi demo yang telah disediakan oleh pengembang

Yii Framework yang tentu saja dibuat dengan Yii Framework. Anda dapat melihat aplikasi tersebut dengan cara memasukkan alamat berikut pada browser anda : <http://localhost/namaFolderYiiFrameworkAnda/demos/>. Cukup banyak contoh aplikasi demo yang ada disana, diantaranya aplikasi “hello word”, blog, game hangman, hingga contoh penggunaan web service dalam aplikasi phone book... Selamat menikmati...

### Membuat Aplikasi Pertama dengan Yii Framework

Setelah berhasil menginstal Yii Framework, sekarang kita akan mencoba membuat aplikasi pertama kita dengan menggunakan Yii Framework. Buka command prompt anda dan masukkan perintah seperti berikut :



```
Administrator: C:\Windows\system32\cmd.exe - yii\framework\yiic webapp namaAplikasiAnda
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

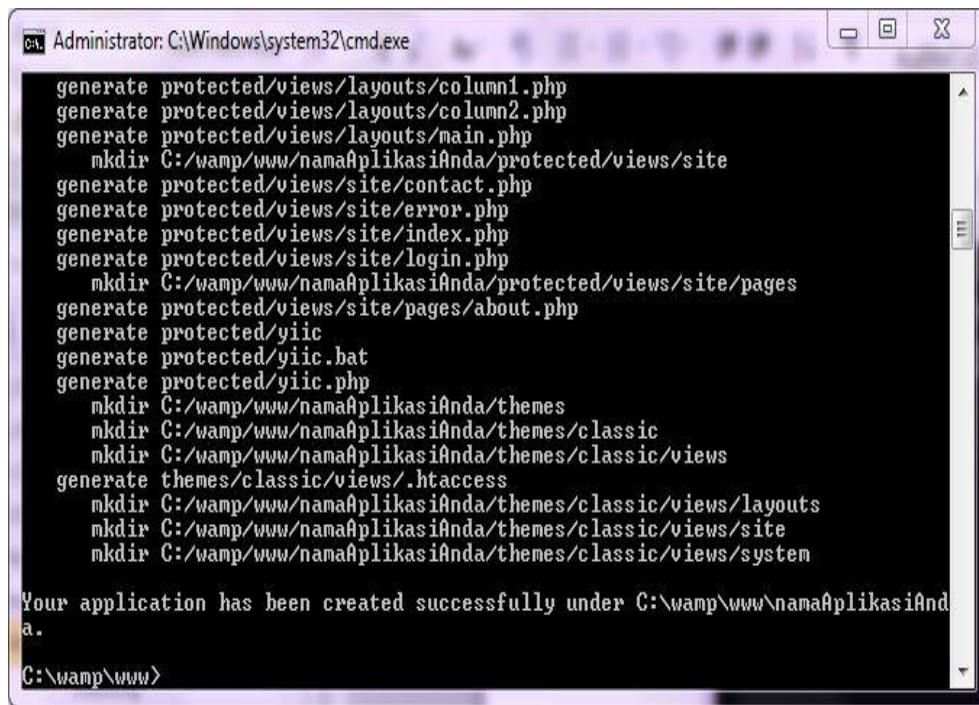
C:\Users\TheArtOfProgramming>cd C:\wamp\www
C:\wamp\www>yii\framework\yiic webapp namaAplikasiAnda
Create a Web application under 'C:\wamp\www\namaAplikasiAnda'? [Yes|No]
```

Gambar 12. Perintah Membuat Aplikasi Yii Framework

Pada baris pertama, kita memerintahkan direktori berpindah ke direktori dimana Yii Framework telah di install. Setelah itu kita akan menjalankan perintah pada file “yiic” yang terdapat di dalam folder yii\framework untuk membuat aplikasi baru dengan nama sesuai dengan yang kita perintahkan. Setelah itu akan muncul



sebuah pesan untuk memastikan kita sudah yakin membuat aplikasi yang kita inginkan, ketikkan perintah “Yes” jika anda sudah yakin untuk membuat aplikasi tersebut. Jika berhasil, maka command prompt akan menampilkan pesan seperti berikut :

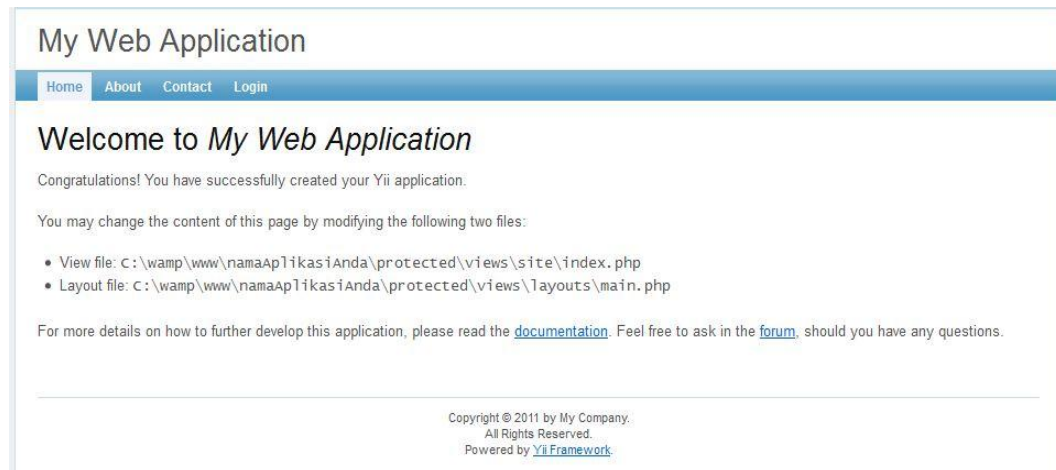


```
Administrator: C:\Windows\system32\cmd.exe
generate protected/views/layouts/column1.php
generate protected/views/layouts/column2.php
generate protected/views/layouts/main.php
mkdir C:/wamp/www/namaAplikasiAnda/protected/views/site
generate protected/views/site/contact.php
generate protected/views/site/error.php
generate protected/views/site/index.php
generate protected/views/site/login.php
mkdir C:/wamp/www/namaAplikasiAnda/protected/views/site/pages
generate protected/views/site/pages/about.php
generate protected/yiic
generate protected/yiic.bat
generate protected/yiic.php
mkdir C:/wamp/www/namaAplikasiAnda/themes
mkdir C:/wamp/www/namaAplikasiAnda/themes/classic
mkdir C:/wamp/www/namaAplikasiAnda/themes/classic/views
generate themes/classic/views/.htaccess
mkdir C:/wamp/www/namaAplikasiAnda/themes/classic/views/layouts
mkdir C:/wamp/www/namaAplikasiAnda/themes/classic/views/site
mkdir C:/wamp/www/namaAplikasiAnda/themes/classic/views/system

Your application has been created successfully under C:\wamp\www\namaAplikasiAnda.
C:\wamp\www>
```

Gambar 13. Pesan Aplikasi Yii Berhasil Dibuat

Jika sudah selesai, maka silahkan buka browser anda dan masukkan alamat berikut : <http://localhost/namaAplikasiAnda/>. Akan terlihat tampilan web seperti berikut :



Gambar 14. Tampilan Web Yii

Yah, itu artinya anda sudah berhasil membuat aplikasi pertama anda... selamat... akan sangat bagus kalo anda mencoba untuk mempelajari dan mendalami kode hasil generate dari Yii tersebut..

### **Model-View-Controller**

Sebelum melangkah lebih jauh, anda harus memahami terlebih dahulu pola desain yang dijalankan oleh Yii Framework yaitu pola desain MVC (Model-View-Controller). Penerapan konsep MVC dimaksudkan untuk memisahkan logika bisnis dan antar muka pada suatu aplikasi. Dengan MVC, pihak pengembang aplikasi dapat lebih mudah mengubah suatu bagian dalam aplikasi tanpa harus mengubah bagian yang lainnya. Dalam MVC, model menggambarkan informasi atau data beserta aturan bisnisnya (validasi, relasi, dll). View atau tampilan menggambarkan antarmuka yang menjadi penghubung antara aplikasi dan pengguna (user). Sedangkan controller akan menjadi jembatan komunikasi antara Model dan View.

## Membuat Model

Seperti telah dijelaskan sebelumnya dalam MVC, model menggambarkan informasi atau data beserta aturan bisnisnya seperti validasi, relasi, tipe data, dan lain-lain. Ada dua jenis model, yaitu model yang disimpan dan dikumpulkan ke dalam database dan model yang setelah dipakai tidak disimpan ke dalam database. Jika kita menggunakan model yang setelah dipakai tidak disimpan ke dalam database, maka kita dapat menggunakan Form Model untuk mendefinisikannya. Sedangkan untuk model yang disimpan dan dikumpulkan ke dalam database, kita dapat menggunakan Active Record. Bingung?? Biar lebih ngerti, mari kita pelajari lewat contoh :

Pada suatu kasus misal saya ingin membuat Form Login. Pada form login user akan diminta menginput username dan password. Apabila username dan password yang dimasukkan sudah benar, maka user bisa masuk ke dalam sistem. Sedangkan jika salah, maka user akan diminta kembali menginput username dan password yang salah. Lalu apakah username dan password yang diinput oleh user akan disimpan?? Tidak. Nah, untuk kasus seperti ini kita dapat menggunakan Form Model untuk mendefinisikan model dari attribute yang digunakan untuk melakukan login. Berikut contoh code untuk mendefinisikan modelnya :

```
<?php  
  
class LoginForm extends CFormModel  
{  
    public $username;  
    public $password;  
    public $rememberMe;  
  
    public function rules()  
    {  
        return array(  
            array('username, password',  
                'required', 'message'=>'{attribute} harus diisi'),
```

```

        array('rememberMe', 'boolean'),
        array('password', 'authenticate'),
    );
}

public function attributeLabels()
{
    return array(
        'username'=>'Username',
        'password'=>'Password',
        'rememberMe'=>'Ingat Saya',
    );
}
}

```

Code di atas adalah contoh sebuah model yang tidak terhubung langsung ke database. Disana kita mendefinisikan variabel public “username”, “password”, dan “rememberMe”. Selain mendefinisikan variabel pada suatu model, model juga mengizinkan kita untuk menetapkan validasi pada masing-masing variabel yang telah kita gunakan tadi. Kita juga dapat menentukan attributeLabel untuk masing-masing variabel.

Pada kasus yang berbeda, saya ingin membuat form yang menginput nama dan alamat user. Dimana nama dan alamat user tersebut akan disimpan dan dikumpulkan ke dalam database. Untuk kasus seperti ini kita dapat menggunakan Active Record untuk mendefinisikan model. Berikut contoh code untuk mendefinisikannya :

```

<?php

/**
 * This is the model class for table "mahasiswa".
 *
 * The followings are the available columns in table 'mahasiswa':
 * @property integer $id
 * @property string $nama
 * @property string $alamat
 */
class Mahasiswa extends CActiveRecord
{
    public static function model($className=__CLASS__)

```

```

    {
        return parent::model($className);
    }

    public function tableName()
    {
        return 'mahasiswa';
    }

    public function rules()
    {
        return array(
            array('nama, alamat', 'required'),
            array('nama', 'length', 'max'=>60),
            array('alamat', 'length', 'max'=>200),
            array('id, nama, alamat', 'safe', 'on'=>'search'),
        );
    }

    public function relations()
    {
        return array(
        );
    }

    public function attributeLabels()
    {
        return array(
            'id' => 'ID',
            'nama' => 'Nama',
            'alamat' => 'Alamat',
        );
    }
}

```

Saya asumsikan nama tabel yang kita gunakan adalah “mahasiswa”. Okeh, mari kita bahas satu persatu maksud dari code active record diatas.

```

<?php

/**
 * This is the model class for table "mahasiswa".
 *
 * The followings are the available columns in table 'mahasiswa':
 * @property integer $id
 * @property string $nama
 * @property string $alamat
 */
class Mahasiswa extends CActiveRecord

```

Code ini menjelaskan property/attribute apa saja yang terdapat dalam tabel mahasiswa. Nama model ini sendiri adalah “Mahasiswa” dimana model ini akan meng-extends class CActiveRecord yang sudah ada pada Yii Framework.

```
public function tableName()  
{  
    return 'mahasiswa';  
}
```

Code di atas akan menghubungkan model yang bersangkutan dengan tabel yang tertera di dalamnya, pada kasus ini tabel “mahasiswa”.

```
public function rules()  
{  
    return array(  
        array('nama, alamat', 'required'),  
        array('nama', 'length', 'max'=>60),  
        array('alamat', 'length', 'max'=>200),  
        array('id, nama, alamat', 'safe', 'on'=>'search'),  
    );  
}
```

Code di atas adalah code untuk memberikan validasi pada attribute-attribute yang ada pada tabel mahasiswa. Berikut maksud dari validasi di atas berdasarkan attributenya :

- “Nama” dan “alamat” wajib diisi
- Panjang attribute “nama” maksimal 60 character
- Panjang attribute “alamat” maksimal 200 character

Validasi di atas adalah control sebagian kecil validasi yang terdapat dalam Yii Framework. Berikut kumpulan semua validasi yang dapat dilakukan dengan class yang ada dari Yii Framework :

- Boolean : validasi attribute bernilai true dan false
- Captcha : validasi attribute harus bernilai sama dengan nilai pada captcha code
- Compare : validasi attribute harus bernilai sama dengan attribute lain
- Date : validasi attribute harus memiliki format tanggal
- Email : validasi attribute harus memiliki format email
- File : validasi attribute harus merupakan sebuah uploaded file yang valid
- In : validasi attribute memiliki range tertentu
- Length : validasi attribute harus memiliki panjang maksimal dan minimal tertentu
- Numerical : validasi attribute harus merupakan sebuah angka
- Required : validasi attribute yang tidak boleh kosong/wajib diisi
- Type : validasi attribute harus merupakan suatu tipe tertentu

```
public function relations ()
{
    return array (
    );
}
```

Code tersebut digunakan untuk mendefinisikan relasi yang ada pada suatu tabel. Tapi karena tabel ini tidak memiliki relasi.. jadi kosong isinya...

```
public function attributeLabels ()
{
    return array (
        'id' => 'ID',
        'nama' => 'Nama',
        'alamat' => 'Alamat',
    );
}
```

Code tersebut digunakan untuk mendefinisikan label yang akan mewakili attribute masing-masing. Jadi kalo anda mau merubah label pada suatu attribute, anda ga perlu merubahnya pada view (tampilan) secara langsung cukup rubah di bagian ini saja.. sekali anda rubah di bagian ini, maka di semua view label masing-masing attribute secara otomatis akan berubah.

### Membuat View (Tampilan)

View adalah antar muka aplikasi yang menjadi jembatan interaksi antara user dan aplikasi. Intinya view adalah tampilan yang akan dilihat oleh user. Membuat tampilan pada Yii Framework sebenarnya tidak ada bedanya dengan membuat form dengan menggunakan script HTML dan PHP, hanya saja pada Yii juga tersedia beberapa kelas pembantu untuk mempermudah membuat tampilan. Beberapa contoh kelas pembantu yang disediakan oleh Yii Framework misalnya :

- Chtml::link : Menampilkan link

Contoh penggunaan : `<?php echo Chtml::link('Tulisan yang ingin ditampilkan', array('controller/action')); ?>`

- Chtml::button : Menampilkan tombol button

Contoh penggunaan : `<?php echo Chtml::button('Tulisan yang ingin ditampilkan', array('submit' => array('controller/action'))); ?>`

- Chtml::dropdownlist : Menampilkan dropdownlist



Contoh penggunaan : `<?php echo CHtml::dropDownList('listname',  
$select, array('M' => 'Male', 'F' => 'Female')); ?>`

Masih banyak lagi yang lainnya.. silahkan anda gali sendiri ya.. atau langsung saja anda lihat di : <http://www.yiiframework.com/doc/api/1.1/CHtml/>

Selain bisa menggunakan kelas yang sudah disediakan Yii Framework , kita juga dapat menggunakan extension untuk memperindah tampilan. Dari Yii Framework sendiri sudah ada extension bernama “Zii” yang benar-benar bisa sangat membantu kita mengatur tampilan. Yah, saya sendiri juga selalu menggunakannya di setiap aplikasi yang saya bangun. Beberapa contoh extension “Zii” yang bisa kita gunakan antara lain CGridView, CListView, CjuiDialog, Cmenu, CdetailColumn, Cbreadcrumbs, dll.... Berikut contoh code penggunaan CGridView pada Yii Framework :

```
<?php  
$haha=array(  
    array('id'=>1, 'nama'=>'Joko'),  
    array('id'=>2, 'nama'=>'Toloi'),  
);  
  
$hasil=new CArrayDataProvider($haha, array(  
    'id'=>'id',  
    'sort'=>array(  
        'attributes'=>array(  
            'id', 'nama',  
        ),  
    ),  
));  
$this->widget('zii.widgets.grid.CGridView', array(  
    'dataProvider'=>$hasil,  
    'columns'=>array(  
        array(  
            'name'=>'Identitas',  
            'value'=>'$data["id"]',  
        ),  
        array(  
            'name'=>'Nama',
```

```

        'value'=>'$data["nama"]',
    ),
),
));
?>

```

Dan hasil dari code di atas adalah :

Displaying 1-2 of 2 result(s).

Identitas	Nama
1	Joko
2	Toloi

Gambar 15. Tampilan CGridView

Jika code di atas masih terasa sangat asing bagi anda, tenang.. kita akan membahasnya nanti.. ☺

### Membuat Controller

Controller adalah jembatan yang menghubungkan antara View dan Model. Controller terdiri dari action (aksi) yang akan melakukan request yang di dalamnya biasanya memerlukan Model dan View. Dengan kata lain, kalo anda mau membuat sebuah aksi yang bisa direquest oleh user, anda harus definisikan aksi tersebut di dalam sebuah controller. Mari kita pelajari sedikit tentang code controller :

```

<?php
class MahasiswaController extends Controller
{
    /**
     * @return array action filters
     */
    public function filters()
    {

```

```

        return array(
            'accessControl',
        );
    }

    /**
     * Specifies the access control rules.
     * This method is used by the 'accessControl' filter.
     * @return array access control rules
     */
    public function accessRules()
    {
        return array(
            array('allow',
                'actions'=>array('index','view'),
                'users'=>array('*'),
            ),
            array('allow',
                'actions'=>array('create','update'),
                'users'=>array('@'),
            ),
            array('allow',
                'actions'=>array('admin','delete'),
                'users'=>array('admin'),
            ),
            array('deny',
                'users'=>array('*'),
            ),
        );
    }

    /**
     * Displays a particular model.
     * @param integer $id the ID of the model to be displayed
     */
    public function actionView($id)
    {
        $this->render('view',array(
            'model'=>$this->loadModel($id),
        ));
    }

    /**
     * Creates a new model.
     * If creation is successful, the browser will be redirected
     to the 'view' page.
     */
    public function createAction()
    {
        $model=new Mahasiswa;

        // Uncomment the following line if AJAX validation is
        needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Mahasiswa']))
        {
            $model->attributes=$_POST['Mahasiswa'];
            if($model->save())

```

```

        $this->redirect(array('view', 'id'=>$model->id));
    }

    $this->render('create', array(
        'model'=>$model,
    ));
}
}

```

Mari kita bahas code di atas :

```

class MahasiswaController extends Controller
{

```

Ini adalah identitas dari controller, disini nama controllernya adalah

MahasiswaController dimana class ini akan meng extends class Controller.

```

/**
 * @return array action filters
 */
public function filters()
{
    return array(
        'accessControl',
    );
}

/**
 * Specifies the access control rules.
 * This method is used by the 'accessControl' filter.
 * @return array access control rules
 */
public function accessRules()
{
    return array(
        array('allow',
            'actions'=>array('index', 'view'),
            'users'=>array('*'),
        ),
        array('allow',
            'actions'=>array('create', 'update'),
            'users'=>array('@'),
        ),
        array('allow',
            'actions'=>array('admin', 'delete'),
            'users'=>array('admin'),
        ),
        array('deny',
            'users'=>array('*'),

```

```
);
```

Code di atas digunakan untuk membuat aturan pada hak akses user. Dengan menggunakan code di atas kita dapat membatasi action-action apa saja yang bisa diakses oleh seorang user. Misal kita ingin membuat action “makan” hanya bisa diakses oleh user yang sudah login, action “minum” bisa diakses oleh siapa saja yang sudah login ataupun belum, action “jalan” bisa diakses oleh user dengan tipe admin, dan sebagainya.. Pada code di atas action “index” dan “view” dapat diakses oleh user dengan tipe “\*”, itu artinya “index” dan “view” bisa diakses oleh semua orang baik yang sudah login maupun belum. Action “create” dan “update” bisa diakses dengan user dengan tipe “@”, itu artinya create dan update hanya bisa diakses oleh user yang sudah login. Jadi kalo belum login, user tidak akan bisa mengakses action tersebut. Action “admin” dan “delete” hanya bisa diakses oleh user dengan tipe “admin”, itu artinya hanya user yang login dengan nama admin lah yang bisa mengakses action tersebut.

Dalam membuat level akses user, akan sangat membantu jika kita memberikan pengkategorian pada masing-masing user. Misalnya si A, B, dan C memiliki level bendahara, D, E, dan F level sekretaris, dan lain-lain.. Tentu saja kita bisa membuat hal tersebut di dalam aplikasi kita. Jadi ada user yang punya level admin, level pimpinan, level bendahara, dan lain-lain.. Dan dengan menggunakan level itulah kita bisa membuat hak akses sesuai dengan levelnya masing-masing. Tenang, kita akan membahas ini nanti saat mempelajari pembuatan aplikasi forum..

```
/**
```

```

    * Displays a particular model.
    * @param integer $id the ID of the model to be displayed
    */
    public function actionView($id)
    {
        $this->render('view', array(
            'model' => $this->loadModel($id),
        ));
    }

```

Code di atas merupakan actionView dimana action tersebut dapat direquest oleh user untuk dijalankan.

```

    public function actionCreate()
    {
        $model=new Mahasiswa;

        if(isset($_POST['Mahasiswa']))
        {
            $model->attributes=$_POST['Mahasiswa'];
            if($model->save())
                $this->redirect(array('view','id'=>$model->id));
        }

        $this->render('create', array(
            'model' => $model,
        ));
    }

```

Sama seperti code sebelumnya, code di atas digunakan untuk melakukan request actionCreate. Bedanya dengan action sebelumnya adalah pada action ini memiliki dua buah kondisi saat mengaksesnya. Kondisi saat pertama kali user merequestnya, dan kondisi dimana user telah merequest action dengan mengirimkan nilai yang berasal dari action itu sendiri.. Bingung?? Hehe, Kalo iya ga usah khawatir, kita akan perdalam lagi masalah ini pada bahasan selanjutnya yak... 😊

## Menjelajah Database dengan Yii Framework

Yii Framework menyediakan banyak cara untuk mempermudah anda bekerja dengan database. Mulai dari mempermudah membuat koneksi ke database, proses query, pagination, hingga meningkatkan security pada pengaksesan database. Beberapa fasilitas yang disediakan untuk mempermudah melakukan operasi database pada Yii Framework antara lain Data Access Objects (DAO), Query Builder, Active Record, dan Relational Active Record. Mari kita bahas satu persatu fasilitas-fasilitas di atas.

### Data Access Object (DAO)

DAO menyediakan API generik yang dapat mengakses data dari Database yang berbeda. Yii DAO sendiri dibangun di atas PHP Data Objects (PDO) yang artinya kalo anda mau menggunakannya dalam membangun aplikasi, anda harus memastikan extension PDO untuk database tertentu sudah terinstal. Kalo anda menggunakan packet server seperti wamp atau xampp, biasanya extension PDO untuk Mysql database sudah secara otomatis terinstal. Untuk menggunakan DAO, kita harus membuat koneksi seperti berikut :

```
$connection=new CDbConnection($namaDatabase,$username,$password);  
$connection->active=true;  
...  
...  
$connection->active=false;
```

Kita harus konfigurasi database kita seperti memasukkan nama database yang dituju, username database, dan password database. Setelah itu kita harus membuka koneksi dengan mengeset active menjadi “true” dan menutup koneksi dengan mengeset active menjadi “false”. Berikut contoh penggunaannya :

```

$dsn='mysql:host=localhost;dbname=namadatabase';
$connection=new CDbConnection($dsn,'root','');
$connection->active=true;
$sql="INSERT INTO mahasiswa(nama, alamat)
VALUES('Itu','Itujuga)";
$command=$connection->createCommand($sql);
$command->execute();
$connection->active=false;

```

Kesulitan dari penerapan kode di atas adalah ketika kita membuat suatu sistem yang besar dimana terjadi ribuan operasi database di dalamnya. Sangat melelahkan ketika harus mendefinisikan koneksi terus menerus sampe ribuan kali, belum lagi ketika suatu waktu terjadi perubahan konfigurasi database misal passwordnya berubah. Nah loh?? Capek tuh kalo kita mau satu persatu ngerubah lagi koneksinya. Eiiitss, tapi tunggu dulu.. ada cara yang lebih mudah untuk membuat koneksinya dimana kita cukup buat satu koneksi dan itu berlaku untuk semuanya. Sebelumnya kita telah membahas cara untuk menghubungkan database dengan aplikasi melauai file yang terletak di “protected/config/main.php”. Karena kita sudah melakukan pengaturan disana, maka kita dapat membuat koneksi database yang lebih simpel dari cara di atas. Berikut contoh penggunaannya :

```

$connection=Yii::app()->db;
$sql="INSERT INTO mahasiswa(nama, alamat)
VALUES('Itu','Itujuga)";
$command=$connection->createCommand($sql);
$command->execute();

```

Lebih simpel bukan?? Kalo pun terjadi perubahan pada konfigurasi database, kita tinggal merubah konfigurasi tersebut pada file config aplikasi kita tadi. Pokoknya lebih ajib dah.. ☺



Untuk menjalankan perintah-perintah SQL, ada beberapa perintah yang dapat kita gunakan dengan menggunakan DAO. Berikut perintah-perintahnya :

- `$rowCount=$command->execute();` : Menjalankan SQL Non Query, seperti contoh di atas
- `$dataReader=$command->query();` : Menjalankan query SQL
- `$rows=$command->queryAll();` : Menjalankan query SQL dan mengembalikan seluruh baris hasil
- `$row=$command->queryRow();` : Menjalankan query SQL dan mengembalikan baris pertama hasil
- `$column=$command->queryColumn();` : Menjalankan query SQL dan mengembalikan kolom pertama hasil
- `$value=$command->queryScalar();` : Menjalankan query SQL dan mengembalikan field pertama dalam baris pertama

Anda juga dapat melihat dokumentasi materi ini langsung dari website Yii di <http://www.yiiframework.com/doc/guide/1.1/id/database.dao>. Saya juga ngambil materinya dari sana kok.. ☺

## Query Builder

Sama seperti DAO, Query Builder juga dibangun di atas DAO. Perbedaan Query Builder dan DAO sendiri adalah cara mendefinisikan query yang berbeda. Jika pada DAO kita mendefinisikan suatu sintaks SQL secara langsung dalam suatu baris query, maka pada Query Builder kita mendefinisikan sintaks SQL secara prosedural menggunakan property dan method yang tersedia dari Yii Framework. Berikut contoh code untuk menggunakan Query Builder :

```
$user = Yii::app()->db->createCommand()  
->select('id, username, profile')  
->from('tbl_user u')  
->join('tbl_profile p', 'u.id=p.user_id')
```

```
->where('id=:id', array(':id'=>$id))
->queryRow()
```

Dan kalo memang masih rada bingung dengan penjelasan saya di atas, lagi-lagi materi di atas juga saya ambil dari Website Yii Framework disini : <http://www.yiiframework.com/doc/guide/1.1/id/database.query-builder>. Jadi anda dapat menggalinya lebih dalam lagi disana.. ☺

### **Active Record (AR)**

Active Record (AR) adalah teknik populer Object-Relational Mapping (ORM). Setiap kelas AR akan mewakili suatu tabel pada database dan propertynya mewakili attribute dari tabel tersebut. Ingat kan materi tentang “Model”?? pada materi tersebut saya membahas tentang cara mendefinisikan model yang disimpan di database dengan menggunakan AR. Dan betul sekali, AR yang sedang kita bahas sekarang adalah AR tersebut. Yah, AR ini adalah model yang kita definisikan agar suatu class dapat terkoneksi ke database dan dapat membuat aturan bisnis sesuai kebutuhan database tersebut. Jadi kalo mau menggunakan AR, kita harus mendefinisikan suatu model menggunakan AR. AR sangat membantu untuk mengurangi waktu kita dalam penulisan sintaks-sintaks SQL. Berikut contoh penggunaan AR pada Yii Framework :

```
$itu=new Mahasiswa;
$itu->nama="Sabit Huraira";
$itu->alamat="Dimana Aja";
$itu->save();
```

Code di atas adalah code untuk melakukan input data baru ke tabel “Mahasiswa”. Dimana dari data tersebut saya menginput attribut nama dan alamat, kemudian menyimpannya ke database.

Yang paling menarik dari AR adalah proses untuk mengambil data yang menjadi sangat mudah. Berikut saya contohkan beberapa operasi melihat data dengan menggunakan AR :

- `$hasil=Mahasiswa::model()->find(); => //mengembalikan baris pertama data pada tabel mahasiswa sesuai dengan kondisi yang ditetapkan`
- `$hasil=Mahasiswa::model()->findByPk(); => //mengembalikan baris pertama data pada tabel mahasiswa yang memiliki Primary Key sesuai dengan kondisi yang ditetapkan`
- `$hasil=Mahasiswa::model()->findByAttributes(); => //mengembalikan baris data pertama pada tabel mahasiswa yang memiliki attribute dengan kondisi yang telah ditetapkan`
- `$hasil=Mahasiswa::model()->findBySql(); => //mengembalikan baris pertama data pada tabel mahasiswa dengan SQL yang telah ditetapkan`

Mungkin anda tidak begitu puas dengan operasi-operasi di atas karena hanya akan mengembalikan data pada baris pertama saja. Tenang, ada operasi lain yang dapat mengembalikan nilai semua row data yang kita butuhkan. Berikut contoh pengoperasiannya :

- `$hasil=Mahasiswa::model()->findAll(); => //mengembalikan semua data pada tabel mahasiswa sesuai dengan kondisi yang ditetapkan`
- `$hasil=Mahasiswa::model()->findAllByPk(); => //mengembalikan semua data pada tabel mahasiswa yang memiliki Primary Key sesuai dengan kondisi yang ditetapkan`
- `$hasil=Mahasiswa::model()->findAllByAttributes(); => //mengembalikan semua data pada tabel mahasiswa yang memiliki attribute dengan kondisi yang telah ditetapkan`

- `$hasil= Mahasiswa::model()->findAllBySql(); =>`  
//mengembalikan baris pertama data pada tabel mahasiswa dengan SQL yang telah ditetapkan

Dan ada juga operasi yang disediakan untuk mengetahui informasi lain dari database, seperti :

- `$jumlah= Mahasiswa::model()->count();` //mengembalikan nilai jumlah baris pada tabel mahasiswa sesuai dengan kondisi yang ditetapkan
- `$jumlah= Mahasiswa::model()->countBySql();`//mengembalikan nilai jumlah baris pada tabel mahasiswa sesuai dengan pernyataan SQL yang ditetapkan
- `$exists= Mahasiswa::model()->exists();` // mengetahui apakah ada satu baris pada tabel mahasiswa yang sesuai dengan kondisi yang ditetapkan

Tidak hanya untuk menampilkan informasi, tersedia juga operasi untuk melakukan operasi update dan delete :

- `Mahasiswa::model()->updateAll(); =>` // memperbaharui semua data yang sama seperti kondisi yang ditetapkan
- `Mahasiswa::model()->updateByPk(); =>` // memperbaharui baris yang sama seperti kondisi dan primary key yang ditetapkan
- `Mahasiswa::model()->deleteAll(); =>` // menghapus baris yang sesuai dengan kondisi yang ditetapkan
- `Mahasiswa::model()->deleteByPk(); =>` // menghapus baris yang sesuai dengan kondisi dan primary key yang ditetapkan

Selain operasi-operasi di atas, AR juga mengizinkan kita untuk membuat operasi sendiri yang bisa kita buat sesuai kebutuhan kita. Hal ini dimaksudkan agar memudahkan kita ketika kita membutuhkan operasi khusus yang kita pakai berulang-ulang. Untuk melakukannya, kita bisa membuat secara langsung code nya pada model yang kita mau. Ada dua cara melakukannya, menggunakan scopes atau membuat fungsi sendiri. Berikut contoh penggunaan scopes :

```

public function scopes ()
{
    return array(
        'panggilJoko'=>array(
            'condition'=>'nama=Joko',
            'limit'=>5,
        ),
    );
}

//Misal code di atas kita masukkan pada Model Mahasiswa,
maka berikut cara memanggilnya
$itu=Mahasiswa::model()->panggilJoko()->findAll();

```

Anda juga tentu dapat membuat fungsi sendiri jika anda mau, misal seperti berikut:

```

public function panggilSiapa($nama=Joko)
{
    //baris untuk mengambil data
    return $data;
}

//Dipanggil dengan cara berikut :
$itu=Mahasiswa::model()->panggilSiapa('Ali');

```

Sangat membantu bukan?? Dengan menggunakan cara-cara tersebut, anda akan menghemat waktu anda untuk memikirkan SQL query yang memusingkan dan terjadi berulang kali..

### **Active Record Relational**

Sama dengan AR, hanya saja Active Record Relational dibuat untuk memudahkan pengelolaan pada tabel-tabel yang memiliki relasi dengan tabel lain. Untuk menggunakan AR Relational sendiri, kita harus memastikan bahwa relational datanya telah didefinisikan pada Model AR. Sebagai contoh seperti berikut :

```

public function relations()
{
    return array(
        'koneksi' => array(self::BELONGS_TO, 'NamaModel',
            'namaAttribute'),
    );
}

```

Pada kode tersebut kita menghubungkan model dengan model lain yang bernama “NamaModel”, dan relasi kedua model tersebut kita beri nama dengan “koneksi”. Contoh penggunaan relasi tersebut seperti berikut :

```
NamaModel::model()->findByPk('1')->koneksi->namaAttribute
```

Sebagai contoh nyata kasus penggunaan AR Relation misal kita punya 2 buah tabel, tabel siswa dan tabel kelas. Dimana masing-masing siswa memiliki kelas yang sesuai dengan pada tabel kelas. Maka dari itu pada tabel mahasiswa, kita harus definisikan relasinya seperti berikut :

```

public function relations()
{
    return array(
        'kelas' => array(self::BELONGS_TO, 'Kelas',
            'kelas_id'),
    );
}

```

Nama relasi yang terdapat dalam aturan di atas adalah “kelas1”. Dimana relasi tersebut akan terhubung dengan model “Kelas” dan yang akan terhubung dengan Primary Key model Kelas. Sedangkan attribute dari model Siswa yang terhubung dengan model kelas adalah “kelas\_id”. Cara Penggunaan untuk mengakses relasinya sendiri sama seperti yang ada pada contoh sebelumnya..

## **Extension**

Untuk mempermudah kita dalam membangun suatu aplikasi, sering kali kita membutuhkan bantuan library yang dapat kita pakai secara langsung. Dan Yii pastinya sudah menyediakan hal tersebut untuk mempermudah para penggunanya. Di dalam Yii juga dikenal sesuatu bernama “Extension”. Extension sendiri adalah library yang sudah dikonfigurasi sedemikian rupa agar dapat mempermudah library tersebut digunakan pada Yii Framework. Hmm untuk penjelasan tata cara penggunaan extension sendiri akan kita bahas pada saat tahap membuat aplikasi forum yak.. 😊

# **PART III**

## **YII IN ACTION**



Ya ya ya.. mungkin anda sudah bosan dengan teori-teori dan udah nggak sabar untuk praktek langsung penerapan aplikasi dengan menggunakan Yii Framework. Okeh okeh.. tanpa banyak ba..bi..bu.. mari kita mulai saja menerapkan penggunaan suatu aplikasi pada Yii Framework...

### **Membuat Database**

Biar lebih mudah melakukan pembelajaran pada ebook ini, mari kita seragamkan skema database yang akan kita gunakan pada aplikasi kita. Saya sendiri menggunakan database MySQL, tetapi tidak masalah jika anda ingin menggunakan database lain. Sekarang silahkan buat sebuah database dan buat sebuah tabel yang bernama “mahasiswa”. Pada tabel mahasiswa, masukkan attribute seperti berikut :

<b>Nama Attribut</b>	<b>Id</b>	<b>Nama</b>	<b>Alamat</b>
<b>Type</b>	Int	Varchar(60)	Varchar(200)
<b>Null</b>	No	No	No
<b>Index</b>	Primary Key	-	-
<b>Auto Increment</b>	Yes	-	-

Okeh, selesai membuat databasenya..

### **Menghubungkan Database dengan Aplikasi Yii Framework**

Pada bahasan sebelumnya kita telah berhasil membuat aplikasi pertama kita dengan Yii Framework. Sekarang mari kita hubungkan aplikasi yang telah kita buat tersebut dengan database kita. Pada folder aplikasi tadi, buka file “protected/config/main.php”. Lihat pada bagian code berikut :

```

        'db'=>array(
            'connectionString' =>
'sqlite:'.dirname(__FILE__).'/../data/testdrive.db',
        ),
        // uncomment the following to use a MySQL database
        /*
        'db'=>array(
            'connectionString' =>
'mysql:host=localhost;dbname=testdrive',
            'emulatePrepare' => true,
            'username' => 'root',
            'password' => '',
            'charset' => 'utf8',
        ),
        */

```

Pada code di atas dapat kita lihat bahwa aplikasi terhubung ke database “sqlite” dan terdapat code yang menghubungkan aplikasi ke “MySQL” database yang diberikan tag “comment” yang artinya code tersebut tidak akan di proses oleh aplikasi saat dijalankan. Sekarang kita akan membuat aplikasi kita dapat terhubung ke database MySQL dan database yang telah kita buat tadi. Rubah source code di atas menjadi seperti berikut :

```

        /*
        'db'=>array(
            'connectionString' =>
'sqlite:'.dirname(__FILE__).'/../data/testdrive.db',
        ),
        */
        'db'=>array(
            'connectionString' =>
'mysql:host=localhost;dbname=namaDatabase',
            'emulatePrepare' => true,
            'username' => 'root',
            'password' => '',
            'charset' => 'utf8',
        ),

```

Pada code di atas saya menjadikan akses ke database sqlite menjadi komentar yang artinya code tersebut tidak akan di proses oleh aplikasi. Sedangkan code untuk akses ke database mysql saya hilangkan tag komentarnya yang artinya code tersebut akan diproses oleh aplikasi. Selain menghilangkan tag komentar, saya

juga melakukan konfigurasi pada nama database untuk mengarahkan aplikasi pada suatu database. Silahkan anda lakukan konfigurasi pada database anda masing-masing dan jangan lupa lakukan konfigurasi juga pada beberapa hal seperti username dan password database anda. Okeh, proses menyatukan aplikasi dengan database selesai...

### **Make It Easy With Gii**

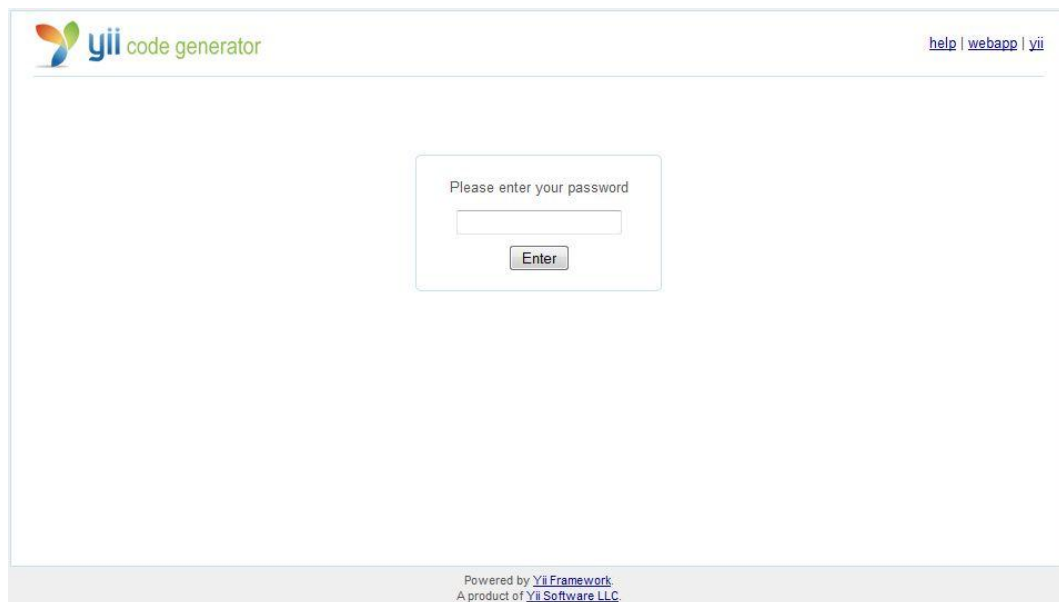
Salah satu fitur yang membuat Yii menjadi lebih menarik untuk dipilih adalah adanya code generate. Code generate? Yah, anda ga perlu pusing mikirin dan capek ngetik code untuk membuat operasi CRUD (Create, Read, Update, Delete) pada suatu tabel di database. Tanpa coding pun anda bisa membuatnya langsung terbentuk secara otomatis hingga pembentukan tampilannya. Itu semua dapat kita lakukan dengan menggunakan “Gii” . Sebelum menikmati fitur ini, tentunya ada beberapa hal yang harus kita konfigurasi terlebih dahulu. Buka kembali file pada “protected/main.config.php”, lalu lihat code berikut :

```
'modules'=>array(  
    // uncomment the following to enable the Gii tool  
    /*  
    'gii'=>array(  
        'class'=>'system.gii.GiiModule',  
        'password'=>'Enter Your Password Here',  
        // If removed, Gii defaults to localhost only. Edit  
        carefully to taste.  
        'ipFilters'=>array('127.0.0.1','::1'),  
    ),  
    */  
)
```

Hilangkan tag komentar yang ada pada code tersebut hingga menjadi seperti berikut :

```
'modules'=>array(  
    'gii'=>array(  
        'class'=>'system.gii.GiiModule',  
        'password'=>'Enter Your Password Here',  
        // If removed, Gii defaults to localhost only. Edit  
        carefully to taste.  
        'ipFilters'=>array('127.0.0.1',':::1'),  
    ),  
)
```

Jangan lupa konfigurasi password anda sesuai dengan keinginan anda. Jika sudah selesai, cobalah masukkan alamat berikut pada browser anda “localhost/namaAplikasiAnda/index.php?r=gii”, akan muncul tampilan seperti berikut :

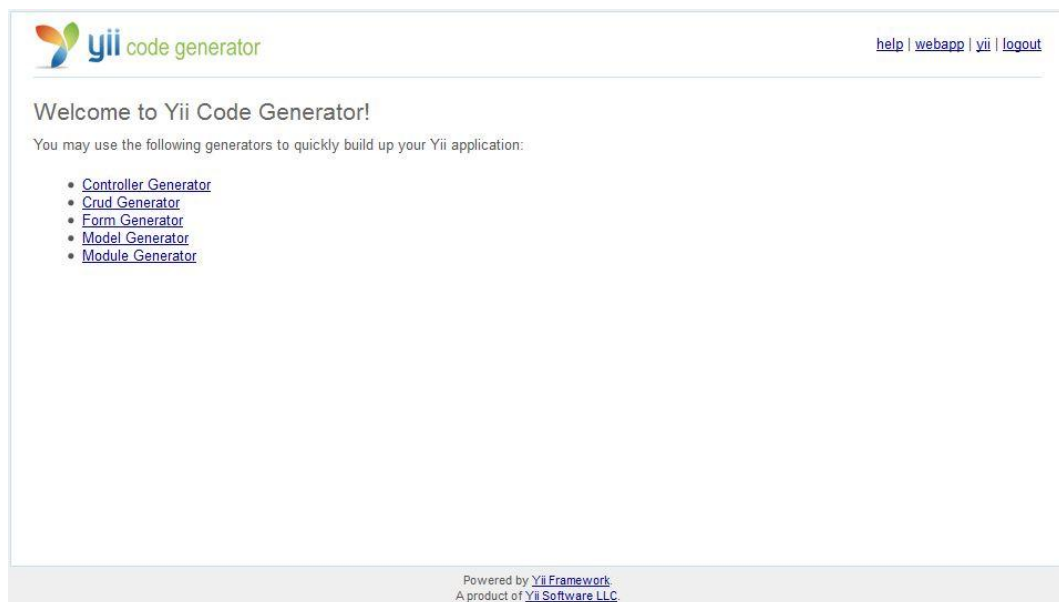


Gambar 16. Tampilan Halaman Gii

Jika sudah sesuai dengan hasil, itu berarti anda sudah bisa menggunakan Gii untuk meng-generate code anda.

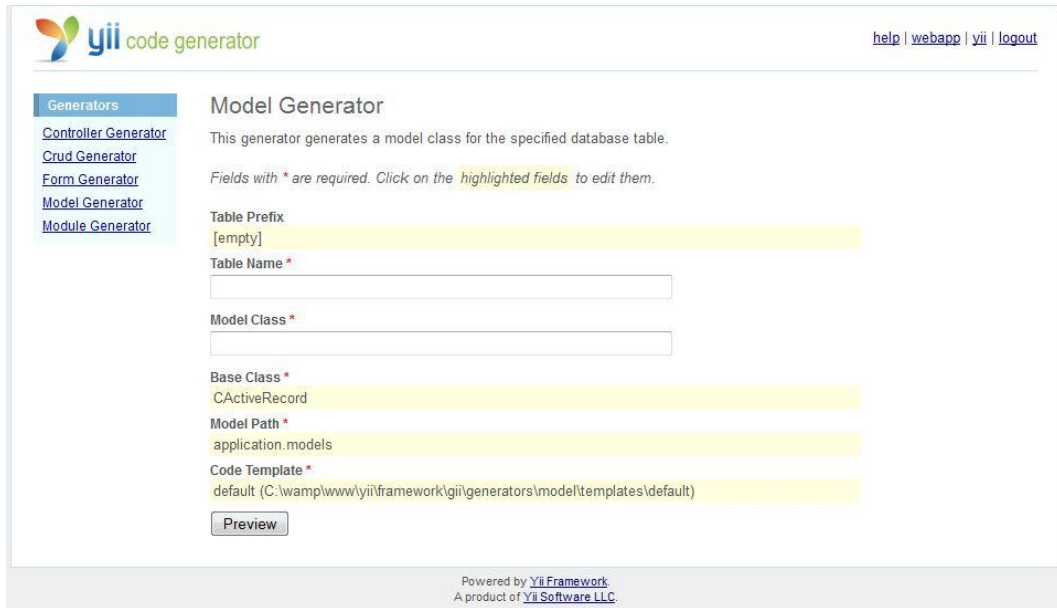
## Membuat Operasi CRUD dengan Gii

Sebelumnya kita telah menghubungkan aplikasi dengan database kita dan mengaktifkan fitur Gii pada aplikasi kita. Sekarang kita bisa membuat code generate yang dapat menjalankan operasi CRUD suatu tabel pada database. Sekarang buka tampilan Gii pada aplikasi anda, lalu masukkan password anda sesuai dengan yang telah anda konfigurasi pada code gii sebelumnya. Akan tampil halaman seperti berikut :



Gambar 17. Tampilan Halaman Utama Gii

Untuk membuat CRUD generator, hal pertama yang harus kita generate adalah model dari suatu tabel. Untuk itu, silahkan memilih menu “Model Generator”. Anda akan melihat tampilan seperti berikut :



Gambar 18. Tampilan Halaman Model Generator Pada Gii

Pada inputan Table Name, masukkan nama table yang ingin anda generate modelnya, kita gunakan tabel “Mahasiswa” yang sudah kita buat tadi. Lalu pilih preview dan kemudian pilih Generate. Jika telah berhasil akan terbentuk sebuah file pada direktori “protected/models/Mahasiswa.php” dalam aplikasi anda. Lalu untuk melanjutkan agar dapat membuat operasi CRUD, pada menu Gii pilih “Crud Generator”. Akan tampil halaman seperti berikut :

The screenshot shows the YII Code Generator interface. The main heading is "Crud Generator". Below the heading, there is a description: "This generator generates a controller and views that implement CRUD operations for the specified data model." A note states: "Fields with \* are required. Click on the highlighted fields to edit them." The form contains the following fields: "Model Class \*" (empty), "Controller ID \*" (empty), "Base Controller Class \*" (highlighted in yellow, containing "Controller"), and "Code Template \*" (highlighted in yellow, containing "default (C:\wamp\www\yii\framework\yii\generators\crud\templates\default)"). A "Preview" button is located below the form. The footer of the page reads: "Powered by Yii Framework A product of Yii Software LLC".

Gambar 19. Tampilan Halaman CRUD Generator Pada Gii

Pada Model Class, masukkan nama model anda. Perlu saya ingatkan agar anda memperhatikan huruf besar dan kecil pada penamaan Model Class. Sesuaikan input pada “Model Class” dengan nama file yang terbentuk pada direktori “protected/models” tadi. Lalu pilih preview, dan pilih generate. Jika telah selesai, silahkan mengakses alamat berikut pada browser anda : <http://localhost/namaAplikasiAnda/index.php?r=mahasiswa> . Anda akan melihat sebuah halaman seperti berikut :



Gambar 20. Tampilan Halaman Tabel Mahasiswa

Agar dapat melihat semua hasil code generate Gii, saya sarankan anda login terlebih dahulu sebagai admin. Setelah itu, silahkan anda coba semua fitur yang telah di generate secara otomatis dari gii. Berikut Beberapa fitur yang disajikan dari hasil code generate Gii :

**Create** : <http://localhost/namaAplikasiAnda/index.php?r=mahasiswa/create>



Gambar 21. Halaman Untuk Membuat Data Baru



Melihat Mahasiswa Yang Memiliki ID tertentu

:<http://localhost/namaAplikasiAnda/index.php?r=mahasiswa/view&id=1>



Gambar 22. Halaman Untuk Melihat Data Dengan ID Tertentu

Memperbaharui/update data mahasiswa dengan ID tertentu :

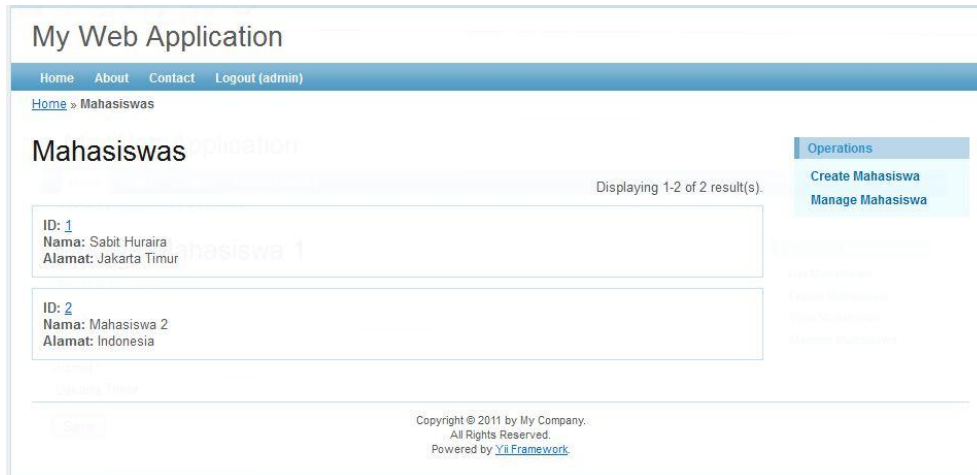
<http://localhost/namaAplikasiAnda/index.php?r=mahasiswa/update&id=1>



Gambar 23. Halaman Untuk Memperbaharui Data Dengan ID Tertentu

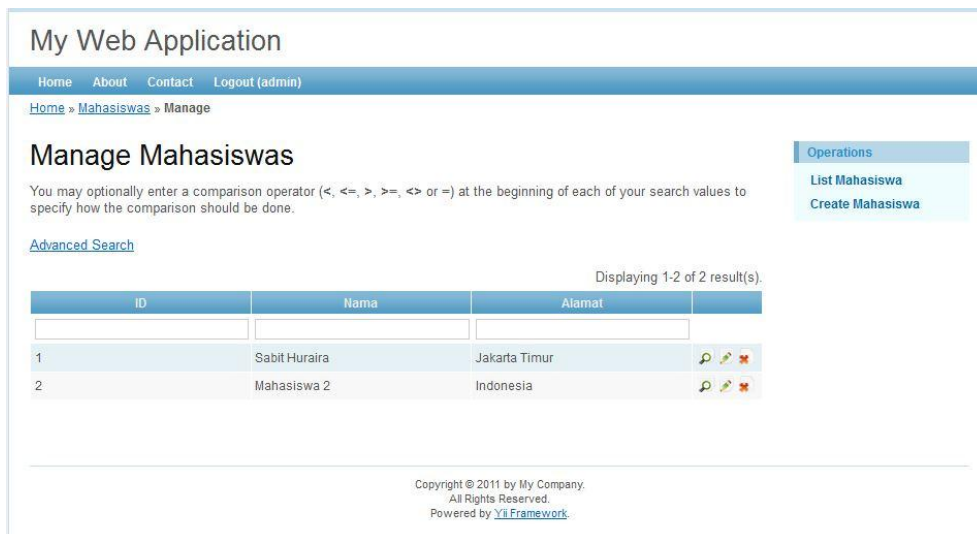
Melihat Semua Data Mahasiswa yang ada di database :

<http://localhost/namaAplikasiAnda/index.php?r=mahasiswa/index>



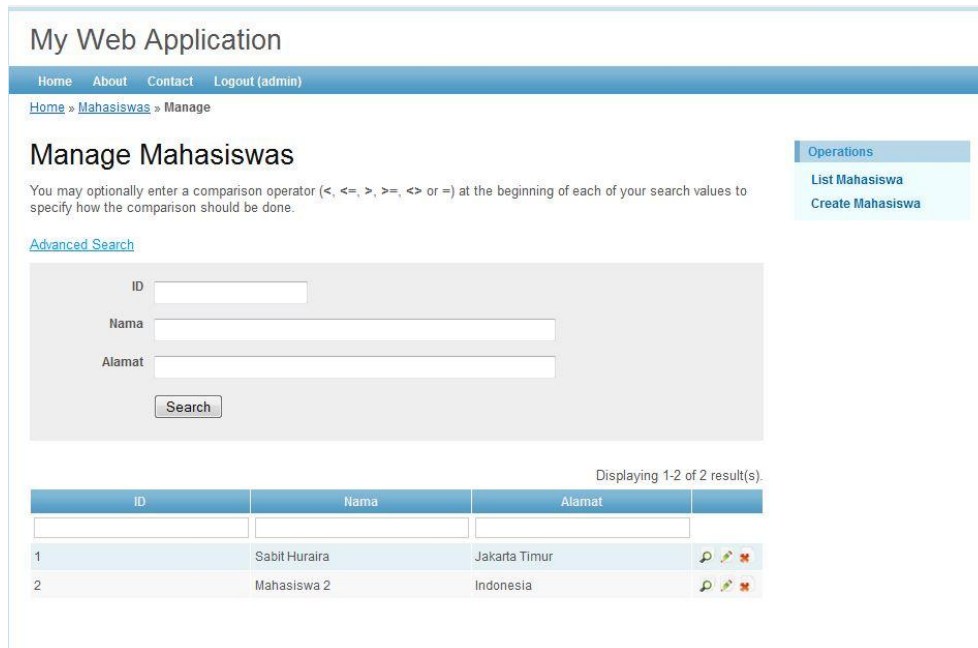
Gambar 24. Halaman Untuk Melihat Data Semua Mahasiswa

Admin : <http://localhost/namaAplikasiAnda/index.php?r=mahasiswa/admin>



Gambar 25. Halaman Admin

Coba anda klik link “Advanced Search” pada menu di atas. Maka akan terlihat tampilan seperti berikut :



Gambar 26. Halaman Admin Dengan Advanced Search

Menarik bukan melihat halaman admin?? Yii bahkan telah membuatkan kita fitur pencarian yang amat memuaskan dengan generate code nya. Tidak hanya itu jika anda sudah menginput banyak data pada tabel mahasiswa, maka data akan ditampilkan dengan pagination. Maksud dari pagination adalah data kita yang banyak tadi tidak ditampilkan sekaligus semuanya, tetapi ditampilkan halaman per halaman. Misal kita punya 200 data, yang ditampilkan pertama kali adalah data ke 1-20. Jika kita memilih halaman ke 2, maka yang tampil data ke 21-40 dan seterusnya... Woww... ini benar-benar akan meringankan pekerjaan anda bukan??

### How About Gii Code??

Ga mungkin kan setelah kita melakukan generate code lantas kita terima aja hasilnya tanpa kita pelajari?? Kecuali kalo anda emang mau mengalami

kesulitan saat harus memodifikasi aplikasi anda. Untuk itu mari kita bahas satu persatu code-code hasil generate Gii.

Yang pertama kali kita generate menggunakan Gii adalah bagian model. Maka pertama-tama mari kita bahas code pada model. Silahkan buka file pada “protected/models/Mahasiswa.php”, akan terlihat code sebagai berikut :

```
<?php

/**
 * This is the model class for table "mahasiswa".
 *
 * The followings are the available columns in table 'mahasiswa':
 * @property integer $id
 * @property string $nama
 * @property string $alamat
 */
class Mahasiswa extends CActiveRecord
{
    /**
     * Returns the static model of the specified AR class.
     * @return Mahasiswa the static model class
     */
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'mahasiswa';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes
        that
        // will receive user inputs.
        return array(
            array('nama, alamat', 'required'),
            array('nama', 'length', 'max'=>60),
            array('alamat', 'length', 'max'=>200),
            // The following rule is used by search().
            // Please remove those attributes that should not be
            searched.
            array('id, nama, alamat', 'safe', 'on'=>'search'),
        );
    }
}
```

```

/**
 * @return array relational rules.
 */
public function relations()
{
    // NOTE: you may need to adjust the relation name and the
    related
    // class name for the relations automatically generated
    below.
    return array(
    );
}

/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
    return array(
        'id' => 'ID',
        'nama' => 'Nama',
        'alamat' => 'Alamat',
    );
}

/**
 * Retrieves a list of models based on the current
    search/filter conditions.
 * @return CActiveDataProvider the data provider that can
    return the models based on the search/filter conditions.
 */
public function search()
{
    // Warning: Please modify the following code to remove
    attributes that
    // should not be searched.

    $criteria=new CDbCriteria;

    $criteria->compare('id',$this->id);
    $criteria->compare('nama',$this->nama,true);
    $criteria->compare('alamat',$this->alamat,true);

    return new CActiveDataProvider(get_class($this), array(
        'criteria'=>$criteria,
    ));
}
}

```

Beberapa bagian code di atas sudah pernah kita bahas di bagian pembahasan tentang AR Model (silahkan buka kembali halaman sebelumnya kalo

anda sudah lupa ☺), jadi saya akan bahas hal yang belum pernah dibahas aja yak.

Okeh, pada code ini :

```
public function search()
{
    // Warning: Please modify the following code to remove
    // attributes that
    // should not be searched.

    $criteria=new CDbCriteria;

    $criteria->compare('id',$this->id);
    $criteria->compare('nama',$this->nama,true);
    $criteria->compare('alamat',$this->alamat,true);

    return new CActiveDataProvider(get_class($this), array(
        'criteria'=>$criteria,
    ));
}
```

Bagian ini dimaksudkan sebagai query yang memfilter data suatu tabel dengan kondisi-kondisi tertentu sesuai dengan inputan dari pengguna. Jadi dengan fungsi ini, kita dapat menciptakan sebuah pencarian yang dinamis sesuai dengan kehendak user.

Setelah membahas model mari kita membahas controller dan view. Agar lebih mudah dan terkelompok, kita akan membahas masing-masing action pada controller beserta view yang diaksesnya. Untuk melihat controller hasil generate Gii, silahkan buka “protected/controllers/MahasiswaController.php”. Akan terlihat kumpulan baris code seperti berikut :

```
<?php

class MahasiswaController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to
     * '//layouts/column2', meaning
     * using two-column layout. See
     * 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
     * @return array action filters
     */
}
```

```

        */
        public function filters()
        {
            return array(
                'accessControl', // perform access control for CRUD
operations
            );
        }

        /**
         * Specifies the access control rules.
         * This method is used by the 'accessControl' filter.
         * @return array access control rules
         */
        public function accessRules()
        {
            return array(
                array('allow', // allow all users to perform 'index'
and 'view' actions
                    'actions'=>array('index','view'),
                    'users'=>array('*'),
                ),
                array('allow', // allow authenticated user to perform
'create' and 'update' actions
                    'actions'=>array('create','update'),
                    'users'=>array('@'),
                ),
                array('allow', // allow admin user to perform 'admin'
and 'delete' actions
                    'actions'=>array('admin','delete'),
                    'users'=>array('admin'),
                ),
                array('deny', // deny all users
                    'users'=>array('*'),
                ),
            );
        }

        /**
         * Displays a particular model.
         * @param integer $id the ID of the model to be displayed
         */
        public function actionView($id)
        {
            $this->render('view',array(
                'model'=>$this->loadModel($id),
            ));
        }

        /**
         * Creates a new model.
         * If creation is successful, the browser will be redirected
to the 'view' page.
         */
        public function createAction()
        {
            $model=new Mahasiswa;

```

```

        // Uncomment the following line if AJAX validation is
        needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Mahasiswa']))
        {
            $model->attributes=$_POST['Mahasiswa'];
            if($model->save())
                $this->redirect(array('view','id'=>$model->id));
        }

        $this->render('create',array(
            'model'=>$model,
        ));
    }

    /**
     * Updates a particular model.
     * If update is successful, the browser will be redirected to
    the 'view' page.
     * @param integer $id the ID of the model to be updated
     */
    public function actionUpdate($id)
    {
        $model=$this->loadModel($id);

        // Uncomment the following line if AJAX validation is
        needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Mahasiswa']))
        {
            $model->attributes=$_POST['Mahasiswa'];
            if($model->save())
                $this->redirect(array('view','id'=>$model->id));
        }

        $this->render('update',array(
            'model'=>$model,
        ));
    }

    /**
     * Deletes a particular model.
     * If deletion is successful, the browser will be redirected
    to the 'admin' page.
     * @param integer $id the ID of the model to be deleted
     */
    public function actionDelete($id)
    {
        if(Yii::app()->request->isPostRequest)
        {
            // we only allow deletion via POST request
            $this->loadModel($id)->delete();

            // if AJAX request (triggered by deletion via admin
            grid view), we should not redirect the browser
            if(!isset($_GET['ajax']))

```



```

        $this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
    }
    else
        throw new CHttpException(400, 'Invalid request. Please
do not repeat this request again.');
```

```

    }

    /**
     * Lists all models.
     */
    public function actionIndex()
    {
        $dataProvider=new CActiveDataProvider('Mahasiswa');
        $this->render('index',array(
            'dataProvider'=>$dataProvider,
        ));
    }

    /**
     * Manages all models.
     */
    public function actionAdmin()
    {
        $model=new Mahasiswa('search');
        $model->unsetAttributes(); // clear any default values
        if(isset($_GET['Mahasiswa']))
            $model->attributes=$_GET['Mahasiswa'];

        $this->render('admin',array(
            'model'=>$model,
        ));
    }

    /**
     * Returns the data model based on the primary key given in
the GET variable.
     * If the data model is not found, an HTTP exception will be
raised.
     * @param integer the ID of the model to be loaded
     */
    public function loadModel($id)
    {
        $model=Mahasiswa::model()->findByPk((int)$id);
        if($model===null)
            throw new CHttpException(404, 'The requested page does
not exist.');
```

```

        return $model;
    }

    /**
     * Performs the AJAX validation.
     * @param CModel the model to be validated
     */
    protected function performAjaxValidation($model)
    {
        if(isset($_POST['ajax']) && $_POST['ajax']==='mahasiswa-
form')
        {

```

```

        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}
}

```

Sama seperti model, kita hanya akan membahas bagian-bagian yang belum kita bahas saja yak.. Jadi kalo anda sudah lupa, silahkan buka kembali halaman sebelumnya yang membahas tentang controller.. Pada bagian berikut :

```

public $layout='//layouts/column2';

```

Baris ini digunakan untuk mengatur layout dari tampilan halaman web. Maksudnya halaman web yang kita maksud akan tampilan layoutnya akan mengacu pada halaman “protected/views/layouts/column2.php”. Jadi kalo anda ingin memodifikasi layout halaman anda, anda dapat melakukannya pada bagian controller secara langsung, atau memodifikasi file “protected/view/layout/column2.php”.

## Action View

```

public function actionView($id)
{
    $this->render('view',array(
        'model'=>$this->loadModel($id),
    ));
}

```

Bagian ini sudah saya jelaskan sedikit pada pembahasan controller, tapi kita akan membahasnya lebih mendalam. Action ini digunakan oleh user ketika ingin melihat suatu data secara detail. User dapat mengakses action ini pada alamat : <http://localhost/namaAplikasiAnda/index.php?r=mahasiswa/view&id=1>. Tampilan halaman ini akan mengakses halaman “view.php” seperti yang terlihat

pada `$this->render('view')`. Halaman `view.php` yang diakses sendiri terdapat pada direktori `“protected/views/mahasiswa/view.php”`. Selain mengakses halaman `“view”`, action ini juga akan mengirimkan nilai yang bernama `“model”` yang isinya adalah `$this->loadModel($id)`. `loadModel($id)` sendiri dapat anda lihat pada controller tersebut yang isinya :

```
public function loadModel($id)
{
    $model=Mahasiswa::model()->findByPk((int)$id);
    if($model===null)
        throw new CHttpException(404,'The requested page does
not exist.');
```

Fungsi `loadMoel` sendiri akan mengembalikan baris data pada tabel mahasiswa yang memiliki Primary Key tertentu sesuai dengan request user. Jadi pada `actionView`, action akan mengirimkan data dengan primary key tertentu kepada halaman `“view”`. Dan data yang telah dikirimkan tadi akan ditampilkan oleh halaman `“view”`. Baris code pada halaman `“view”` sendiri adalah seperti berikut :

```
<?php
$this->breadcrumbs=array(
    'Mahasiswas'=>array('index'),
    $model->id,
);

$this->menu=array(
    array('label'=>'List Mahasiswa', 'url'=>array('index')),
    array('label'=>'Create Mahasiswa', 'url'=>array('create')),
    array('label'=>'Update Mahasiswa', 'url'=>array('update',
'id'=>$model->id)),
    array('label'=>'Delete Mahasiswa', 'url'=>'#',
'linkOptions'=>array('submit'=>array('delete','id'=>$model-
>id),'confirm'=>'Are you sure you want to delete this item?')),
    array('label'=>'Manage Mahasiswa', 'url'=>array('admin')),
);
?>

<h1>View Mahasiswa #<?php echo $model->id; ?></h1>

<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'id',
```

```

        'nama',
        'alamat',
    ),
)); ?>

```

- `$this->breadcrumbs` : tampilan breadcrumbs pada halaman web, breadcrumbs adalah sejenis tautan yang menunjukkan jejak kita pada suatu aplikasi.
- `$this->menu` : tampilan menu pada halaman web
- `$this->widget("Zii.widget.CDetailView")` : menampilkan list data yang akan ditampilkan pada halaman. CdetailView adalah extension dari Yii Framework yang digunakan untuk mempermudah developer menampilkan suatu data.

## Action Create

```

public function actionCreate()
{
    $model=new Mahasiswa;

    // Uncomment the following line if AJAX validation is
    needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['Mahasiswa']))
    {
        $model->attributes=$_POST['Mahasiswa'];
        if($model->save())
            $this->redirect(array('view','id'=>$model->id));
    }

    $this->render('create',array(
        'model'=>$model,
    ));
}

```

Kode diatas memiliki dua buah kondisi, kondisi biasa dan kondisi ketika user merequest POST kondisi. Kondisi biasa adalah kondisi dimana user mengakses tautan actionCreate secara langsung. Action akan memanggil model

Mahasiswa dan mengirimkannya ke halaman “create” yang dapat anda lihat pada “protected/views/mahasiswa/create.php”.

```
<?php
$this->breadcrumbs=array(
    'Mahasiswas'=>array('index'),
    'Create',
);

$this->menu=array(
    array('label'=>'List Mahasiswa', 'url'=>array('index')),
    array('label'=>'Manage Mahasiswa', 'url'=>array('admin')),
);
?>

<h1>Create Mahasiswa</h1>

<?php echo $this->renderPartial('_form', array('model'=>$model));
?>
```

Hampir mirip dengan tampilan view sebelumnya, tapi di dalam file ini terdapat kode

```
<?php echo $this->renderPartial('_form', array('model'=>$model));
?>
```

Ini berarti halaman “create” akan memanggil halaman “\_form” yang terdapat pada “protected/views/mahasiswa/\_form.php”.

```
<div class="form">

<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'mahasiswa-form',
    'enableAjaxValidation'=>false,
)); ?>

<p class="note">Fields with <span class="required">*</span>
are required.</p>

<?php echo $form->errorSummary($model); ?>

<div class="row">
    <?php echo $form->labelEx($model,'nama'); ?>
    <?php echo $form-
>textField($model,'nama',array('size'=>60,'maxlength'=>60)); ?>
    <?php echo $form->error($model,'nama'); ?>
</div>

<div class="row">
```

```

        <?php echo $form->labelEx($model, 'alamat'); ?>
        <?php echo $form-
>textField($model, 'alamat', array('size'=>60, 'maxlength'=>200)); ?>
        <?php echo $form->error($model, 'alamat'); ?>
    </div>

    <div class="row buttons">
        <?php echo CHtml::submitButton($model->isNewRecord ?
'Create' : 'Save'); ?>
    </div>

<?php $this->endWidget(); ?>

</div><!-- form -->

```

Bagian di atas akan membentuk tampilan yang terdiri dari form input data. Bila anda terbiasa menggunakan script HTML untuk membuat form input data, maka pada Yii Framework kita akan dipermudah dengan menggunakan class yang sudah tersedia dari Yii. Terdapat labelEx, textField, Error, dan submitButton. Untuk lebih memahami masing-masing penggunaan bantuan tersebut, anda dapat lihat secara langsung bagaimana efeknya pada tampilan anda.

Saat actionCreate dijalankan dalam kondisi biasa, action ini hanya akan menampilkan form input data yang sudah disesuaikan dengan model. Bagaimana ketika dijalankan dalam kondisi POST ?? Kondisi POST sendiri akan dijalankan ketika kita menekan “submitButton” yang ada pada halaman form. Ketika tombol “submitButton” ditekan, maka actionCreate akan menjalankan kode berikut :

```

    if(isset($_POST['Mahasiswa']))
    {
        $model->attributes=$_POST['Mahasiswa'];
        if($model->save())
            $this-
            >redirect(array('view','id'=>$model->id));
    }

```

Pada code di atas akan dilakukan penyimpanan data baru sesuai dengan yang diinput oleh user. Itulah bedanya, ketika dalam kondisi biasa action hanya akan

menampilkan form input data, sedangkan pada kondisi POST action akan menyimpan data yang telah diinput oleh user.

## Action Update

```
public function actionUpdate($id)
{
    $model=$this->loadModel($id);

    // Uncomment the following line if AJAX validation is
    needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['Mahasiswa']))
    {
        $model->attributes=$_POST['Mahasiswa'];
        if($model->save())
            $this->redirect(array('view','id'=>$model->id));
    }

    $this->render('update',array(
        'model'=>$model,
    ));
}
```

Action Update tidak jauh berbeda dengan action Create. Bedanya adalah pada action update form input data akan secara otomatis berisi data yang akan di update. Dan pada kondisi POST nya, aplikasi akan memperbaharui data yang telah dipilih.

## Action Delete

```
public function actionDelete($id)
{
    if(Yii::app()->request->isPostRequest)
    {
        // we only allow deletion via POST request
        $this->loadModel($id)->delete();

        // if AJAX request (triggered by deletion via admin
        grid view), we should not redirect the browser
        if(!isset($_GET['ajax']))
    }
}
```

```

        $this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
    }
    else
        throw new CHttpException(400, 'Invalid request. Please
do not repeat this request again.');
```

Action delete hanya dapat diakses ketika di request dalam kondisi POST.

Ini dapat anda lihat pada kondisi berikut :

```

    if(Yii::app()->request->isPostRequest)
    {
    }
    else
        throw new CHttpException(400, 'Invalid request. Please do
not repeat this request again.');
```

Jadi ketika kita merequest tidak dalam kondisi POST, maka akan diperingatkan bahwa request invalid. Dan jika sudah di request dalam kondisi POST, maka akan dijalankan kode berikut :

```

        $this->loadModel($id)->delete();

        // if AJAX request (triggered by deletion via admin
grid view), we should not redirect the browser
        if(!isset($_GET['ajax']))
            $this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
```

Akan dihapus data dengan Primary Key tertentu yang sudah direquest oleh user, kemudian setelah dihapus halaman akan di-redirect kembali ke halamannya sendiri atau ke halaman admin.

## Action Index

```

public function actionIndex()
{
    $dataProvider=new CActiveDataProvider('Mahasiswa');
    $this->render('index', array(
        'dataProvider'=>$dataProvider,
    ));
}
```



```
}
```

Action Index akan menampilkan semua data yang ada pada tabel mahasiswa. Semua data yang ada pada tabel mahasiswa tersebut akan dikirim ke halaman “index” untuk ditampilkan. Halaman index sendiri dapat anda lihat pada “protected/views/mahasiswa/index.php”.

```
<?php  
$this->breadcrumbs=array(  
    'Mahasiswas',  
);  
  
$this->menu=array(  
    array('label'=>'Create Mahasiswa', 'url'=>array('create')),  
    array('label'=>'Manage Mahasiswa', 'url'=>array('admin')),  
);  
?>  
  
<h1>Mahasiswas</h1>  
  
<?php $this->widget('zii.widgets.CListView', array(  
    'dataProvider'=>$dataProvider,  
    'itemView'=>'_view',  
)); ?>
```

Sama seperti halaman create yang memiliki breadcrumbs dan menu. Lalu anda dapat lihat disana terdapat widget “zii.widgets.CListView” yang merender kepada halaman “\_view.php” yang ada pada “protected/views/mahasiswa/\_view.php”:

```
<div class="view">  
  
    <b><?php echo CHtml::encode($data->getAttributeLabel('id'));  
?></b>  
    <?php echo CHtml::link(CHtml::encode($data->id), array('view',  
'id'=>$data->id)); ?>  
    <br />  
  
    <b><?php echo CHtml::encode($data->getAttributeLabel('nama'));  
?></b>  
    <?php echo CHtml::encode($data->nama); ?>  
    <br />  
  
    <b><?php echo CHtml::encode($data->  
>getAttributeLabel('alamat')); ?></b>  
    <?php echo CHtml::encode($data->alamat); ?>  
    <br />  
</div>
```

Anda dapat melihat item-item apa saja yang akan ditampilkan. Dan data pada halaman ini akan di tampilkan sesuai dengan data yang sudah kita kirimkan di halaman “index” sebelumnya. Saya rasa dengan mencoba dan melihat secara langsung akan dapat memberi pemahaman yang lebih bagi anda..

## Action Admin

```
public function actionAdmin()
{
    $model=new Mahasiswa('search');
    $model->unsetAttributes();
    if(isset($_GET['Mahasiswa']))
        $model->attributes=$_GET['Mahasiswa'];

    $this->render('admin',array(
        'model'=>$model,
    ));
}
```

Action admin juga memiliki dua buah kondisi, yaitu kondisi biasa dan kondisi GET. Kondisi biasa akan menampilkan semua data yang ada pada tabel mahasiswa. Sedangkan kondisi GET akan menampilkan semua baris data yang sesuai dengan kondisi yang di tentukan oleh user. Action admin akan memanggil fungsi “search” yang ada pada Model Mahasiswa dimana action ini akan memanggil halaman “admin” yang terdapat pada direktori “protected/views/mahasiswa/admin.php”.

```
<?php
$this->breadcrumbs=array(
    'Mahasiswas'=>array('index'),
    'Manage',
);

$this->menu=array(
    array('label'=>'List Mahasiswa', 'url'=>array('index')),
    array('label'=>'Create Mahasiswa', 'url'=>array('create')),
);

Yii::app()->clientScript->registerScript('search', "
$('.search-button').click(function(){
```

```

        $('.search-form').toggle();
        return false;
    });
    $('.search-form form').submit(function() {
        $.fn.yiiGridView.update('mahasiswa-grid', {
            data: $(this).serialize()
        });
        return false;
    });
    ");
?>

```

**<h1>Manage Mahasiswas</h1>**

**<p>**

**You may optionally enter a comparison operator (**<b>&lt;</b>**, **<b>&lt;=</b>**, **<b>&gt;</b>**, **<b>&gt;=</b>**, **<b>&lt;&gt;</b>** or **<b>=</b>**) at the beginning of each of your search values to specify how the comparison should be done.**

**</p>**

```

<?php echo CHtml::link('Advanced
Search', '#', array('class'=>'search-button')); ?>
<div class="search-form" style="display:none">
<?php $this->renderPartial('_search', array(
    'model'=>$model,
)); ?>
</div><!-- search-form -->

```

```

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'mahasiswa-grid',
    'dataProvider'=>$model->search(),
    'filter'=>$model,
    'columns'=>array(
        'id',
        'nama',
        'alamat',
        array(
            'class'=>'CButtonColumn',
        ),
    ),
)); ?>

```

Pada tampilan tersebut terdapat toggle yang dapat di hide and show.

Toggle itu sendiri mengacu pada halaman `_search` yang terdapat pada

“protected/views/mahasiswa/\_search.php” :

```

<div class="wide form">
<?php $form=$this->beginWidget(' CActiveForm', array(
    'action'=>Yii::app()->createUrl($this->route),
    'method'=>'get',
)); ?>

```

```

<div class="row">
    <?php echo $form->label($model, 'id'); ?>
    <?php echo $form->textField($model, 'id'); ?>
</div>

<div class="row">
    <?php echo $form->label($model, 'nama'); ?>
    <?php echo $form-
>textField($model, 'nama', array('size'=>60, 'maxlength'=>60)); ?>
</div>

<div class="row">
    <?php echo $form->label($model, 'alamat'); ?>
    <?php echo $form-
>textField($model, 'alamat', array('size'=>60, 'maxlength'=>200)); ?>
</div>

<div class="row buttons">
    <?php echo CHtml::submitButton('Search'); ?>
</div>

<?php $this->endWidget(); ?>

</div><!-- search-form -->

```

Halaman `_search` akan menampilkan form input data yang bila kita melakukan input di dalamnya maka akan mengembalikan semua data sesuai dengan yang telah di filter sesuai kondisi yang dimasukkan oleh user. Kembali ke halaman “admin”, pada halaman ini akan memanggil sebuah widget, yaitu widget `CGridView` :

```

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'mahasiswa-grid',
    'dataProvider'=>$model->search(),
    'filter'=>$model,
    'columns'=>array(
        'id',
        'nama',
        'alamat',
        array(
            'class'=>'CButtonColumn',
        ),
    ),
)); ?>

```

Widget ini digunakan untuk menampilkan data dalam bentuk tabel. Anda dapat melihat secara langsung pada action admin bagaimana tampilan dari CGridView dan efek seperti apa saja yang dapat kita lihat disana..

Okeh-okeh.. itulah sedikit penjelasan mengenai code yang telah dihasilkan dari Gii. Jika memang penjelasan dalam ebook ini kurang memuaskan bagi anda, tidak ada salahnya anda otak-atik secara langsung code pada aplikasi anda.. Saran saya cobalah seluk beluk operasi CRUD tersebut dan lihatlah dan pahami code yang ada di balik setiap proses tersebut.. Mungkin dari situ anda dapat mengerti lebih dalam akan code yang dihasilkan dari generate Gii...

# **PART IV**

# **MEMBUAT APLIKASI FORUM DENGAN YII FRAMEWORK**

Sejauh ini pembahasan yang ada pada ebook ini hanya terkait masalah konsep-konsep dan contoh-contoh kecil penggunaan Yii Framework pada beberapa contoh kasus. Saya rasa akan lebih baik jika kita mempelajari bagaimana membangun aplikasi real dengan Yii Framework. Pada ebook ini, saya akan membahas pembuatan real aplikasi yang dalam ebook ini adalah aplikasi forum dengan menggunakan Yii Framework. Kenapa aplikasi forum?? Hmm, sebenarnya ini adalah keputusan subjektif saya kok, walau sebelumnya pertimbangan saya adalah membuat aplikasi simpel social network atau forum.. Dan karena lagi pengen buat aplikasi forum, jadi saya memilih membuat aplikasi forum.. Tapi jangan khawatir, saya rasa aplikasi forum sudah mencakup banyak hal-hal umum yang biasa kita lakukan dalam pembuatan kebanyakan aplikasi web. Selain itu populernya website seperti Kaskus (saya kaskuser juga lho) saat ini mungkin bisa menjadi pancingan agar anda lebih bersemangat untuk mengupas pembuatan aplikasi forum seperti kaskus, walaupun yang akan kita bahas masih sangat jauh dari itu. Yang akan saya bahas sendiri bukanlah aplikasi forum yang sudah kompleks layaknya Kaskus. Saya hanya membahas dasar-dasarnya dari aplikasi forum yang paling tidak sudah bisa digunakan untuk berinteraksi antar satu user dengan user yang lain, saling berkomentar di suatu thread, memberi penilaian pada suatu thread, dan lain-lain. Jikalau anda memang berminat untuk mengembangkannya lebih dalam lagi, saya rasa itu adalah hal yang bagus.. dan jangan lupa dibagi-bagi ya ilmunya ...

## **Mau Aplikasi Seperti Apa??**

Sebelum membuat aplikasi, tentunya kita harus tentukan dulu bagaimana aplikasi yang mau kita buat. Kalo dalam pembuatan aplikasi yang sebenarnya, sebelum membuat suatu aplikasi biasanya kita harus buat rancangan ERD, DFD/diagram UML, Diagram Context, Rancangan GUI, blablabla.... Apalagi kalo buat skripsi/TA, kita diwajibkan buat proposal, latar belakang, masalah, diagram inilah, itulah, minta tanda tangan dosen, disuruh pembimbing revisi lah dan sebagainya sebagainya.. Tapi karena ditakutkan bisa membuat ebook ini menembus angka ribuan halaman kalo saya bahas perancangan ABC, maka saya rasa kita akan membahas “aplikasi seperti apa yang kita inginkan” hanya dari fitur-fiturnya saja. Berikut fitur-fitur dari aplikasi yang akan kita buat :

- User dapat melakukan register dan login
- Jika belum login, semua orang bisa melihat forum tetapi tidak bisa membuat postingan baru atau membuat komentar pada suatu postingan
- Jika telah berhasil melakukan login, user dapat membuat postingan baru dan menambahkan komentar
- Tiap postingan dikelompokkan berdasarkan kategori
- Halaman pertama akan memperlihatkan top member, list postingan user dan berita (news) yang diposting oleh admin
- User terdiri dari 3 level yaitu Admin, moderator, dan Member biasa
- Admin dan moderator dapat menambah, mengedit, dan menghapus “posting”



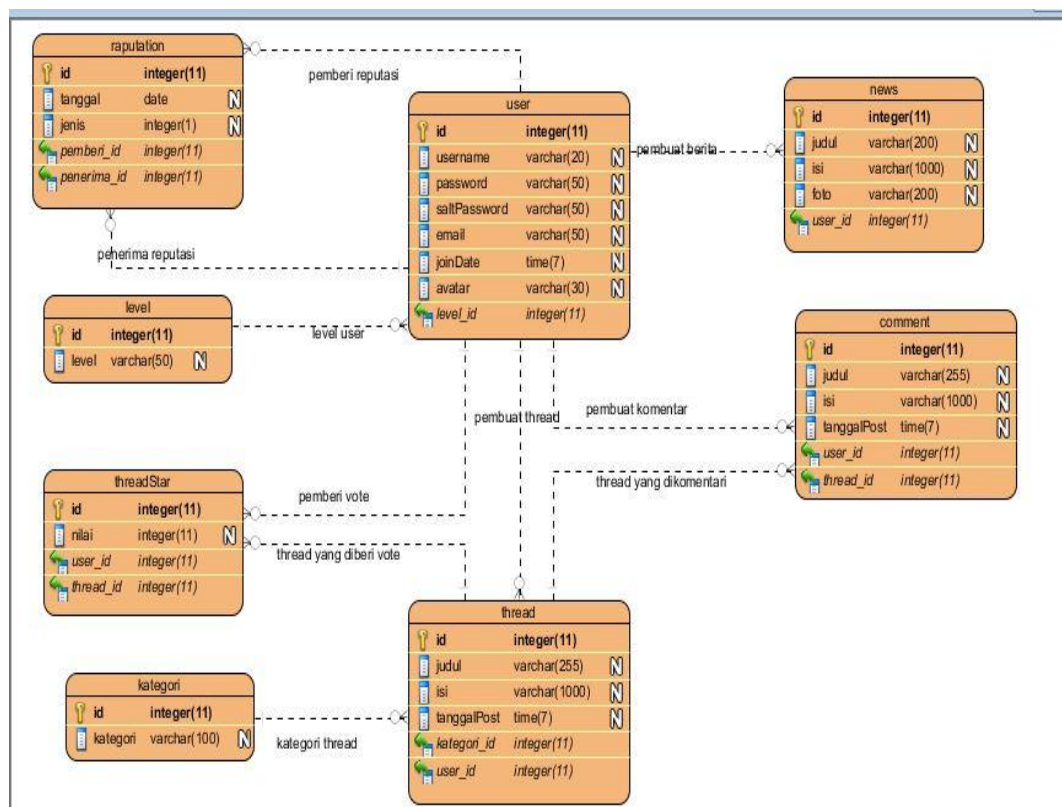
- Admin dan moderator dapat mengelola “kategori” dan “berita”
- User yang telah melakukan login dapat memberi komentar pada suatu thread
- User dapat memberikan reputasi pada user lain

Okeh, itulah fitur-fitur yang ingin kita capai pada pembangunan aplikasi ini.. Semoga bisa tercapai semuanya.. ☺

### Merancang Database

Setelah menentukan fitur-fitur, sekarang saatnya kita merancang database.

Berikut rancangan database yang telah saya buat :



- Level : Tabel level digunakan sebagai identitas level pada User. Seperti yang telah ada pada fitur bahwa user dalam aplikasi ini terdiri dari 3 jenis user, yaitu Admin, Moderator dan User Biasa. Dengan adanya tabel level, menjadi memungkinkan jika kita ingin memodifikasi level pada user.
- Kategori : Untuk membagi-bagi thread yang ada ke dalam kategori-kategori tertentu. Kategori juga dapat ditambah, dikurangi dan dimodifikasi sesuai kebutuhan.
- User : Tabel ini digunakan untuk menyimpan informasi user. Tabel user juga lah yang digunakan untuk proses autentikasi ketika user melakukan proses login.
- Thread : Menyimpan semua informasi thread yang terdapat dalam forum.
- Comment : Menyimpan semua komentar dari setiap thread
- News : Menyimpan informasi berita yang diinput oleh admin yang kemudian akan ditampilkan pada halaman utama aplikasi.
- Raputation : Menyimpan semua penilaian yang diberikan oleh user kepada seorang user.
- Threadstar : Menyimpan informasi penilaian pada suatu thread yang dilakukan oleh user.

Berikut SQL database di atas yang telah saya buat :

```
CREATE TABLE IF NOT EXISTS `comment` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `judul` varchar(255) DEFAULT NULL,
  `isi` text NOT NULL,
  `user_id` int(11) NOT NULL,
  `thread_id` int(11) NOT NULL,
```

```

    `tanggalPost` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (`id`),
    KEY `user_id` (`user_id`),
    KEY `thread_id` (`thread_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `kategori` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `kategori` varchar(100) NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `level` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `level` varchar(50) NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `news` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `judul` varchar(200) NOT NULL,
    `isi` text NOT NULL,
    `foto` varchar(200) NOT NULL,
    `user` int(11) NOT NULL,
    PRIMARY KEY (`id`),
    KEY `user` (`user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `raputation` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `tanggal` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    `jenis` tinyint(1) NOT NULL,
    `pemberi_id` int(11) NOT NULL,
    `penerima_id` int(11) NOT NULL,
    PRIMARY KEY (`id`),
    KEY `pemberi_id` (`pemberi_id`),
    KEY `penerima_id` (`penerima_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `thread` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `judul` varchar(255) NOT NULL,
    `isi` text NOT NULL,
    `user_id` int(11) NOT NULL,
    `kategori_id` int(11) NOT NULL,
    `tanggalPost` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (`id`),
    KEY `user_id` (`user_id`),
    KEY `kategori_id` (`kategori_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

CREATE TABLE IF NOT EXISTS `threadstar` (
  `is` int(11) NOT NULL AUTO_INCREMENT,
  `nilai` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `thread_id` int(11) NOT NULL,
  PRIMARY KEY (`is`),
  KEY `user_id` (`user_id`),
  KEY `thread_id` (`thread_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

CREATE TABLE IF NOT EXISTS `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(20) NOT NULL,
  `password` varchar(50) NOT NULL,
  `saltPassword` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `joinDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
  `level_id` int(11) NOT NULL,
  `avatar` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`),
  KEY `level_id` (`level_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

ALTER TABLE `comment`
  ADD CONSTRAINT `comment_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `user` (`id`) ON UPDATE CASCADE,
  ADD CONSTRAINT `comment_ibfk_2` FOREIGN KEY (`thread_id`)
REFERENCES `thread` (`id`) ON UPDATE CASCADE;

```

```

ALTER TABLE `news`
  ADD CONSTRAINT `news_ibfk_1` FOREIGN KEY (`user`) REFERENCES
`user` (`id`) ON UPDATE CASCADE;

```

```

ALTER TABLE `raputation`
  ADD CONSTRAINT `raputation_ibfk_1` FOREIGN KEY (`pemberi_id`)
REFERENCES `user` (`id`),
  ADD CONSTRAINT `raputation_ibfk_2` FOREIGN KEY (`penerima_id`)
REFERENCES `user` (`id`);

```

```

ALTER TABLE `thread`
  ADD CONSTRAINT `thread_ibfk_3` FOREIGN KEY (`user_id`)
REFERENCES `user` (`id`) ON UPDATE CASCADE,
  ADD CONSTRAINT `thread_ibfk_4` FOREIGN KEY (`kategori_id`)
REFERENCES `kategori` (`id`) ON UPDATE CASCADE;

```

```

ALTER TABLE `threadstar`
  ADD CONSTRAINT `threadstar_ibfk_3` FOREIGN KEY (`user_id`)
REFERENCES `user` (`id`) ON UPDATE CASCADE,

```

```
ADD CONSTRAINT `threadstar_ibfk_4` FOREIGN KEY (`thread_id`)
REFERENCES `thread` (`id`) ON UPDATE CASCADE;
```

```
ALTER TABLE `user`
ADD CONSTRAINT `user_ibfk_1` FOREIGN KEY (`level_id`) REFERENCES
`level` (`id`) ON UPDATE CASCADE;
```

Okeh, kalo ga mau repot buat database nya anda bisa copas langsung script di atas kok.. Terus langsung deh pindahin ke database anda.. gitu aja kok reppoot..

## **Membuat Aplikasi, Menghubungkan Database, dan Membuat CRUD**

### **Operation**

Setelah anda selesai membuat database, silahkan buat aplikasi Yii Framework baru dengan nama “forum” (kalo lebih seneng nama lain silahkan, tapi saya asumsikan namanya aplikasinya “forum” yak..). Setelah itu mari kita hubungkan aplikasi tersebut dengan database yang telah kita buat tadi, dan aktifkan fitur Gii pada Yii Framework. Gimana caranya?? Kalo anda lupa, silahkan lihat kembali pembahasan sebelumnya. Jika anda sudah berhasil melakukan keduanya, silahkan anda generate CRUD operation pada semua tabel yang ada pada database anda. Gimana caranya?? Silahkan baca lagi pembahasan sebelumnya... ☺

### **Membuat Operasi Register, Login dan Logout**

Okeh, saya harap ga ada kendala pada proses koneksi database dan generate CRUD operation pada masing-masing tabel. Nah, sekarang kita akan membahas tahapan-tahapan pada proses autentikasi user yang terdiri dari operasi register, login, dan logout.

Pertama-tama mari kita membuat fungsi untuk melakukan “register”. Operasi ini adalah operasi yang dilakukan apabila user ingin mendaftar pada aplikasi kita. Setelah dia melakukan pendaftaran, maka data user akan disimpan ke database pada tabel “user”. Untuk menginput data ke tabel user, kita dapat menggunakan action pada UserController dan actionCreate(). Berikut hak akses yang terdapat pada UserController :

```
public function accessRules()
{
    return array(
        array('allow',
            'actions'=>array('create','captcha'),
            'users'=>array('*'),
        ),
        array('allow',
            'actions'=>array('update','view'),
            'users'=>array('@'),
        ),
        array('allow',
            'actions'=>array('admin','index','delete'),
            'expression'=>'$user->getLevel()<=1',
        ),
        array('deny',
            'users'=>array('*'),
        ),
    );
}
```

Action “Create” adalah menu register, action ini dapat diakses oleh semua pengunjung yang mengunjungi aplikasi web ini. Captcha juga diizinkan untuk semua jenis pengunjung, captcha sendiri adalah class yang mengizinkan kita untuk menggunakan captcha pada form register. Captcha sendiri digunakan sebagai konfirmasi pada saat pendaftaran. Update dan view adalah menu agar user dapat memperbaharui profilnya dan melihat profilnya. Untuk melakukan action ini, maka user harus login terlebih dahulu. Admin, index dan delete adalah menu

yang hanya dapat oleh admin. Disana terdapat code `'expression'=>'$user->getLevel()<=1'`, Kode ini akan kita bahas pada bahasan selanjutnya.

Pada tabel user terdapat attribute-attribute seperti berikut :

- Id = attribute ini tidak diinput oleh user melainkan secara otomatis akan digenerate oleh database
- Username = username user
- Password dan saltPassword = password dan saltPassword adalah dua attribute yang berhubungan. Password sendiri akan diinput oleh user, sedangkan saltPassword akan di generate secara otomatis oleh aplikasi. Hanya saja pada attribute password, data yang masuk ke dalam database tidak akan sama dengan data yang diinput oleh user. Data yang akan masuk ke dalam attribute password sendiri adalah data yang sudah dienkripsi antara input user dan saltPassword. Jadi jangan heran jika pada database nanti anda melihat attribute password dan saltPassword adalah sekumpulan karakter yang tidak pernah anda tahu. Kenapa kita menerapkan mekanisme ini?? Ini dilakukan untuk meningkatkan keamanan pada aplikasi. Bahkan pihak yang mengelola aplikasi pun tidak bisa mengetahui password dari seorang user.
- Email = Email dari user
- Joindate= tanggal melakukan register. Attribute ini secara otomatis akan mengambil tanggal user melakukan register.
- Level\_id = level pada seorang user. Attribute ini secara default akan di set bernilai 3, yaitu member biasa
- Avatar = attribute yang berisi link ke file gambar avatar seorang user. Pada tampilannya, user akan disuruh memilih sebuah file gambar untuk dijadikan avatarnya.

Pada saat login, kita akan meminta user memasukkan password dua kali.

Jika antara password1 dan password2 berbeda, maka user diminta mengulangi

lagi proses memasukkan password. Lalu kita meminta user menginput captcha konfirmasi kepada user. Maka dari itu kita perlu menambah dua buah variabel, yaitu variabel password2 dan captcha yang kita beri nama verifyCode. Maka dari itu tambahkan attribute berikut pada model anda :

```
class User extends CActiveRecord
{
    public $password2;
    public $verifyCode;
```

Setelah itu kita buat validasi pada model user sesuai dengan aturan bisnis yang sudah kita tentukan :

```
public function rules()
{
    return array(
        array('username, password,
email,password2,verifyCode', 'required', 'message'=>'{attribute}
Tidak Boleh Kosong'),
        array('verifyCode', 'captcha',
'allowEmpty'=>!extension_loaded('gd')),
        array('level_id', 'numerical', 'integerOnly'=>true),
        array('username', 'length', 'max'=>20),
        array('password, saltPassword, email', 'length',
'max'=>50),
        array('avatar', 'file', 'types'=>'gif,png,jpg'),
        array('id, username, password, saltPassword, email,
joinDate, level_id, avatar, isActive', 'safe', 'on'=>'search'),
    );
}
```

- Aturan pertama : attribute username, password, email, password2, verivyCode wajib diisi.
- Aturan kedua : VerifyCode harus sesuai dengan sama dengan kode yang ditampilkan oleh captcha.
- Aturan ketiga : level\_id harus berupa angka
- Aturan keempat : Panjang maksimal karakter username adalah 20



- Aturan kelima : Panjang maksimal password, saltPassword dan email maksimal 50
- Aturan keenam : attribute avatar harus merupakan file dan ekstensi yang diizinkan untuk file tersebut adalah gif, jpg, dan png.

Yah, karena kita sudah menambahkan attribute baru pada aturan register di model user, maka kita juga perlu melakukan konfigurasi pada tampilan kita agar sesuai dengan aturan tersebut :

```
<div class="form">
<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'user-form',
    'enableAjaxValidation'=>false,
    'htmlOptions'=>array('enctype'=>'multipart/form-data'),
)); ?>

<p class="note">Fields with <span class="required">*</span>
are required.</p>

<?php echo $form->errorSummary($model); ?>

<div class="row">
    <?php echo $form->labelEx($model, 'username'); ?>
    <?php echo $form->
>textField($model, 'username', array('size'=>20, 'maxlength'=>20));
?>
    <?php echo $form->error($model, 'username'); ?>
</div>

<div class="row">
    <?php echo $form->labelEx($model, 'password'); ?>
    <?php echo $form->
>passwordField($model, 'password', array('size'=>50, 'maxlength'=>50)
); ?>
    <?php echo $form->error($model, 'password'); ?>
</div>

<div class="row">
    <?php echo $form->labelEx($model, 'password2'); ?>
    <?php echo $form->
>passwordField($model, 'password2', array('size'=>50, 'maxlength'=>50
)); ?>
    <?php echo $form->error($model, 'password2'); ?>
</div>
```

```

        <div class="row">
            <?php echo $form->labelEx($model, 'email'); ?>
            <?php echo $form-
>textField($model, 'email', array('size'=>50, 'maxlength'=>50)); ?>
            <?php echo $form->error($model, 'email'); ?>
        </div>

        <div class="row">
            <?php echo $form->labelEx($model, 'avatar'); ?>
            <?php echo $form-
>fileField($model, 'avatar', array('size'=>30, 'maxlength'=>30)); ?>
            <?php echo $form->error($model, 'avatar'); ?>
        </div>

        <?php if (extension_loaded('gd')): ?>
            <div class="row">
                <?php echo CHtml::activeLabelEx($model, 'verifyCode')
?>

                <div>
                    <?php $this->widget('CCaptcha'); ?><br/>
                    <?php echo
CHtml::activeTextField($model, 'verifyCode'); ?>
                </div>
                <div class="hint">Ketik tulisan yang ada pada gambar .
                <br/>Tulisan tidak case sensitive</div>
            </div>
            <?php endif; ?>

            <div class="row buttons">
                <?php echo CHtml::submitButton($model->isNewRecord ?
'Create' : 'Save'); ?>
            </div>

        <?php $this->endWidget(); ?>
    </div><!-- form -->

```

Diatas terdapat kode untuk menggunakan captcha dan untuk mengaktifkan captcha kita harus menambahkan kode berikut pada controller user :

```

public function actions()
{
    return array(
        'captcha'=>array(
            'class'=>'CCaptchaAction',
            'backColor'=>0xFFFFFF,
        ),
    );
}

```

Pada code di atas saya menambahkan input data untuk attribute password2 dan verivyCode. Saya juga melakukan pengaturan agar pada attribute avatar input data yang muncul adalah memerintahkan user untuk memilih file gambar yang akan di upload. Setelah melakukan pengaturan pada menu input data, sekarang kita sesuaikan juga code pada controller sesuai dengan aturan register.

Pada action register, kita akan melakukan generate nilai pada attribute saltPassword dan melakukan enkripsi attribute password sesuai dengan data password yang diinput oleh user dan nilai saltPassword. Karena itu kita harus melakukan pengaturan tersebut pada Model "User". Tambahkan fungsi berikut pada model tersebut :

```
public function validatePassword($password)
{
    return $this->hashPassword($password,$this->saltPassword)=== $this->password;
}

public function hashPassword($password,$salt)
{
    return md5($salt.$password);
}

public function generateSalt()
{
    return uniqid('', true);
}
```

Setelah itu, pada User Controller tepatnya pada actionCreate buat menjadi seperti berikut :

```
public function actionCreate()
{
    $model=new User;

    // Uncomment the following line if AJAX validation is
    needed
    // $this->performAjaxValidation($model);
}
```

```

        if(isset($_POST['User']))
        {
            $model->attributes=$_POST['User'];

            $dua=$model->password;
            $model->saltPassword=$model->generateSalt();
            $model->password=$model->hashPassword($dua,$model->
>saltPassword);
            $model->level_id=3;
            $model->isActive=0;

            $sss;

if(strlen(trim(CUploadedFile::getInstance($model,'avatar')) > 0)
        {
            $sss=CUploadedFile::getInstance($model,'avatar');
            $model->avatar=$model->username.'.'.$sss-
>extensionName;
        }

        if($model->save())
        {
            if(strlen(trim($model->avatar)) > 0)
                $sss->saveAs(Yii::app()->basePath .
'../avatar/' . $model->avatar);
            $this->redirect(array('view','id'=>$model->id));
        }
    }

    $this->render('create',array(
        'model'=>$model,
    ));
}

```

Baiklah, selesai membuat actionCreate.. Silahkan anda coba jalankan fungsi tersebut.. Jika ada yang error... selamat memperbaiki aplikasi anda ya.. Happy coding.. ☺

Sekarang kita akan membuat action untuk melakukan login. Saat pertama kali kita melakukan generate code aplikasi Yii, Yii secara otomatis telah menyediakan form dan action login. Daripada membuat dari awal, lebih baik kita modifikasi action login yang sudah disediakan oleh Yii. Anda bisa lihat code yang memproses action login di “protected/controllers/SiteController.php”, “protected/view/site/login.php”, “protected/models/LoginForm.php”, dan

“protected/Components/UserIdentify.php”. Walau ada banyak file yang terkait action login, namun yang akan kita modifikasi hanya UserIdentify.php.

```
class UserIdentity extends CUserIdentity
{
    private $_id;

    public function authenticate()
    {
        $username = strtolower($this->username);
        $user = User::model()->find('LOWER(username)=?',
array($username));
        if($user===null)
            $this->errorCode=self::ERROR_USERNAME_INVALID;
        else if(!$user->validatePassword($this->password))
            $this->errorCode = self::ERROR_PASSWORD_INVALID;
        else
        {
            $this->_id = $user->id;
            $this->username = $user->username;
            $this->errorCode = self::ERROR_NONE;
        }
        return $this->errorCode == self::ERROR_NONE;
    }

    public function getId()
    {
        return $this->_id;
    }
}
```

Yang dilakukan kode di atas adalah mengambil nilai dari tabel user dan mencocokkan username dan password yang telah diinput oleh user. Tentunya password yang akan di cocokkan adalah password yang telah di enkripsi. Jika username dan password yang diinput user sudah sesuai dengan username dan password yang ada pada database, maka user berhasil melakukan login.

Okeh, selesai sudah kita membuat menu register, login, dan logout. Menu logout sudah tersedia langsung dari Yii dan tidak perlu kita modifikasi lagi. Silahkan anda coba action-action tersebut... Semoga langsung berjalan dengan

lancar yak... Tapi kalo ada yang gagal.. Happy codiiiiing... Error adalah Success Running yang tertunda...

## Membuat User Level Akses

Salah satu hal penting dalam suatu aplikasi adalah user level akses. User level akses adalah pengkategorian hak akses user berdasarkan level user tersebut. Sebagai contoh kita hanya mengizinkan user dengan level admin membuat berita baru, mengizinkan member yang telah login membuat thread baru, dan lain-lain. Pertama kali yang harus anda buat adalah sebuah proses untuk mengetahui level user yang sedang melakukan aktivitas di dalam aplikasi kita. Pada “protected/components/” buatlah sebuah file dengan ekstensi “.php” dengan nama “EwebUser.php”. Lalu masukkan kode berikut pada file tersebut :

```
<?php
class EwebUser extends CWebUser{

    protected $_model;

    protected function loadUser()
    {
        if ( $this->_model === null ) {
            $this->_model = User::model()->findByPk($this->id);
        }
        return $this->_model;
    }

    function getLevel()
    {
        $user=$this->loadUser();
        if($user)
            return $user->level_id;
        return 100;
    }
}
```

Kode di atas menurunkan class CwebUser yang ada pada Yii. Pada kode diatas fungsi getLevel() akan mengembalikan level user yang sedang login dari tabel “User” pada attribute “level\_id”. Setelah membuat kode di atas, kita dapat membatasi fungsi-fungsi tertentu berdasarkan levelnya seperti contoh berikut:

```
public function accessRules ()
{
    return array(
        array('allow',
            'actions'=>array('index','view'),
            'users'=>array('*'),
        ),
        array('allow',
            'actions'=>array('create','update'),
            'users'=>array('@'),
        ),
        array('allow',
            'actions'=>array('admin','delete'),
            'expression'=>'$user->getLevel()<=1',
        ),
        array('deny',
            'users'=>array('*'),
        ),
    );
}
```

Kode di atas mengizinkan action index dan view diakses oleh semua orang baik yang sudah login ataupun tidak. Action create dan update hanya bisa diakses oleh user yang sudah login. Action admin dan delete hanya bisa diakses oleh user yang memiliki level bernilai  $\leq 1$ . Nilai level sendiri akan diambil dari tabel user. Pada aplikasi ini attribute level\_id akan terhubung dengan attribute level yang ada pada tabel level, dan level 1 merupakan user dengan tipe admin. Nah, dengan menggunakan metode tersebut anda dapat menentukan sendiri aturan level akses anda masing-masing. Untuk aplikasi ini tentu kita harus menyesuaikan dengan kebutuhan aplikasi kita.

Selain bisa mengatur suatu akses level, kita juga bisa menggunakan level akses user untuk mengatur akses suatu tampilan. Misal kalo login dengan level admin akan nampil tulisan A, sedangkan kalo login dengan level user biasa akan nampil tulisan B. Berikut contoh penggunaanya

```
if(Yii::app()->user->getLevel()<=2)
{
    echo "itu";
}
else
{
    echo "ini";
}
```

### **Memfaatkan Extension Untuk Tampilan**

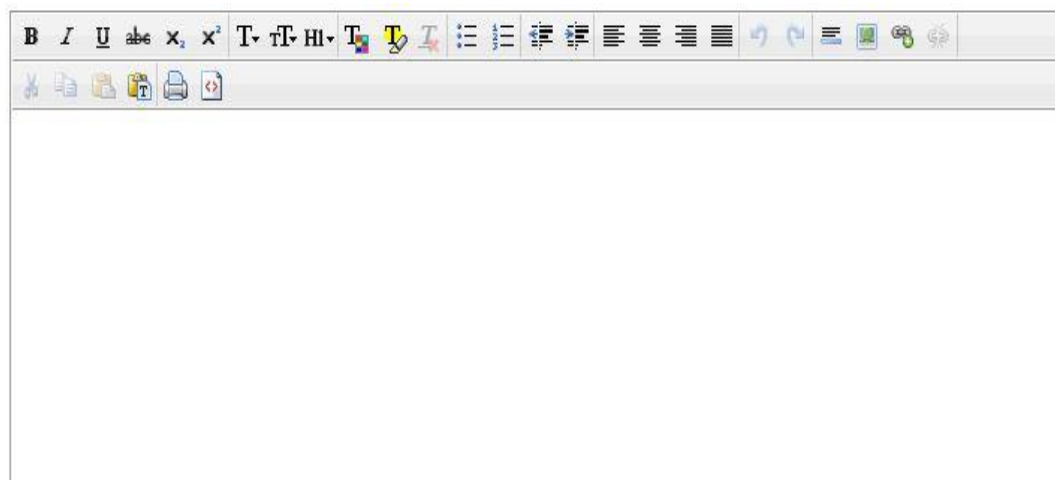
Pada saat member membuat postingan baru dalam aplikasi forum ini, kita ingin tampilan form input post mirip dengan tampilan Microsoft Office misalnya. Dimana disana sudah tersedia fitur untuk mempermudah user membuat text dengan format-format tertentu yang ia inginkan. Paling tidak user bisa membuat suatu tulisan berhuruf tebal, huruf miring, dan lain-lain. Mau buat sendiri?? Bisa keburu tua mas kalo buat sendiri.. Tapi tenang, di luar sana banyak orang yang telah membuat komponen seperti itu dan berbaik hati mengizinkan kita menggunakannya secara gratis.. tis.. tis... Dan ada juga orang yang sudah berbaik hati untuk mempermudah user Yii Framework menggunakan komponen-komponen itu secara lebih mudah dengan membuat komponen tersebut menjadi extension pada Yii.. Wah jadi terharu dengan rekan-rekan komunitas Yii Framework yang sudah berbaik hati.. semoga amal kebaikan kalian dibalas oleh Tuhan.. ☺ Ada banyak sekali extension yang ada pada Yii Framework, jika anda



ingin melihat-lihat extension pada Yii Framework anda dapat berkunjung ke : <http://www.yiiframework.com/extensions/>. Okeh, mari kita sempurnakan aplikasi kita dengan memanfaatkan beberapa extension yang sudah ada..

### **CLEditor**

Cleditor adalah extension yang digunakan untuk membuat form input data yang sudah dilengkapi dengan fitur untuk pengaturan suatu text. Kurang lebih tampilan form input data anda akan seperti berikut jika menggunakan Cleditor :



Gambar 27. Tampilan Extension CLEditor

Keren bukan?? Okeh, untuk menggunakan extension tersebut silahkan download extension di : <http://www.yiiframework.com/extension/cleditor>. Setelah itu unpack file yang telah anda download dan pindahkan ke dalam aplikasi anda tepatnya pada direktori “protected/extensions/”. Lalu buka kode tampilan form input data anda, misal saya gunakan saat membuat form input data thread pada “protected/views/thread/\_form.php”. Kode anda yang sebelumnya seperti berikut :

```

<div class="row">
    <?php echo $model->getAttributeLabel('isi').' :'; ?>
    <?php echo $form-
>textField($model, 'isi', array('size'=>60, 'maxlength'=>255)); ?>
    <?php echo $form->error($model, 'isi'); ?>
</div>

```

Rubah menjadi seperti berikut :

```

<div class="row">
    <?php echo $model->getAttributeLabel('isi').' :'; ?>
    <?php $this-
>widget('application.extensions.cleditor.ECLEditor', array(
        'model'=>$model,
        'attribute'=>'isi',
        'options'=>array(
            'width'=>'700',
            'height'=>250,
            'useCSS'=>true,
        ),
        'value'=>$model->isi,));
    ?>
    <?php echo $form->error($model, 'isi'); ?>
</div>

```

Dan lihat hasilnya pada tampilan anda.. Selain pada form input “thread”, saya juga menyisipkan kode tersebut pada form input data “comment” dan “news”. Jika anda rasa perlu, silahkan menggunakan komponen tersebut di tempat yang anda rasa perlu..

### **MbMenu**

MbMenu digunakan untuk menampilkan menu yang lebih menarik yang dapat menampilkan submenu. Berikut contoh tampilan yang dihasilkan menggunakan MbMenu :



Gambar 28. Tampilan Extension MbMenu

Extension MbMenu sendiri dapat anda download di : <http://www.yiiframework.com/extension/mbmenu>. Setelah di download, letakkan Mbmenu pada direktori extension tempat kita meletakkan Cleditor tadi. Pengaturan menu terdapat pada direktori “protected/views/layouts/main.php”. Kode yang sebelumnya seperti berikut :

```

        $this->widget('zii.widgets.CMenu', array(
            'items'=>array(
                array('label'=>'Home',
                    'url'=>array('/site/index')),
                array('label'=>'About', 'url'=>array('/site/page',
                    'view'=>'about')),
                array('label'=>'Contact',
                    'url'=>array('/site/contact')),
                array('label'=>'Login',
                    'url'=>array('/site/login'), 'visible'=>Yii::app()-
                    >isGuest),
                array('label'=>'Logout ('.Yii::app()->user-
                    >name.)', 'url'=>array('/site/logout'), 'visible'=>!Yii::app()-
                    >user->isGuest)
            ),
        ));

```

Rubah menjadi seperti berikut :

```

$kategoriAll=array();
        $kategori=Kategori::model()->findAll();
        foreach ($kategori as $i=>$ii)
        {

$kategoriAll[]=array('label'=>$ii["kategori"], 'url'=>array('/kate
gori/view', 'id'=>$ii["id"]));
        }

```

```

        $this->widget('ext.mbmenu.MbMenu', array(
            'items'=>array(
                array('label'=>'Home',
                    'url'=>array('/site/index')),
                array('label'=>'Forum',
                    'url'=>array('/kategori/index'),
                    'items'=>$kategoriAll,
                ),
                array('label'=>'Kelola Kategori',
                    'url'=>array('/kategori/admin'), 'visible'=>Yii::app()->user-
                    >getLevel()<=2),
                array('label'=>'Kelola Thread',
                    'url'=>array('/thread/admin'), 'visible'=>Yii::app()->user-
                    >getLevel()<=2),
                array('label'=>'Kelola Komentar',
                    'url'=>array('/comment/admin'), 'visible'=>Yii::app()->user-
                    >getLevel()<=2),
                array('label'=>'Kelola Berita',
                    'url'=>array('/news/admin'), 'visible'=>Yii::app()->user-
                    >getLevel()<=2),
                array('label'=>'Login',
                    'url'=>array('/site/login'), 'visible'=>Yii::app()->user-
                    >isGuest),
                array('label'=>'Register',
                    'url'=>array('/user/create'), 'visible'=>Yii::app()->user-
                    >isGuest),
                array('label'=>'Profile ('.Yii::app()->user-
                    >name.)', 'url'=>array('/user/view', 'id'=>Yii::app()->user->id),
                    'visible'=>!Yii::app()->user->isGuest),
                array('label'=>'Logout',
                    'url'=>array('/site/logout'), 'visible'=>!Yii::app()->user-
                    >isGuest)
            ),
        ));

```

Pada kode tersebut saya menambahkan menu lain dan ada beberapa menu yang saya ambil langsung dari database.

### S3slider

Digunakan untuk membuat tampilan slide yang akan menampilkan slide show foto beserta text yang akan ditampilkan. Silahkan anda download extensionnya disini : <http://www.yiiframework.com/extension/s3slider/>. Saya akan meletakkan slider pada halaman utama, yaitu pada “protected/views/site/index.php”. Berikut kodenya :

```

$sterserah=array();
foreach($dataProvider->getData() as $i=>$ii)
{
    $situ=Chtml::link($ii['judul'],array('news/view','id'=>$ii['id']));
    $awal='a/./news/'. $ii['id']. $ii['foto'];
    $sterserah[$i]=array($awal,$situ);
}
$this->widget('application.extensions.s3slider.S3Slider',
    array(
        'images' => $sterserah,
        'width' => '590',
        'height' => '375',
    )
);

```

Pada kode di atas, aplikasi akan mengambil foto dan text dari lima data terakhir di tabel “news”. Dimana data tersebut akan di ambil dari controller, berikut kode controller yang mengambil nilai dari tabel “news” :

```

public function actionIndex()
{
    $dataProvider=Thread::model()->lastNew();

    $this->render('index',array(
        'dataProvider'=>$dataProvider,
    ));
}

```

Kita mengambil lima data terakhir pada tabel news dengan memanggil fungsi yang telah kita buat di model “Thread” yang bernama lastNew(). Maka dari itu pada “models/Thread.php”, tambahkan sebuah fungsi yang mengembalikan nilainya seperti berikut :

```

public function lastNew()
{
    $sql='SELECT * FROM news order by id desc';
    $dataProvider=new CSqlDataProvider($sql,array(
        'keyField' => 'id',
        'pagination'=>array(
            'pageSize'=>5,
        ),
    ));
    return $dataProvider;
}

```

```
}
```

Ada sebuah SQL yang mengambil semua nilai dari tabel “news” dan data yang diambil akan dibatasi hanya lima data pada CsqlDataProvider..

### **Beresin Aplikasi..**

Sebelumnya kita sudah menentukan spesifikasi yang ada pada aplikasi kita, mari kita review apakah sudah tercapai semua spesifikasi tersebut :

- User dapat melakukan register dan login
- Jika belum login, semua orang bisa melihat forum tetapi tidak bisa membuat postingan baru atau membuat komentar pada suatu postingan
- Jika telah berhasil melakukan login, user dapat membuat postingan baru dan menambahkan komentar
- Tiap postingan dikelompokkan berdasarkan kategori
- Halaman pertama akan memperlihatkan top member, list postingan user dan berita(news) yang diposting oleh admin
- User terdiri dari 3 level yaitu Admin, moderator, dan Member biasa
- Admin dan moderator dapat menambah, mengedit, dan menghapus “posting”
- Admin dan moderator dapat mengelola “kategori” dan “berita”
- User yang telah melakukan login dapat memberi komentar pada suatu thread
- User dapat memberikan reputasi pada user lain

Point pertama sudah bisa kita lakukan. Point dua, tiga dan tujuh menyangkut level akses user, dimana kita hanya mengizinkan action Create pada ThreadController dan CommentController hanya bisa diakses oleh orang yang sudah login. Untuk itu silahkan buka kedua controller tersebut, lalu berikan level aksesnya :

```
public function accessRules()
{
    return array(
        array('allow', // allow all users to perform 'index'
and 'view' actions
            'actions'=>array('index','view'),
            'users'=>array('*'),
        ),
        array('allow', // allow authenticated user to perform
'create' and 'update' actions
            'actions'=>array('create','update'),
            'users'=>array('@'),
        ),
        array('allow', // allow admin user to perform 'admin'
and 'delete' actions
            'actions'=>array('admin','delete'),
            'expression'=>'$user->getLevel()<=2',
        ),
        array('deny', // deny all users
            'users'=>array('*'),
        ),
    );
}
```

Kode di atas adalah kode access rule pada commentController dan threadController. Kita membatasi operasi “create” dan “update” hanya dapat diakses oleh orang yang sudah login. Sedangkan operasi “admin” dan “delete” hanya dapat diakses oleh user yang memiliki level kurang dari sama dengan 2, yaitu level admin dan moderator.

Pada point ke empat, aplikasi akan mengkategorikan masing-masing thread atau postingan pada forum dengan kategori yang sudah ada pada tabel kategori. Maka dari itu setiap dilakukan pembuatan thread baru diwajibkan untuk menginput kategori thread tersebut. Kurang lebih alur pembuatan thread baru sebagai berikut :

1. User Memilih Salah Satu Kategori



Gambar 29. Tampilan Halaman Kategori

2. User Masuk ke Tampilan Kategori, lalu membuat thread baru dengan cara memilih tombol “Buat Thread baru”



Home / Kategori / Lounge			
Buat Thread Baru			
Judul	Rate	By	Total Komentar
Izin Gabung	0/5	sabit6	0
Bakar Diri di Depan Istana	0/5	sabit3	1
[ASK]Cara Hack FB?	0/5	sabit	1
[Pool] Pilih Messi apa Ronaldo gan??	4/5	sabit	0
Gan, kira2 siapa yg ada di balik topeng madara?	2/5	admin	4

Copyright © 2011 by My Company.  
All Rights Reserved.  
Powered by Yii Framework.

Gambar 30. Tampilan Suatu Kategori Thread

Dengan cara seperti itu, user tidak akan menyadari kalau aplikasi sudah meminta dirinya menginput “kategori” pada thread tersebut dikarenakan sistem akan menangkap sendiri thread tempat user ingin menambahkan thread baru. Lalu bagaimana dengan code nya?? Okeh, mari kita bahas..

1. Berikan Parameter Kategori pada actionCreate yang terdapat pada ThreadController.php

```
public function actionCreate($id)
{
    $model=new Thread;

    // Uncomment the following line if AJAX validation is
    needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['Thread']))
    {
        $model->attributes=$_POST['Thread'];
        $model->kategori_id=$id;
        $model->user_id=Yii::app()->user->id;
        if($model->save())
            $this->redirect(array('view','id'=>$model->id));
    }

    $this->render('create',array(
```

```

        'model'=>$model,
    ));
}

```

Action di atas memiliki sebuah parameter \$id yang digunakan untuk membuat kategori pada thread yang dibuat. Kode di atas juga sudah membuat sebuah aturan dimana thread yang akan ditambahkan memiliki “kategori\_id” sesuai dengan parameter \$id tersebut.

## 2. Memberikan Link/Tautan yang Menyisipkan Parameter “Kategori” Sesuai Kategori yang Ada

```

<?php
$this->breadcrumbs=array(
    'Kategori'=>array('index'),
    $model->kategori,
);

$this->menu=array(
    array('label'=>'List Kategori', 'url'=>array('index')),
    array('label'=>'Create Kategori', 'url'=>array('create')),
    array('label'=>'Update Kategori', 'url'=>array('update',
'id'=>$model->id)),
    array('label'=>'Delete Kategori', 'url'=>'#',
'linkOptions'=>array('submit'=>array('delete', 'id'=>$model->id),
'confirm'=>'Are you sure you want to delete this item?')),
    array('label'=>'Manage Kategori', 'url'=>array('admin')),
);
?>

<?php $form=$this->beginWidget('CActiveForm', array(
)); ?>

<?php echo CHtml::link('Buat Thread
Baru', array('thread/create', 'id'=>$model->id), array('class'=>'btn
success')) ?>

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'thread-grid',
    'dataProvider'=>$allt->search(),
    //'filter'=>$allt,
    'emptyText'=>'Belum ada thread pada kategori ini',
    'summaryText'=>'',
    'columns'=>array(
        array(
            'name'=>'Judul',

```

```

        'type'=>'raw',
        'value'=>'Chtml::link($data-
>judul,array(\'thread/view\',\'id\'=>$data->id))',
    ),
    array(
        'name'=>'Rate',
        'type'=>'raw',
        'value'=>'Threadstar::model()->rate($data->id)."/5"',
    ),
    array(
        'name'=>'By',
        'type'=>'raw',
        'htmlOptions'=>array('style'=>'text-align: center'),
        'value'=>'Chtml::link(User::model()->findByPk($data-
>user_id)->username,array("user/view","id"=>$data->user_id))',
    ),
    array(
        'name'=>'Total Komentar',
        'htmlOptions'=>array('style'=>'text-align: center'),
        'value'=>'Comment::model()-
>countByAttributes(array(\'thread_id\'=>$data->id))',
    ),
    ),
); ?>

<?php $this->endWidget(); ?>

```

Setiap user membuka detail(actionView) pada suatu kategori, maka halaman tersebut akan memperlihatkan semua thread yang terdapat pada kategori tersebut. Selain itu juga akan sebuah link yang akan mengarahkan user pada actionCreate Thread baru dimana pada link itu sudah disisipkan parameter kategori sesuai dengan kategori tempat user membuka thread tersebut. Anda dapat lihat link tersebut pada kode :

```

<?php echo Chtml::link('Buat Thread
Baru',array('thread/create','id'=>$model->id),array('class'=>'btn
success')) ?>

```

Dengan cara seperti itu, setiap thread yang ditambahkan oleh user akan otomatis dikategorikan oleh sistem.

Pada point kelima kita ingin menampilkan daftar user yang paling banyak membuat thread, membuat komentar, daftar thread paling banyak dikomentari, thread terbaru, dan 5 berita terbaru dalam bentuk slide view. Hal pertama yang perlu kita lakukan adalah membuat query yang mengembalikan data-data tersebut:

- a. Mengambil lima nama user yang paling banyak membuat komentar

```
public function mostUC()
{
    $sql4='SELECT count(id), user_id FROM comment GROUP BY
user_id order by count(id) DESC';
    $dataProvider4=new CSqlDataProvider($sql4,array(
        'keyField' => 'user_id',
        'pagination'=>array(
            'pageSize'=>5,
        ),
    ));
    return $dataProvider4;
}
```

- b. Mengambil lima nama user yang paling banyak membuat thread

```
public function mostUT()
{
    $sql3='SELECT count(id), user_id FROM thread GROUP BY
user_id order by count(id) DESC';
    $dataProvider3=new CSqlDataProvider($sql3,array(
        'keyField' => 'user_id',
        'pagination'=>array(
            'pageSize'=>5,
        ),
    ));
    return $dataProvider3;
}
```

- c. Mengambil lima thread yang paling banyak komentar

```
public function topThread()
{
    $sql5='SELECT count(thread_id),thread_id FROM comment
GROUP BY thread_id order by count(thread_id) DESC';
    $dataProvider5=new CSqlDataProvider($sql5,array(
        'keyField' => 'thread_id',
        'pagination'=>array(
```

```

        'pageSize'=>5,
    ),
));

    return $dataProvider5;
}

```

d. Mengambil lima thread terakhir yang telah dibuat

```

public function lastThread()
{
    $sql2='SELECT * FROM thread order by id desc';
    $dataProvider2=new CSqlDataProvider($sql2,array(
        'keyField' => 'id',
        'pagination'=>array(
            'pageSize'=>5,
        ),
    ));

    return $dataProvider2;
}

```

e. Mengambil lima berita terakhir yang telah dibuat

```

public function lastNew()
{
    $sql='SELECT * FROM news order by id desc';
    $dataProvider=new CSqlDataProvider($sql,array(
        'keyField' => 'id',
        'pagination'=>array(
            'pageSize'=>5,
        ),
    ));
    return $dataProvider;
}

```

Kelima fungsi di atas saya letakkan pada model Thread, tapi kalo anda mau meletakkannya pada model lain, itu sah sah saja kok.. Tujuan saya letakkan di model thread biar saya gampang mengingatnya saja..

Setelah membuat semua fungsi yang mengembalikan nilai-nilai yang kita inginkan, kita harus definisikan juga nilai-nilai tersebut pada suatu action untuk kemudian ditampilkan kepada user. Silahkan buka actionIndex pada SiteController.php. Buat actionIndex tersebut mendefinisikan nilai yang ingin kita ambil, lalu kirim nilainya ke view.

```

public function actionIndex ()
{
    $dataProvider=Thread::model()->lastNew();
    $dataProvider2=Thread::model()->lastThread();
    $dataProvider3=Thread::model()->mostUT();
    $dataProvider4=Thread::model()->mostUC();
    $dataProvider5=Thread::model()->topThread();

    $this->render('index', array(
        'dataProvider'=>$dataProvider,
        'dataProvider2'=>$dataProvider2,
        'dataProvider3'=>$dataProvider3,
        'dataProvider4'=>$dataProvider4,
        'dataProvider5'=>$dataProvider5,
    ));
}

```

Setelah itu atur tampilan pada halaman index :

```

<?php CHtml::encode(Yii::app()->name); ?>
<table>
    <tr>
        <th>
            <div class="alert-message info">
                News
            </div>
        <?php
            $terserah=array();
            foreach($dataProvider->getData() as $i=>$ii)
            {
                $situ=Chtml::link($ii['judul'], array('news/view', 'id'=>$ii['id']));
                $awal='a/./news/'. $ii['id']. $ii['foto'];
                $terserah[$i]=array($awal, $situ);
            }
            $this->widget('application.extensions.s3slider.S3Slider',
                array(
                    'images' => $terserah,
                    'width' => '590',
                    'height' => '375',
                )
            );
        ?>
        </th>
        <th style='vertical-align:top;'>
            <?php
            $this->widget('zii.widgets.grid.CGridView', array(
                'id'=>'news-grid',
                'dataProvider'=>$dataProvider3,
                'summaryText'=>'',
                'columns'=>array(
                    array(
                        'name'=>'Most Thread',

```

```

        'type'=>'raw',
        'headerHtmlOptions'=>array('style'=>'text-align:
left','colspan'=>'2'),
        'htmlOptions'=>array('width'=>'190px'),
        'value'=>'Chtml::link(User::model()-
>findByPk($data["user_id"])->username,
        array("user/view","id"=>$data["user_id"]))',
    ),
    array(
        'headerHtmlOptions'=>array('style'=>'display:none'),
        'htmlOptions'=>array('style'=>'text-align:
center','width'=>'70px'),
        'type'=>'raw',
        'value'=>'$data["count(id)"]',
    ),
),
);
echo '<hr/>';

$this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'news-grid',
    'dataProvider'=>$dataProvider4,
    'summaryText'=>'',
    'columns'=>array(
        array(
            'type'=>'raw',
            'headerHtmlOptions'=>array('style'=>'display:none'),
            'htmlOptions'=>array('width'=>'190px'),
            'value'=>'Chtml::link(User::model()-
>findByPk($data["user_id"])->username,
            array("user/view","id"=>$data["user_id"]))',
        ),
        array(
            'name'=>'Most Comment',

            'headerHtmlOptions'=>array('colspan'=>'2','style'=>'text-
align:left'),
            'htmlOptions'=>array('style'=>'text-align:
center','width'=>'70px'),
            'type'=>'raw',
            'value'=>'$data["count(id)"]',
        ),
    ),
);
?>
</th>
</tr>
</table>

<table>
<tr>
<th>
<?php
$this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'news-grid',
    'dataProvider'=>$dataProvider5,
    'summaryText'=>'',
    'columns'=>array(
        array(

```

```

        'name'=>'Top Thread',
        'type'=>'raw',
        'value'=>'Chtml::link(Thread::model()-
>findByPk($data["thread_id"])->judul."
(".$data["count(thread_id)"].")',
        array("thread/view","id"=>$data["thread_id"]))',
    ),
),
));
?>
</th>
<th>
<?php
        $this->widget('zii.widgets.grid.CGridView', array(
        'id'=>'news-grid',
        'dataProvider'=>$dataProvider2,
        'summaryText'=>',
        'columns'=>array(
            array(
                'name'=>'Last Thread',
                'type'=>'raw',

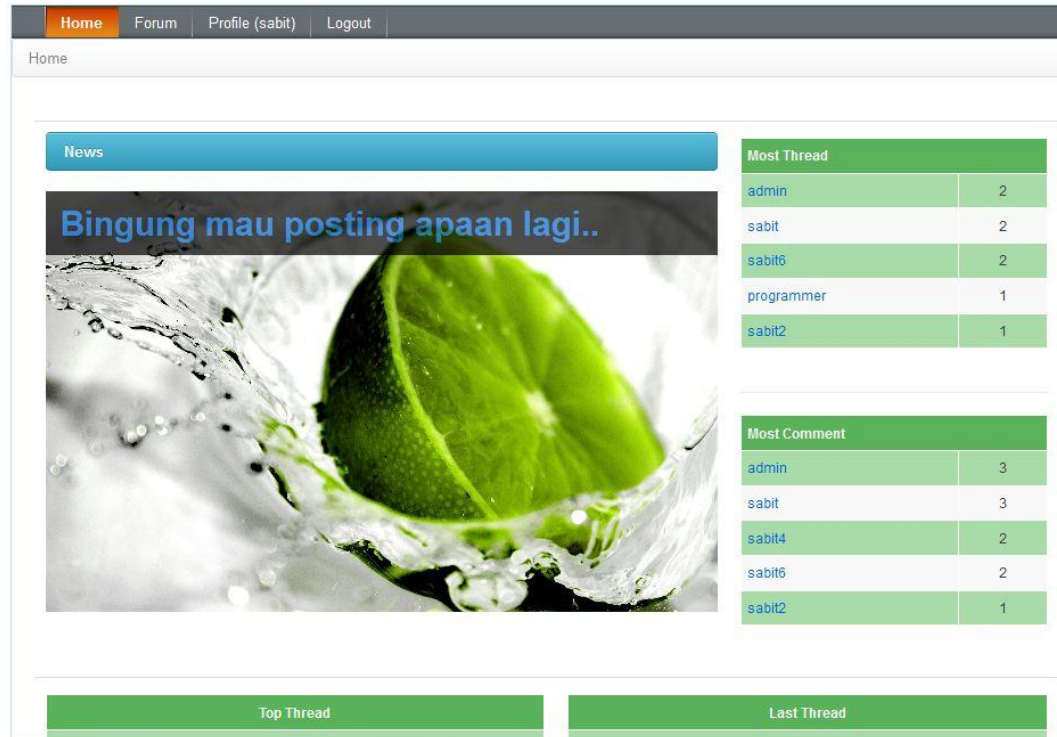
                'value'=>'Chtml::link($data["judul"],array("thread/view","id"=>$da
ta["id"]))',
            ),
        ),
    ));
?>
</th>
</tr>
</table>

```

Setelah di atur dengan CSS yang saya punya, akan tampil halaman seperti

berikut :





Gambar 31. Halaman Utama Aplikasi

Pada point keenam, kita ingin user terdiri dari tiga level.. Anda tinggal menginput data pada tabel level yang terdiri dari “Admin”, “Moderator”, dan “Member”. Pada point ke sembilan, kita ingin memberikan hak kepada user yang sudah melakukan login untuk memberikan komentar pada suatu thread. Setiap user memberikan komentar, kita harus mengetahui thread mana yang dikomentari oleh user. Maka kita harus menambahkan parameter identitas thread yang dikomentari pada `actionCreate` komentar :

```

public function actionCreate($id)
{
    $model=new Comment;

    if(isset($_POST['Comment']))
    {
        $model->attributes=$_POST['Comment'];
    }
}

```

```

        $model->user_id=Yii::app()->user->id;
        $model->thread_id=$id;
        if($model->save())
            $this->redirect(array('thread/view','id'=>$model->thread_id));
    }

    $this->render('create',array(
        'model'=>$model,
    ));
}

```

Hampir mirip dengan metode input pada “Thread” bukan?? Setelah itu pada tampilan actionView “Thread”, kita harus menyediakan suatu link/tautan dimana jika user memilihnya, maka user akan membuat komentar baru dan parameter thread akan secara otomatis dikirimkan pada action tersebut :

```

<?php
$skate=Kategori::model()->findByPk($model->kategori_id);

$this->breadcrumbs=array(
    $skate->kategori=>array('kategori/view','id'=>$skate->id),
    $model->judul,
);

$this->menu=array(
    array('label'=>'List Thread', 'url'=>array('index')),
    array('label'=>'Create Thread', 'url'=>array('create')),
    array('label'=>'Update Thread', 'url'=>array('update',
'id'=>$model->id)),
    array('label'=>'Delete Thread', 'url'=> '#',
'linkOptions'=>array('submit'=>array('delete','id'=>$model->id),
'confirm'=>'Are you sure you want to delete this item?')),
    array('label'=>'Manage Thread', 'url'=>array('admin')),
);
?>
<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'thread-form',
    'enableAjaxValidation'=>false,
)); ?>
<?php
    $this->widget('CStarRating',array(
        'name'=>'rating',
        'minRating'=>1,
        'maxRating'=>5,
        'starCount'=>5,
        'value'=>$rate,
    ));
    echo CHtml::submitButton(" Vote ");
?>

```

```

<?php $this->endWidget(); ?>

<table class="alert-message success">
  <tr>
    <th colspan="2">
      <h1><?php echo $model->judul; ?></h1>
    </th>
  </tr>
  <tr>
    <td width="15%">
      <div class="tengah">
        <?php
          $TS=User::model()->findByPk($model->user_id);
          echo CHtml::link($TS->username,
            array('user/view','id'=>$model->user_id));
        ?>
      </div>
      <br/>
      <div class="tengah">
        <?php
          $link='a/../../avatar/'. $TS->avatar;
          echo CHtml::link(CHtml::image($link, 'DORE',
            array("width"=>80)), '#');
        ?>
      </div>
    </td>
    <td><?php echo $model->isi; ?></td>
  </tr>
</table>
<?php
  echo CHtml::link('Tambahkan Komentar',
    array('comment/create','id'=>$model->id), array('class'=>'btn
success'));
?>

<?php $form=$this->beginWidget('CActiveForm', array(
)); ?>

<?php $this->widget('zii.widgets.CListView', array(
  'dataProvider'=>$allt->search(),
  'itemView'=>'_viv',
  'emptyText'=>'Belum ada komentar pada thread ini'
)); ?>
<?php $this->endWidget(); ?>

```

Disana terdapat ClistView yang mengarah pada halaman “\_viv.php”, jadi kita harus buat sebuah file bernama \_viv.php dan masukkan kode berikut :

```

<div class="alert-message block-message info">
  <table>
    <tr>
      <td width="15%" class="tengah">

```

```

        <?php
            $haha=User::model()->findByPk($data->user_id);
            echo CHtml::image('a/../avatar/'.$haha-
>avatar,'DORE', array("width"=>100)).'<br/>';
            echo $haha->username.'<br/>';
            echo $data->tanggalPost;
        ?>
    </td>
    <td>
        <?php echo $data->judul;?><hr/>
        <?php echo $data->isi; ?>
    </td>
</tr>
</table>
</div>

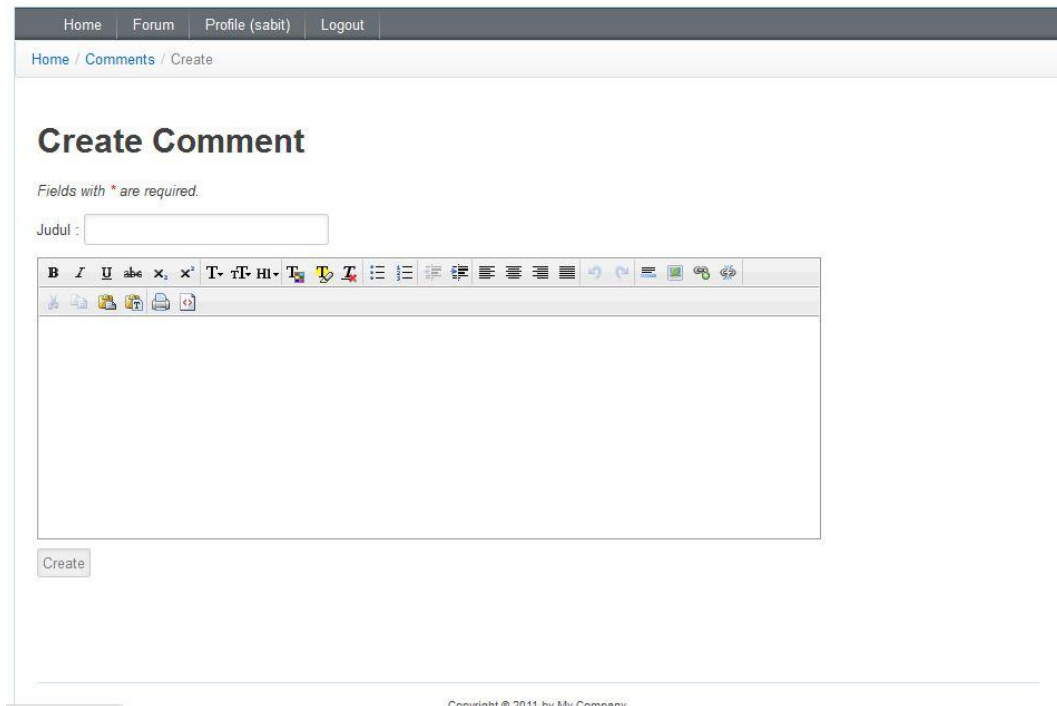
```

Jika sudah, akan tampil halaman seperti berikut :



Gambar 32. Tampilan Halaman Suatu Thread

Jika kita memilih “Tambahkan Komentar”, maka akan tampil halaman seperti berikut :



Gambar 33. Tampilan Halaman Membuat Komentar Baru

Anda juga melihat di halaman view thread tadi terdapat tombol untuk memberikan “rate” pada suatu thread bukan? Yah, tentu itu akan dapat berfungsi sebagaimana mestinya. Tapi modifikasi kode pada actionView di Thread controller menjadi seperti berikut :

```

public function actionView($id)
{
    $allt=new Comment('search');
    $allt->unsetAttributes();
    $allt->thread_id=$id;
    if(isset($_POST['rating']))
    {
        $udah=Threadstar::model()->countByAttributes(
            array('user_id'=>Yii::app()->user-
>id,'thread_id'=>$id));
        if($udah==0)
        {
            $itu=new Threadstar;
            $itu->nilai=$_POST['rating'];
            $itu->user_id=Yii::app()->user->id;
            $itu->thread_id=$id;
            $itu->save();
        }
    }
}

```

```

    }
}

$this->render('view', array(
    'model'=>$this->loadModel($id),
    'allt'=>$allt,
    'rate'=>Threadstar::model()->rate($id),
));
}

```

Untuk memastikan fungsi “vote” sudah berjalan, silahkan anda coba berikan vote pada suatu thread..

Pada point ke sepuluh, kita ingin user yang telah login dapat memberikan penilaian kepada user lain. Mari kita buat fitur tersebut... Pertama Sediakan link untuk mengarah pada action memberikan reputasi pada halaman profile user. Pada user/view.php, masukkan kode berikut :

```

<?php
$this->breadcrumbs=array(
    'Users'=>array('index'),
    $model->username,
);

$this->menu=array(
    array('label'=>'List User', 'url'=>array('index')),
    array('label'=>'Create User', 'url'=>array('create')),
    array('label'=>'Update User', 'url'=>array('update',
'id'=>$model->id)),
    array('label'=>'Delete User', 'url'=>'#',
'linkOptions'=>array('submit'=>array('delete', 'id'=>$model-
>id), 'confirm'=>'Are you sure you want to delete this item?')),
    array('label'=>'Manage User', 'url'=>array('admin')),
);
?>

<?php $form=$this->beginWidget('CActiveForm', array(
)); ?>

<h1>Profile #<?php echo $model->username; ?></h1>
<?php echo CHtml::link('Berikan
Reputasi', array('raputation/create', 'id'=>$model->id),
    array('class'=>'btn success')) ?>
<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'username',
        'email',
        'joinDate',

```

```

        'level_id',
        array(
            'label'=>'Avatar',
            'type'=>'raw',
            'value'=>Chtml::image('a/../../avatar/'.$model-
>avatar,'DORE', array("width"=>100)),
        ),
    ),
); ?>
Reputasi :
<?php echo Raputation::model()->reput($model->id) ?>

<?php $this->endWidget(); ?>

```

Ga terlalu banyak yang berubah dari kode hasil generate Yii kok.. kita hanya menambahkan link dan sedikit memodifikasi CdetailView..

Setelah itu, mari kita buat action yang mengizinkan memberikan reputasi kepada seorang user, tepatnya di controller reputation actionCreate :

```

public function actionCreate($id)
{
    $model=new Raputation;

    if(isset($_POST['Raputation']))
    {
        $model->attributes=$_POST['Raputation'];
        $model->tanggal=date('Y-m-d');
        $model->pemberi_id=Yii::app()->user->id;
        $model->penerima_id=$id;
        if($model->save())
            $this->redirect(array('user/view','id'=>$id));
    }

    $this->render('create',array(
        'model'=>$model,
        'id'=>$id,
    ));
}

```

Terdapat sebuah parameter pada action tersebut yaitu \$id, \$id sendiri adalah parameter yang digunakan untuk mengetahui siapa user yang akan diberikan reputasi. Action create reputasi akan mengarah ke halaman create :

```

<?php
$this->breadcrumbs=array (
    'Raputations'=>array('index'),
    'Create',
);

$this->menu=array (
    array('label'=>'List Raputation', 'url'=>array('index')),
    array('label'=>'Manage Raputation', 'url'=>array('admin')),
);
?>

<h1>Create Raputation</h1>

<?php echo $this->renderPartial('_form',
array('model'=>$model, 'id'=>$id)); ?>

```

Dan halaman create akan mengarah ke halaman \_form :

```

<div class="form">

<?php $form=$this-
>beginWidget('ext.bootstrap.widgets.BootActiveForm', array (
    'id'=>'reputation-form',
    'enableAjaxValidation'=>false,
)); ?>

    <?php echo $form->errorSummary($model); ?>
    <b>
    <?php echo "Berikan Reputasi Untuk :<h1>".User::model()-
    >findByPk($id)->username."</h1>"; ?>
    </b>

    <div class="row">
        <?php echo $form->dropDownList($model, 'jenis',
            array(-1=>'Bad', 1=>'Good'), array()); ?>
        <?php echo $form->error($model, 'jenis'); ?>
    </div>

    <div class="row buttons">
        <?php echo CHtml::submitButton($model->isNewRecord ?
'Create' : 'Save'); ?>
    </div>

<?php $this->endWidget(); ?>

</div><!-- form -->

```

Terjadi sedikit modifikasi pada halaman form, silahkan anda sesuaikan dengan code anda..



Okeh, semua fitur2 sudah kita buat.. Sekarang saatnya anda mencoba aplikasi anda.. Dan dengan selesainya pembangunan aplikasi forum tersebut, berakhir juga pembahasan dari ebook ini..

Yah saya harap ebook ini dapat membantu teman-teman semua untuk menggali ilmu teman-teman dalam Yii Framework. Anda dapat mendapatkan source code dari aplikasi forum yang kita bahas di atas disini : <http://freebookyii.codeplex.com/>.

## DAFTAR PUSTAKA

- <http://www.yiiframework.com/>
- <http://sabitlabscode.wordpress.com/>
- <http://www.kaskus.us>

## TENTANG PENULIS



### SABIT HURAIRA

Penulis dilahirkan di Palembang pada tanggal 23 Agustus 1989 dari pasangan Rusydi Hasir Syukri (Alm) dan Fariko. Penulis merupakan anak kedua dari tiga bersaudara. Pada tahun 1995 penulis masuk pendidikan dasar di SD Kartika II-2 Kota Palembang dan lulus pada tahun 2001. Penulis melanjutkan pendidikan di SMP Negeri 9 Kota Palembang dan lulus pada tahun 2004. Tahun 2007, penulis berhasil menyelesaikan pendidikan di SMA Negeri 6 Kota Palembang. Pada tahun yang sama, penulis berkesempatan untuk melanjutkan pendidikan ke Sekolah Tinggi Ilmu Statistik (STIS) dengan jurusan Komputasi Statistik. Saat ini penulis bekerja di Badan Pusat Statistik dan juga aktif bekerja sebagai “pengusaha code” freelance.

Awal mula dunia programming dengan bahasa C/C++, kemudian mulai mendalami C#, Java, ASP.NET MVC, hingga PHP dengan Yii Framework. Hingga saat ini

penulis aktif menulis beberapa artikel mengenai programming pada blognya <http://sabitlabscodewordpress.com/>.

Penulis dapat dihubungi melalui email [sabitzhabit@yahoo.com](mailto:sabitzhabit@yahoo.com) dan [sabitzhabit@gmail.com](mailto:sabitzhabit@gmail.com).

*Penulis mempersembahkan ebook ini untuk semua rekan-rekan komunitas Yii Framework...*



## **Masa Depan Ebook Ini**

Penulis sangat menyadari bahwa ebook ini masih jauh dari sempurna, pun materi yang disampaikan tidaklah mencakup seluk beluk dari Yii Framework. Selain itu ilmu pengetahuan yang dimiliki penulis yang terbatas juga membuat tidak semua materi dapat tercakup dalam ebook ini. Untuk itu ke depannya ebook ini akan diperbaharui dimana setiap pembaharuan akan ditambahkan materi-materi terbaru, contoh-contoh code, dan lain-lain terkait Yii Framework.

## **Anda berminat bergabung untuk menulis ebook ini ?**

Untuk ke depan ebook ini akan terus disempurnakan baik itu perbaikan-perbaikan maupun penambahan materi baru. Bagi anda yang berminat bergabung dengan penulis untuk melanjutkan pengembangan ebook ini dapat langsung menghubungi penulis di [sabitzhabit@yahoo.com](mailto:sabitzhabit@yahoo.com) atau [sabitzhabit@gmail.com](mailto:sabitzhabit@gmail.com). Tetapi perlu saya tekankan project ini dilakukan bukan dengan tujuan komersil, tetapi sukarela..  
Yah, itung-itung amal jariyah lah untuk kita semua ☺