

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI
AKADEMIK STT-NF MODUL RENCANA STUDI
BERBASIS WEB MENGGUNAKAN MVC
FRAMEWORK**

**EDO RIANSYAH
0110212048**



Program Studi Sarjana Teknik Informatika

**STT TERPADU NURUL FIKRI
JAKARTA**

2015

HALAMAN PENGESAHAN

PROPOSAL PENELITIAN ANALISIS DAN PERANCANGAN SISTEM INFORMASI AKADEMIK STT-NF MODUL RENCANA STUDI BERBASIS WEB MENGGUNAKAN MVC FRAMEWORK

OLEH :
EDO RIANSYAH
0110212048

PROPOSAL

Diajukan sebagai salah satu syarat untuk menyusun Skripsi

Jakarta, Oktober 2015
Program Studi Teknik Informatika

Pembimbing I

Ketua Program Studi

(Sirojul Munir, S.Si, M.Kom.)
NIDN 0414047101

(Ahmad Rio Adriansyah, S.Si. M.Si.)
NIDN 0413128601

DAFTAR ISI

HALAMAN PENGESAHAN	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Perumusan Masalah	3
1.3. Tujuan dan Manfaat Penelitian	4
1.4. Batasan Masalah	4
1.5. Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1. Tinjauan Pustaka	6
2.1.1. Pengertian Sistem Informasi Akademik	6
2.1.2. Unified Process	6
2.1.3. UML (Unified Modeling Language)	7
2.1.4. MVC (Model-View-Controller)	10
2.1.5. Framework	11
2.1.6. Yii Framework	12
2.1.7. Black Box Testing	12
2.2. Penelitian Terkait	13
BAB III METODOLOGI PENELITIAN	16
3.1. Tahapan Penelitian	16
3.2. Rancangan Penelitian	19
3.2.1. Metode Pengembangan	19

3.2.2.	Metode Penelitian.....	20
3.2.3.	Sumber dan Tahapan Pengumpulan Data	20
3.2.4.	Rancangan Analisis	21
3.2.5.	Bahan dan Alat.....	23
3.2.6.	Tempat/Lokasi dan Waktu Penelitian	23
	DAFTAR PUSTAKA	26

DAFTAR GAMBAR

Gambar 1 MVC Architecture.....	10
Gambar 2 Unified Process.....	16
Gambar 3 Unified Process Phase.....	17
Gambar 4 Rancangan Penelitian.....	22
Gambar 5 Timeline Penelitian.....	25

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Sekolah Tinggi Teknologi Terpadu Nurul Fikri (biasa disingkat STT Terpadu NF atau STT NF) merupakan lembaga pendidikan tinggi swasta di Indonesia yang memadukan antara keilmuan praktis di bidang teknologi informasi dengan pengembangan kepribadian islami. Pada tahun 2012, STT Terpadu Nurul Fikri resmi berdiri dengan SK Menteri Pendidikan dan Kebudayaan No. 269/E/O/2012. Salah satu tujuan berdirinya STT Terpadu Nurul Fikri adalah “Mencetak Sarjana Komputer Berakhhlak Mulia, Profesional, dan Bersertifikasi IT” yang memenuhi kebutuhan sarjana teknologi informasi dengan tetap konsisten mengusung nilai-nilai profesionalisme dan keislaman.

Semakin meningkatnya jumlah mahasiswa di STT NF, maka dibutuhkan peningkatan kualitas pelayanan kepada mahasiswa. Salah satu bentuk peningkatan kualitas pelayanan kepada mahasiswa adalah dengan membangun sebuah Sistem Informasi Akademik. Dalam melayani terlaksananya sistem belajar mengajar maka sistem pelayanan akademik sangat dibutuhkan dan merupakan sebuah hal yang sangat vital untuk terlaksananya proses belajar mengajar secara kontinuitas dan berkualitas. Tanpa adanya sebuah sistem informasi akademik maka proses belajar mengajar akan menjadi bukan sebuah proses belajar mengajar [1]. Sistem Informasi Akademik mampu menangani kegiatan-kegiatan akademik dari suatu badan organisasi (dalam hal ini perguruan tinggi). Kegiatan-kegiatan akademik seperti pencatatan data mahasiswa dan dosen, absensi mahasiswa, kartu rencana studi, kartu hasil studi, administrasi keuangan bisa dilakukan secara mudah dengan menggunakan Sistem Informasi Akademik (disingkat SIAK). Selain itu bisa mengurangi frekuensi terjadinya kesalahan pada saat pengolahan data akademik.

Berkaitan dengan penyediaan informasi akademik bagi semua *stakeholder*, dalam perancangan sistem informasi akademik terdapat pula modul-modul yang berkaitan dengan perencanaan sumber daya yang diperlukan untuk menghasilkan informasi akademik [2]. Modul-modul tersebut saling terintegrasi satu sama lain sehingga membentuk suatu sistem besar yaitu Sistem Informasi Akademik. Modul-modul tersebut antara lain Modul PMB, Modul Administrasi Keuangan, Modul Kurikulum, Modul Penjadwalan, Modul Rencana Studi, Modul Absensi, Master Data.

Salah satu modul yang ada pada Sistem Informasi Akademik adalah modul Rencana Studi. Dalam modul ini salah satu fungsinya adalah mahasiswa dapat menginput data pengajuan mata kuliah yang akan diambil per semester atau Kartu Rencana Studi (KRS) dan dapat melihat serta mencetak nilai hasil studi atau Kartu Hasil Studi (KHS) dari seluruh mata kuliah yang sudah ditempuh. Saat ini pengelolaan KRS dan KHS di STT NF sudah menggunakan aplikasi SIAK berbasis web, namun aplikasi ini masih banyak kekurangan seperti pada sisi keamanan yang sangat rentan terhadap *SQL Injection*, *XSS (Cross-site Scripting)* sehingga aplikasi belum bisa diakses secara *online*. Kekurangan lainnya adalah *output* yang dihasilkan oleh sistem belum sesuai dengan kebutuhan akademis. Untuk melakukan pengembangan pada aplikasi ini juga sangat sulit karena aplikasi tidak dibangun dengan menggunakan teknologi yang mudah dikembangkan dalam jangka panjang salah satunya adalah menggunakan MVC framework.

Framework digunakan oleh *developer* untuk memudahkan pembangunan aplikasi web yang dapat berupa sekumpulan *library* yang berisi fungsi, *tools*, ataupun class-class, dan digunakan sebagai kerangka dalam pembangunan aplikasi web. Penggunaan framework diharapkan membuat pengembangan aplikasi menjadi rapi dan bersih, memiliki struktur yang optimal, dan *reusable*. Struktur aplikasi yang lebih rapi dan teratur, biasanya menggunakan struktur MVC yang melakukan pemisahan antara business logic dengan presentation. Pemisahan dilakukan dengan tujuan agar setiap perubahan yang terjadi pada presentation logic atau business logic tidak memberikan pengaruh satu sama lainnya yang kompleks [3].

Dari uraian tersebut maka diperlukan Sistem Informasi Akademik modul rencana studi berbasis web yang dibangun menggunakan teknologi yang mudah dikembangkan dalam jangka panjang (menggunakan MVC *framework*), aplikasi akan lebih *secure* dan *reusable*, karena framework dibangun dari pengalaman membangun aplikasi.

1.2. Perumusan Masalah

Masalah yang diangkat dalam Tugas Akhir ini adalah:

“Apakah dengan membangun Aplikasi SIAK modul rencana studi dengan menggunakan teknologi MVC framework bisa menjawab kebutuhan STT Nurul Fikri dalam membangun aplikasi SIAK yang mudah dikembangkan dengan sistem yang lebih aman?”

Sesuai dengan latar belakang yang telah dijelaskan, maka akan diuraikan pokok-pokok masalah yang akan dikaji adalah sebagai berikut :

1. Apakah dengan membangun kembali aplikasi SIAK modul rencana studi dengan teknologi MVC framework bisa menjawab kebutuhan STT NF dalam membangun aplikasi SIAK yang memberikan kemudahan pengembangan untuk jangka panjang?
2. Apakah dengan membangun aplikasi SIAK menggunakan teknologi framework bisa menjamin keamanan sistem dari kerentanan yang umum ditemukan?
3. Apakah modul rencana studi yang akan dibangun dapat memenuhi kebutuhan organisasi STT NF pada pengolahan data KRS dan KHS mahasiswa saat ini dan juga bisa sebagai salah satu bentuk peningkatan kualitas pelayanan kepada civitas akademik?
4. Apakah Aplikasi SIAK yang dibangun dapat diterapkan pada aplikasi client pada berbagai platform aplikasi seperti web, mobile dan desktop?

1.3. Tujuan dan Manfaat Penelitian

Penyusunan tugas akhir ini memiliki tujuan dan manfaat untuk.

1. Menghasilkan sebuah aplikasi SIAK modul rencana studi yang mudah dikembangkan.
2. Menghasilkan sebuah aplikasi Sistem Informasi Akademik modul rencana studi untuk STT Terpadu Nurul Fikri yang aman dari kerentanan dan bisa diakses secara *online* baik melalui perangkat komputer maupun perangkat *mobile*.
3. Membantu Bagian Akademik di dalam melakukan pengolahan data KRS dan KHS mahasiswa.
4. Membantu mempermudah Mahasiswa di dalam mengajukan KRS dan melihat hasil studi dari awal semester hingga semester akhir perkuliahan.
5. Membantu Dosen dalam pengolahan data nilai Mahasiswa.

1.4. Batasan Masalah

Batasan masalah pada Tugas Akhir ini antara lain:

1. Pembuatan SIAK modul rencana studi menggunakan Yii Framework dengan tahapan pengembangan melalui metode pendekatan Unified Process.
2. Alur sistem informasi ini dibuat sesuai dengan sistem yang sedang berjalan di STT NF.
3. Pembuatan sistem informasi akademik modul Rencana Studi berbasis web ini mengambil obyek penelitian di STT NF.
4. Penelitian ini tidak melakukan pengujian keamanan aplikasi.

1.5. Sistematika Penulisan

Semua kegiatan yang mendukung Tugas Akhir ditulis dengan sistematika sebagai berikut :

Bab I : Pendahuluan

Bab ini berisi latar belakang dari penulisan proposal tugas akhir, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah, dan sistematika dari penulisan proposal tugas akhir ini

Bab II : Landasan Teori

Bab ini berisikan mengenai pembahasan teori pengembangan sistem dengan metode pendekatan *unified process*, teori mengenai desain sistem menggunakan UML, teori mengenai pengembangan sistem menggunakan framework, konsep MVC, dan pembahasan mengenai Yii Framework.

Bab III : Metodologi Penelitian

Bab ini berisi tahapan yang dilakukan dalam penelitian diantaranya tahapan pengembangan sistem dengan metode *unified process*. Rancangan penelitian yang menjelaskan mengenai cara menyelesaikan permasalahan, jenis penelitian yang dilakukan, rancangan analisis dan pengumpulan data, tempat lokasi serta waktu dilaksanakannya penelitian.

BAB II

LANDASAN TEORI

Dalam bab ini akan dijelaskan mengenai teori/algoritma/model/fokus teknologi yang digunakan, teori / definisi terkait dan relevan dengan judul tugas akhir.

2.1. Tinjauan Pustaka

2.1.1. Pengertian Sistem Informasi Akademik

Sistem informasi akademik adalah suatu sistem yang mengolah data-data akademik pada suatu instansi pendidikan baik formal maupun informal dari tingkat dasar sampai tingkat perguruan tinggi. Secara umum data-data yang diolah dalam sistem informasi akademik meliputi data guru, data siswa, data mata pelajaran dan jadwal mengajar dan data-data lain yang bersifat umum berdasarkan kebutuhan masing-masing lembaga pendidikan. Secara singkat sistem informasi akademik dapat diartikan aplikasi untuk membantu memudahkan pengelolaan data-data dan informasi yang berkaitan dengan instansi pendidikan [4].

2.1.2. Unified Process

Unified Process merupakan salah satu *software development* proses yang telah mendukung konsep berorientasi objek, *Unified Process* menggunakan *Unified Modeling Language* (UML) sebagai notasi dalam menggambarkan proses pengembangan perangkat lunak. Dalam Unified Process, pengembangan perangkat lunak dilakukan dalam sejumlah siklus yang berkelanjutan. Setiap siklus akan berakhir pada release sebuah produk. Setiap siklus tersebut terdiri atas empat fase yang berbeda, yaitu inception, elaboration, construction, dan transition. Setiap fase dapat terdiri atas sejumlah iterasi workflow (requirement, analysis, design, implementation, dan test). Selanjutnya, workflow menentukan urutan aktivitas yang perlu dilakukan dan artifak apa yang harus dihasilkan [5].

2.1.3. UML (Unified Modeling Language)

UML adalah bahasa untuk menspesifikasi, menvisualisasikan, membangun dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti dalam pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep berorientasi object. UML dibuat oleh Grady Booch, James Rumbaugh dan Ivar Jacobson dibawah bendera rational Software Corp. UML menyediakan notasi-notasi yang membantu memodelkan perangkat lunak, namun hampir dalam sebuah bidang yang membutuhkan pemodelan [6].

UML menyediakan kemampuan untuk menangkap karakteristik sebuah sistem dengan menggunakan notasi-notasi tertentu. UML menyediakan sederetan notasi sederhana yang mudah dipahami untuk mendokumentasikan sistem berdasarkan prinsip-prinsip perancangan berorientasi objek. UML sudah menjadi standardized modelling language yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan software agar bisa menyelesaikan tugas-tugas seperti [6]:

1. Spesifikasi
2. Visualisasi
3. Desain Arsitektur
4. Konstruksi
5. Simulasi dan testing
6. Dokumentasi
7. Sketsa

Didalam UML terdapat pengelompokan diagram sistem. Diagram ini berbentuk grafik yang menunjukan elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram

merupakan bagian dari view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu.

Jenis-jenis diagram dalam UML [6] :

a. *Use Case Diagram*

Menggambarkan sejumlah eksternal actors dan hubungannya ke Use Case yang diberikan oleh sistem. Use Case adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dari dokumentasi dari Use Case symbol namun juga dapat dilakukan dalam activity diagrams. Use Case digambarkan hanya yang dilihat dari luar oleh actor (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

b. *Class Diagram*

Menggambarkan struktur statis class di dalam sistem. Class dapat berhubungan dengan yang lain melalui berbagai cara : associated (terhubung satu sama lain), depended (satu class tergantung atau menggunakan class yang lain), specialized(satu class merupakan spesialisasi dari class yang lainnya), atau packed (group bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa class diagram.

c. *State Diagram*

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan, state class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

d. *Sequence Diagram*

Menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan sejumlah pesan yang dikirim antara

object juga interaksi antar object , sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

e. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti sequence diagrams. Dalam menunjukan pertukaran pesan, collaboration diagram menggambarkan object dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan, gunakan sequence diagram tapi jika penekanannya pada konteks gunakan collaboration diagrams.

f. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti Use Case atau interaksi.

g. Component Diagram

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa source code, komponen biner atau class yang diimplementasikan sehingga membuat pemetaan dari logical view ke component view.

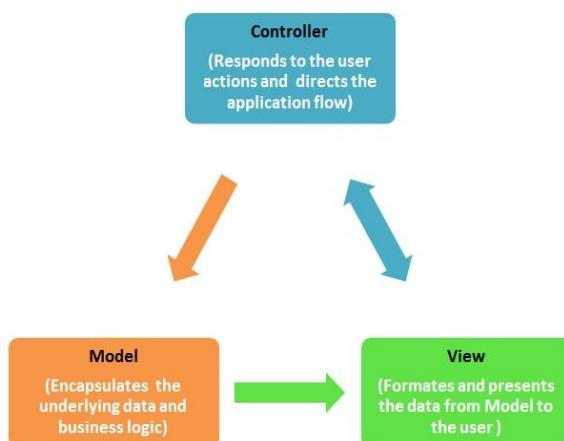
h. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Di dalam nodes, executable dan object yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

2.1.4. MVC (Model-View-Controller)

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu [7] :

- Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
- View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
- Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.



Gambar 1. MVC Architecture

Secara umum control flow arsitektur MVC melakukan hal seperti :

- Pengguna berinteraksi dengan user interface dengan beberapa cara (sebagai contoh, menekan tombol mouse).
- Controller menangani event input melalui user interface, biasanya via registered handler atau callback, kemudian mengkonversi event menjadi user action yang tepat dan dapat dimengerti oleh model.
- Controller memberitahukan kepada model mengenai user action yang memungkinkan terjadinya perubahan pada state model. (Sebagai contoh, controller meng-update shopping cart user.)
- Query model digunakan untuk men-generate user interface dengan tepat (contohnya, list konten yang ada pada shopping cart). View mendapatkan data melalui model, dalam beberapa implementasi, controller memberikan instruksi kepada view untuk me-render hasil query. Namun ada juga model yang secara otomatis memberitahukan view mengenai perubahan state yang membutuhkan update pada tampilan.
- User Interface akan menunggu interaksi lebih lanjut dari pengguna, dimana akan memulai kembali siklus control flow.

2.1.5. Framework

Framework adalah kerangka kerja. Framework digunakan oleh developer untuk memudahkan pembangunan aplikasi web yang dapat berupa sekumpulan library yang berisi fungsi, tools, ataupun class-class, dan digunakan sebagai kerangka dalam pembangunan aplikasi web. Umumnya didalam framework tersebut telah menyediakan solusi untuk akses database, authentication, templating, controls, dan fungsi-fungsi lainnya. Penggunaan framework diharapkan membuat pengembangan aplikasi menjadi rapi dan bersih, memiliki struktur yang optimal, dan reusable. Struktur aplikasi yang lebih rapi dan teratur, biasanya menggunakan struktur MVC yang melakukan pemisahan antara business logic dengan presentation. Pemisahan dilakukan dengan tujuan agar setiap perubahan yang terjadi pada presentation logic atau

business logic tidak memberikan pengaruh satu sama lainnya yang kompleks [3].

2.1.6. Yii Framework

Yii adalah kerangka kerja PHP berbasis-komponen dengan performansi tinggi untuk pengembangan aplikasi Web berskala-besar. Ia menyediakan resuabilitas maksimum dalam pemrograman Web dan bisa mengakselerasi proses pengembangan secara signifikan. Nama Yii (dieja sebagai /i:/) singkatan dari easy, efficient dan extensible (mudah, efisien, dan bisa diperluas). Yii adalah kerangka kerja pemrograman umum Web yang bisa dipakai untuk mengembangkan semua jenis aplikasi Web. Oleh karena Yii sangat ringan dan dilengkapi dengan solusi caching yang memuaskan, ia sangat cocok untuk pengembangan aplikasi dengan lalu lintas-tinggi, seperti portal, forum, sistem manajemen konten (CMS), sistem e-commerce, dll.

Seperti kebanyakan PHP framework, Yii adalah MVC framework. Yii melampaui PHP framework lain dalam hal efisiensi, kaya-fitur, dan didokumentasikan dengan jelas. Yii didesain dengan hati-hati dari awal agar sesuai untuk pengembangan aplikasi Web secara serius. Yii bukan berasal dari produk pada beberapa proyek maupun konglomerasi pekerjaan pihak ketiga. Yii adalah hasil dari pengalaman kaya para pembuat pada pengembangan aplikasi Web dan investigasi serta refleksi kerangka kerja pemrograman Web paling populer dan aplikasi.

Yii saat ini mempunyai 2 versi utama: 1.1 dan 2.0. Versi 1.0 adalah generasi lama dan sekarang dalam mode pemeliharaan. Versi 2.0 adalah penulisan ulang lengkap dari Yii. Mengadopsi teknologi terbaru dan protokol, termasuk composer, PSR, namespaces, trait, dan sebagainya. Versi 2.0 merupakan generasi sekarang dari framework Yii [8].

2.1.7. Black Box Testing

Black box testing, dilakukan tanpa pengetahuan detil struktur internal dari sistem atau komponen yang dites. juga disebut sebagai behavioral testing,

specification-based testing, input/output testing atau functional testing. Black box testing berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software. Black-box testing merupakan pendekatan pengujian yang ujinya diturunkan dari spesifikasi program atau komponen. Sistem merupakan ‘kotak hitam’ yang perlakunya hanya dapat ditentukan dengan mempelajari input dan output yang berkaitan. Nama lain untuk pengujian ini ialah pengujian fungsional karena penguji hanya berkepentingan dengan fungsionalitas dan bukan implementasi perangkat lunak (Sommerville 2001) [9].

2.2. Penelitian Terkait

Berikut ini penulis menjabarkan penelitian yang terkait dengan penelitian yang dilakukan oleh penulis. Berdasarkan semua penelitian terkait yang diambil menyimpulkan bahwa konsep MVC framework dapat mempermudah developer dalam membangun suatu sistem dan banyak keuntungan lainnya yang bisa didapat oleh developer seperti penghematan waktu dalam pengembangan sistem, struktur program menjadi rapi. Penelitian ini juga menggunakan pendekatan *unified process* untuk pengembangan perangkat lunak. Metode ini menghasilkan dokumentasi pengembangan aplikasi yang baik dengan menggunakan UML.

1. Pada penelitian “Pengembangan Aplikasi E-Commerce PT. Global Elektronik Semarang Dengan Metode Unified Process” disimpulkan bahwa menggunakan metode Unified Process menghasilkan dokumentasi pengembangan aplikasi yang lebih mudah dipahami dengan adanya UML pada metode Unified Process [10].
2. Pada penelitian ”Implementasi framework Code Igniter untuk aplikasi situs portal berita” disimpulkan bahwa dengan mengimplementasikan Framework codeigniter, seorang yang bukan webmaster sekalipun dapat melakukan pengelolaan web menjadi dinamis sehingga penyajian informasi dapat dilakukan dengan cepat. Dengan menggunakan Framework codeigniter, proses update informasi tidak hanya dimonopoli oleh seorang administrator saja. Dengan demikian akan didapatkan penghematan waktu, tenaga dan

biaya, selain itu proses administrasi situs dapat dilakukan secara mudah karena bentuk pengelolaannya terdistribusi [11].

3. Pada Penelitian “PURCHASE REQUESTION BERBASIS WEB MENGGUNAKAN METODE MODEL VIEW CONTROLLER DENGAN FRAMEWORK CODEIGNITER Studi Kasus : PT. Batamindo Investment Cakrawala” disimpulkan bahwa aplikasi yang menerapkan MVC sebagai metode dasar pembuatan aplikasinya, mempermudah pengelolaan dan maintenance aplikasi. Aplikasi purchase Requestion berbasis web pada PT. Batamindo Investment Cakrawala mengimplementasikan metode MVC yang menggunakan codeigniter sebagai framework memudahkan karyawan dalam proses rencana pembelian barang (Purchase Requestion) dan membantu penyimpanan data barang kedalam database dan penerapan metode MVC ini dalam aplikasi dapat meminimalisir waktu apabila terjadi kesalahan program, dan apabila satu komponen dikembangkan tidak akan mengganggu komponen lain. Karena pemisahan antara file model, view, dan controller [12].
4. Pada penelitian “ANALISIS DAN PERANCANGAN SISTEM INFORMASI BERBASIS WEBSITE MENGGUNAKAN ARSITEKTUR MVC DENGAN FRAMEWORK CODEIGNITER (Studi Kasus : Ikatan Pelajar Mahasiswa Kepulauan Riau Yogyakarta)”, disimpulkan bahwa menggunakan arsitektur MVC ini diharapkan programmer dapat memahami bahwa ini merupakan arsitektur yang menstrukturkan perancangan aplikasi. Pemeliharaan aplikasi ini memudahkan programer bila membutuhkan pengembangan dan penambahan lainnya [13].
5. Pada Penelitian “ANALISA DAN PERANCANGAN APLIKASI DORMITORY MANAGEMENT MENGGUNAKAN UNIFIED SOFTWARE DEVELOPMENT PROCESS” disimpulkan bahwa Metode Unified Software Development Process, yang disingkat USDP atau UP, dapat digunakan untuk membangun Aplikasi berbasis Web dengan pendekatan berorientasi obyek. Metode USDP merupakan metode pengembangan perangkat lunak yang use-case driven, incremental dan iterative.

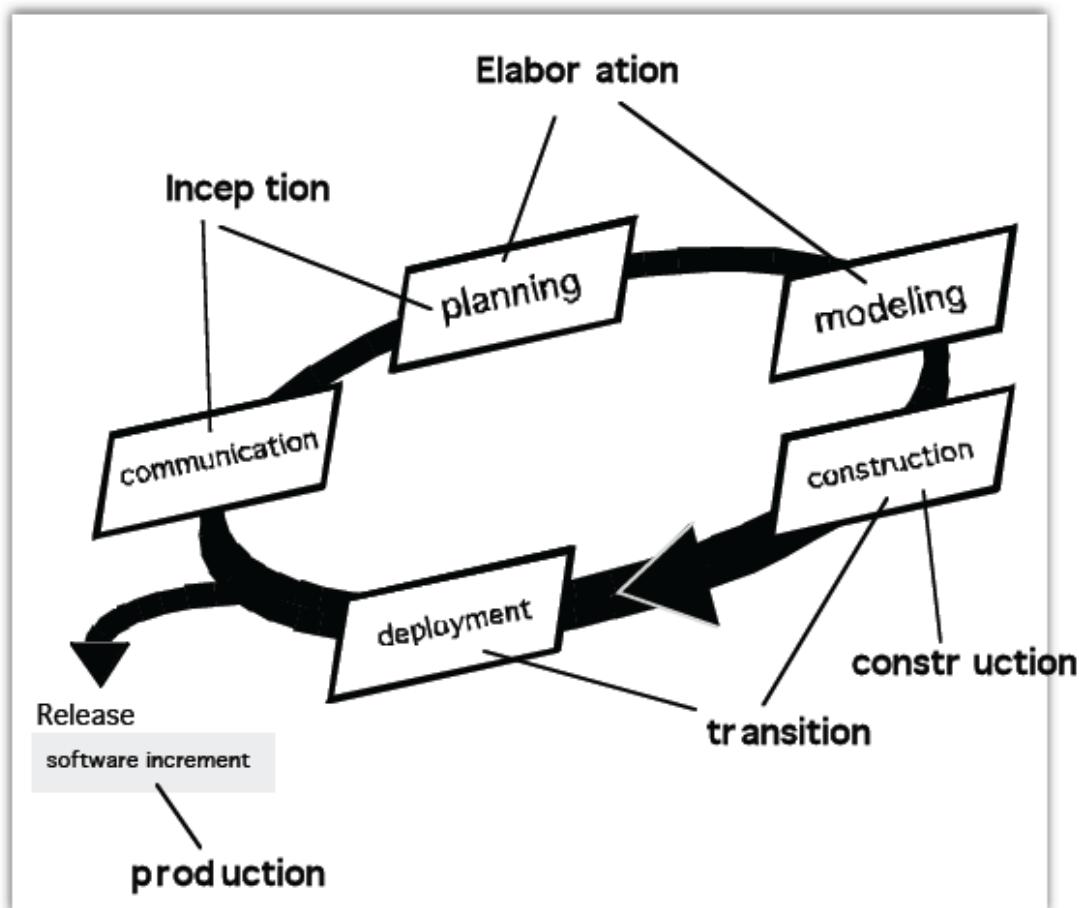
Karakteristik metode ini mengharuskan keterlibatan stakeholder secara aktif [14].

BAB III

METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Metode pengembangan rekayasa perangkat lunak yang digunakan oleh penulis adalah *unified process*. Metode ini dipilih karena menekankan pada use-case dan pendekatan iteratif untuk siklus pengembangan perangkat lunak dan aktifitasnya berfokus pada pengembangan model yang menggunakan UML, sehingga dapat dengan mudah melakukan perbaikan berulang-ulang hingga sesuai dengan kebutuhan user.



Gambar 2. Unified Process

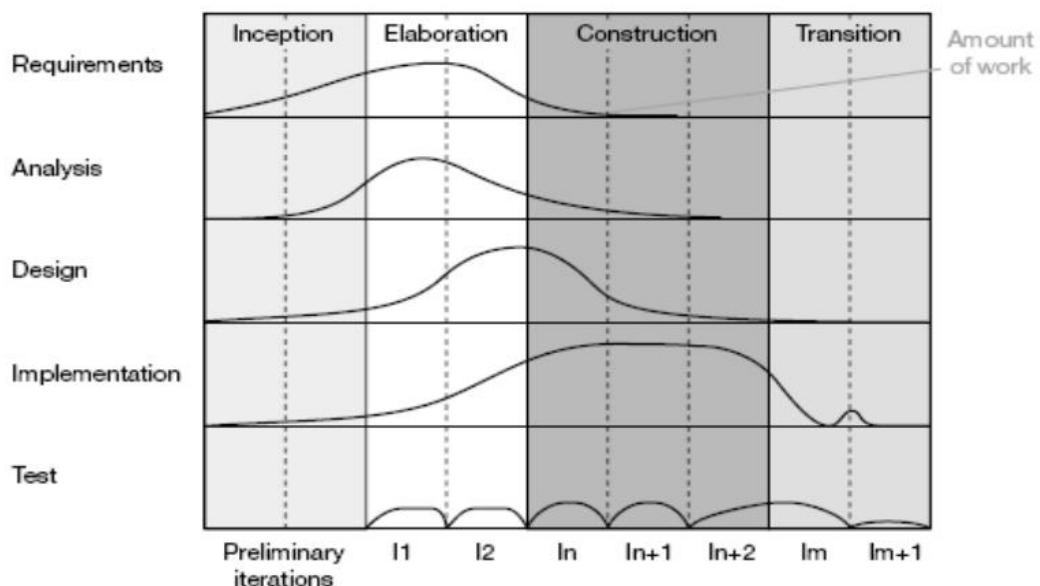
Metode ini digambarkan dalam 2 grafik dimensi, penjelasannya adalah sebagai berikut :

- Dimensi pertama (horizontal)

Mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Setiap fase dapat terdiri dari satu atau beberapa iterasi. Yang termasuk dimensi ini adalah: *Inception*, *Elaboration*, *Construction*, dan *Transition*.

- Dimensi kedua (vertikal)

Mewakili aspek-aspek statis dari proses pengembangan perangkat lunak. Dimensi ini terdiri atas *Requirement*, *Analysis*, *Design*, *Implementation*, *Test*.



Gambar 3. Unified Process Phase

Ada 4 fase yang terdapat dalam unified process, berikut penjelasannya :

1. Fase Inception

- a. Melakukan komunikasi (*Communication*)
 - i. Berkolaborasi dengan *stakeholders*
 - ii. Mengidentifikasi kebutuhan bisnis, yaitu:
 1. Menentukan batasan-batasan dalam proyek. Anggaran dan biaya proyek.
 2. Menentukan tipe model yang akan dipakai dalam proses pengembangan software itu sendiri.
 3. Mendeskripsikan persiapan untuk use case yang akan dibuat

b. Perencanaan (*Planning*)

- i. Merencanakan model proyek yang iterative dan incremental, seperti mengidentifikasi sumber, resiko-resiko yang mungkin terjadi, membuat jadwal proyek.

2. Fase Elaboration

a. Planning

- i. Menganalisa problem, memastikan kembali skup proyek.
- ii. Mengembangkan perencanaan project.
- iii. Menghilangkan kemungkinan munculnya resiko dalam proses perkembangan project itu sendiri, dikarenakan jika sampai terjadi perubahan terhadap project dalam fase berikutnya, akan menyulitkan pengembang project tersebut untuk kembali meninjau ulang.

b. Modeling

- i. Pengimplementasian use case, sebagai perwujudan dari arsitektur system software.

3. Fase Construction

a. Membuat code (Architectural model => operational use-cases)

- i. Pendefinisian kebutuhan dan disain model harus sudah selesai. Versi final dari software increment.
- ii. Semua fungsi dan fitur diimplementasikan ke dalam source code.

b. Testing

- i. Unit test didisain dan dijalankan
- ii. Termasuk tes integrasi (Software Integration Test)
- iii. Use-case digunakan saat tes untuk melihat kesesuaian hasil software dengan use-case

4. Fase Transition

a. Testing

- i. Beta testing, yaitu testing yang dilakukan pihak end user setelah software di-install di perusahaan.
- b. Deployment
 - i. Membuat dokumentasi pendukung (contoh: user manual, panduan instalasi, dll).
- c. End of transition phase
 - i. Yaitu pada saat software sudah siap untuk di-release

3.2. Rancangan Penelitian

3.2.1. Metode Pengembangan

Metode pengembangan pada penelitian menggunakan metode *unified process*. Sifatnya yang iteratif dan bertahap membuat metode ini dipilih oleh penulis, karena melakukan perbaikan berulang-ulang sampai sesuai kebutuhan user. Metode *unified process* ini memiliki tahapan-tahapan utama di setiap fasanya (*core workflow*) antara lain :

1. Requirement

Tujuan dari requirement adalah untuk menemukan dan mencapai persetujuan mengenai apa yang harus dilakukan sistem yang diungkapkan dalam bahasa user. Selanjutnya, requirement sistem tersebut akan digambarkan dalam istilah use case, sehingga output dari requirement ini berupa use case model yang terdiri atas use case, aktor, GUI prototype, dan non-functional requirement.

2. Analysis

Analisis bertujuan untuk menerjemahkan requirement ke dalam bahasa developer dan mengidentifikasi elemen atau entitas utama dari sistem yang diperlukan untuk memenuhi user requirement.

3. Design

Desain berusaha untuk menspesifikasikan secara penuh bagaimana fungsionalitas akan diimplementasikan dengan menggunakan model yang telah dihasilkan dari workflow analisis.

4. Implementation

Implementasi bertujuan untuk mentransformasikan design model ke dalam executable code. Source code program dibuat pada workflow ini.

5. Testing

Tujuan dari pengujian adalah untuk memastikan bahwa sistem dapat menyediakan fungsionalitas yang diperlukan. Pengujian yang dilakukan hanya pengujian secara black box, yaitu menguji fungsionalitas dari perangkat lunak, tanpa harus mengetahui struktur internal program.

3.2.2. Metode Penelitian

Penelitian penulisan merupakan penelitian dengan pendekatan studi kasus maka dari itu penulis menggunakan jenis penelitian kualitatif-deskriptif. Studi kasus termasuk kedalam penelitian analisis deskriptif, yaitu penelitian yang dilakukan proses analisis secara cermat dan tuntas. Penelitian ini tidak menggunakan data angka sehingga menggunakan pendekatan kualitatif. Metode pengembangan sistem penulis menggunakan pendekatan *unified process*.

3.2.3. Sumber dan Tahapan Pengumpulan Data

Sumber data dalam penelitian yang dilakukan oleh penulis dalam tugas akhir ini adalah data primer. Data primer adalah data yang didapatkan secara langsung oleh penulis, proses untuk mendapatkan data primer ini melalui observasi (pengamatan) dan melakukan proses wawancara terhadap *user* yang terlibat langsung dengan penggunaan aplikasi. Data-data ini sangat berguna bagi penulis dalam melakukan proses penelitian tugas akhir ini. Berikut ini adalah tahapan penulis dalam melakukan proses pengumpulan data :

a. Untuk data primer dilakukan proses :

1. Observasi

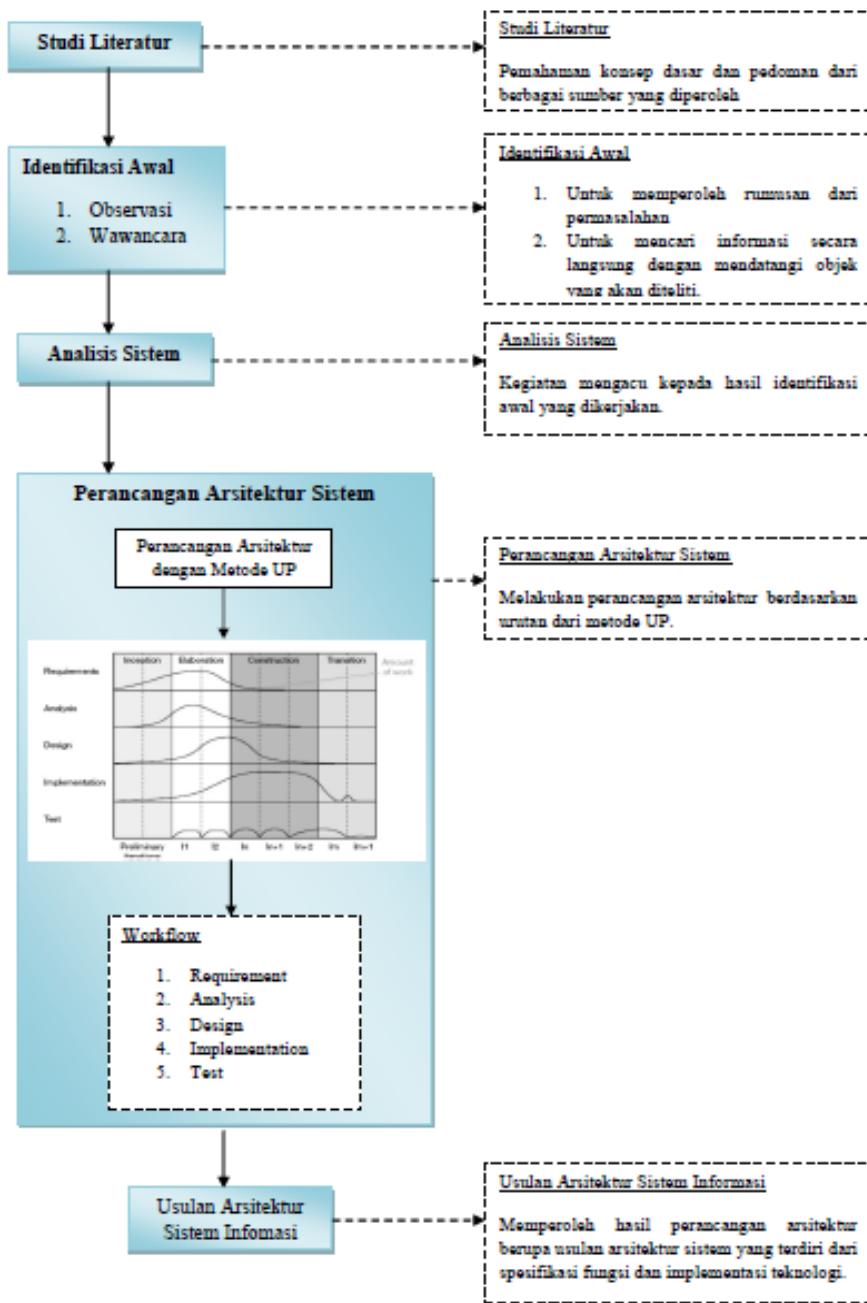
Pada tahapan ini dilakukan proses pengamatan terhadap cara kerja sistem yang sedang berjalan di STT-NF saat ini dan mengamati bagaimana user memanfaatkan aplikasi tersebut.

2. Wawancara

Pada tahapan ini penulis melakukan proses wawancara atau tanya jawab dengan user yang terlibat langsung dalam pengoperasian aplikasi SIAK. proses wawancara ini berguna untuk mengetahui apa saja kendala yang dihadapi oleh user selama pengoperasian, masukan dari user akan sangat membantu dalam proses penggalian kebutuhan yang nantinya akan digunakan oleh penulis dalam penelitian.

3.2.4. Rancangan Analisis

Pada gambar 4 menggambarkan rancangan analisis penelitian dari penulis yang berguna untuk mencapai tujuan yang telah direncanakan sebelumnya.



Gambar 4. Rancangan Penelitian

3.2.5. Bahan dan Alat

Konfigurasi Sistem :

- Sistem akan berbasiskan web framework, aplikasi nantinya akan bisa diakses via web browser.

Adapun perangkat keras yang digunakan untuk membangun aplikasi SIAK antara lain :

- Komputer dengan prosesor Intel-Core i5
- Kapasitas minimal RAM 2 GB
- Kapasitas Hardisk minimal 500 GB
- Sistem Operasi Linux

Software yang digunakan :

- a. *Apache Web Server 2.x*
- b. *Minimal PHP 5.4*
- c. *PostgreSQL 9.x.x* :
 - a. *PGAdmin*
 - b. *PhpPgAdmin*
- d. *Yii 2 Framework*
- e. *IDE : Sublime Text 3*
- f. *PHP Composer*
- g. *Yii2 Library :: Github Repository*

Alat desain yang digunakan :

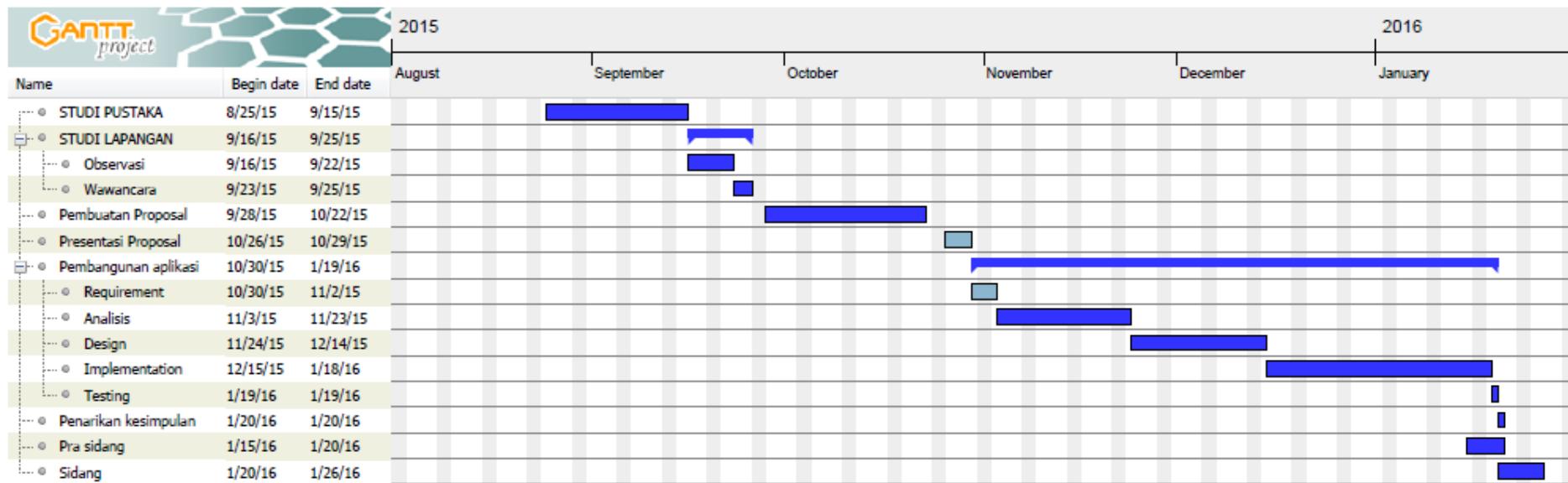
- a. *Enterprise Architect (UML 2.0)*
Execute Query :: Query Designer

3.2.6. Tempat/Lokasi dan Waktu Penelitian

Lokasi dari penelitian ini dilakukan di STT Terpadu Nurul Fikri dengan fokus pada studi kasus analisa dan perancangan sistem informasi akademik STT-NF modul rencana studi. Waktu penelitian ini berlangsung sejak 25

Agustus 2015. Pada gambar 5 merupakan *timeline* dari penelitian yang dilakukan oleh penulis.

Gantt Chart



Gambar 5. Timeline Penelitian

DAFTAR PUSTAKA

- [1] C. Anwar, "SISTEM INFORMASI AKADEMIK ONLINE SEBAGAI PENUNJANG," *Fakultas Teknologi Informasi, Universitas Budi Luhur*, vol. V, no. 1, p. 5, 2010.
- [2] M. Ibnu Choldun R, "PERANCANGAN SISTEM INFORMASI AKADEMIK DENGAN MENIMPLEMENTASIKAN ERP", Laboratorium Sistem Informasi Politeknik Pos Indonesia, Bandung, 2006.
- [3] Yanuar Firdaus AW,Warih Maharani, "ANALISIS PERFORMANSI FRAMEWORK PRADO DAN CAKEPHP PADA APLIKASI WEB AJAX", Departemen Teknik Informatika Institut Teknologi Telkom, Bandung, 2008.
- [4] Julian Chandra W, "IMPLEMENTASI SISTEM INFORMASI AKADEMIK (STUDI KASUS : SMP NEGERI 20 BANDUNG)", Universitas Komputer Indonesia, Bandung, 2012
- [5] Y. Alim, S. and P. S. Sasongko, "PENGEMBANGAN SISTEM INFORMASI ADMINISTRASI PEMERIKSAAN PASIEN DI INSTALASI RADIOLOGI RSUD KAJEN DENGAN UNIFIED PROCESS," *Jurnal Masyarakat Informatika*, vol. II, no. 6, p. 4, 2011.
- [6] H. Bangun, "PEMBANGUNAN WEBSITE SEBAGAI MEDIA PROMOSI KATALOG ONLINE PADA TOKO BUKU DAN PENYEWAAN KOMIK AI READ," *jbptunikompp*, vol. I, no. 1, pp. 49-50, 2009.
- [7] Fajar Y. Zebua, "Framework CodeIgniter – Part 1", Yogyakarta, 2012.
- [8] <http://www.yiiframework.com/doc-2.0/guide-README.html> (diakses pada tanggal 23 Oktober 2015)
- [9] R. M. R. Attamimi, "PEMBANGUNAN SISTEM INFORMASI MANAJEMEN INVENTARISASI BARANG MILIK NEGARA DENGAN METODE AGILE UNIFIED PROCESS (AUP)," IPB, Bogor, 2013.
- [10] A. R. Dhani, S. and B. Noranita, "Pengembangan Aplikasi E-Commerce PT. Global Elektronik Semarang Dengan Metode Unified Process," *Journal of Informatics and Technology*, vol. I, no. 4, pp. 29-39, 2012.
- [11] S. K. Fajriyah, "IMPLEMENTASI FRAMEWORK CODE IGNITER UNTUK APLIKASI SITUS PORTAL BERITA", AMIKOM, Yogyakarta, 2010.
- [12] Vivin Andryan Daya, Eka Suswaini, ST, MT, Martaleli Bettiza, S.Si., M.Sc, "PURCHASE REQUESITION BERBASIS WEB MENGGUNAKAN METODE MODEL VIEW CONTROLLER DENGAN FRAMEWORK CODEIGNITER Studi Kasus : PT. Batamindo Investment Cakrawala", Teknik Perangkat Lunak, Fakultas Teknik, Universitas Maritim Raja Ali Haji, Tanjungpinang, 2013.

- [13] M. IQBAL TANJUNG, “ANALISIS DAN PERANCANGAN SISTEM INFORMASI BERBASIS WEBSITE MENGGUNAKAN ARSITEKTUR MVC DENGAN FRAMEWORK CODEIGNITER Studi Kasus : Ikatan Pelajar Mahasiswa Kepulauan Riau Yogyakarta”, AMIKOM, Yogyakarta, 2011.
- [14] Stanley Karouw, “ANALISA DAN PERANCANGAN APLIKASI DORMITORY MANAGEMENT MENGGUNAKAN UNIFIED SOFTWARE DEVELOPMENT PROCESS”, Universitas Sam Ratulangi, 2012.