

# Sistem Informasi Akademik Subsistem Master Data Mahasiswa Dengan Menggunakan *Framework CodeIgniter*

Irsyad Purbha Irwansyah<sup>1\*)</sup>

<sup>1</sup>Jurusan Teknik Informatika, Fakultas Sains Dan Teknologi, Universitas Islam As-syafi'iyah, Bekasi

<sup>1</sup>Jln. Raya Jatiwaringin No.12 Pondok Gede Jati Cempaka, Kota Bekasi, 17411, Indonesia

email: <sup>1</sup>irsyadpurbha@gmail.com

Received: 30 April 2018; Revised: 1 Oktober 2018; Accepted: 17 Oktober 2018

Copyright ©2018, Politeknik Harapan Bersama, Tegal

**Abstract** – In designing an academic information system in the wider scope, the first thing to be built is an information system that can process master data. Examples of such master data is student data, course data, and lecturer data. This student data becomes very important because student data will become a reference or data source for other information system incorporated in an academic information system as a whole which is found in Islamic University As-syafi'iyah. To build an academic information system that capable of processing master data student required an adequate framework. Codeigniter is a very popular framework as it is very light and has very complete documentation making it easy to use. Codeigniter also uses the MVC approach (Model, View, Controller) in the concept of application development so it will be easier if the development of this information system is growing or transferable in the development process. Based on the above, it is necessary to develop an academic information system with master data subsystem of students at Islamic University As-syafi'iyah which can manage all student data with Codeigniter framework.

**Abstrak** – Dalam perancangan suatu sistem informasi akademik dalam cakupan yang lebih luas hal yang pertama kali harus dibangun adalah sistem informasi yang dapat mengolah data master. Salah satu contoh dari data master tersebut adalah data mahasiswa, data mata kuliah, dan data dosen. Data mahasiswa ini menjadi sangat penting karena data mahasiswa akan menjadi acuan atau sumber data untuk sistem informasi lainnya yang tergabung dalam suatu sistem informasi akademik secara keseluruhan yang terdapat pada Universitas Islam As-syafi'iyah. Untuk membangun sistem informasi akademik yang mampu mengolah master data mahasiswa diperlukan suatu framework yang memadai. Codeigniter adalah merupakan framework yang sangat populer karena sangat ringan dan memiliki dokumentasi yang sangat lengkap sehingga menjadi mudah untuk digunakan. Codeigniter juga menggunakan pendekatan MVC (Model, View, Controller) dalam konsep pengembangan aplikasinya sehingga akan lebih mudah jika pengembangan sistem informasi ini mengalami pertumbuhan ataupun dipindahtangankan dalam proses pengembangannya. Berdasarkan hal tersebut maka perlu dikembangkan suatu sistem informasi akademik dengan subsistem master data mahasiswa di Universitas Islam As-

syafi'iyah yang dapat memanajemen seluruh data mahasiswa dengan framework Codeigniter.

**Kata Kunci** – Sistem Informasi Akademik, UIA, CodeIgniter.

## I. PENDAHULUAN

Keberadaan suatu Teknologi Informasi (TI) didalam suatu institusi perguruan tinggi adalah merupakan suatu kebutuhan yang mutlak atau secara ideal memang sudah harus ada. Keberadaan TI dapat membantu menyelesaikan suatu permasalahan yang ada sehingga TI banyak dimanfaatkan oleh suatu perusahaan, pemerintahan maupun pendidikan [1]. Seiring dengan berjalannya waktu, teknologi informasi tidak saja mempengaruhi sebuah sistem informasi namun juga mempengaruhi dalam kehidupan sehari-hari [2].

Universitas Islam As-syafi'iyah adalah suatu universitas yang memiliki beberapa Fakultas. Salah satu diantara fakultas-fakultas tersebut adalah Fakultas Sains dan Teknologi yang dimana didalam fakultas ini terdapat beberapa program studi. Namun pemanfaatan TI pada Universitas Islam As-syafi'iyah terutama pada Fakultas Sains Dan Teknologi dirasa masih sangat kurang. Pada setiap tahun ajaran dimulai banyak mahasiswa baru yang mendaftar pada Fakultas Sains dan Teknologi. Hal ini menyebabkan data mahasiswa Universitas Islam As-syafi'iyah terus bertambah setiap tahun. Data mahasiswa ini berasal dari data penerimaan mahasiswa baru. Namun hingga saat ini proses pencatatan data mahasiswanya masih bersifat manual yaitu menggunakan *Microsoft Excel* sehingga sering sekali terjadi permasalahan data yang diantaranya adalah ketersediaan akses ke data master mahasiswa tersebut dan masalah validitasi dan integritas dari data tersebut sehingga diperlukan suatu sistem informasi akademik.

Sistem informasi akademik adalah suatu sistem yang dibuat oleh manusia untuk mengolah data dan informasi yang berkaitan dengan akademik pada suatu organisasi atau instansi pendidikan baik formal maupun informal dari tingkat dasar sampai tingkat perguruan tinggi [3]. Untuk mengembangkan suatu sistem informasi akademik yang berbasis *website* dapat menggunakan beberapa alat bantu. Salah satunya adalah dengan menggunakan *framework*.

\*) **Corresponding author:** (Irsyad Purbha Irwansyah)

Email: irsyadpurbha@gmail.com

*Framework* adalah kerangka kerja untuk membantu dalam mengembangkan proyek TI untuk menangani masalah seperti menghubungkan ke pemrograman *database* dan memanggil variabel dan *file* [4].

Pada saat sekarang ini penggunaan *framework* adalah merupakan hal yang umum atau lazim digunakan oleh para pengembang *website*. Salah satu *framework* yang populer adalah *Codeigniter*. Banyak hal yang menyebabkan *Codeigniter* menjadi suatu *framework* yang cukup populer, diantaranya adalah sifatnya yang *open source* dan dukungan dokumentasinya yang cukup lengkap.

Berdasarkan uraian diatas maka diperlukan pengembangan suatu sistem informasi akademik yang dapat mengakomodir kebutuhan untuk manajemen data-data mahasiswa tersebut sehingga dapat digunakan untuk pengembangan sistem informasi akademik secara berkelanjutan.

## II. PENELITIAN YANG TERKAIT

Beberapa penelitian yang terkait dengan sistem informasi akademik diantaranya adalah penelitian yang dilakukan oleh Indrayani, Etin (2013). Penelitian ini dilakukan untuk merencanakan dan mengorganisasikan sistem informasi akademik pada perguruan tinggi di Kota Bandung. Selain itu tujuan dari dilakukannya penelitian ini adalah untuk menerapkan sistem informasi akademik pada perguruan tinggi di Kota Bandung dan melakukan pemantauan dan evaluasi pada sistem informasi tersebut. Teknik sampling yang digunakan adalah teknik purposif sampling dimana sampelnya berasal dari perguruan-perguruan tinggi yang telah mengadopsi ICT kedalam sistem informasi akademik mereka. Hasil dari penelitian ini adalah menunjukkan bahwa tingkat efektifitas perencanaan dan pengorganisasian sistem informasi akademik pada perguruan-perguruan tinggi di Kota Bandung sebagian besar efektif [5].

Sebuah penelitian juga dilakukan oleh Syahputra, Pane Eddis., dkk (2015) dengan penelitiannya yang terkait dengan perancangan Website Sistem Informasi Akademik Sekolah Menengah Kejuruan (SMK) Negeri 5 Pekanbaru. Penelitian ini mengangkat permasalahan yang ada pada SMK Negeri 5 Pekanbaru yaitu penggunaan media internet pada SMK tersebut hanya sebatas untuk mencari informasi terbaru untuk mencari bahan dan tugas siswa. Sekolah Menengah Kejuruan (SMK) Negeri 5 Pekanbaru juga belum menggunakan media internet dalam hal pemberian informasi baik kepada siswa sendiri, lingkungan sekolah, orang tua, dan masyarakat luas. Hasil dari penelitian ini adalah dengan adanya sistem informasi akademik berbasis *web* pada SMKN 5 Pekanbaru mampu meningkatkan mutu pengolahan data sehingga penyampaian informasi dapat diterima secara cepat dan berjalan lancar [6].

Sebuah penelitian dilakukan oleh Alpiandi, M.R (2016) terkait Sistem Informasi Akademik berbasis web di SMP Negeri 2 Kecamatan Gaung Anak Serka. Penelitian ini mengangkat permasalahan yang terjadi di SMP Negeri 2 tersebut yaitu sistem pendataan siswanya dirasakan masih banyak kekurangan yang terjadi dikarenakan sistem yang ada masih menggunakan lembaran kertas dan arsip sehingga dapat menyebabkan data-data yang ada mudah hilang ataupun rusak. Data-data sekolah seperti siswa, guru, dan nilai terkadang susah dicari saat dibutuhkan. Informasi sekolah

juga diumumkan melalui media majalah dinding sehingga bagi siswa yang jarang melihat mading akan ketinggalan informasi. Pembaruan data juga sangat lambat karena sistem belum berbasis *online*. Hasil dari penelitian ini adalah dengan adanya sistem informasi akademik berbasis *web* ini sangat membantu pihak sekolah dalam mengelola administrasi data sekolah seperti data guru, data siswa, data nilai, data pengajar, data mata pelajaran, dan data jadwal pelajaran. Hasil dari penelitian ini adalah juga dapat membantu membuat data-data akademik selalu *update* sehingga tidak mengganggu proses pembelajaran [7].

Sebuah penelitian juga telah dilakukan oleh Susanti, Melan (2016) terkait perancangan Sistem Informasi Akademik berbasis web pada SMK Pasar Minggu Jakarta. Penelitian tersebut mengangkat permasalahan yang terjadi yaitu tidak terdapatnya sistem informasi yang terkomputerisasi, sehingga untuk mendapatkan informasi orang tua dan siswa harus datang ke sekolah. Hasil dari penelitian ini adalah bahwa penggunaan sistem informasi akademik berbasis web pada SMK Pasar Minggu tersebut dapat memberikan solusi yang lebih cepat dalam pengolahan nilai, proses pembuatan laporan yang lebih efektif dan efisien karena penyimpanan data sudah dalam database [8].

Sebuah penelitian telah dilakukan oleh Putra Membara, Eko., dkk (2014) mengenai Sistem Informasi Akademik SMP Negeri 2 Talang Empat berbasis web. Penelitian ini bertujuan untuk merancang dan merealisasikan sistem informasi akademik pada SMP Negeri 2 Talang Empat berbasis web dengan menggunakan bahasa pemrograman PHP dan database MySQL. Hasil dari penelitian ini adalah dengan adanya sistem informasi akademik berbasis website pada Sekolah Menengah Pertama Negeri Talang 4, maka sekolah telah memiliki media alternatif untuk penyedia informasi, sehingga mempermudah pencarian data dan sebagainya [9].

## III. PHP, MYSQL, CODEIGNITER & FRAMEWORK

### A. PHP: Hypertext Preprocessor

Dalam beberapa tahun terakhir, PHP telah berkembang menjadi platform *web* yang paling tersebar luas di dunia, beroperasi di lebih dari sepertiga *web server* di seluruh dunia. Pertumbuhan PHP tidak hanya kuantitatif tetapi juga kualitatif. Semakin banyak perusahaan-perusahaan bergantung pada PHP aplikasi bisnis mereka yang bersifat krusial. Hal ini menciptakan banyak lapangan kerja baru untuk para pengembang PHP. Tingkat kesulitan pengembangan sistem berbasis PHP cenderung tidak berubah dan sangat mudah, namun fitur yang ditawarkan oleh PHP saat ini memungkinkan pengembang untuk membuat aplikasi yang jauh dari sekedar aplikasi HTML sederhana. [10].

Perbaikan dalam konsep objek-model pada PHP membuat proyek berskala besar dapat dikerjakan secara lebih efisien dengan menggunakan metodologi berorientasi objek standar. Dukungan pada XML terbaru juga menjadikan PHP bahasa pemrograman terbaik untuk memproses XML dan digabungkan dengan dukungan SOAP terbaru menjadikan *PHP platform* yang paling ideal untuk membuat layanan *web* [10].

### B. MySQL

Basis data merupakan komponen yang sangat penting dalam pembangunan Sistem Informasi (SI), karena menjadi

tempat untuk menampung dan mengorganisasikan seluruh data yang ada didalam sistem, sehingga dapat diekspansi untuk menyusun informasi-informasi dalam berbagai bentuk [11].

MySQL adalah sistem basis data yang bersifat sangat cepat, relasional, dan kuat (*Relational Database Management System RDBMS*). MySQL adalah sebuah sistem basis data yang membantu untuk menyimpan, melakukan pencarian, penarikan, dan pengurutan data secara efisien. *MySQL Server* mengatur akses ke data dan memastikan beberapa pengguna dapat mengakses data tersebut secara bersama-sama dalam waktu yang sama. *MySQL Server* juga membantu pengguna agar dapat mengakses data tersebut dengan cepat dan memastikan hanya pengguna yang memiliki otorisasi yang dapat memiliki akses ke data tersebut. Oleh karena itulah MySQL adalah sistem basis data yang bersifat *multi-user*, dan *multi-threaded server* [12].

MySQL menggunakan SQL (*Structure Query Language*), bahasa *query* yang menjadi standar sistem basis data diseluruh dunia. MySQL telah resmi di rilis dan tersedia sejak tahun 1996, namun pengembangannya sudah mulai dilakukan sejak tahun 1979. MySQL saat ini merupakan pemenang dari penghargaan Linux Journal Readers Chose Award tiga tahun berturut-turut. MySQL saat ini tersedia dalam *open source license*, namun versi *commercial license* juga tersedia jika dibutuhkan [12].

### C. CodeIgniter

*CodeIgniter* adalah kerangka kerja untuk pengembangan aplikasi berbasis *web* dengan menggunakan bahasa PHP. *CodeIgniter* memiliki banyak fitur yang membuatnya lebih menonjol dari *framework* yang lain. Tidak seperti kerangka kerja atau *framework* lainnya yang mungkin dapat dijumpai, *CodeIgniter* memiliki dokumentasi yang sangat lengkap yang meliputi setiap aspek pada kerangka kerjanya. *CodeIgniter* juga dapat berjalan di lingkungan *shared hosting* karena memiliki indikasi penggunaan yang sangat minim namun dengan fungsi yang maksimal. Di sisi pemrograman *CodeIgniter* juga kompatibel dengan bahasa pemrograman PHP4 dan bahasa pemrograman PHP5 sehingga akan dapat berjalan di sebagian besar *web hosting* yang ada di seluruh dunia [13].

*CodeIgniter* juga menggunakan mekanisme perancangan dengan konsep Model, View, Controller (MVC) yang merupakan konsep untuk mengatur aplikasi menjadi 3 bagian. Model adalah lapisan abstraksi dari sistem basis data. View adalah *front end* dari aplikasi *web*. Sedangkan Controller adalah logika bisnis atau *business logic* dari aplikasi *web*. *CodeIgniter* juga memiliki implementasi dari pola *Active Record*. Hal ini membuat penulisan *query SQL* yang kompleks menjadi lebih mudah dan membuat aplikasi jadi lebih mudah dibaca. Sistem *Active Record* juga membuat jika suatu saat aplikasi tersebut ada perubahan sistem basis data [13].

### D. Framework

Keuntungan yang dapat diperoleh dari penggunaan *framework* adalah:

- Waktu pembuatan aplikasi *website* jauh lebih singkat.

- Kode aplikasi *website* menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok, detailnya adalah kode dari *framework*.
- *Website* menjadi lebih mudah diperbaiki, karena tidak perlu fokus kesemua komponen kode *website*, terutama kode sistem *framework*,
- Tidak perlu lagi membuat kode penunjang aplikasi *website* seperti koneksi *database*, validasi *form*, GUI, dan keamanan.
- Pikiran pengembang menjadi lebih terfokus ke kode alur permasalahan *website*, apa yang ditampilkan dan layanan apa saja yang diberikan dari aplikasi *website* tersebut.
- Jika dikerjakan *team-work* maka akan lebih terarah karena sistem *framework* mengharuskan adanya keteraturan peletakan kode. Seperti bagian pengambilan *database* terpisah dengan bagian pengaturan tampilan untuk pengunjung [14].

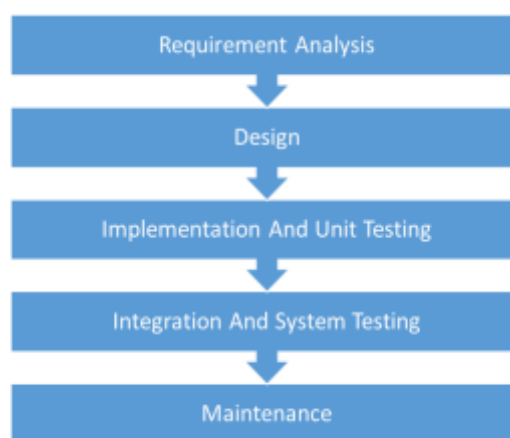
## IV. METODE PENELITIAN

### A. Teknik Pengumpulan Data

1) *Wawancara*: dilakukan dengan cara datang langsung ke Universitas Islam As-syafi'iyah untuk melakukan sesi tanya jawab dengan Ketua Jurusan Program Studi Teknik Informatika dan Dekan Fakultas Sains & Teknologi Universitas Islam As-syafi'iyah dalam rangka untuk mendapatkan informasi yang sesuai dengan fokus penelitian yang dilakukan penulis.

2) *Observasi*: dilakukan dengan cara terjun langsung untuk menjadi partisipan dalam berbagai kegiatan yang berkaitan dengan penggunaan data mahasiswa terutama mahasiswa yang ada di Fakultas Sains & Teknologi Universitas Islam As-syafi'iyah.

3) *Studi Pustaka*: dilakukan dengan cara mencari dan mengutip informasi dari beberapa buku dan jurnal yang dapat dijadikan sumber referensi untuk penelitian ini.



Gbr. 1 Tahapan Metode Waterfall [16].

### B. Metode Pengembangan Sistem

Metode pengembangan sistem yang dilakukan pada penelitian ini adalah menggunakan metode pengembangan *waterfall*. Pengembangan sistem dengan metode *waterfall* adalah metode pengembangan sistem secara statis dan lebih menyerupai metode pengembangan sistem secara berurutan

atau linear, menyelesaikan satu kegiatan sebelum kegiatan yang lain [15].

Beberapa keunggulan dari metode pengembangan sistem dengan metode *waterfall* adalah:

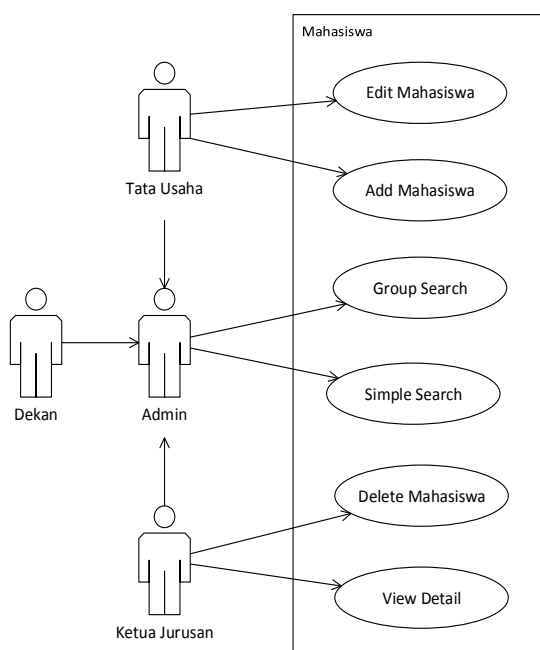
- Metode ini menyediakan struktur untuk mengatur dan mengendalikan proyek pengembangan perangkat lunak.
- Detail rancangan dan kesalahan dianalisa oleh model sebelum kode program ditulis sehingga menghemat waktu selama pengembangan.
- Karena sifatnya yang berurutan dan kesalahan linear dalam suatu fase dapat dideteksi sebelum pindah ke yang lain [16].

## V. HASIL DAN PEMBAHASAN

### A. Use Case Diagram

Pada sistem informasi akademik subsistem master data mahasiswa ini terdapat 3 buah aktor yaitu Dekan, Ketua Jurusan, Tata Usaha dan 1 buah aktor yang bersifat generalisasi yaitu Admin.

- Dekan adalah seseorang yang memimpin sebuah fakultas dalam suatu universitas. Dekan dapat menjadi pimpinan yang bersifat administratif maupun pimpinan yang bersifat keakademikan.
- Ketua Jurusan (biasa disebut Kajur) adalah seseorang yang menjalankan kebijakan akademik dan standar mutu dari pendidikan yang telah ditetapkan oleh suatu universitas.
- Tata Usaha adalah suatu bagian yang terdiri dari 1 orang atau lebih yang menjalankan tugas atau fungsi administratif yang ada dalam suatu institusi maupun suatu organisasi.
- Admin adalah spesialisasi dari aktor yang dipartisipasikan ke semua *usecase* yang diasosiasikan dengan aktor yang lebih bersifat *general*.



Gbr. 2 Use Case Diagram sistem yang diusulkan

TABEL I  
 DESKRIPSI USECASE ADD MAHASISWA

Use Case	Add Mahasiswa
Brief Description	Use Case ini memungkinkan bagian Tata Usaha menambahkan mahasiswa baru kedalam sistem informasi akademik. Kegiatan ini biasa dilakukan sebelum tahun ajaran dimulai pada semester ganjil. Namun dapat pula dilakukan ketika tahun ajaran baru dimulai dengan kebijakan-kebijakan tertentu. Fakultas Sains Dan Teknologi Universitas Islam As-syafi'iyah tidak menyelenggarakan kegiatan penerimaan mahasiswa baru pada semester genap sehingga proses <i>input</i> data mahasiswa baru hanya dilakukan pada semester ganjil. Untuk mahasiswa yang sifatnya pindahan dari universitas atau institusi lainnya jika akan dimasukan di semester ganjil saja. Namun sistem informasi akademik tidak melakukan suatu verifikasi atau validasi tertentu terhadap periode <i>penginputan</i> data master mahasiswa sehingga pada dasarnya proses <i>penginputan</i> data master mahasiswa dapat dilakukan kapan saja oleh bagian Tata Usaha Universitas Islam As-syafi'iyah.
Actor	Tata Usaha
Pre Condition	Tata Usaha membuka halaman View Mahasiswa lalu memilih menu Add.
Main Flow	Tata Usaha memasukan data NIM, nama mahasiswa, tempat lahir, tanggal lahir, memilih jenis kelamin, memilih agama, memilih status marital, memilih golongan darah, memilih jenis kewarganegaraan, memasukan data alamat yang tertera di KTP, alamat tinggal saat ini, nomor telpon genggam, nama ibu kandung, nomor KTP, alamat <i>email</i> , memilih status masuk, memilih info, mengisi asal sekolah, tanggal masuk, kontak dalam keadaan darurat, memilih hubungan, memilih jenis kelas, memilih tahun ajaran, memilih status, dan memilih kurikulum.
Alternative Flow	Tata Usaha dapat memilih menu <i>refresh</i> untuk menyegarkan tampilan atau memilih menu <i>back</i> untuk kembali kehalaman <i>view</i> mahasiswa.
Post Condition	Setelah Tata Usaha menekan tombol <i>save</i> maka sistem akan melakukan pengecekan apakah nama dan tanggal lahirnya sudah ada atau belum disistem. Jika nama dan tanggal lahir ditemukan ada yang sama persis maka data yang dimasukan terindikasi ganda. Sistem lalu akan memberikan notifikasi atau pemberitahuan bahwa data tersebut sudah ada dan proses penyimpanan tidak akan dilanjutkan oleh sistem. Jika nama dan tanggal lahir tidak ditemukan maka sistem akan melakukan penyimpanan data ke sistem basis data dan tampilan sistem informasi akademik akan kembali ke menu <i>view</i> mahasiswa.

TABEL II  
 DESKRIPSI USECASE EDIT MAHASISWA

Use Case	Edit Mahasiswa
Brief Description	Use Case ini memungkinkan bagian Tata Usaha untuk merubah data mahasiswa yang telah dimasukan. Perubahan data dapat terjadi karena ada kesalahan pada waktu memasukan data atau data mahasiswa yang kurang lengkap.
Actor	Tata Usaha
Pre Condition	Tata Usaha membuka halaman View Mahasiswa lalu melakukan pencarian terhadap mahasiswa tertentu yang ingin diubah datanya. Setelah ditemukan dapat memilih menu Edit.
Main Flow	Tata Usaha mengubah data NIM, nama mahasiswa, tempat lahir, tanggal lahir, mengubah jenis kelamin, mengubah kelas, mengubah agama, mengubah status marital, mengubah golongan darah, mengubah status, mengubah kurikulum, mengubah alamat KTP, mengubah alamat tinggal, mengubah nomor telpon genggam, mengubah nomor KTP, mengubah warga negara, mengubah status masuk, mengubah asal sekolah, mengubah nomor kontak dalam keadaan darurat, mengubah hubungan, mengubah ibu kandung, mengubah info dan mengubah alamat email.
Alternative Flow	Tata Usaha dapat memilih menu <i>refresh</i> untuk menyegarkan tampilan atau memilih menu <i>back</i> untuk kembali kehalaman <i>view</i> mahasiswa.
Post Condition	Setelah Tata Usaha menekan tombol <i>save</i> maka sistem akan melakukan perubahan data yang ada di sistem basis data. Lalu tampilan di sistem informasi akademik akan kembali ke menu <i>view</i> mahasiswa.

TABEL III  
 DESKRIPSI USECASE GROUP SEARCH

Use Case	Group Search
Brief Description	Use Case ini memungkinkan Admin untuk melakukan pencarian data mahasiswa secara berkelompok.
Actor	Admin
Pre Condition	Admin membuka halaman View Mahasiswa lalu melakukan pencarian data mahasiswa dengan kategori tertentu.
Main Flow	Pencarian dapat dilakukan berdasarkan kategori kurikulum yang diambil, status mahasiswa (aktif, non aktif, cuti, lulus), atau berdasarkan jenis kelas (kelas reguler, kelas malam, kelas karyawan).
Alternative Flow	Admin dapat memilih menu <i>refresh</i> untuk menyegarkan tampilan atau memilih menu <i>add</i> untuk menambahkan data mahasiswa.
Post Condition	Jika data ditemukan, sistem akan menampilkan data dalam bentuk tabel yang terdiri dari beberapa kolom yaitu: no, nama mahasiswa, NIM, tempat lahir, tanggal lahir, nomor telpon genggam, dan <i>tools</i> yang berisi menu-menu lainnya.

TABEL IV  
 DESKRIPSI USECASE SIMPLE SEARCH

Use Case	Simple Search
Brief Description	Use Case ini memungkinkan Admin untuk melakukan pencarian data mahasiswa secara langsung dengan memasukan kata kunci yang dicari.
Actor	Admin
Pre Condition	Admin membuka halaman View Mahasiswa lalu melakukan pencarian terhadap mahasiswa tertentu.
Main Flow	Pencarian dapat dilakukan berdasarkan NIM, nama mahasiswa, alamat KTP, alamat tinggal, nomor telpon genggam, ibu kandung, tempat lahir, email, dan asal sekolah. Hasil pencarian juga dapat di urutkan berdasarkan NIM, nama mahasiswa, alamat KTP, alamat tinggal, nomor telpon genggam, ibu kandung, tempat lahir, email, dan asal sekolah.
Alternative Flow	Admin dapat memilih menu <i>refresh</i> untuk menyegarkan tampilan atau memilih menu <i>add</i> untuk menambahkan data mahasiswa.
Post Condition	Jika data ditemukan, sistem akan menampilkan data dalam bentuk tabel yang terdiri dari beberapa kolom yaitu: no, nama mahasiswa, NIM, tempat lahir, tanggal lahir, nomor telpon genggam, dan <i>tools</i> yang berisi menu-menu lainnya.

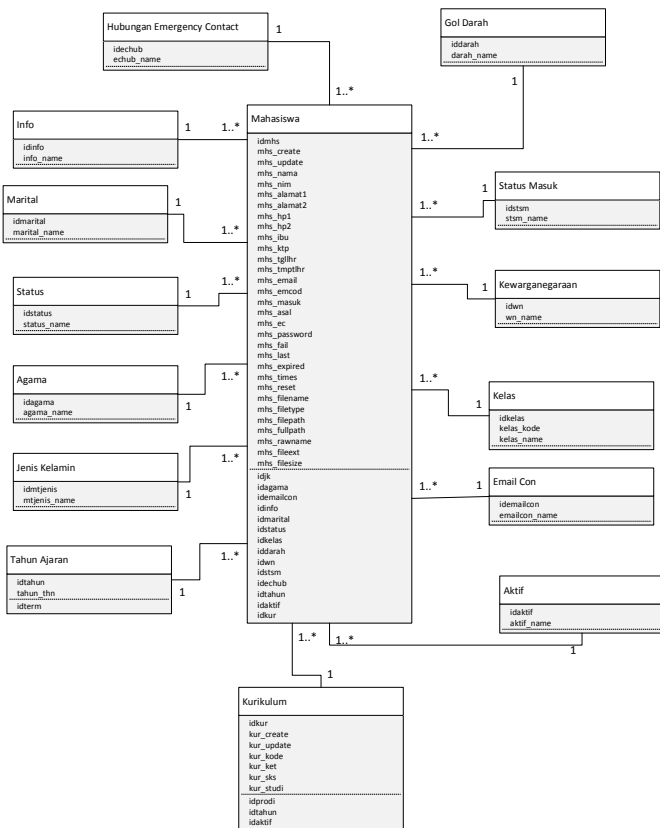
TABEL V  
 DESKRIPSI USECASE DELETE MAHASISWA

Use Case	Delete Mahasiswa
Brief Description	Use Case ini memungkinkan Ketua Jurusan untuk menghapus data mahasiswa tertentu yang ada di sistem informasi akademik. Jarang sekali terjadi data mahasiswa yang perlu dihapus. Hal ini terjadi umumnya jika terdapat mahasiswa yang <i>cancel join</i> (tidak jadi bergabung).
Actor	Ketua Jurusan
Pre Condition	Ketua Jurusan membuka halaman View Mahasiswa lalu melakukan pencarian data mahasiswa tertentu yang ingin dihapus dari sistem.
Main Flow	Didalam menu <i>Delete</i> Mahasiswa, sistem akan melakukan pengecekan apakah data mahasiswa tersebut ada hubungan dengan data nilai. Jika ada maka data mahasiswa tersebut tidak akan bisa di hapus. Sistem juga akan melakukan pengecekan terhadap data transkrip. Jika mahasiswa tersebut ada relasi dengan data transkrip maka data mahasiswa tersebut tidak dapat dihapus dari sistem.
Alternative Flow	Admin dapat memilih menu <i>refresh</i> untuk menyegarkan tampilan atau memilih menu <i>back</i> untuk kembali ke menu <i>view</i> mahasiswa.
Post Condition	Sistem akan menghapus data mahasiswa tersebut dan tampilan akan kembali ke menu <i>view</i> mahasiswa.

TABEL VI  
 DESKRIPSI USECASE VIEW DETAIL

Use Case	View Detail
Brief Description	Use Case ini memungkinkan Ketua Jurusan untuk melihat detail data mahasiswa tertentu.
Actor	Ketua Jurusan
Pre Condition	Ketua Jurusan membuka halaman View Mahasiswa lalu melakukan pencarian data mahasiswa. Jika data mahasiswa ditemukan, Ketua Jurusan dapat memilih menu Detail.
Main Flow	Didalam menu Detail, data mahasiswa ditampilkan dalam bentuk tabulasi-tabulasi yang terdiri dari tabulasi data pribadi, tabulasi kontak dalam keadaan darurat, tabulasi status masuk, tabulasi email, tabulasi login, dan tabulasi data akademik.
Alternative Flow	Ketua Jurusan dapat memilih menu <i>refresh</i> untuk menyegarkan tampilan atau memilih menu <i>back</i> untuk kembali ke menu view mahasiswa.
Post Condition	-

B. Class Diagram



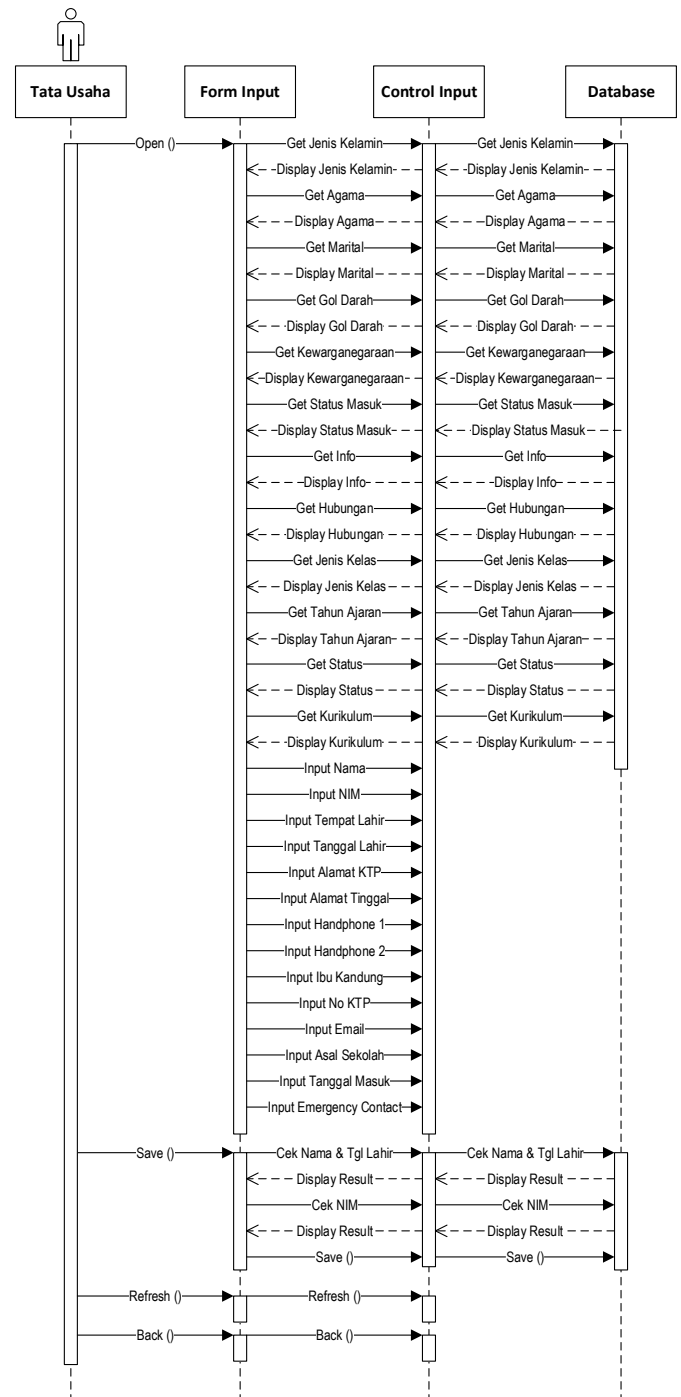
Gbr. 3 Class Diagram.

Pada sistem yang dibuat sesuai Gbr.3, *Class Mahasiswa* berelasi *many to one* dengan kelas kurikulum. Hal ini disebabkan karena setiap mahasiswa baru dalam suatu program studi yang ada di Universitas Islam As-syafiiyah harus didaftarkan pada suatu kurikulum. Misalnya mahasiswa

bernama budi didaftarkan pada kurikulum INF2016 yang berarti kurikulum informatika tahun 2016. Sedangkan *Class Kurikulum* berelasi *one to many* dengan *Class Mahasiswa* yang artinya suatu kurikulum yang ada di suatu program studi di Universitas Islam As-syafiiyah harus didaftarkan di satu atau lebih mahasiswa pada program studi tersebut.

C. Sequence Diagram

Perancangan penggambaran *Sequence diagram* diperlihatkan pada Gbr. 4.



Gbr. 4 Sequence Diagram Input Mahasiswa.

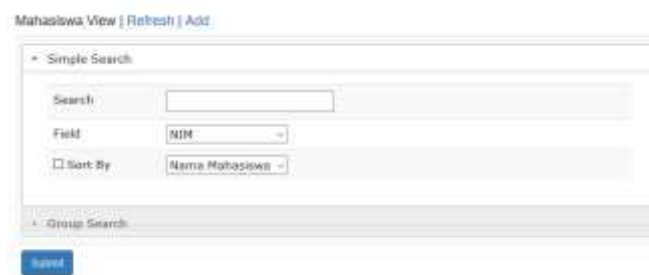
TABEL VII  
 DESKRIPSI SEQUENCE DIAGRAM INPUT MAHASISWA

Message	Description
Get Jenis Kelamin	Message ini berfungsi untuk memerintakan sistem mengambil data list jenis kelamin dari database.
Display Jenis Kelamin	Message ini berfungsi untuk memerintahkan sistem menampilkan data list jenis kelamin dari database ke form input.
Get Agama	Message ini berfungsi untuk memerintakan sistem mengambil data list agama dari database.
Display Agama	Message ini berfungsi untuk memerintahkan sistem menampilkan data list agama dari database ke form input.
Get Marital	Message ini berfungsi untuk memerintakan sistem mengambil data list marital dari database.
Display Marital	Message ini berfungsi untuk memerintahkan sistem menampilkan data list marital dari database ke form input.
Get Golongan Darah	Message ini berfungsi untuk memerintakan sistem mengambil data list golongan darah dari database.
Display Golongan Darah	Message ini berfungsi untuk memerintahkan sistem menampilkan data list golongan darah dari database ke form input.
Get Kewarganegaraan	Message ini berfungsi untuk memerintakan sistem mengambil data list kewarganegaraan dari database.
Display Kewarganegaraan	Message ini berfungsi untuk memerintahkan sistem menampilkan data list kewarganegaraan dari database ke form input.
Get Status Masuk	Message ini berfungsi untuk memerintakan sistem mengambil data list status masuk dari database.
Display Status Masuk	Message ini berfungsi untuk memerintahkan sistem menampilkan data list status masuk dari database ke form input.
Get Info	Message ini berfungsi untuk memerintakan sistem mengambil data list info dari database.
Display Info	Message ini berfungsi untuk memerintahkan sistem menampilkan data list info masuk dari database ke form input.
Get Hubungan	Message ini berfungsi untuk memerintakan sistem mengambil data list hubungan dari database.
Display Hubungan	Message ini berfungsi untuk memerintahkan sistem menampilkan data list hubungan dari database ke form input.

Get Jenis Kelas	Message ini berfungsi untuk memerintakan sistem mengambil data list jenis kelas dari database.
Display Jenis Kelas	Message ini berfungsi untuk memerintahkan sistem menampilkan data list jenis kelas dari database ke form input.
Get Tahun Ajaran	Message ini berfungsi untuk memerintakan sistem mengambil data list tahun ajaran dari database.
Display Tahun Ajaran	Message ini berfungsi untuk memerintahkan sistem menampilkan data list tahun ajaran dari database ke form input.
Get Status	Message ini berfungsi untuk memerintakan sistem mengambil data list status dari database.
Display Status	Message ini berfungsi untuk memerintahkan sistem menampilkan data list status dari database ke form input.
Get Kurikulum	Message ini berfungsi untuk memerintakan sistem mengambil data list kurikulum dari database.
Display Kurikulum	Message ini berfungsi untuk memerintahkan sistem menampilkan data list kurikulum dari database ke form input.

#### D. Membuat Tampilan Website

1) *Halaman Mahasiswa View*: Halaman ini berfungsi sebagai *form* untuk pencarian data mahasiswa. Model pencarian ada dua macam yaitu *simple search* dan *group search*. Data mahasiswa akan menjadi sering digunakan kedepannya sehingga model pencariannya harus bersifat fleksibel yaitu pencarian secara spesifik atau pencarian secara berkategori atau kelompok.



Gbr. 5 Interface Simple Search.

Pencarian dengan cara *Simple Search* pada Gbr.5 dilakukan dengan memilih kategori pencarian yang ingin dicari, misal mencari NIM atau mencari nama mahasiswa. Setelah itu memasukkan kata kunci yang ingin dicari, misalnya seluruh mahasiswa dengan NIM yang mengandung angka 342016 (34 berarti kode program studi Teknik Informatika dan 2016 berarti mahasiswa yang masuk pada tahun 2016).

No	Nama Mahasiswa	NIM	Tempat Lahir	Tanggal Lahir	Handphone	Tools
1						Edit Detail Detail Transkrip KHS
2						Edit Detail Detail Transkrip KHS
3						Edit Detail Detail Transkrip KHS
4						Edit Detail Detail Transkrip KHS

Gbr. 6 Simple Search Result.

No	Nama Mahasiswa	NIM	Tempat Lahir	Tanggal Lahir	Handphone	Tools
1						Edit Detail Detail Transkrip KHS
2						Edit Detail Detail Transkrip KHS
3						Edit Detail Detail Transkrip KHS

Gbr. 7 Group Search.

No	Nama Mahasiswa	NIM	Tempat Lahir	Tanggal Lahir	Handphone	Tools
1						Edit Detail Detail Transkrip KHS
2						Edit Detail Detail Transkrip KHS
3						Edit Detail Detail Transkrip KHS

Gbr. 8 Group Search Result.

Pada Gbr.6 Jika pencarian ditemukan maka sistem akan menampilkan tersebut secara acak atau dengan urutan yang acak. Jika ingin melakukan pengurutan maka dapat dilakukan dengan memberikan tanda *checkbox* pada pilihan *Sort By* lalu memilih untuk mengurutkan berdasarkan kolom apa, misalnya mengurutkan berdasarkan NIM atau berdasarkan nama mahasiswa. Data yang ditampilkan hanya nama mahasiswa, NIM, tempat lahir, tanggal lahir, dan nomor handphone saja. Jika ingin melihat data mahasiswa tersebut secara lengkap dapat memilih menu detail yang terdapat pada kolom *tools*. Data detail akan ditampilkan per mahasiswa

saja. Pencarian dengan cara *Group Search* seperti pada Gbr.7 menggunakan kolom-kolom yang sering digunakan sebagai acuan pengelompokan, misalnya kolom kurikulum, kolom status, dan kolom jenis kelas. Jika ingin mencari mahasiswa dengan jenis kelas reguler maka dapat dilakukan dengan memberi tanda *checkbox* pada Kelas dan pilih jenis kelasnya kelas reguler.

Kombinasi pencarian pada pencarian data dapat dilakukan dengan model *Group Search* seperti pada Gbr. 8. Pencarian secara kombinasi ini dapat dilakukan dengan cara memberi tanda *checkbox* lebih dari satu. Misalnya untuk menampilkan seluruh data mahasiswa yang terdapat pada kelas karyawan yang berstatus aktif yang tergabung pada kurikulum INF2014 maka dapat dilakukan dengan memberi tanda *checkbox* pada Kelas dan memilih jenis kelas karyawan, lalu memberi tanda *checkbox* pada Status dan memilih status mahasiswa aktif, setelah itu beri tanda *checkbox* pada Kurikulum dan memilih kurikulum INF2014.

Pada sistem informasi akademik ini juga dapat melakukan kombinasi pencarian *Simple Search* dan *Group Search* misalnya ingin melakukan pencarian secara spesifik yang dapat memiliki *result* dengan jumlah lebih dari satu row. Contohnya misalnya ingin melakukan pencarian data seluruh mahasiswa angkatan tahun 2016 yang masuk dalam kelas karyawan yang masih berstatus aktif yang tergabung dalam kurikulum Informatika Tahun 2014 dan data diurutkan berdasarkan NIM. Hal ini bisa dilakukan dengan cara melakukan pengaturan pencarian pada *Group Search* dan *Simple Search* sekaligus atau secara bersamaan.

2) *Halaman Mahasiswa Add*: Halaman pada Gbr.9 ini berfungsi sebagai *form* untuk menambahkan data mahasiswa. Pada saat menginput data, sistem akan melakukan validasi untuk meminimalisir kesalahan-kesalahan penginputan yang dapat terjadi. Validasi-validasi ini diantaranya adalah:

- Kolom NIM hanya dapat diinput angka saja dan tidak boleh dikosongkan.
- Kolom Tanggal Lahir hanya dapat diinput tanggal saja dengan cara memunculkan tampilan kalender atau *datepicker* dan tidak boleh dikosongkan.
- Kolom Alamat KTP tidak boleh dikosongkan.
- Kolom Alamat Tinggal tidak boleh dikosongkan.
- Kolom Tempat Lahir tidak boleh dikosongkan.
- Kolom Handphone 1 tidak boleh dikosongkan dan hanya dapat diinput dengan angka saja.
- Kolom Handphone 2 boleh dikosongkan namun jika diisi maka hanya dapat diinput angka saja.
- Kolom Nama Ibu Kandung tidak boleh dikosongkan.
- Kolom No KTP tidak boleh dikosongkan dan hanya dapat diinput angka saja.
- Kolom Email boleh dikosongkan namun jika diinput maka hanya dapat diinput dengan format penulisan email yang benar saja.
- Kolom Tanggal Masuk boleh dikosongkan namun jika diisi maka hanya dapat diinput tanggal saja dengan cara memunculkan tampilan kalender atau *datepicker*.
- Kolom Emergency Contact boleh dikosongkan namun jika diinput maka hanya dapat diinput angka saja.



3) *Halaman Mahasiswa Delete*: Halaman ini berfungsi sebagai form untuk menghapus data mahasiswa. Seperti pada Gbr.10 halaman ini akan menampilkan data NIM, nama, dan tanggal lahir dari data mahasiswa yang akan dihapus. Halaman ini juga berfungsi untuk melakukan pengecekan apakah data yang akan dihapus mempunyai relasi dengan data lainnya misalnya data transkrip dan data bimbingan. Jika data mahasiswa yang akan dihapus memiliki relasi dengan data lainnya maka sistem akan memberikan informasi tersebut dan tombol untuk menghapus data tidak akan muncul. Jika data mahasiswa yang akan dihapus sudah tidak memiliki relasi apapun dengan data-data lainnya maka sistem akan menampilkan pesan konfirmasi apakah sudah yakin untuk menghapus data tersebut. Hal ini penting karena data yang telah dihapus tidak akan bisa ditarik kembali.

4) *Halaman Mahasiswa Edit*: Halaman pada Gbr.11 ini berfungsi sebagai form untuk mengubah data mahasiswa yang telah diinput. Pada dasarnya semua kolom dalam data master mahasiswa dapat diubah datanya namun ada beberapa kolom yang tidak dapat diubah datanya. Diantaranya yaitu kolom kurikulum. Kolom kurikulum dalam data mahasiswa tidak dapat diubah datanya karena seorang mahasiswa yang sudah menjalani perkuliahan tidak dapat melakukan perubahan kurikulum. Kurikulum yang ada di Universitas Islam As-Syafiyah artinya adalah susunan mata kuliah yang seorang mahasiswa ambil dari mulai perkuliahan sampai dengan mahasiswa tersebut lulus. Sehingga jika seorang mahasiswa di *assign* kedalam suatu kurikulum maka kurikulum dari mahasiswa tersebut sudah tidak dapat diubah lagi. Namun untuk beberapa mata kuliah tertentu dapat diambil dari kurikulum lain atau diluar dari kurikulum yang didaftarkan dengan beberapa kondisi misalnya sudah tidak ada dosen yang mampu mata kuliah tersebut karena mata kuliah tersebut sudah digantikan mata kuliah yang lain di kurikulum yang baru.

Mahasiswa Add | Refresh | Back

Field	Value
NIM	<input type="text"/>
Nama Mahasiswa	<input type="text"/>
Tempat Lahir	<input type="text"/>
Tanggal Lahir	<input type="text"/>
Jenis Kelamin	PIRA
Agama	ISLAM
Mantai	BELUM MENIKAH
Golongan Darah	D
Warga Negara	WNI
Alamat KTP	<input type="text"/>
Alamat Tinggal	<input type="text"/>
Handphone 1	<input type="text"/>
Handphone 2	<input type="text"/>
Ibu Kandung	<input type="text"/>
No KTP	<input type="text"/>
Email	<input type="text"/>
Status Masuk	ASLI
Info	TEMAN
Asal Sekolah	<input type="text"/>
Tanggal Masuk	<input type="text"/>
Emergency Contact	<input type="text"/>
Hubungan	BAPAK
Kelas	KELAS REGULER
Tahun	1990 - 01
Status	AKTIF
Kurikulum	BIO2013

Submit

Gbr. 9 Interface halaman "Mahasiswa Add".

Mahasiswa Delete | Refresh | Back

Field	Value
NIM	<input type="text"/>
Nama Mahasiswa	<input type="text"/>
Tanggal Lahir	<input type="text"/>

Submit

Gbr. 10 Interface bagian "Mahasiswa delete".

Mahasiswa Edit | Refresh | Back

Field	Value
NIM	
Nama Mahasiswa	
Tanggal Lahir	
Tempat Lahir	
Jenis Kelamin	
Kelas	
Agama	
Mantel	
Golongan Darah	
Status	
Kurikulum	
Alamat KTP	
Alamat Tinggal	
Handphone 1	
Handphone 2	
No KTP	
Warga Negara	
Status Masuk	
Acal Sekolah	
Emergency Contact	
Hubungan	
Ibu Kandung	
Info	
Email	

Submit

Gbr. 11 Interface bagian "Mahasiswa edit"

## VI. KESIMPULAN

Berdasarkan uraian yang telah diuraikan sebelumnya, disimpulkan bahwa data master mahasiswa adalah data yang sangat penting yang dibutuhkan dalam membangun suatu sistem informasi akademik dalam cakupan yang luas. Hal ini karena data master mahasiswa digunakan sebagai data acuan dalam proses bisnis yang lain dalam lingkungan pengelolaan data akademik. Data master mahasiswa yang masih manual akan mengakibatkan terhambatnya pengembangan sistem informasi akademik lainnya yang terdiri dari beberapa atau banyak sub sistem yang sifatnya paralel.

Selain itu saran untuk selanjutnya adalah data master mahasiswa harus di-maintenance secara real time yang

artinya data mahasiswa harus selalu dalam keadaan ter-update sehingga penggunaannya untuk sistem informasi yang lain dalam lingkup sistem informasi akademik dapat menjadi valid dan dapat diandalkan.

## UCAPAN TERIMA KASIH

Terima kasih kepada semua pihak yang telah membantu penulisan jurnal ini yang tidak dapat disebutkan satu persatu.

## DAFTAR PUSTAKA

- [1] Ariansyah dan Ahmat Josi, "Pengembangan Framework Yii Dalam Pembangunan Sistem Inventaris STMIK Prabumulih Dengan Konsep User Centered Design (UCD)," *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, Vol.03, No.1, Januari 2017.
- [2] Imaniawan, Fabrian Fandi Dwi dan Andi Dwi Riyanto, "Perancangan Sistem Informasi Akademik Berbasis Web Pada Akademi Pertanian HKTI Banyumas," *Jurnal Telematika*, Vol.8, No.1, Februari 2015.
- [3] Suryandani, Farid, Basori dan Dwi Maryono, "Pengembangan Sistem Informasi Akademik Berbasis Web Sebagai Sistem Pengolahan Nilai Siswa Di SMK Negeri 1 Kudus," *JIPTEK*, Vol.X, No.1, Januari 2017.
- [4] Hustinawari, Albert Kurnia Himawan dan Latifah, "Performance Analysis Framework Codeigniter And CakePHP in Website Creation," *International Journal Of Computer Applications*, Vol.94, No.20, May 2014.
- [5] Indrayani, Etin, "Management Of Academic Information System (AIS) at Higher Education In The City Of Bandung," *Procedia – Social And Behavioral Science 103 (2013) 628-636* 13<sup>th</sup> International Educational Technology Conference.
- [6] Pane, Eddis Syahputra dan Muhamad Sadar, "Rancangan Website Sistem Informasi Akademik Sekolah Menengah Kejuruan (SMK) Negeri 5 Pekanbaru," *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, Vol.6, No.2, November 2015.
- [7] Alpiandi, M. Rizki, "Sistem Informasi Akademik Berbasis Web Di SMP Negeri 2 Kecamatan Gaung Anak Serka," *Jurnal SISTEMASI*, Vol.5, No.3, September 2016.
- [8] Susanti, Melan, "Perancangan Sistem Informasi Akademik Berbasis Web Pada SMK Pasar Minggu Jakarta," *Jurnal Informatika*, Vol.3 No.1, April 2016.
- [9] Membara, Eko Putra, Liza Yulianti dan Indra Kanedi, "Sistem Informasi Akademik SMP Negeri Talang Empat Berbasis Web," *Jurnal Media Infotama*, Vol.10, No.1, Februari 2014.
- [10] Gutmans, Andi, Stig Saether Bakken and Derick Rethans, *PHP 5 Power Programming*, Indianapolis: Pearson Education, 2004.
- [11] Djaelangara, Recky T, Rizal Sengkey dan Oktavian A. Lantang, "Perancangan Sistem Informasi Akademik Sekolah Berbasis Web Studi Kasus Sekolah Menengah Atas Kristen 1 Tomohon," *E-Jurnal Teknik Elektro Dan Komputer*, 2015.
- [12] Welling, Luke and Laura Thomson, *PHP and MySQL Web Development*, Indianapolis: Sams Publishing, 2001.
- [13] Griffiths, Adam, *CodeIgniter 1.7 Professional Development Become a CodeIgniter expert with professional tools, techniques, and extended libraries*, Birmingham: Packt Publishing, 2010.
- [14] Octavian, D. Tri, "Web Multi E-Commerce Berbasis Framework CodeIgniter," *Jurnal Teknologi Dan Informatika (TEKNOMATIKA)*, Vol. 5, No.1, Januari 2015.
- [15] Adenowo, Adetokunbo A.A and Basirat A. Adenowo, "Software Engineering Methodologies: A Review Of The Waterfall Model And Object-Oriented Approach," *International Journal of Scientific & Engineering Research*, Vol.4, July 2013.
- [16] Kannan, Vaishnavi, Smita Jhajharia and Seema Verma "Agile vs Waterfall: A Comparative Analysis," *International Journal of Science, Engineering and Technology Research (IJSETR)*, Vol.3, October 2014.