

SISTEM KOMPUTERISASI PENGELOLAAN TUGAS AKHIR MAHASISWA MENGGUNAKAN KONSEP MODEL VIEW CONTROLLER PADA STMIK KHARISMA MAKASSAR

Henry Liputra¹, Junaedy², Abd. Munir S³

^{1,2,3}Teknik Informatika, STMIK KHARISMA Makassar

e-mail: ¹henry_13@kharisma.ac.id, ²junaedy@kharisma.ac.id, ³abdulmunir@kharisma.ac.id

Abstrak

Penelitian ini bertujuan untuk membangun sistem komputersisasi pengolahan tugas akhir mahasiswa menggunakan konsep Model View Controller pada STMIK Kharisma Makassar. Untuk tujuan tersebut, maka diusulkan penggunaan Framework Codeigniter dimana framework Codeigniter menggunakan pendekatan MVC (Model, View, Controller) yang memisahkan pengembangan aplikasi menjadi Model (Data Logic), View (Presentation Logic) dan Controller (Process Logic), sehingga pengembangan aplikasi dengan menggunakan konsep MVC dapat lebih terstruktur. Berdasarkan hasil penelitian, penulis telah berhasil membangun sebuah aplikasi pengolahan tugas akhir mahasiswa berbasis web menggunakan konsep MVC pada STMIK Kharisma Makassar. Aplikasi ini dapat membantu prodi serta panitia ujian dalam mengolah judul tugas akhir mahasiswa STMIK Kharisma, serta memudahkan mahasiswa yang ingin mengajukan judul tugas akhir.

Kata kunci: judul tugas akhir, framework codeigniter, Model View Controller.

Abstract

This research aims to build a computerized Management system of college student thesis by using Model View Controller concept at STMIK KHARISMA Makassar. For that aims, it is proposed to use CodeIgniter Framework, where framework Codeigniter use MVC (Model, View, Controller) method to separates between Model (data logic) from View (presentation logic) and Controller (process logic) so that the application development using MVC concept can be more structured. Based on research result, the author has managed to build a software application of college student thesis Management by using MVC concept in STMIK Kharisma Makassar. This Application can be used by Prodi and final exam committee to manage STMIK Kharisma Makassar college student thesis and make college student more easy to submit thesis.

Keywords: thesis, framework Codeigniter, Model View Controller

1. Pendahuluan

Dalam STMIK KHARISMA Makassar terdapat beberapa masalah dalam pendaftaran Tugas Akhir yang menjadi salah satu syarat dalam menyelesaikan studi. Terkadang beberapa mahasiswa semester akhir telah memiliki pekerjaan tetap sehingga tidak memungkinkan untuk datang ke kampus mengurus pendaftaran judul tugas akhir mereka, dan terkadang dosen terkait yang bertugas untuk melakukan pendaftaran judul tugas akhir tidak berada pada tempatnya seperti dosen Ketua Prodi bidang studi atau Panitia Ujian.

Berdasarkan permasalahan yang telah di paparkan di atas, penulis ingin mencoba membuat suatu sistem berbasis aplikasi yang dapat digunakan untuk pendaftaran judul tugas akhir yang berfungsi untuk mempermudah mahasiswa semester akhir untuk memasukkan judul-judul mereka dimana saja dan kapan saja tanpa mengganggu pekerjaan mereka. Dalam aplikasi ini penulis akan menggunakan konsep MVC.

MVC sendiri merupakan kepanjangan dari Model View Controller merupakan sebuah konsep untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller). Saat ini kebanyakan framework dalam aplikasi website adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web.

Aplikasi ini akan dirancang pada komputer dengan sistem operasi Windows dan berbasis web. Dengan adanya aplikasi ini diharapkan dapat mengatasi masalah pada pengelolaan judul tugas akhir sehingga mempermudah mahasiswa semester akhir untuk memasukkan judul tugas akhir mereka. Berdasarkan uraian latar belakang masalah, yang menjadi rumusan masalahnya adalah “Bagaimana cara mengimplementasikan konsep MVC dalam membangun sistem komputerisasi dan pengelolaan Tugas Akhir mahasiswa pada STMIK KHARISMA MAKASSAR?”.

Terdapat batasan masalah dalam penelitian ini, yaitu : Aplikasi yang dibuat hanya menangani pengelolaan tugas akhir mahasiswa berdasarkan inputan masing-masing mahasiswa dan memproses pembagian jadwal ujian di input manual oleh panitia ujian akhir

Penelitian ini bertujuan untuk membangun sistem komputerisasi pengolahan tugas akhir mahasiswa menggunakan konsep MVC pada STMIK KHARISMA Makassar.

2. Landasan Teori

Model View Controller

Model View Controller atau MVC merupakan sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Saat ini kebanyakan framework dalam aplikasi website adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web.

Dalam metode MVC terdapat tiga komponen yaitu:

- a) **Model.** *Model* berhubungan dengan *database* sehingga biasanya dalam *model* akan berisi *class* ataupun fungsi untuk membuat (*create*), melakukan pembaruan (*update*), menghapus data (*delete*), mencari data (*search*), dan mengambil data (*select*) pada *database*.

- b) **View.** *View* adalah bagian *User Interface* atau bagian yang nantinya merupakan tampilan untuk *end-user*. Karena metode yang dipakai merupakan MVC sehingga dalam *view* tidak boleh terdapat pemrosesan data ataupun pengaksesan yang berhubungan dengan *database*. Sehingga *view* hanya menampilkan data-data hasil dari *model* dan *controller*.
- c) **Controller.** *Controller* merupakan penghubung antara *view* dan *model*, maksudnya adalah karena *model* tidak dapat berhubungan langsung dengan *view* ataupun sebaliknya, jadi *controller* inilah yang digunakan sebagai jembatan dikeduanya. Sehingga tugas *controller* adalah sebagai pemrosesan data atau alur *logic program*, menyediakan variabel yang akan ditampilkan di *view*, pemanggilan *model* sehingga *model* dapat mengakses *database*, *error handling*, validasi atau *check* terhadap suatu inputan [2].

Framework CodeIgniter

CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis PHP dibandingkan jika menulis semua kode program dari awal [1]. CodeIgniter menyediakan banyak *library* untuk mengerjakan tugas-tugas yang umumnya ada pada sebuah aplikasi berbasis *web*. Selain itu, struktur dan susunan logis dari CodeIgniter membuat aplikasi yang dibuat menjadi semakin teratur dan rapi sehingga *developer* dapat lebih fokus pada fitur-fitur yang dibutuhkan oleh aplikasi yang dibangun dengan membuat kode program seminimal mungkin. CodeIgniter menggunakan pendekatan *Model-View-Controller* (MVC).

MySQL

Pengertian MySQL menurut Welling (2001), MySQL adalah relational *database management system* yang cepat dan kuat. MySQL sendiri merupakan sebuah basis data yang dapat memberikan pengguna fitur untuk menyimpan, mencari, mengurutkan dan mendapatkan data secara cepat dan mudah[10]. MySQL memiliki server tersendiri yang berfungsi untuk mengendalikan akses-akses ke dalam data untuk memastikan bahwa banyak pengguna dapat menggunakan data-data dalam server secara bersamaan yang dibatasi dengan hak akses tertentu.

MySQL menggunakan Bahasa *Structured Query Language*, sebuah bahasa query basis data. MySQL tersedia dalam lisensi *Open Source* dan lisensi komersil.

Terdapat beberapa kelebihan dari MySQL;

- a) Performance

MySQL memiliki performa yang sangat cepat dalam memproses data-data.

b) Low Cost

MySQL dapat digunakan tanpa mengeluarkan biaya sedikit pun dengan catatan menggunakan lisensi Open Source. Dan memiliki biaya yang sangat rendah jika pengguna berencana menggunakan lisensi komersil.

c) Easy of Use

Mudah digunakan karena saat ini banyak para developer aplikasi menggunakan MySQL sebagai database utama sehingga mudah beradaptasi dalam penggunaannya

d) Probability

MySQL dapat digunakan dalam berbagai system operasi seperti Windows dan UNIX.

Pengujian Blackbox

Menurut Pressman (2002), pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak, yang memungkinkan perekrutan perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya semua persyaratan fungsional untuk suatu program^[12].

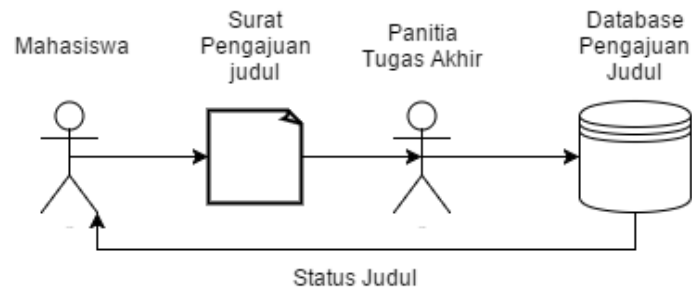
Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

- a. fungsi-fungsi yang tidak benar atau hilang,
- b. kesalahan *interface*,
- c. kesalahan dalam struktur data atau akses *database* eksternal,
- d. kesalahan kinerja,
- e. inisialisasi dan kesalahan terminasi.

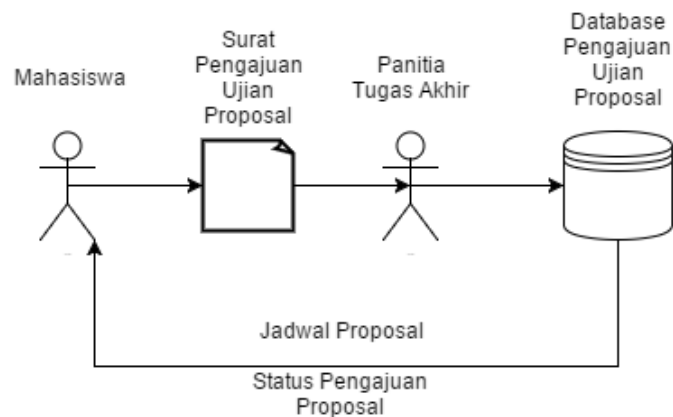
3. Analisa Dan Design Sistem

Pada STMIK KHARISMA MAKASSAR saat ini, proses pengajuan dan pengumpulan judul ujian tugas akhir masih menggunakan sistem manual, dimana mahasiswa datang kepada panitia yang mengatur tugas akhir untuk bertanya apakah dosen yang ingin di angkat sebagai pembimbing masih kosong atau sudah penuh, dan mahasiswa juga terkadang mengalami kesulitan dalam menerima tanda tangan Pembimbing yang menjadi salah satu syarat bahwa judul yang mereka ajukan telah di lihat oleh Pembimbing yang akan membimbing selama tugas akhir di kerjakan.

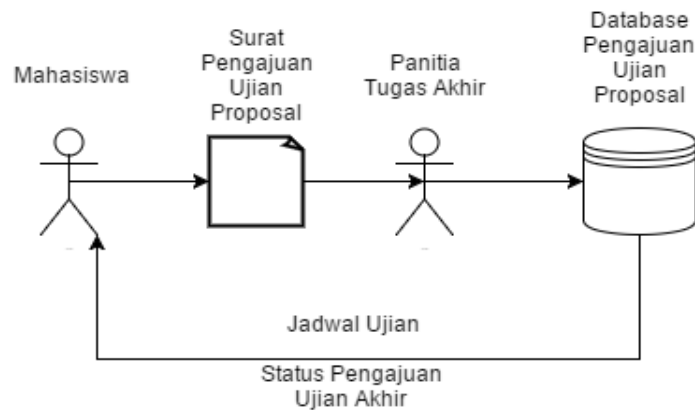
Karena proses pengajuan yang lama serta tidak pasti, maka membuat minat mahasiswa tingkat akhir untuk mengurus tugas akhir mereka menurun. Dalam penelitian ini, proses pengajuan yang membutuhkan waktu lama dapat diselesaikan dengan memakan waktu yang relative singkat serta dapat di lakukan di mana saja dan kapan saja.



Gambar 1. Alur pengajuan Judul Tugas Akhir



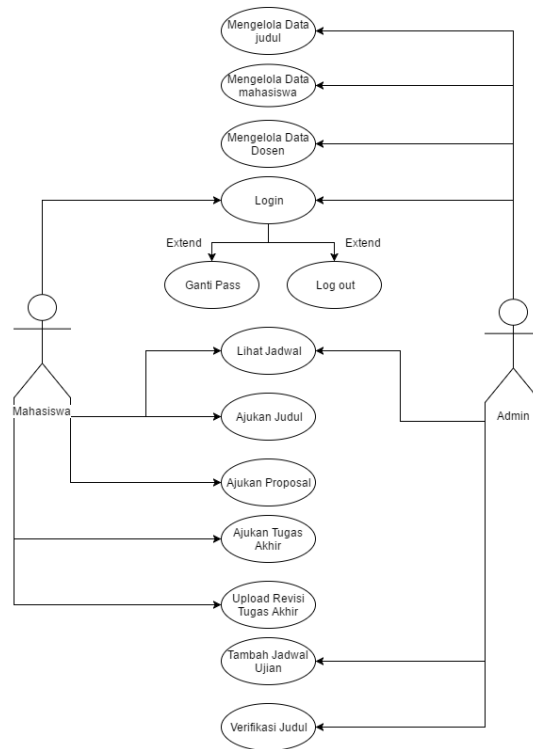
Gambar 2. Alur Pengajuan Ujian Proposal



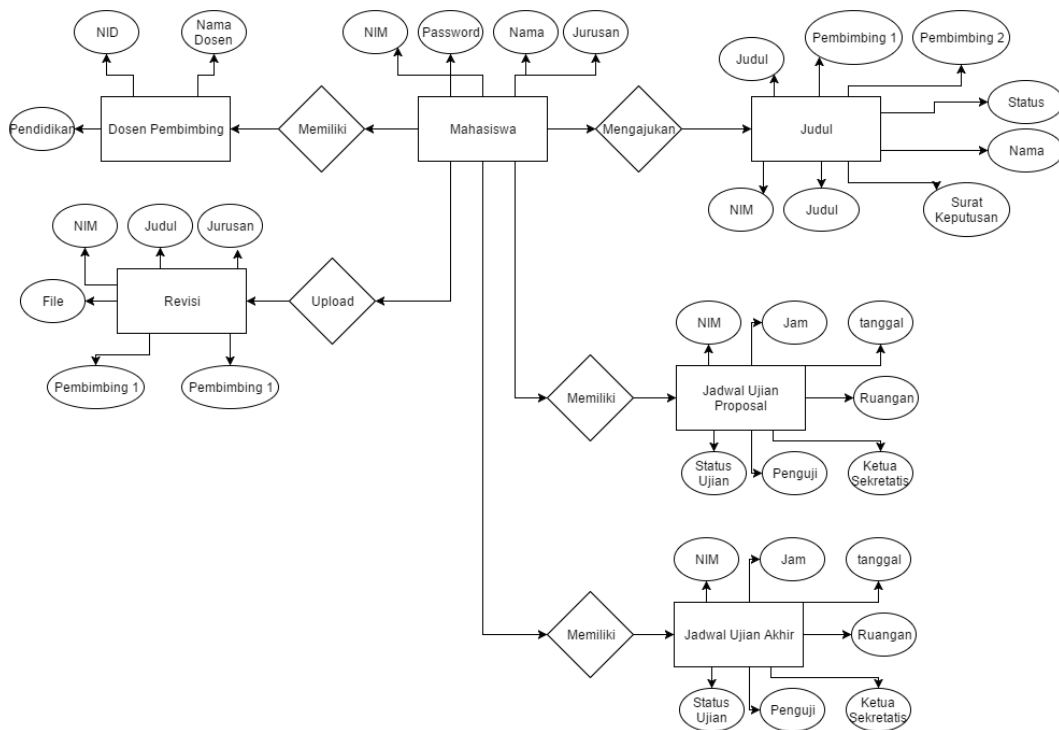
Gambar 3. Alur Pengajuan Ujian Akhir

Penyimpanan data dilakukan oleh panitia secara manual dengan menggunakan Microsoft Excel. Hal ini kadang menyulitkan panitia dalam mengelola judul yang dimasukkan oleh mahasiswa. Contohnya jika banyak mahasiswa secara bersamaan mengumpulkan judul maka panitia harus mengecek dosen terkait secara satu per satu untuk melihat apakah dosen yang di pilih mahasiswa sebagai pembimbing dapat membimbing mahasiswa tersebut. Belum lagi jika ada pergantian pembimbing maka panitia harus mengedit satu per satu data yang telah di masukkan sebelumnya.

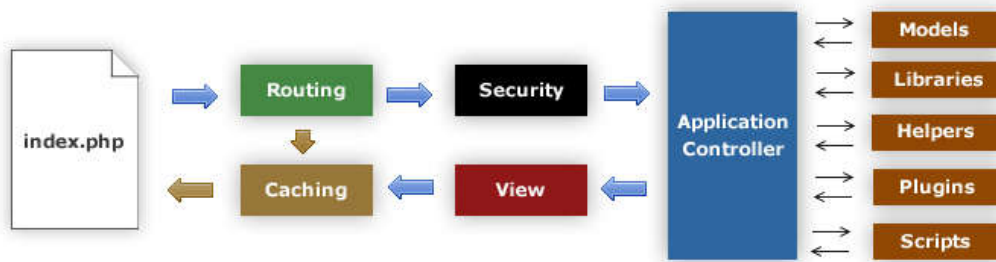
Rancangan Sistem



Gambar 4. Use Case Diagram



Gambar 5. Entity Relationship Diagram



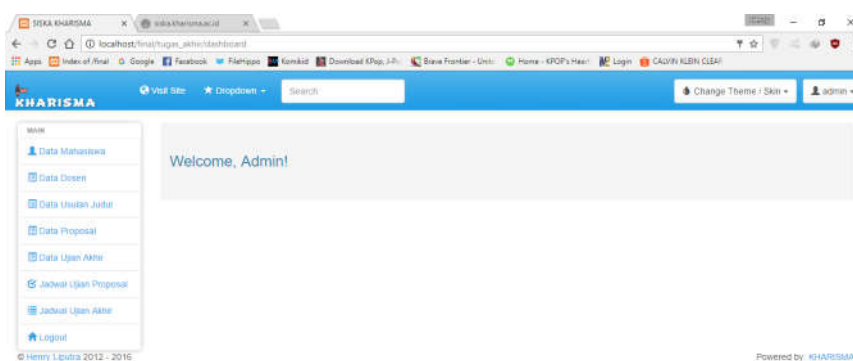
Gambar 6. Arsitektur CodeIgniter

Keterangan gambar:

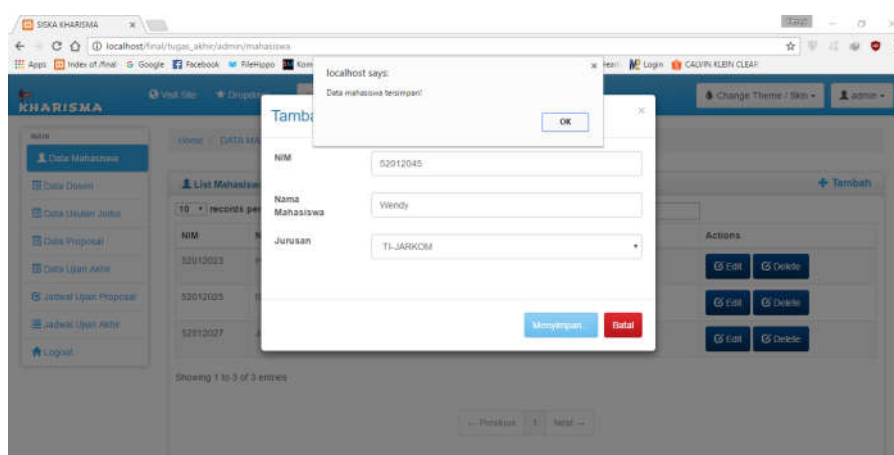
1. Index.php berfungsi sebagai *front-controller*, menginisialisasi sumber dasar yang dibutuhkan untuk menjalankan CodeIgniter.
2. *Router* ini meneliti permintaan HTTP untuk menentukan apa yang harus dilakukan dengannya.
3. Jika *file cache* ada, dikirim langsung ke *browser*, melewati eksekusi sistem normal.
4. *Security*. Sebelum *controller* aplikasi dimuat, permintaan HTTP dan setiap pengguna data yang diajukan disaring untuk keamanan.
5. *Controller* memuat *model*, *library* inti, *helper*, dan sumber daya lain yang diperlukan untuk memproses permintaan khusus.
6. *View* yang diselesaikan diberikan kemudian dikirim ke *web browser* untuk dilihat. Jika *caching* diaktifkan, *view* di-*cache* terlebih dahulu sehingga pada permintaan selanjutnya dapat disajikan.

4. Pengujian Sistem

Metode pengujian yang digunakan pada penelitian ini adalah pengujian *black box*. Pada pengujian *black box*, sistem dipandang sebagai sebuah kotak hitam yang tidak diketahui isi dan fungsinya. Pengujian dilakukan dengan memasukkan input pada sistem dan mengamati apakah hasil output yang diberikan sesuai dengan yang diharapkan. Apabila sistem memberikan output yang tidak sesuai, berarti telah terjadi kesalahan dalam sistem. Kriteria yang menjadi tolak ukur keberhasilan sistem adalah apabila program dapat dikompilasi dengan baik dan memberikan output yang sesuai dengan skenario program.



Gambar 4. Proses Pengujian menu utama



Gambar 5. Proses Pengujian Simpan Data

5. Penutup

Kesimpulan

Berdasarkan hasil pembahasan dan informasi yang diperoleh, maka penulis dapat menarik kesimpulan, antara lain adalah:

1. Penulis telah berhasil membangun sebuah Sistem Komputerisasi Pengolahan Tugas Akhir Mahasiswa Menggunakan Konsep Model View Controller Pada STMIK KHARISMA Makassar
2. Aplikasi yang telah dibuat dapat membantu Panitia Ujian akhir dalam mengolah data judul tugas akhir mahasiswa STMIK Kharisma, serta memudahkan mahasiswa yang ingin mengajukan judul tugas akhir dan melihat jadwal ujian.
3. Dengan menggunakan konsep MVC dalam pembuatan sistem pengolahan judul tugas akhir mahasiswa ini, penulisan program menjadi lebih terstruktur.

Saran

Penulis mengharapkan agar penelitian ini dapat dikembangkan ke penelitian lanjutan guna mengembangkan dan memperbaiki kekurangan sistem. Pengembangan yang dapat dilakukan

pada penelitian ini adalah penambahan fitur yang masih di anggap kurang dan di buat aplikasi untuk platform android atau ios.

Daftar Pustaka

- [1] Basuki, Awan Pribadi. 2010. *Membangun Web Berbasis PHP Dengan Framework CodeIgniter*. Yogyakarta: Lokomedia.
- [2] Hidayatullah, Priyanto dan Jauhari Khairul Kawistara. 2015. *Pemrograman Web*. Bandung: Informatika Bandung.
- [3] Endah Wulansari. 2013. *Perancangan Sistem Informasi Manajemen Tugas Akhir Berbasis Web*. Jurnal Sistem dan Teknologi Informasi Volume 2, No 2013
- [4] Heri Kusharyanto, Dwi Retnoningsih, Dahlan Susilo. 2013. *Sistem komputersisasi pengolahan data tugas akhir mahasiswa*. Surakarta:UNIV. Sahid Surakarta
- [5] Simarmata, Janner dan Iman Paryudi. 2006. *Basis Data*. Yogyakarta: ANDI
- [6] Raharjo, Budi, dkk. 2010. *Modul Pemrograman Web (HTML, PHP, dan MySQL)*. Bandung: Modula.
- [7] M, Syafii, *Membangun aplikasi berbasis PHP dan MySQL*. Yogyakarta : Andi, 2004
- [8] Basuki, Awan Pribadi. 2010. *Membangun Web Berbasis PHP Dengan Framework CodeIgniter*. Yogyakarta: Lokomedia.
- [9] Hidayatullah, Priyanto dan Jauhari Khairul Kawistara. 2015. *Pemrograman Web*. Bandung: Informatika Bandung.
- [10] Welling, Luke dan Laura Thomson. 2009. *PHP and MySQL Web Development (4th Edition)*. USA: Pearson Education, Inc.
- [11] Whitten, Jeffrey L. dan Lonnie D. Bentley. 2007. *System Analysis and Design Methods (Seventh Edition)*. New York: McGraw-Hill
- [12] Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku 1)*. Yogyakarta: ANDI.