

**Perancangan dan Implementasi Sistem Informasi
Audit Mutu Akademik Internal berbasis *Web*
menggunakan *Framework Codeigniter*
(Studi Kasus : LPMAI UKSW)**

Artikel Ilmiah



Peneliti:

Alvind Wayne Luhukay (672013129)

Dr. Sri Yulianto J.P.,S.Si,M.Kom

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN SATYA WACANA SALATIGA
Januari 2017**



PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama : ALVINDI WAYME LUHUKAY
NIM : 672013129 Email : 672013129@student.uksw.edu
Fakultas : TEKNOLOGI INFORMASI Program Studi : TEKNIK INFORMATIKA
Judul tugas akhir : PERANCANGAN DAN IMPLEMENTASI SISTEM INFORMASI AUDIT MUTU
AKADEMIK INTERNAL BERBASIS WEB MENGGUNAKAN FRAMEWORK
CODIGNITER (STUDI KASUS : LPMI UKSW)
Pembimbing : 1. Dr. SPI YULIANTU JOKO PRASTYO, S.Si., M.Kom.
2. _____

Dengan ini menyatakan bahwa:

1. Hasil karya yang saya serahkan ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar kesarjanaan baik di Universitas Kristen Satya Wacana maupun di institusi pendidikan lainnya.
2. Hasil karya saya ini bukan saduran/terjemahan melainkan merupakan gagasan, rumusan, dan hasil pelaksanaan penelitian/implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan narasumber penelitian.
3. Hasil karya saya ini merupakan hasil revisi terakhir setelah diujikan yang telah diketahui dan disetujui oleh pembimbing.
4. Dalam karya saya ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali yang digunakan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka.

Pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari terbukti ada penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya saya ini, serta sanksi lain yang sesuai dengan ketentuan yang berlaku di Universitas Kristen Satya Wacana.

Salatiga, 6 FEBRUARI 2017





PERNYATAAN PERSETUJUAN AKSES

Saya yang bertanda tangan di bawah ini:

Nama : ALVIND WAYNE LOHUKAY
NIM : 672013129 Email : 672013129@student.uksw.edu
Fakultas : TEKNOLOGI INFORMASI Program Studi : TEKNIK INFORMATIKA
Judul tugas akhir : PERANCANGAN DAN IMPLEMENTASI SISTEM INFORMASI AUDIT MUTU
AKADEMIK INTERNAL BERBASIS WEB MENGGUNAKAN FRAMEWORK
CODEIGNITER (STUDI KASUS : LPMI Usw)

Dengan ini saya menyerahkan hak *non-eksklusif** kepada Perpustakaan Universitas – Universitas Kristen Satya Wacana untuk menyimpan, mengatur akses serta melakukan pengelolaan terhadap karya saya ini dengan mengacu pada ketentuan akses tugas akhir elektronik sebagai berikut (beri tanda pada kotak yang sesuai):

- a. Saya mengizinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA
- b. Saya tidak mengizinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA**

* Hak yang tidak terbatas hanya bagi satu pihak saja. Pengajar, peneliti, dan mahasiswa yang menyerahkan hak *non-eksklusif* kepada Repositori Perpustakaan Universitas saat mengumpulkan hasil karya mereka masih memiliki hak copyright atas karya tersebut

** Hanya akan menampilkan halaman judul dan abstrak. Pilihan ini harus dilampiri dengan penjelasan/ alasan tertulis dari pembimbing TA dan diketahui oleh pimpinan fakultas (dekan/kaprodi).

Demikian pernyataan ini saya buat dengan sebenarnya.

Salatiga, 6 FEBRUARI 2017

Mengetahui,

Dr. SPI YULIANTO Ioko P.S.Si., M.Kom.

Tanda tangan & nama terang pembimbing 1

ALVIND WAYNE LOHUKAY
Tanda tangan & nama terang mahasiswa

Tanda tangan & nama terang pembimbing 2

**Perancangan dan Implementasi Sistem Informasi
Audit Mutu Akademik Internal berbasis Web
menggunakan Framework Codeigniter
(Studi Kasus : LPMAI UKSW)**

Artikel Ilmiah

Oleh:
Alvind Wayne Luhukay
NIM: 672013129

Telah disetujui diuji:
Tanggal: 09 Januari 2017.....

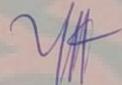
Pembimbing


Dr. Sri Yulianto Joko Prasetyo, S.Si., M.Kom.

Lembar Pengesahan

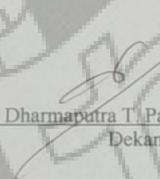
Judul Tugas Akhir : Perancangan dan Implementasi Sistem Informasi Audit Mutu Akademik Internal berbasis *Web* menggunakan *Framework Codeigniter*
(Studi Kasus : LPMAT UKSW)
Nama Mahasiswa : Alvind Wayne Luhukay
NIM : 672013129
Program Studi : Teknik Informatika
Fakultas : Teknologi Informasi

Menyetujui,

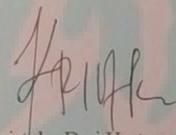


Dr. Sri Yulianto Joko Prasetyo, S.Si., M.Kom.
Pembimbing

Mengesahkan,



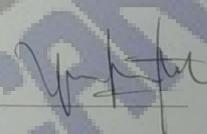
Dr. Dharmaputra T. Palekahelu, M.Pd.
Dekan



Dr. Kristpko Dwi Hartomo, M.Kom.
Ketua Program Studi

Dinyatakan Lulus Tanggal: 2 Februari 2019

Reviewer :

- Yos Richard Beeh, S.T., M.Cs.
- 

1. Pendahuluan

Sistem Penjaminan Mutu Internal (SPMI) adalah kegiatan sistemik penjaminan mutu pendidikan tinggi di Perguruan Tinggi oleh Perguruan Tinggi, untuk mengawasi penyelenggaraan pendidikan tinggi oleh Perguruan Tinggi secara berkelanjutan [1]. Dalam SPMI konsep yang digunakan ialah konsep PDCA (*Plan, Do, Check, Action*) dalam Kaizen. Kaizen berasal dari bahasa Jepang yang artinya ‘penyempurnaan’ atau ‘perbaikan’ berkesinambungan yang melibatkan semua orang, baik manajemen puncak, manajer maupun seluruh karyawan, karena kaizen adalah tanggungjawab setiap individu/orang [2]. Konsep PDCA dalam Kaizen dilakukan untuk menjaga standar yang ada dan mencapai tujuan yang diharapkan. Tahap *Plan* sendiri adalah tahapan yang bertujuan untuk merumuskan atau merencanakan suatu standar penilaian. Tahapan *Do*, pada tahap ini dilakukan pembuatan dokumen SPMI sesuai dengan standar yang telah dirumuskan. Selanjutnya pada tahapan *Check*, dilakukannya evaluasi terhadap perguruan tinggi terkait. Tahap terakhir adalah *Action*. Pada tahap ini, fokus terhadap hasil evaluasi dan mulai melakukan perbaikan sesuai dengan hasil evaluasi yang didapat.

AMAI (Audit Mutu Akademik Internal) adalah suatu kegiatan penjaminan mutu dan konsultasi dalam bidang akademik yang diatur oleh lembaga internal milik perguruan tinggi yang bertujuan untuk menjamin hasil dari audit sesuai dengan rencana yang telah ditetapkan. AMAI juga merupakan hasil dari rumusan standar – standar yang ada pada SPMI yang membuat AMAI digunakan sebagai sistem pada tahapan *Check* untuk evaluasi kinerja operasional suatu perguruan tinggi.

Lembaga Penjaminan Mutu Akademik dan Audit Internal (LPMAI) UKSW dibentuk pada bulan Maret 2006 dan secara efektif mulai berjalan mulai 1 April 2006. Sebelumnya ada dua biro yaitu Sumber Daya Dosen (SDD) dan Akreditasi telah dilebur, sehingga tugas dan tanggung jawab biro-biro tersebut dialihkan pada kantor LPMAI di bawah koordinasi Pembantu Rektor I bidang Akademik. Sebagai lembaga penjaminan mutu di tingkat universitas, LPMAI mempunyai tanggung jawab untuk menyusun sistem manajemen mutu, menjamin pelaksanaan sistem manajemen mutu, dan melakukan evaluasi, dan terus-menerus meningkatkan serta mengembangkan sistem penjaminan mutu, baik itu pada tingkat universitas hingga program studi dan sumber daya dosen [3].

Penelitian ini dilakukan terhadap Sistem Informasi AMAI LPMAI UKSW. Sistem Informasi AMAI pada LPMAI UKSW yang telah ada masih merupakan *web* tanpa *framework* sehingga mempengaruhi waktu kerja dari auditor, dengan demikian proses kerja tidaklah efisien dan dapat menurunkan sistem kerja atau SOP (*Standard Operating Procedure*) dari LPMAI UKSW.

Berdasarkan penelitian yang telah dilakukan pada LPMAI UKSW, maka dibuatlah Sistem Informasi Audit Mutu Akademik Internal berbasis *Web* menggunakan *Framework Codeigniter*. Tampilan dari *Framework Codeigniter* (CI) dipilih sebagai kerangka kerjanya karena CI tergolong sebagai *framework* yang ringan jadi tidak memberikan beban berlebih pada server. CI juga sudah mendukung Model View Controller (MVC), sehingga pengembangan aplikasi menjadi lebih terstruktur dan untuk *database* digunakan MySQL.

2. Kajian Pustaka

Penelitian yang membahas mengenai sistem penjaminan mutu telah dilakukan sebelumnya. Salah satunya adalah penelitian yang berjudul “Perancangan Sistem Informasi Audit Mutu Internal Akmi Baturaja”. Penelitian yang dilakukan terhadap Akmi Baturaja ini, membahas perancangan sistem informasi dengan membuat suatu aplikasi *desktop* yang digunakan khusus untuk bagian internal penjaminan mutu. Aplikasi yang dibuat meliputi pengelolaan data auditor dan data klausa [4].

Penelitian kedua adalah Peningkatan Kualitas Evaluasi Mutu Akademik Universitas Muhammadiyah Malang melalui Sistem Informasi Mutu (SIMUTU). Pada penelitian ini membahas optimasi atau pengoptimalan waktu pelaksanaan evaluasi dengan menggunakan SIMUTU. SIMUTU merupakan sistem AMAI dari Universitas Muhammadiyah Malang, yang dibuat berbasis *web*. Hasil dari penelitian ini adalah *web prototype* SIMUTU dengan melalui *website* www.testomato.com [5].

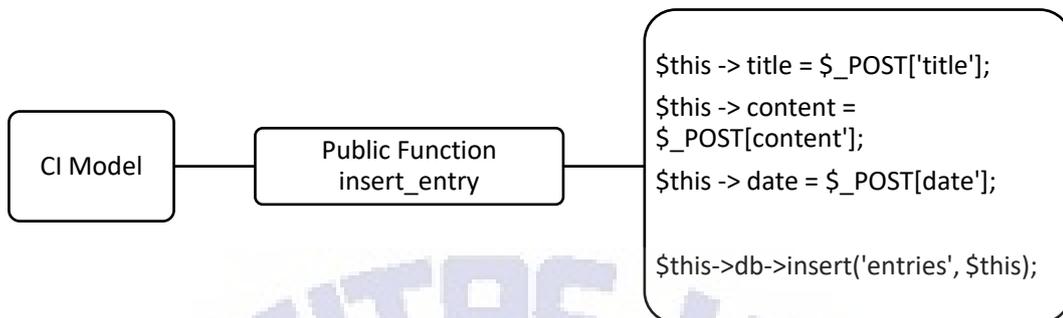
Penelitian ketiga membahas Analisis dan Perancangan Aplikasi Sistem Informasi Audit Mutu Internal dan Dokumentasi Penjaminan Mutu Perguruan Tinggi. Penelitian ini dilakukan terhadap Institut Bisnis dan Informatika (IBI) Darmajaya, Bandar Lampung. Pada penelitian ini membahas peningkatan efisiensi waktu kerja dalam pelaksanaan evaluasi akademik pada IBI Darmajaya. Hasil dari penelitian ini adalah Sistem Informasi penjaminan mutu berbasis *web* yang dibuat berdasarkan ISO 9001:2008 untuk membantu meningkatkan kualitas proses pelaksanaan evaluasi akademik pada IBI Darmajaya [6].

Berdasarkan penelitian-penelitian yang pernah dilakukan sebelumnya, hal ini menjadi acuan untuk turut serta dalam Perancangan Sistem Informasi Audit Mutu Akademik Internal yang berbeda dari sebelumnya untuk menaikkan mutu dari sisi akademik internal UKSW. Pada penelitian ini, penulis mengembangkan Sistem Informasi Audit Mutu Akademik Internal berbasis *Web* dengan menggunakan *Framework Codeigniter* yang belum pernah dikembangkan dan juga diperuntukan bagi auditor akademik.

Framework Codeigniter

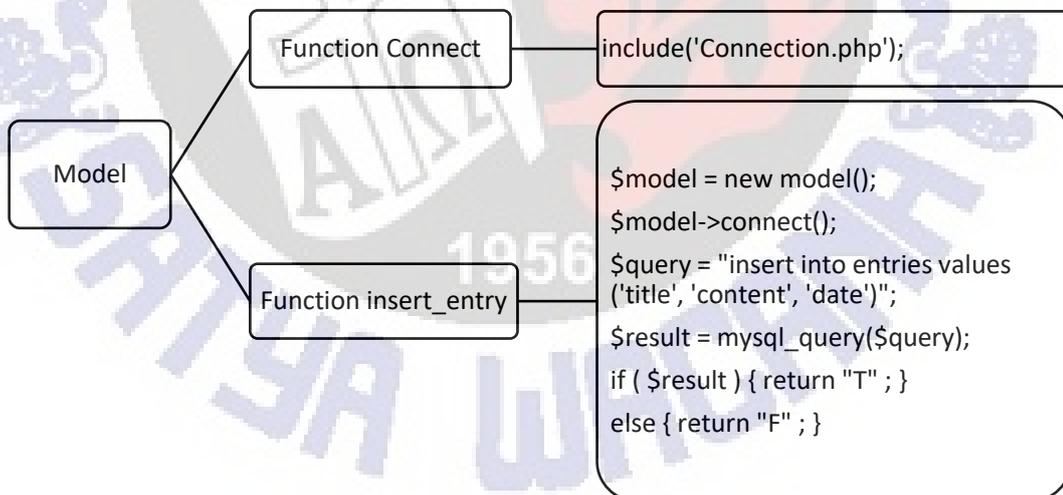
Framework Codeigniter (CI) adalah salah satu *Frameworks* yang ringan jadi tidak memberikan beban berlebih pada server. CI juga sudah mendukung Model View Controller (MVC), sehingga pengembangan aplikasi menjadi lebih terstruktur dan mudah terbaca.

Model adalah bagian yang bertanggung jawab terhadap operasi database, baik itu *create*, *read*, *update* atau *delete* [7]. Ia berupa fungsi – fungsi operasional *database* yang dapat dipanggilkan oleh Controller. Halaman Model pada CI telah tersedia secara langsung oleh CI dan ditandai dengan penggunaan fungsi *extends* dari *CI_Model*.



Gambar 1. Bentuk Diagram Halaman Model pada CI

Proses CRUD pada CI tidak harus menggunakan perintah SQL (*Structured Query Language*), karena pada CI telah tersedia fungsi CRUD yang dapat membantu pengoperasian *database*. Seperti pada Gambar 1, fungsi *insert* (*Create*) pada CI untuk memasukan data kedalam *database* adalah `$this->db->insert()`, berbeda dengan halaman *web* yang tidak menggunakan CI, dimana masih menggunakan perintah SQL. Fungsi ini juga cukup membantu, terutama bagi *developer* yang tidak mahir dalam menggunakan perintah SQL. Selain itu, CI juga telah menyediakan pengaturan koneksi ke *database* yang digunakan agar tidak harus untuk mengatur koneksinya pada tiap halaman yang ada.



Gambar 2. Bentuk Diagram Halaman Model tanpa CI

View merupakan bagian yang menangani tampilan. Pada bagian inilah yang bertugas untuk mempresentasikan data kepada user [7]. Ia berbentuk struktur HTML yang berisikan variabel data yang dikirimkan oleh Controller. Untuk bagian *View* pada CI dan halaman *web* yang tanpa CI masih sama, dikarenakan penggunaan bahasa masih menggunakan struktur Bahasa HTML.

Controller adalah bagian yang mengatur hubungan antara Model dan View, dan juga mengatur HTTP Request [7]. Ia adalah otak dari kinerja aplikasi. Ia terdiri dari fungsi-fungsi yang bersifat operasional dan logikal. Saat ada request yang

masuk, ia akan menangani dan memprosesnya untuk kemudian ditampilkan dalam View. Halaman Controller pada CI dapat dilihat dari penggunaan *extends* terhadap CI_Controller.

Kode Program 1. Contoh Penulisan Kode Controller

```
1. <?php
2. class Evaluation extends CI_Controller {
3.     public function __construct()
4.     {
5.         parent::__construct();
6.         $this->load->model('Model');
7.     }
8.     public function index(){
9.         $this->load->view('home');
10.    }
```

Pada contoh kode *Controller* yaitu Kode Program 1, halaman *Controller* dapat menghubungkan Model dan *View*. Penggunaan atau pemanggilan halaman Model dilakukan pada fungsi *Construct*, agar setiap fungsi yang berada pada halaman *Controller* dapat menggunakan objek dari kelas Model. Fungsi *function index* digunakan untuk memanggil halaman beranda atau *home* secara langsung, karena setiap halaman *Controller* yang diakses akan menjalankan *function index* dengan sendirinya.

3. Metodologi Penelitian dan Perancangan

Penelitian ini akan dilakukan dan diselesaikan melalui 6 tahapan penelitian yaitu: (1). Identifikasi Masalah, (2). Perumusan Masalah, (3). Rancangan Penelitian, (4). Pengumpulan Dat, (5). Pengolahan Data, (6) Penyimpulan Hasil.



Gambar 6. Metodologi Penelitian [8]

Tahap pertama dilakukan analisis terhadap permasalahan yang terkait dengan sistem penjaminan mutu LPMIA UKSW yaitu Audit Mutu Akademik Internal (AMAI). Dalam tahapan ini untuk dapat mengidentifikasi permasalahan yang ada pada AMAI LPMIA UKSW, maka dilakukan grup diskusi dengan pihak

LPMIA UKSW dan juga Auditor agar dapat mengetahui permasalahan yang ada dan juga kebutuhan dari sistem informasi nya nanti.

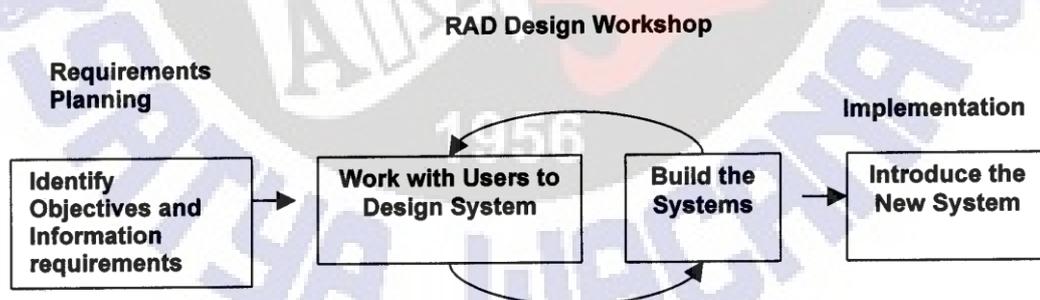
Tahap selanjutnya adalah perumusan masalah. Perumusan masalah dilakukan agar proses penelitian terarah sesuai dengan rumusan masalah yang ada. Pada tahap ini dirumuskan bersama Kepala Bidang Audit Internal dan Mutu sehingga pada proses selanjutnya dapat menjawab tiap masalah dengan tepat

Setelah perumusan masalah dilakukan, tahap berikutnya yang dilakukan adalah rancangan penilitan. Rancangan penilitan dilakukan untuk menentukan hal – hal yang dilakukan dalam penelitian ini, serta waktu yang diperlukan dalam tahapan penilitan.

Tahap berikutnya yaitu pengumpulan data terkait dengan permasalahan pada penelitian yang dilakukan. Pengumpulan data didapatkan dengan melakukan konsultasi terkait data – data audit yang telah ada sebelumnya dan juga pemahaman mengenai buku panduan AMAI.

Data yang didapat berupa dokumen audit mutu yang telah dilakukan sebelumnya begitu juga dengan proses pelaksanaan audit dan buku panduan melaksanakan AMAI. Data yang telah terkumpul kemudian diolah melalui tahap pengujian pada sistem yang akan dibangun.

Adapun metode perancangan sistem yang digunakan dalam penelitian ini adalah model *Rapid Application Development* (RAD). Model RAD adalah salah satu metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat [9]. Tujuan menggunakan RAD sebagai metode perancangan sistem adalah proses penelitian berjalan lebih singkat mengingat penelitian ini bersifat *deadline*, selain itu *user* dapat secara aktif memberikan pandangan atau masukan terkait pengembangan sistem yang dikembangkan. Bagan mengenai RAD dapat dilihat pada Gambar 7.



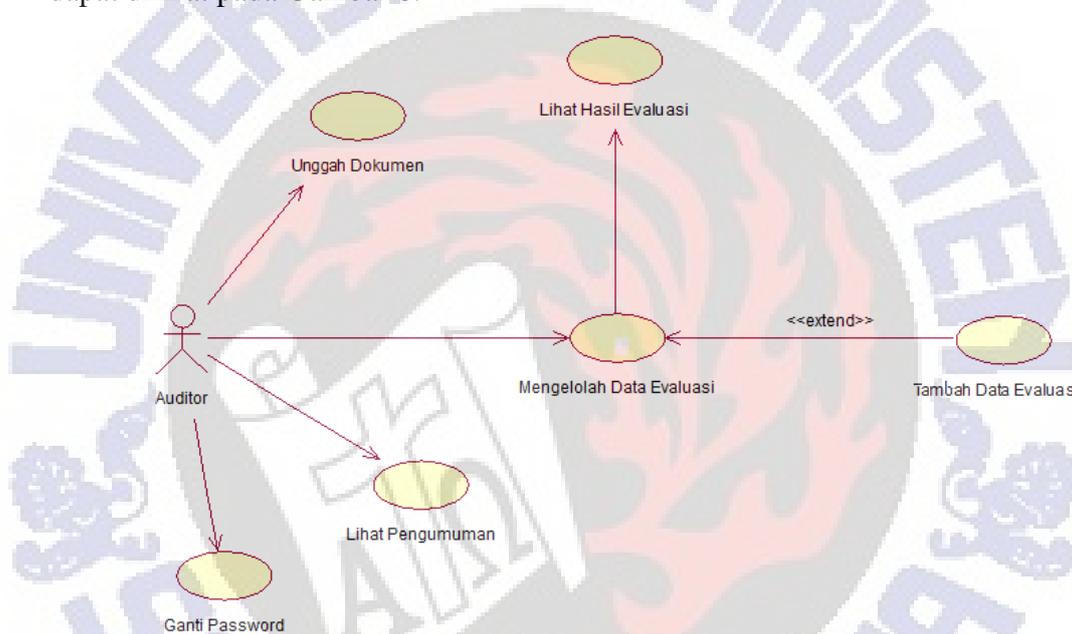
Gambar 7. RAD Model [9]

Metode RAD memiliki 3 tahap: (1) *Requirements Planning*, (2) *RAD Design Workshop*, (3) *Implementation*. Pada tahap pertama metode RAD dilakukan pengidentifikasian terhadap sistem oleh peneliti dan juga *user*, dengan tujuan dapat menentukan kebutuhan serta syarat – syarat yang ada pada sistem yang dibuat. Tahap kedua adalah tahap dimana dibangun dan menunjukkan *visual* atau tampilan yang telah dibuat kepada *user* dan juga menunjukkan cara kerja dari sistem yang telah dibuat. Pada tahap kedua ini juga memungkinkan untuk *user* dapat menilai sistem yang telah dibuat, jika ada fitur pada sistem yang tidak disetujui *user*, maka *user* dapat memberikan masukan terhadap sistem untuk peneliti. Tahap ini akan berulang hingga baik peneliti maupun *user* telah sepakat dengan sistem yang dibangun. Tahap ketiga atau tahap terakhir dalam metode RAD dilakukan setelah

sistem telah disetujui oleh *user*. Sistem yang telah selesai akan diperkenalkan untuk organisasi atau lembaga yang menggunakan sistem tersebut.

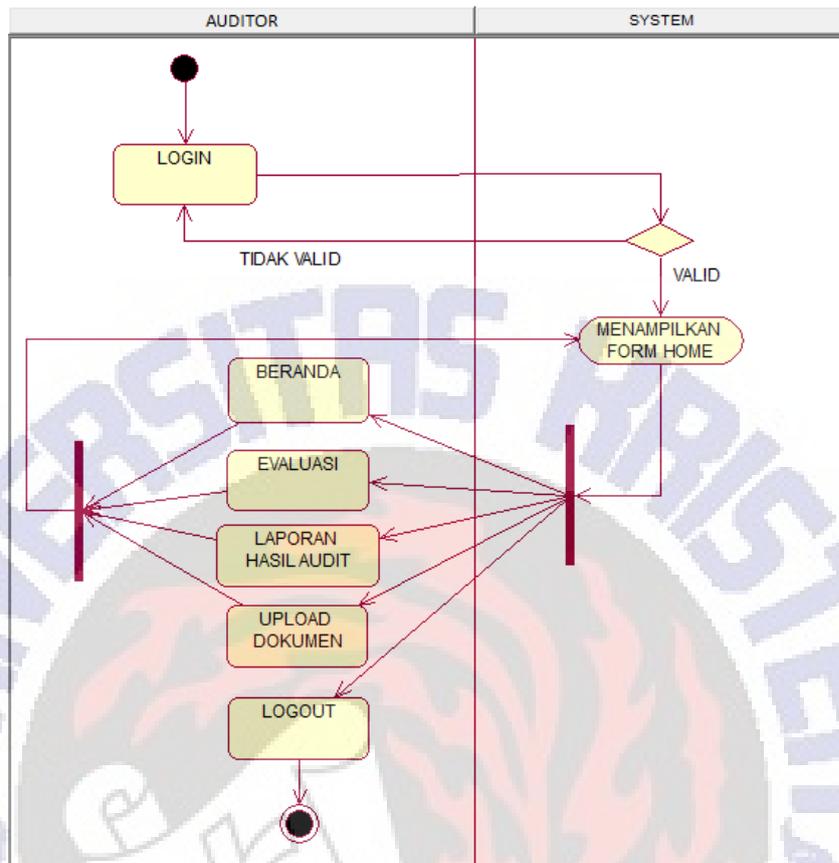
Perancangan Sistem Informasi Audit Mutu Akademik Internal (AMAI) menggunakan *framework Codeigniter* dirancang menggunakan *Unified Modelling Language* (UML) yang terdiri dari *uses case*, activity diagram, dan deployment diagram.

Pada Uses Case Diagram, *user* (auditor) memiliki hak untuk melihat pengumuman terkait audit mutu, pengolahan data evaluasi (audit), melihat hasil dan laporan evaluasi yang telah diinputkan, dan juga penambahan fitur *Ganti Password* dimana memberikan hak untuk *user* untuk mengubah kata kunci (*password*) yang digunakan pada saat *Login*. Secara umum dan fungsionalitas Use Case Diagram dapat dilihat pada Gambar 8.



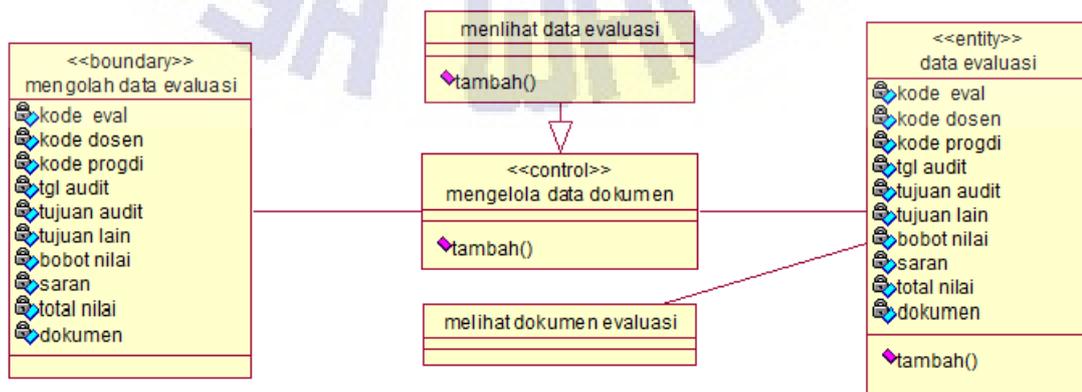
Gambar 8. Use Case Diagram

Activity Diagram yaitu menjelaskan aktivitas *user* dan aplikasi saat pertama kali dijalankan hingga selesai. Dalam perancangan sistem ini proses dimulai pada saat *user* (auditor) akan melakukan *login*, jika *password* tidak sesuai maka akan kembali ke tampilan *login*. Jika tidak berhasil melakukan *login*, maka auditor akan masuk ke tampilan utama Sistem Informasi Audit Mutu Akademik Internal (AMAI) berbasis *Web*. Setelah masuk ke halaman utama, auditor dapat melakukan pemilihan menu berupa Beranda, Evaluasi, Laporan Hasil Audit dan Upload Dokumen.



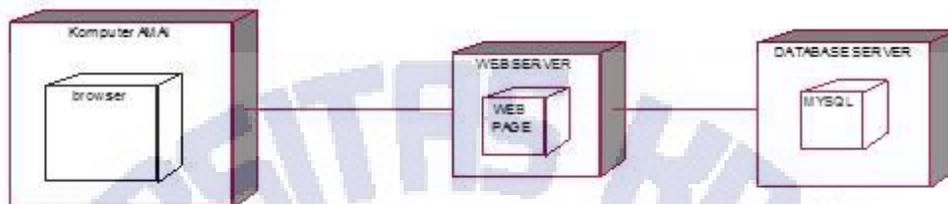
Gambar 9. Activity Diagram

Class Diagram digunakan untuk menunjukkan setiap entitas yang terdapat pada tabel didalam database. Didalam tabel mengelola data evaluasi terdapat entitas seperti berikut: kode eval, kode dose, kode progdi, tgl audit, tujuan audit, tujuan lain, bobot nilai, saran, total nilai, dokumen. Untuk mengelola data evaluasi, terdapat fungsi tambah evaluasi dan melihat hasil evaluasi. Setelah melakukan proses evaluasi hasil akan masuk ketabel data evaluasi yang entitas nya seperti berikut ini: kode eval, kode dose, kode progdi, tgl audit, tujuan audit, tujuan lain, bobot nilai, saran, total nilai, dokumen



Gambar 10. Class Diagram

Gambar 10 adalah Gambar *Deployment Diagram*. *Deployment Diagram* adalah pandangan yang terkait dengan penyebaran fisik aplikasi. Hal ini termasuk persoalan layout jaringan dan lokasi komponen-komponen dalam jaringan. *Deployment View* berisikan prosesor-prosesor, peralatan-peralatan, proses-proses dan hubungan antar prosesor dan antar perlatan.



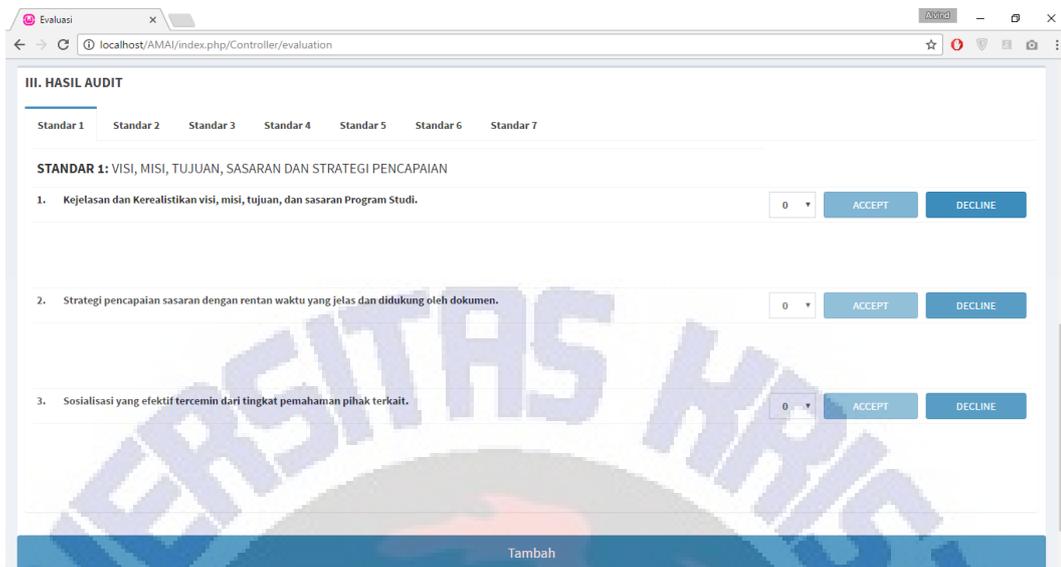
Gambar 11. *Deployment Diagram*

4. Hasil dan Pembahasan

Pembuatan Sistem Informasi AMAI menggunakan *framework* CI dikarenakan dapat membantu proses kerja pembangunan sistem informasi AMAI menjadi lebih cepat.



Gambar 12. Halaman Evaluasi AMAI (1)



Gambar 13. Halaman Evaluasi AMAI (2)

MVC yang dimiliki oleh *Framework* CI sangat membantu dalam proses pengerjaan Sistem Informasi AMAI. MVC secara garis besar berfungsi untuk, memisahkan halaman yang berhubungan langsung dengan *database*, dengan halaman yang berguna untuk berkomunikasi dengan *user* atau biasa dikenal dengan sebutan GUI, dan halaman yang sebagai penerima *request* yang diterima dari *user* dan mengaturnya.

Kode Program 2. Model

```

1. <?php
2.     class Model extends CI_Model{
3.         function getFakultas(){
4.             return $this->db->get('datafakultas');
5.         }
6.         function getProgdi($kode){
7.             $where = array (
8.                 "kodeFakultas" => $kode
9.             );
10.            $data = $this->db->get_where('dataprogdi', $where);
11.            return $data->result();

```

Kode Program 3. Controller

```

1. <?php
2.     defined('BASEPATH') OR exit('No direct script access allowed');
3.     class Evaluation extends CI_Controller {
4.         public function __construct()
5.         {
6.             parent::__construct();
7.             $this->load->model('Model');
8.         }
9.         public function index(){
10.            $data["fakultas"] = $this->Model->getFakultas()->result();
11.            $this->load->view('evaluation', $data);
12.        }

```

Halaman Model pada Sistem Informasi AMAI menggunakan *framework* CI sesuai dengan Kode Program 2, memiliki 2 fungsi yaitu *getFakultas* dan

getProgdi. Fungsi *getFakultas* adalah mengambil semua data fakultas dari tabel fakultas yang nantinya akan dikirimkan ke halaman *Controller* untuk ditampilkan pada *View*. Fungsi *getProgdi* bertujuan untuk mengambil data progdi dengan menggunakan kondisi (*\$where*). Kondisi yang dipakai pada *getProgdi* adalah kode Fakultas, tujuannya adalah agar data progdi yang didapat sesuai dengan fakultas yang terpilih.

Halaman *Controller* bertujuan untuk mengatur setiap data yang akan dikirimkan dari Model ke halaman *View* dapat diterima, begitu juga sebaliknya. Seperti pada Kode Program 3, data fakultas yang telah diterima dari Model dengan menggunakan perintah `$data["fakultas"]=$this->Model->getFakultas()->result()`, akan diteruskan ke halaman *View* menggunakan perintah `$this->load->view`.

Uji Coba dan Perbandingan Kecepatan CI

Pengujian dan perbandingan kecepatan akses data (*load data*) pada penelitian ini dilakukan terhadap proyek yang menggunakan halaman CI dan yang tidak menggunakan CI. Pengujian dan perbandingan ini dilaksanakan dengan menggunakan *Google Chrome* sebagai *browser*-nya. Terdapat empat kriteria yang akan dibandingkan antara CI dan PHP tanpa CI, yaitu *Request File*, *Transferred*, *Finish*, *Load Time*. *Request File* merupakan sejumlah data (gambar, CSS, *font*) yang diakses ke *server* untuk ditampilkan ke *user*. *Transferred* merupakan total ukuran data dari *Request File*. *Finish* dan *Load Time* merupakan total waktu yang dibutuhkan untuk mengakses sebuah halaman, namun *Finish time* terkadang akan lebih kecil dibandingkan *Load Time*, karena *Finish time* hanya mengakses *file* yang bertugas mengatur informasi pada *server* (*back-end*) sedangkan *Load Time* mengakses *file* untuk design antarmuka dengan *user* (*front-end*).

Tabel 1. Pengujian halaman Evaluasi dari Sistem Informasi AMAI

VARIABLE	REQUEST FILE	TRANSFERRED	FINISH	LOAD TIME
CI	43	123 KB	491 ms	860 ms
TANPA CI	43	1.4 MB	2.32 s	2.32 s

Sesuai dengan data pada Tabel 1, ketika pertama kali halaman Evaluasi diakses, halaman Evaluasi pada web tanpa CI dengan jumlah *Request File* 43 data memiliki *Transferred rate* 1.4 *MegaByte* (MB), waktu *finish* 2.32 *second* (s) dan *load time* 2.32 s. Sedangkan pada CI dengan total *Request File* yang sama dengan web tanpa CI, memiliki *Transferred rate*, *Finish time*, dan *Load Time* lebih kecil dari web yang tidak menggunakan CI yaitu, 123 *KiloByte* (KB), 491 *millisecond* (ms), dan 860 ms. Pengujian tersebut membuktikan bahwa menggunakan *framework* CI lebih cepat dibandingkan dengan web tanpa CI meskipun memiliki jumlah *request file* yang sama.

Tabel 2. Pengujian halaman Evaluasi Proses Input dari Sistem Informasi AMAI

VARIABLE	REQUEST FILE	TRANSFERRED	FINISH	LOAD TIME
CI	44	124 KB	486 ms	963 ms
TANPA CI	44	164 KB	11.23 s	6.85 s

Pada tabel 2 dilakukan uji coba ketika proses pengujian pada penyimpanan data evaluasi dari *user* ke *database*. Dilihat dari *Transferred*, waktu *finish* dan *load time*, memiliki waktu yang lebih lama dari pada *framework* CI yaitu. 164 KB, 11.23 s dan 6.85 s. Sedangkan pada CI adalah 124 KB, 486 ms, dan 963 ms.

Berdasarkan hasil uji coba sistem yang dikembangkan melalui *browser Google Chrome*, sistem yang menggunakan *framework* CI dan tanpa menggunakan *framework* CI terdapat perbedaan kecepatan yang cukup signifikan. Dari tabel pengujian, *request file* sistem yang menggunakan *framework* CI dan *request file* yang Tanpa CI memiliki jumlah sama, namun kecepatan *Transferred Data*, *Finish* dan *Load Time* jauh lebih cepat sistem yang menggunakan *framework* CI dari pada sistem yang dikembangkan dengan tanpa CI.

5. Kesimpulan dan Saran

Penelitian ini membuat Sistem Informasi Audit Mutu Akademik Internal (AMAI) berbasis *Web* menggunakan *framework Codeigniter*. Dengan Sistem Informasi AMAI dapat membantu proses kerja dari auditor LPMAI UKSW.

Penggunaan *framework Codeigniter* (CI) sebagai kerangka kerja sangat mempermudah dan mempercepat proses pengerjaan dari Sistem Informasi AMAI.

Framework CI memiliki waktu *finish* dan *load time* lebih cepat dibanding sistem yang tidak menggunakan *framework* CI.

Pengembangan yang dapat dilakukan pada penelitian selanjutnya adalah memperbaiki sistem perhitungan agar lebih kongkret dan mengembangkannya pada *Framework* yang lain atau berbasis *platform* yang berbeda.

6. Daftar Pustaka

- [1] <http://fip.unesa.ac.id/gpm/sistem-penjaminan-mutu-internal-spmi>, diakses tanggal 11 Desember 2016.
- [2] Hakim, M. Arifin, & Musadieg, M. Al, & Nurtahjono, Gunawan Eko. 2016. Pengaruh Budaya Kaizen Terhadap Motivasi dan Kinerja. *Jurnal Administrasi Bisnis (JAB)* Vol. 35, No. 1.
- [3] <http://ppma.uksw.edu/read/sejarah>, diakses tanggal 5 Desember 2016.
- [4] Pujianto. 2014. Perancangan Sistem Informasi Audit Mutu Internal Akmi Baturaja. *Jurnal Media Informatika dan Komputer* Vol. 4, No. 1.
- [5] Wicaksono, Galih W. & Al-Rizki, Andi. 2016. Peningkatan Kualitas Evaluasi Mutu Akademik Universitas Muhammadiyah Malang melalui Sistem Informasi Mutu (SIMUTU). *Jurnal KINETIK* Vol 1, No. 1.
- [6] Darmawan, Abdi & Hasibuan, M. Said. 2014. Analisa dan Perancangan Aplikasi Sistem Informasi Audit Mutu Internal dan Dokumentasi Penjaminan Mutu Perguruan Tinggi. *Jurnal Generic*, Vol. 9, No. 2.
- [7] https://www.codeigniter.com/user_guide/overview/mvc.html, diakses tanggal 23 Desember 2016.

- [8] Hasibuan, Zainal A. 2007. Metode Penelitian Pada Bidang Ilmu Komputer dan Teknologi Informasi : Konsep, Teknik dan Aplikasi. Jakarta : Fakultas Ilmu Komputer Universitas Indonesia.
- [9] Noertjahyana, Agustinus. 2002. Studi Analisis *Rapid Application Development* Sebagai Salah Satu Alternatif Metode Pengembangan Perangkat Lunak. Jurnal Informatika Vol. 3, No. 2.

