

PENGEMBANGAN SITUS KOMUNITAS DENGAN *FRAMEWORK* CAKEPHP

Hendrik

Laboratorium Sistem Informasi dan Rekayasa Perangkat Lunak (SIRKEL),
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia
Jl. Kaliurang Km. 14 Yogyakarta 55501; Telp. (0274) 895287 ext. 122, Faks. (0274) 895007 ext. 148
E-mail: javanehese2002@yahoo.com

ABSTRAKSI

Perkembangan teknologi internet dan web telah memunculkan cabang ilmu baru yakni *web engineering* (*rekayasa web*). Guna memperdalam pemahaman mahasiswa terhadap matakuliah *rekayasa web* yang diampu penulis, maka penulis mengembangkan sebuah situs komunitas yang dapat digunakan mahasiswa sebagai wadah repository proyek-proyek web yang mereka kerjakan dan media diseminasi pengetahuan terkait dengan proyek pengembangan web. Pengembangan situs ini memanfaatkan *framework CakePHP* yang diharapkan dapat digunakan untuk mengembangkan aplikasi berbasis web secara cepat. Selain itu, karena *framework* ini dikembangkan oleh komunitas *open-source* maka dimungkinkan pulabagi kita untuk berpartisipasi dalam pengembangannya.

Kata kunci: situs komunitas, *framework*, *CakePHP*, *PHP*, *rekayasa web*

1. PENDAHULUAN

Saat ini situs web atau pun aplikasi berbasis web telah menjadi semakin populer seiring berkembangnya teknologi internet. Hal ini mengakibatkan munculnya cabang ilmu baru berkaitan dengan proses pengembangan web. Kalau dulu dikenal istilah *software engineering* untuk menyebut cabang ilmu yang mempelajari proses perancangan dan pengembangan suatu aplikasi atau perangkat lunak, maka saat ini muncul *web engineering* (*rekayasa web*) yang menjadi cabang ilmu terpisah yang mempelajari proses perancangan dan pengembangan situs web atau pun aplikasi berbasis web.

Untuk lebih memperdalam pemahaman mahasiswa serta meningkatkan kemampuan dan pengalaman praktis mereka dalam mengelola sebuah proyek web, maka dalam matakuliah *rekayasa web* yang diampu oleh penulis diberikan sebuah *final project* untuk mengembangkan situs web yang menekankan pada pemanfaatan teknologi terkini seperti penggunaan *framework*. Hal ini dimaksudkan agar mahasiswa lebih memahami proyek pengembangan web dengan paradigma berbasis objek. *Framework* menyediakan suatu lingkungan pengembangan berbasis objek yang memungkinkan pemanfaatan kode-kode yang sudah ada sehingga akan mempercepat proses pengembangan. Selanjutnya penulis berpendapat diperlukan suatu media yang dapat dijadikan sebagai *repository* bagi proyek-proyek yang dikerjakan mahasiswa dan diseminasi pengetahuan berkaitan dengan proyek pengembangan web. Media yang dikembangkan tersebut berupa sebuah situs komunitas yang di dalamnya dimungkinkan bagi mahasiswa untuk memasukkan proyek-proyek yang mereka kerjakan sehingga mahasiswa lainnya dapat memberikan penilaian maupun komentar terhadap hasil karya rekan-rekannya. Selain itu, mahasiswa juga dapat membagi pengalaman maupun pengetahuan yang

mereka miliki dengan mengirimkan artikel-artikel terkait proses pengembangan web beserta isunya.

Adapun penelitian ini bertujuan untuk mengembangkan sebuah situs komunitas yang memanfaatkan sebuah *framework* guna memberikan contoh nyata bagi mahasiswa berkaitan dengan pemanfaatan *framework* dalam mengembangkan situs web secara cepat. Pemilihan *CakePHP* sebagai *framework* pengembangan situs komunitas ini karena *CakePHP* merupakan salah satu *framework* yang dikembangkan oleh komunitas *open-source* yang menurut penulis cukup mudah dipelajari dan memanfaatkan konsep MVC yang memisahkan dengan jelas antara data dan presentasi sehingga dapat mempercepat proses pengembangan dan mempermudah dalam pengelolaan.

2. LANDASAN TEORI

CakePHP atau disebut *Cake* merupakan *framework* Pengembangan Cepat untuk *PHP* yang gratis dan dikembangkan oleh komunitas *open-source*. Adapun *framework* sendiri dapat diartikan sebagai suatu struktur pustaka-pustaka, kelas-kelas dan infrastruktur *run-time* yang dapat digunakan oleh programmer untuk mengembangkan aplikasi web secara cepat. Tujuan penggunaan *framework* adalah untuk mempermudah pengembang web mengembangkan aplikasi web yang *robust* secara cepat tanpa kehilangan fleksibilitas.

CakePHP awalnya dikembangkan oleh Michal Tatarynowicz yang menulis versi minimal dari *framework* ini dan kemudian mempublikasikan *framework* ini di bawah lisensi MIT dan menyebarkannya ke komunitas pengembang yang saat ini mengelola pengembangan *CakePHP* ini. *Framework* ini dikembangkan dengan menggunakan beberapa prinsip dari *Ruby on Rails*, salah satu *framework* yang dikembangkan untuk bahasa *Ruby*. Beberapa pola desain yang dikenal secara umum dalam pengembangan web seperti

ActiveRecord, Association Data Mapping, Front Controller, dan MVC (*Model-View-Controller*) digunakan dalam framework ini.

Hal pertama dalam memahami Cake adalah salah satu tujuan framework ini yakni menghindari pengubahan file konfigurasi yang panjang dan kompleks untuk menjalankan lingkungan pengembangan. Pendekatan ini menggunakan kaidah-kaidah tertentu pada konfigurasi. Hal ini nampak sebagai suatu hal yang membatasi, namun pada kenyataannya justru kaidah-kaidah tersebut membuat segala sesuatunya menjadi lebih sederhana. Cake menggunakan beberapa aturan-aturan dasar agar dapat dijalankan dengan baik. Untuk dapat memahami dengan baik aturan-aturan dasarnya, perlu dipahami terlebih dulu konsep MVC dan struktur direktori Cake yang menggambarkan kerangka dari setiap aplikasi CakePHP.

2.1 Pola MVC (*Model-View-Controller*)

- Model

Model mewakili bagian yang sangat penting dari informasi dan isi aplikasi web. Model dapat dibayangkan sebagai suatu objek yang dapat mengumpulkan informasi dan isi dari aplikasi web dari sumber tertentu, misal basisdata.

- View

View hanya bertanggungjawab menampilkan informasi yang dikumpulkan oleh model. View berfungsi sebagai presentasi dari aplikasi web. Pada semua framework aplikasi web berbasis MVC, hanya view yang berisi kode (X)HTML sehingga dapat diubah kapan pun tanpa perlu mengubah kode yang berkaitan dengan logika bisnis dari aplikasi yang dikembangkan.

- Controller

Controller dapat dianggap sebagai “otak” dari pola MVC. Umumnya controller berisi logika bisnis dari aplikasi web. Controller bertanggungjawab memerintahkan model untuk mengumpulkan informasi dan meneruskan informasi yang diperoleh ke view untuk ditampilkan.

Dengan pola MVC ini nampaknya aplikasi menjadi lebih rumit, namun kelebihanannya adalah dengan terjadinya pemisahan antara logika bisnis dan presentasi mempermudah dalam pengelolaan aplikasi web terutama jika aplikasi yang dikembangkan telah berkembang semakin besar.

2.2 Struktur direktori CakePHP

Berikut adalah struktur direktori dari framework CakePHP.

```
/
---app/
-----config/
-----controllers/
-----components/
-----models/
```

```
-----plugins/
-----views/
-----elements/
-----errors/
-----helpers/
-----layouts/
-----pages/
-----webroot/
-----css/
-----files/
-----img/
-----js/
---cake/
-----config/
-----inflections/
-----docs/
-----libs/
-----controller/
-----generator/
-----model/
-----view/
-----scripts/
---tmp/
-----cache/
-----distro/
-----logs/
-----tests/
---vendors/
```

Direktori/cake merupakan direktori yang berisi pustaka-pustaka internal yang umumnya tidak perlu dilakukan perubahan. Direktori/tmp digunakan untuk menyimpan file-file temporer dan direktori/vendors digunakan untuk menempatkan script-script dan pustaka-pustakan pihak ketiga yang akan digunakan dalam aplikasi. Adapun direktori/app merupakan direktori utama yang berisi 95% dari aplikasi yang dibuat.

Penjelasan detail untuk direktori /app ini adalah sebagai berikut:

/config/

Direktori ini berisi file-file konfigurasi untuk dapat menjalankan sistem. Hanya ada 5 file konfigurasi yang ada direktori ini:

- acl.ini.php: file ini harus diubah jika menggunakan sistem ACL (*Access Control List*) untuk aplikasi yang dikembangkan. File ini berisi konfigurasi ijin-ijin akses bagi pengguna, kelompok pengguna dan hak-hak mereka. Umumnya hanya digunakan untuk situs kecil dengan pengguna yang telah dikenal dengan baik.
- core.php: file ini dapat diubah untuk mengganti beberapa opsi default, seperti tingkat pesan kesalahan dan peringatan yang akan ditampilkan.
- database.php.default: file ini harus diubah menjadi database.php jika aplikasi yang dikembangkan menggunakan basisdata. File ini berisi konfigurasi jenis basisdata yang digunakan (mysql, postgres, sqlite, maupun basisdata lainnya yang didukung oleh pustaka AdoDB), nama user, *password*, dan nama basisdata.
- paths.php: file ini memberitahukan kepada Cake lokasi file-file CSS, *controller*, javascript, dan lain-lain. Umumnya file ini

tidak perlu diubah jika menggunakan struktur direktori standar Cake.

- e. `routes.php`: berisi konfigurasi “sistem *routing*” untuk URL yang *user-friendly*, sehingga URL aplikasi yang dikembangkan nantinya akan nampak sebagai berikut:
`http://nama_situs/nama_kontroler/nama_aksi/vent/aksi/parameter`

`/controllers/`

Direktori ini berisi file-file kontroler yang digunakan untuk mengelola logika bagian tertentu dari aplikasi yang dikembangkan. Sebagai contoh jika situs yang dikembangkan mengelola koleksi video, maka akan terdapat `VideoController` dan `RentalController` untuk mengelola bagian koleksi video dan peminjaman. Cake membuat aturan bahwa nama kontroler berbentuk jamak. Kontroler ini merupakan kelas-kelas perluasan dari kelas `AppController`. Suatu kelas kontroler berisi aksi-aksi yakni fungsi yang digunakan untuk menampilkan *views*.

Berikut adalah contoh salah satu kelas kontroler

```
class VideosController extends AppController
{
    function view($id)
    {
        Aksi yang dilakukan
    }

    function rent($customer_id, $video_id)
    {
        Aksi yang dilakukan
    }

    function search($query)
    {
        Aksi yang dilakukan
    }
}
```

Untuk mengakses aksi-aksi tersebut digunakan URL seperti berikut:

`http://www.namasitus.com/videos/view/253`
`http://www.namasitus.com/videos/rent/5124/0-235253`
`http://www.namasitus.com/videos/search/hudsucker+proxy`

`/models/`

Direktori ini berisi file-file model. Model merupakan suatu objek yang digunakan untuk mengakses basisdata. Pada Cake setiap tabel dalam basisdata yang terlibat dalam aplikasi harus dibuatkan file modelnya. Ketentuan dalam Cake bahwa nama tabel berbentuk jamak (dalam bahasa Inggris) sedangkan nama model tunggal. Hal ini nampaknya agak memberatkan, namun seperti disebutkan sebelumnya bahwa tujuan ketentuan tersebut agar pengembangan aplikasi menjadi lebih sederhana.

Berikut adalah contoh isi dari sebuah file model.

```
<?php
class Article extends AppModel
{
    var $name = 'Article';
}
?>
```

Terlihat pada isi dile tersebut suatu kelas dengan nama `Article` yang merupakan perluasan dari kelas `AppModel` dan berhubungan dengan table `Articles`. Selain menunjukkan nama table, dalam file model juga dapat didefinisikan aturan-aturan validasi, informasi asosiasi, dan metode-metode tertentu untuk tabel yang menggunakannya.

`/plugins/`

Direktori ini berisi *script-script* tambahan yang dikembangkan oleh pengembang. Berbeda dengan file-file yang diletakkan pada direktori `/vendor`, file-file pada direktori ini merupakan suatu aplikasi yang dikembangkan secara khusus untuk berjalan di lingkungan Cake.

`/views/`

Direktori ini digunakan untuk meletakkan file-file tampilan dari aplikasi yang dibuat (*views*). Suatu file *views* merupakan file yang berisi tag-tag HTML dan beberapa tag PHP di dalamnya yang digunakan untuk menampilkan informasi yang diperoleh oleh suatu model dan kontroler.

Beberapa ketentuan berkaitan dengan *view*:

- View* diberi nama sesuai dengan aksi yang ada pada suatu kontroler sehingga memungkinkan kontroler merujuk pada *view* tertentu secara otomatis. Misal sebuah kontroler `Articles` berisi aksi-aksi: `index()`, `view()`, `add()`, `edit()`, `delete()`, maka nama-nama file *view* yang dibuat sesuai dengan nama aksi-aksi tersebut. Beberapa *view* yang sama dapat digunakan oleh sebuah kontroler namun penggunaan harus diset secara manual atau dengan menggunakan suatu layout.
- File-file *view* tersebut diletakkan dalam sebuah subfolder yang diberi nama sesuai dengan kontrolernya. Dalam kasus di atas file-file *view* tersebut diletakkan di subdirektori bernama `Articles`.
- Semua file *view* berekstensi `.thtml`

Untuk template tampilan, digunakan file *view* yang diletakkan pada direktori `/layouts`. File `default.thtml` merupakan template global untuk aplikasi yang dikembangkan. Judul halaman dan file *view* tertentu akan diambil secara otomatis dengan menggunakan variabel `$title_for_layout` dan `$content_for_layout`.

`/webroot/`

Direktori ini berisi subdirektori-subdirektori seperti `css` untuk menyimpan file-file `css`, `javascript` untuk menyimpan file-file `javascript`, dan `img` untuk menyimpan file-file gambar yang digunakan pada aplikasi web.

2.3 Fitur-fitur

Cake memiliki beberapa fitur yang membantu dalam proses pengembangan aplikasi web yang cepat:

- Scaffolding:** merupakan suatu fitur yang memungkinkan untuk menguji aplikasi tanpa perlu menuliskan kode CRUD (*Create, Read, Update, Delete*). Hal ini dimungkinkan dengan cara menset beberapa variabel pada tempat yang tepat, dan kemudian Cake akan *generate* mekanisme dasar untuk proses menambah, mengubah, menampilkan dan menghapus record pada basisdata sepanjang berkaitan dengan file-file view yang berasosiasi.
- Bake:** sekumpulan skrip yang digunakan untuk membuat dasar dari suatu aplikasi web.
- ACL (*Access Control List*):** fitur yang memungkinkan aplikasi web diperluas dan digunakan untuk membatasi area-area khusus dari aplikasi Cake terhadap pengguna atau kelompok pengguna tertentu.
- Kompatibel dengan PHP4 dan PHP5.**

Secara umum proses pengembangan suatu aplikasi web dengan menggunakan Cake dapat dilakukan melalui beberapa langkah berikut:

- Membuat basisdata dengan tabel-tabelnya. Tabel-tabel tersebut memiliki beberapa field tambahan, yakni : id yang merupakan kunci primer *auto_increment*, *created* dan *modified* dengan tipe *datetime*.
- Mengkonfigurasi file *database.php* pada direktori */config* agar Cake dapat mengakses basisdata.
- Membuat kelas-kelas model sesuai dengan tabel-tabel yang terdapat dalam basisdata.
- Membuat kelas kontroler dan metode-metode yang diperlukan untuk berinteraksi dengan file *view*.
- Membuat suatu *view* yang merupakan file php dengan ekstensi *.html* yang berisi tag-tag HTML dan sedikit tag PHP.

3. PENGEMBANGAN SITUS KOMUNITAS

3.1 Perancangan

Pada tahapan ini dilakukan perancangan terhadap situs komunitas yang akan dikembangkan. Situs komunitas ini ditujukan bagi mahasiswa yang mengambil mata kuliah rekayasa web. Pada situs ini mahasiswa yang mengambil matakuliah rekayasa web mendaftarkan diri menjadi anggota komunitas. Dua hingga tiga orang mahasiswa terdaftar dalam satu kelompok yang mengerjakan sebuah proyek untuk tugas akhir. Setelah mereka terdaftar, ketua kelompok akan memiliki kewenangan sebagai seorang manajer. Adapun hak dari **manajer** adalah membuat kelompok dan menambahkan atau mengubah link alamat situs dari proyek yang mereka kerjakan. Sedangkan anggota lainnya yang memiliki hak sebagai **user** mempunyai hak untuk mengirimkan artikel

berkaitan dengan proses pengembangan suatu situs web. Topik artikel bisa berupa desain, implementasi, pengembangan, pengujian, dan lain sebagainya. Adapun dosen berperan sebagai **admin** yang mengelola situs komunitas ini.

Dari analisis pendahuluan didapatkan diagram *use-case* sebagaimana terlihat pada Gambar 1. Adapun diagram kelas dari hasil analisis *use-case* terlihat pada Gambar 2.

3.2 Implementasi

Pada tahap ini dilakukan proses pengembangan sistem. Beberapa perangkat lunak yang dibutuhkan dalam proses pengembangan ini adalah:

- Apache2triad sebagai perangkat lunak AMP (Apache/2.0.49 sebagai web server, MySQL 4-0-18 sebagai server basisdata, dan PHP 4.3.6 sebagai *scripting language*)
- Editplus sebagai editor pengembangan
- Browser internet : Mozilla Firefox dan Internet Explorer

Berikut adalah beberapa hasil implementasi yang dilakukan:

a. Model: *article.php*

```
<?php
/**
 * kelas model Article
 */

class Article extends AppModel {
    var $name = 'Article';
    var $hasAndBelongsToMany = 'User';

    var $validate = array(
        'title'=>VALID_NOT_EMPTY,
        'content'=>VALID_NOT_EMPTY
    );
}
?>
```

b. Controller: *articles_controller.php*

```
<?
/**
 * kelas kontroler Articles
 */

Class ArticlesController extends
AppController {

    var $name='Articles';

    function index()
    {
        $this->set('data', $this->Article-
>findAll());
        $content_for_right = $this-
>requestAction('users/welcome', array
('return'));
        $content_for_right .= $this-
>requestAction('users/menu', array('r
eturn'));
        $this-
>set('content_for_right', $content_fo
r_right);
    }

    function view($id)
    {
        $this->Article->setId($id);
    }
}
```

```

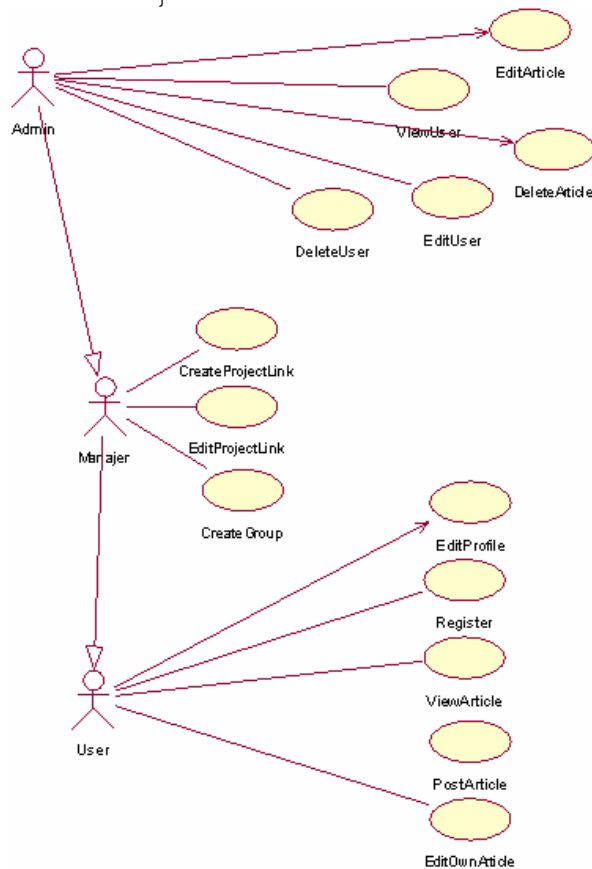
        $this->set('data', $this->
>Article->read());
    }
    function add()
    {
        if (empty($this->
>params['data']))
        {
            $this->render();
        }
        else
        {
            if ($this->Article->
>save($this->params['data']))
            {
                $this->
>flash('Artikel anda telah
tersimpan.', '/articles');
            }
            else
            {
                $this->set('data',
$this->params['data']);
                $this->
>validateErrors($this->Article);
                $this->render();
            }
        }
    }
    function delete($id)
    {
        if ($this->Article->
>del($id))
        {
            $this->flash('Artikel
dengan id: '.$id.' telah dihapus.',
'/articles');
        }
    }
}

```

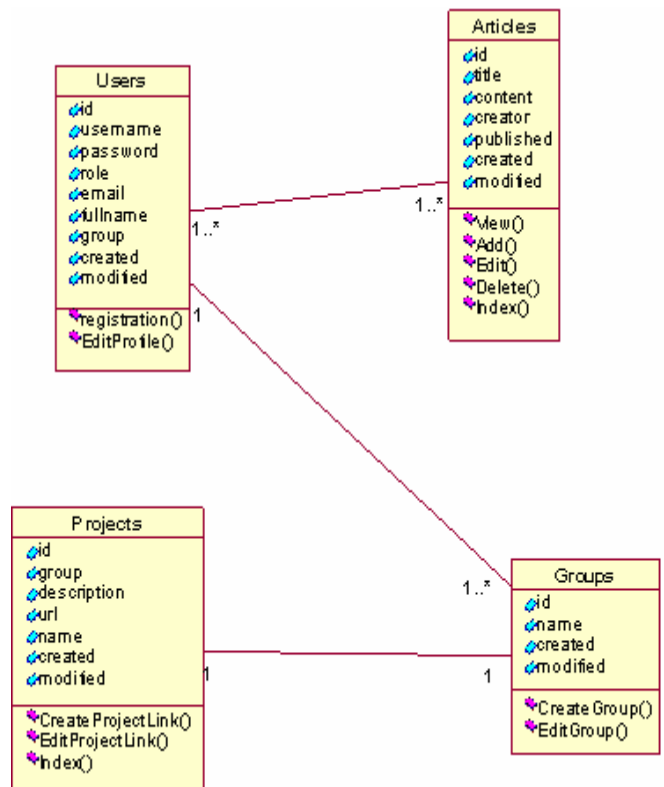
```

function edit($id=null)
{
    if (empty($this->
>params['data']))
    {
        $this->Article->
>setId($id);
        $this->params['data'] =
$this->Article->read();
        $this->set('data',
$this->params['data']);
        $this->render();
    }
    else
    {
        $this->Article->
>set($this->params['data']);
        if ($this->Article->
>save())
        {
            $this->
>flash('Artikel anda telah
diperbarui.', '/articles');
        }
        else
        {
            $this->set('data',
$this->params['data']);
            $this->
>validateErrors($this->Article);
            $this->render();
        }
    }
}
?>

```



Gambar 1. Diagram use case situs komunitas rekayasa web



Gambar 2. Diagram kelas situs komunitas rekayasa web

c. View: *index.html*

```
<h1>Daftar Artikel</h1>
<table>
  <tr>
    <th>Id</th>
    <th>Judul</th>
    <th>Penulis</th>
    <th></th>
  </tr>
  <? foreach($data as $article): ?>
    <tr>
      <td><?=$article['Article']['id'];
      ?></td>
      <td><?=$html->link(
      $article['Article']['title'],
      "/articles/view/{$article['Article']
      ['id']}" ); ?></td>
      <td><?=$
      $article['Article']['creator'];
      ?></td>
      <td>
      <?=$html->link('Ubah',
      "/articles/view/{$article['Article']
      ['id']}" ); ?> |
      <?=$html-
      >link('Hapus',"/articles/delete/{$ar
      ticle['Article']['id']}", null,
      "Anda yakin akan mengubah artikel
      '{$article['Article']['title']}'?");
      ?>
      </td>
    </tr>
  <? endforeach; ?>
</table>

<?=$html->link('Tambah
Artikel',"/articles/add/"); ?>
```

4. PENGUJIAN

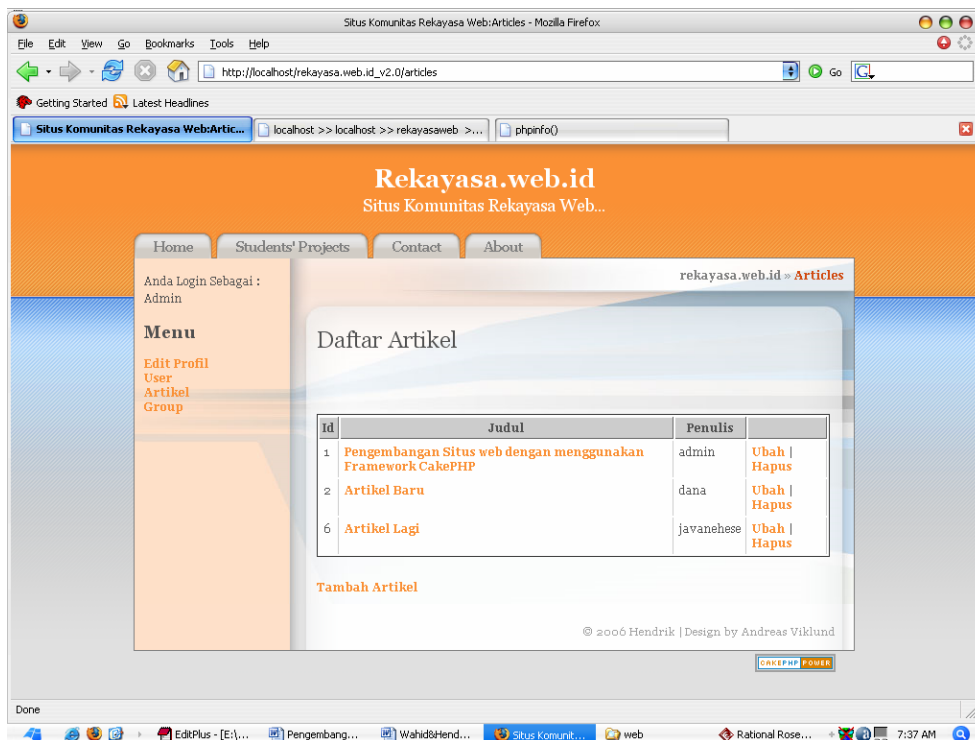
Hasil pengujian akhir dapat dilihat pada Gambar 3.

5. SIMPULAN

Setelah dilakukan proses pengembangan dan pengujian terhadap situs komunitas ini, disimpulkan bahwa penggunaan *framework* CakePHP dapat mempercepat proses pengembangan dan pengelolaan web. Hal ini sangat bermanfaat terutama jika digunakan untuk mengembangkan aplikasi berbasis web berskala besar. Dengan CakePHP pengembang hanya perlu berkonsentrasi pada fungsionalitas aplikasi. Adapun komponen-komponen lainnya dapat memanfaatkan *repository* dari komunitas pengembang CakePHP. Selain itu dengan menggunakan CakePHP yang dikembangkan oleh komunitas *open-source* memungkinkan juga bagi kita untuk mempelajari serta berpartisipasi dalam proses pengembangan *framework* ini.

DAFTAR PUSTAKA

<http://en.wikipedia.org/wiki/Framework>
<http://cakephp.org>
<http://cakeforge.org/>
<http://h3rald.com/articles/view/cakephp/>



Gambar 3. Diagram kelas situs komunitas rekayasa web