

**ANALISA DAN PERANCANGAN SISTEM INFORMASI
AKADEMIK STT-NF MODUL ADMINISTRASI
KEUANGAN BERBASIS WEB MENGGUNAKAN MVC
*FRAMEWORK***

**WATIK ADIDAYA
0110212042**



Program Studi Sarjana Teknik Informatika

**STT TERPADU NURUL FIKRI
JAKARTA**

2015

HALAMAN PENGESAHAN

PROPOSAL PENELITIAN ANALISA DAN PERANCANGAN SISTEM INFORMASI AKADEMIK STT-NF MODUL ADMINISTRASI KEUANGAN BERBASIS WEB FRAMEWORK

OLEH :

WATIK ADIDAYA

0110212042

PROPOSAL

Diajukan sebagai salah satu syarat untuk menyusun Skripsi

Jakarta, Oktober 2015

Program Studi Teknik Informatika

Pembimbing I

Ketua Program Studi

(

)

(

)

DAFTAR ISI

Halaman Pengesahan

Daftar Isi

Daftar Gambar

Daftar Tabel

BAB I PENDAHULUAN

1.1.Latar Belakang Masalah

1.2.Perumusan Masalah

1.3.Tujuan dan Manfaat Penelitian

1.4.Batasan Masalah

1.5.Sistematika Penulisan

BAB II LANDASAN TEORI

2.1.Tinjauan Pustaka

 2.1.1.Sistem Informasi Akademik.....

 2.1.2.UML

 2.1.3.MVC.....

 2.1.4.Framework

 2.1.5.Yii.....

 2.1.6.*Unified Process*

 2.1.7.*Black-box Testing*.....

2.2.Penelitian Terkait

BAB III METODOLOGI PENELITIAN

3.1.Tahapan Penelitian.....
3.2.Rancangan Penelitian.....
3.2.1.Metode Pengembangan
3.2.2.Metode Penelitian.....
3.2.3.Sumber dan Tahapan Pengumpulan Data
3.2.4.Rancangan Analisis.....
3.2.5.Bahan dan Alat
3.2.6.Tempat/Lokasi dan Waktu Penelitian

DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 1 modul-modul di SIAK-NF
Gambar 2 ilustrasi MVC
Gambar 3 Hubungan fase dan Workflow
Gambar 4 Daur Hidup <i>Unified Process</i>
Gambar 5 <i>Timeline</i> Penelitian.....

DAFTAR TABEL

Table 1 Penelitian Terkait.....

BAB I

PENDAHULUAN

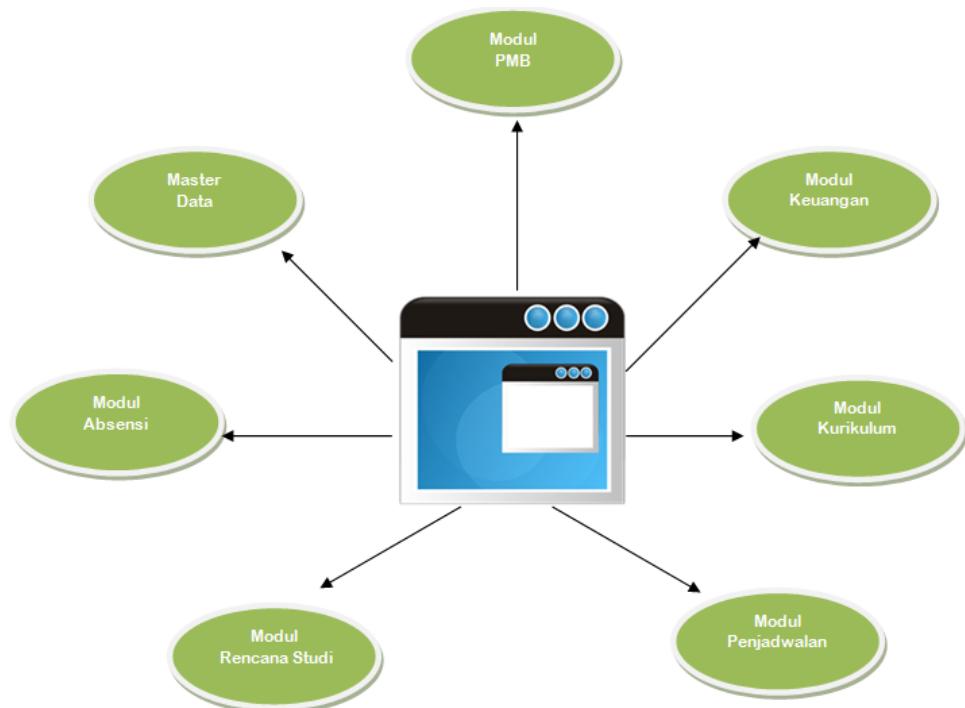
Pada bab ini berisi latar belakang dari penulisan proposal tugas akhir, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah, dan sistematika dari penulisan proposal tugas akhir ini

1.1. Latar Belakang Masalah

Sistem informasi akademik merupakan “tiang utama dalam mengatur segala hal yang berkaitan dengan penyelenggaraan perkuliahan maupun hal-hal yang lainnya” dan akademik yaitu “suatu rangkaian kegiatan yang disusun secara sistematis untuk mahasiswa sebagai input agar dapat menghasilkan lulusan yang berkualitas” [1]. Sistem Informasi Akademik sangat dibutuhkan oleh perguruan tinggi, karena memiliki banyak keuntungan yang dapat membantu pihak perguruan tinggi dalam menyelenggarakan proses pendidikan, beberapa keuntungan sistem informasi akademik yang disebutkan oleh Chairil Anwar di dalam jurnal penelitiannya antara lain : mahasiswa terhindarkan dari kemungkinan salah mengambil matakuliah yang akan diambil karena dari sistem aplikasi ini otomatis terhubung ke database dan mahasiswa hanya perlu melihat nama matakuliah yang mereka ambil adapun kode matakuliah sendiri ada di sampingnya, memudahkan staff universitas dalam melihat data-data mahasiswa yang telah melakukan pengisian KRS maupun data-data pembayaran mahasiswa, jika mahasiswa mengambil satu matakuliah dan jadwal matakuliah tersebut bentrok dengan jadwal matakuliah yang sudah dipilih sebelumnya maka sistem juga akan memberikan warning dan menolak untuk memasukkan matakuliah tersebut [2].

Sistem informasi yang dimiliki oleh perguruan tinggi terdiri dari komponen-komponen didalamnya, dan masing-masing komponen terkait membentuk suatu alur

sistem informasi lembaga pendidikan tertinggi [1], sama seperti aplikasi SIAK Nurul Fikri sendiri tersusun dari banyak modul-modul didalamnya, seperti yang digambarkan pada gambar 1. Pada gambar 1 digambarkan bahwa modul-modul penyusun aplikasi SIAK antara lain modul PMB (penerimaan mahasiswa baru), modul keuangan, modul kurikulum, modul penjadwalan, modul rencana studi, modul absensi, dan master data.



Gambar 1 Modul-modul di SIAK-NF

Modul-modul tersebut saling terkait satu sama lain sehingga membentuk suatu sistem kesatuan yang disebut sebagai Sistem Informasi Akademik yang digunakan oleh STT Terpadu Nurul Fikri saat ini. Sayangnya aplikasi SIAK Nurul Fikri saat ini masih memiliki banyak kekurangan seperti halnya dari segi keamanan, aplikasi SIAK ini masih tergolong rentan salah satu kerentanannya yakni *SQL Injection*. Berdasarkan pernyataan Moh Dahlan, ST, MT dan tim didalam penelitiannya, dikatakan bahwa serangan *SQL Injection* sangat berbahaya karena penyerang yang telah berhasil memasuki database sistem dapat melakukan manipulasi data yang ada pada database sistem. Proses manipulasi

data yang tidak semestinya oleh penyerang dapat menimbulkan kerugian bagi pemilik website yang terinjeksi, kebocoran data dan informasi merupakan hal yang fatal. Data-data tersebut dapat disalahgunakan oleh pihak yang tidak bertanggung jawab [3], dari pernyataan tersebut bisa disimpulkan bahwa aplikasi yang rentan terhadap SQL Injection sangat berbahaya, apalagi aplikasi SIAK-NF juga rentan terhadap serangan *SQL injection* hal ini bisa membahayakan data-data akademik STT-NF yang tersimpan di database.

Kekurangan selanjutnya yakni belum diterapkannya prinsip *responsive design*, sehingga ketika aplikasi SIAK tersebut diakses melalui perangkat mobile tampilannya belum optimal. Pada penelitian yang berjudul *Perancangan Website Menggunakan Responsive Web Design* dijelaskan dengan adanya konfigurasi *responsive design* maka membuat situs web dapat beradaptasi secara optimal dalam jenis perangkat apapun yang digunakan oleh pengguna karena sifatnya yang dapat merespon secara otomatis terhadap ukuran layar [4], dari penelitian tersebut dapat disimpulkan bahwa *responsive design* sangat diperlukan untuk mendapatkan informasi akademik yang optimal ketika aplikasi SIAK-NF ini diakses oleh civitas melalui perangkat yang berbeda.

Kekurangan lainnya yakni pengembangan aplikasi SIAK Nurul Fikri yang digunakan saat ini belum menggunakan *framework* menyebabkan kode program belum terorganisir dengan baik sehingga dalam melakukan proses *maintenance* kode program membutuhkan *effort* yang lebih. Berdasarkan penelitian yang dilakukan oleh Safarudin Sabti dijelaskan bahwa pada *framework* menganut konsep MVC sehingga membuat data dapat terorganisir dengan baik, karena MVC mampu memisahkan antara penggunaan database (*Models*), tampilan antar muka (*Views*), dan alur kerja (*Controller*), serta Extension sebagai pendukung didalam aplikasi web [5], dari penelitian tersebut dapat disimpulkan bahwa penggunaan *framework* dengan konsep MVC menjadi sangat penting dalam melakukan pengorganisasiran data-data akademik STT-NF apalagi ketika

mahasiswa sudah semakin banyak maka diperlukan pengelolaan data yang terstruktur serta mudah untuk dilakukan proses *maintenance*.

Modul-modul yang berada pada aplikasi SIAK Nurul Fikri juga belum sepenuhnya bisa memenuhi kebutuhan dari para civitas, salah satu contohnya adalah modul administrasi keuangan. Pada modul administrasi keuangan, laporan keuangannya masih direkap secara manual. Berdasarkan penuturan Samsudin selaku user yang menggunakan modul tersebut, untuk melakukan proses rekap laporan keuangan membutuhkan waktu kurang lebih satu minggu, fitur laporan rekapitulasi yang terdapat pada SIAK saat ini tidak digunakan karena formatnya tidak sesuai dengan kebutuhan bagian keuangan dalam melakukan rekapitulasi pembayaran mahasiswa. Selain itu STT Terpadu Nurul Fikri saat ini sudah menggunakan virtual account untuk melakukan transaksi pembayaran uang kuliah mahasiswa, aplikasi SIAK yang digunakan saat ini belum mendukung untuk fitur *upload file* dari bank terkait data mahasiswa yang melakukan transaksi pembayaran. Kwitansi atau bukti pembayaran mahasiswa masih dilakukan pengeditan secara manual karena format yang disediakan oleh aplikasi SIAK saat ini belum memenuhi kebutuhan dari bagian keuangan.

Berkaitan dengan banyaknya modul-modul pada aplikasi SIAK NF yang telah digambarkan serta paparan mengenai kekurangan dari aplikasi SIAK-NF maupun kekurangan dari modul keuangan yang sudah disebutkan sebelumnya, pada tugas akhir ini penulis membatasi hanya mengangkat modul keuangan dalam penelitian tugas akhir ini, penulis memberikan solusi berupa pembangunan kembali modul administrasi keuangan SIAK berbasis *web framework* dengan konsep MVC. Penulis memilih *web framework* dengan konsep MVC sebagai solusi dikarenakan pentingnya penggunaan MVC *framework* di dalam pengembangan aplikasi SIAK-NF saat ini, karena dengan menggunakan MVC *framework* dapat menyelesaikan permasalahan yang telah disebutkan sebelumnya terkait kekurangan aplikasi SIAK-NF khususnya modul keuangan.

Pada penelitian yang dilakukan oleh Ary Budi warsito dan tim berangkat dari permasalahan yang sama dengan penulis terkait responsif pada website perguruan tinggi, Ary Budi Warsito memberikan solusi berupa pembangunan kembali menggunakan *web framework* yang menggunakan konsep MVC yaitu dengan *Yii framework* versi 1.1.10 dan hasil yang didapatkan tampilan website menjadi lebih responsif, Ary Budi Warsito juga menyimpulkan dengan menggunakan *framework* akan menghemat waktu penggerjaan aplikasi [6], apalagi dalam mengerjakan modul keuangan ini penulis terbatas dengan waktu yang kurang dari 4 bulan maka penggunaan *framework* akan sangat membantu penulis dalam menyelesaikan kebutuhan dari bagian keuangan yang telah disebutkan sebelumnya.

Penelitian selanjutnya yang dilakukan oleh Safrudin juga menggunakan *web framework* dengan konsep MVC disitu safrudin menyimpulkan pengimplementasian *Yii framework* pada studi kasusnya membuat aplikasi forumnya memiliki tingkat keamanan yang cukup baik dari berbagai macam ancaman yang sering dialami aplikasi berbasis web, seperti *SQL Injection* dan *XSS attack*, selain data di dalam aplikasi akan lebih aman, data juga dapat terorganisir dengan baik karena di dalam framework terdapat MVC yang mampu memisahkan antara penggunaan database (Models), tampilan antarmuka (Views) dan Proses atau alur kerja (Controller), dan juga komponen-komponen serta Extension sebagai pendukung di dalam aplikasi berbasis web [5], dengan begitu ketika aplikasi SIAK-NF nanti sudah menggunakan MVC *framework* maka *effort* programmer dalam melakukan proses *maintenance* terhadap kode program tidak seberat ketika belum menggunakan *framework* serta tingkat kemanan aplikasi SIAK-NF menjadi lebih aman khususnya dari serangan *SQL injection*.

Atas dasar penelitian yang telah penulis paparkan sebelumnya terkait penggunaan MVC *framework*, akhirnya penulis memberi judul tugas akhir ini

dengan “**Analisa Dan Perancangan Sistem Informasi Akademik STT-NF Modul Administrasi Keuangan Berbasis Web Dengan Menggunakan MVC Framework**”.

1.2. Perumusan Masalah

Dari paparan latar belakang mengenai beberapa permasalahan yang telah bahas diatas, maka masalah yang dapat dirumuskan sebagai berikut :

“Apakah analisa dan perancangan aplikasi SIAK-NF khususnya modul keuangan berbasis web dengan menggunakan MVC framework sudah membuatnya lebih responsive dari sistem yang sebelumnya dan dapat memenuhi kebutuhan dari bagian keuangan ?”

1.3. Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini antara lain :

1. Untuk mengetahui bagaimana sistem akademik khususnya modul keuangan yang sedang berjalan saat ini dan dapat memberikan solusi yang lebih baik dari sistem sebelumnya.
2. Untuk menghasilkan sebuah produk aplikasi sistem akademik berbasis *web framework* yang aman dan responsif khususnya modul keuangan agar dapat dimplementasikan di STT NF
3. Untuk memberikan laporan mengenai informasi keuangan yang akurat dan cepat kepada pihak *top level management*, dan mahasiswa.

Manfaat yang diperoleh dari penelitian mengenai pengembangan aplikasi SIAK modul keuangan ini antara lain :

1. Proses pembuatan *report* data keuangan menjadi lebih efisien karena memakan waktu yang lebih singkat dari sistem sebelumnya.

2. Membantu pihak keuangan dalam melakukan pengelolaan data keuangan kampus secara berkala.
3. Memberikan tampilan informasi yang optimal bagi para mahasiswa maupun bagian keuangan ketika mengakses aplikasi SIAK khususnya modul keuangan diberbagai device karena sifatnya yang responsif.

1.4. Batasan Masalah

Batasan dari penelitian ini adalah :

1. Aplikasi SIAK yang akan dibangun hanya berfokus pada pengembangan modul keuangan karena waktu pengembangan yang sangat singkat yaitu 4 bulan
2. Modul keuangan yang akan dibangun menggunakan framework yii 2.0
3. Modul keuangan sangat bergantung dengan data yang berada pada modul PMB
4. Modul keuangan yang akan dibangun berfokus pada fitur : Pembayaran mahasiswa, Retur, Pelaporan
5. Mahasiswa hanya dapat melihat history pembayaran dari awal semester sampai akhir semester beserta tagihan yang harus dibayarkan

1.5. Sistematika Penulisan

Penyusunan tugas akhir ini mengikuti sistematika berikut ini :

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.

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

Pada 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.

2.1 Tinjauan Pustaka

2.1.1 Sistem Informasi Akademik

Sistem informasi akademik merupakan “tiang utama dalam mengatur segala hal yang berkaitan dengan penyelenggaraan perkuliahan maupun hal-hal yang lainnya” dan akademik yaitu “suatu rangkaian kegiatan yang disusun secara sistematis untuk mahasiswa sebagai input agar dapat menghasilkan lulusan yang berkualitas”. Sebuah perguruan tinggi tentunya memiliki sistem informasi pada instansi pendidikan, serta komponen-komponen yang terkandung didalamnya, kemudian kaitannya masing-masing komponen tersebut pada alur sistem informasi lembaga pendidikan tertinggi [1].

Dewasa ini kebutuhan akan sistem informasi akademik bagi perguruan tinggi menjadi hal yang penting apalagi jika jumlah mahasiswa suatu perguruan tinggi sangat banyak, dengan adanya sistem informasi akademik membuat pihak perguruan tinggi dapat menyediakan layanan informasi yang lebih baik dan efektif kepada komunitasnya serta dapat menghemat waktu, pikiran dan tenaga.

Sistem informasi ini akan mendukung kinerja perguruan tinggi yang bersangkutan baik dalam pelayanan terhadap mahasiswa sampai karyawan. Namun sesudah perkembangan teknologi informasi yang sedemikian pesatnya, perguruan tinggi harus setiap saat mengupdate sistem informasinya sehingga dalam kinerjanya akan menuju ke titik yang lebih baik [1].

Adapun keuntungan – keuntungan yang dapat diperoleh dari adanya sistem aplikasi akademik online ini antara lain [2] :

1. Ketika melakukan proses pengisian KRS sistem telah melakukan antisipasi seperti misalnya ketika mahasiswa akan mengambil satu matakuliah tetapi mahasiswa tersebut belum mengambil matakuliah prasyaratnya maka otomatis sistem akan menolak proses pengambilan mata kuliah tersebut.
2. Sistem akan memberikan peringatan dan menolak permintaan mahasiswa yang akan mengambil matakuliah yang bentrok dengan jadwal matakuliah yang sudah dipilih sebelumnya.
3. Adanya proses pembatasan banyaknya mahasiswa yang dapat mengambil suatu matakuliah, ketika suatu matakuliah sudah mencapai batas maksimum maka mahasiswa yang ingin mengambil matakuliah tersebut akan ditolak karena kapasitas kelas yang telah terpenuhi agar proses belajar-mengajar menjadi nyaman.
4. Aplikasi akademik memudahkan staff universitas dalam melihat data-data mahasiswa yang telah melakukan pengisian KRS maupun data-data pembayaran mahasiswa.

Banyaknya keuntungan yang telah disebutkan semakin mengutamakan betapa pentingnya penggunaan sistem informasi akademik didalam perguruan tinggi, terutama untuk bagian keuangan yang menjadi pusat transaksi dari setiap pembayaran kuliah yang dilakukan oleh mahasiswa, dengan adanya sistem informasi akademik membuat proses validasi status mahasiswa menjadi lebih cepat karena data-data akademik dari bagian lain terintegrasi didalam satu aplikasi sehingga mudah mengaksesnya.

2.1.2 UML (*Unified Modelling Language*)

UML merupakan bahasa pemodelan yang digunakan 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 [1].

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 [7] :

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 [1].

Jenis-jenis diagram dalam UML [1] :

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), specialed(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. Kegunaanya 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 menunjukkan 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*

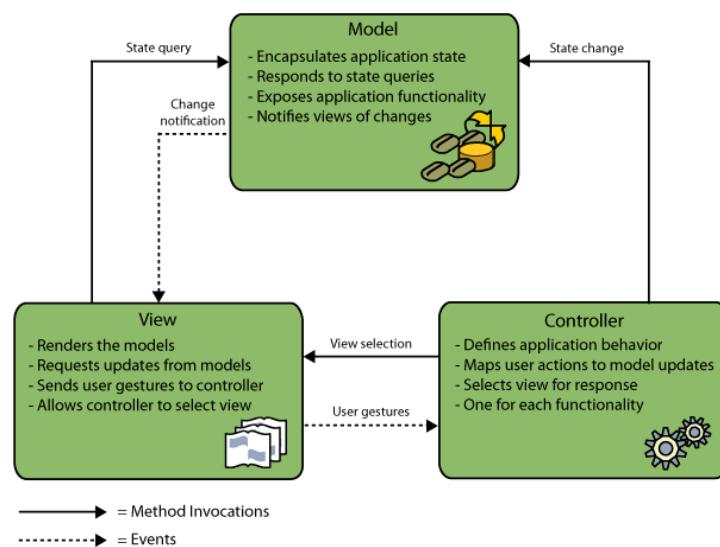
Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa source code, komponen biner atau class yang diimplementasikan sehingga membuat pemetaan dari logical view ke component view.

h. *Deployment Diagram*

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan 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.

2.1.3 MVC

Konsep MVC (Model-View-Controller) merupakan pola yang terbukti lebih efektif dalam melakukan pembangunan suatu proyek perangkat lunak, caranya dengan melakukan pemilihan komponen dalam proyek yang akan dijadikan model, view, dan controller. Arsitektur Web framework memiliki keterikatan yang kuat dengan konsep MVC, gambaranya terlihat seperti pada gambar 2 [5].



Gambar 2. Ilustrasi MVC

Pada MVC terdapat lapisan yang kita sebut dengan model, model digunakan untuk menyimpan data beserta aturan bisnisnya seperti validasi, relasi dan lain-lain. Model merupakan data objek yang tunggal, ini bisa menjadi sebuah baris dalam sebuah tabel database, atau membentuk form html dengan menginputkan field oleh user, setiap field data objek diwakili oleh atribut dari Model, atribut ini memiliki label dan dapat divalidasi terhadap aturan-aturan dalam Yii Framework [6].

Layer selanjutnya yaitu *view* pada layer ini mengandung keseluruhan detail dari implementasi user interface. Disini, komponen grafis menyediakan representasi

proses internal aplikasi dan menuntun alur interaksi user terhadap aplikasi. Tidak ada layer lain yang berinteraksi dengan user, hanya View. Penggunaan layer View memiliki beberapa kelebihan : Pertama, memudahkan pengabungan divisi desain dalam development team. Divisi desain dapat berkonsentrasi pada style, look & feel, dan sebagainya, dalam aplikasi tanpa harus memperhatikan lebih pada detail yang lain. Dan juga, memiliki layer View yang terpisah memungkinkan ketersediaan multiple interface dalam aplikasi. Jika inti dari aplikasi terletak pada bagian lain (dalam Model), multiple interfaces dapat dibuat (Swing, Web, Console), secara keseluruhan memiliki tampilan yang berbeda namun mengeksekusi komponen Model sesuai fungsionalitas yang diharapkan [8].

Terakhir, arsitektur MVC memiliki layer Controller. Layer ini menyediakan detail alur program dan transisi layer, dan juga bertanggung jawab atas penampungan event yang dibuat oleh user dari View dan melakukan update terhadap komponen Model menggunakan data yang dimasukkan oleh user. Kelebihan penggunaan layer Controller secara terpisah : Pertama, dengan menggunakan komponen terpisah untuk menampung detail dari transisi layer, komponen View dapat didesain tanpa harus memperhatian bagian lain secara berlebih. Hal ini memudahkan team pengembang multiple interface bekerja secara terpisah dari yang lain secara simultan. Interaksi antar komponen View terabstraksi dalam Controller. Kedua, dengan menggunakan layer terpisah yang melakukan update terhadap komponen Model, detail tersebut dihapus dari layer presentasi. Layer presentasi kembali pada fungsi utamanya untuk menampilkan data kepada user. Detail tentang bagaimana data dari user mengubah ketetapan aplikasi disembunyikan oleh Controller. Hal ini memisahkan dengan jelas antara presentation logic dengan business logic [8].

2.1.4 Framework

Berdasarkan kamus Inggris – Indonesia yang disusun oleh Jhon M. Echols dan Hassan Sadily *framework* berarti kerangka kerja, disebut kerangka

kerja sebab pekerjaan yang dilakukan harus sesuai dengan kerangka tersebut dan tidak boleh lari dari jalur atau kerangka yang telah ditetapkan [1].

Framework merupakan sekumpulan pustaka atau *library* yang diorganisasikan pada sebuah rancangan arsitektur guna memberikan kecepatan, ketepatan, kemudahan dan konsistensi di dalam pengembangan proyek perangkat lunak tersebut [1].

Manfaat dari framework antara lain [9] :

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola standar yang ada).
3. Umumnya framework telah menyediakan fasilitas-fasilitas penunjang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple database, scaffolding, pengaturan session, error handling, dan lain – lain).
4. Jika dikerjakan secara team work, maka akan menjadi lebih terarah karena system framework mengharuskan adanya keteraturan peletakan kode.

2.1.5 **Yii**

Yii Framework dimulai oleh Qiang Xue pada tanggal 1 Januari 2008. Yii merupakan akronim dari “Yes, it is” selalu menekankan nilai easy, efficient dan extensible. Belakangan mereka mengganti menjadi fast, secure, professional. Namun nilai-nilai lamanya tidaklah hilang. Yii selalu disebut sebagai framework PHP 5 yang high-performance dan component-based. Yii dirancang dengan memperhatikan sisi performa dan memanfaatkan konsep component-based yang dimiliki Prado. Selain itu Yii juga memanfaatkan even tetapi tidak tergantung sepenuhnya [5].

Selain mengambil konsep dari Prado, Yii juga memanfaatkan konsep Ruby on Rails yang mengutamakan konvensi daripada konfigurasi serta implementasi

desain Active Record (AR). Yii juga memasukkan konsep Joomla dalam desain modular dan skema penerjemahan pesan serta sistem Symfony pada desain filter dan arsitektur plug-in. Yii Framework dibandingkan yang lainnya mungkin tergolong cukup muda karena dimulai pada tahun 2008. Tetapi walaupun mungkin masih muda, Yii Framework sendiri sudah memiliki segudang fitur yang bisa mempermudah pekerjaan kita [5].

Menurut pengembang dan komunitas Yii Framework dari situs web resminya, berikut ini adalah fitur yang terdapat pada Yii [10] :

1. Menggunakan pola MVC. Yii mengadopsi pola standar pengembangan aplikasi yang memisahkan tampilan (view). Logika program (controller), dan modelnya (model).
2. Data Access Objects (DAO), Query Builder, Active Record, dan DB Migration. Yii memungkinkan pengembang menggunakan berbagai model database dengan objek yang dapat mengurangi kompleksitas menulis pernyataan SQL berulang serta memudahkan interaksi antardatabase.
3. Form input dan validasi. Yii menyediakan form input dan validasi yang memudahkan pengembang untuk bekerja dengan form dan melakukan validasi input dari form.
4. Authentication dan authorization. Dukungan autentikasi dan autoritas internal sehingga memudahkan pengembangan aplikasi dengan fitur autentikasi dan autorisasi.
5. Keamanan. Yii dilengkapi dengan langkah-langkah keamanan untuk membantu mencegah aplikasi Web dari serangan seperti SQL injection, cross-site scripting (XSS), Cross-site forgery (CSRF), dan cookie tampering.
6. Penghasil kode otomatis. Yii menyediakan penggunaan beragam perintah otomatis seperti meng-generate struktur dasar aplikasi, model dan CRUD.

7. Librari ekstension. Yii menyediakan dukungan ekstensi dan komponen tambahan sehingga beragam fitur tambahan dapat ditambahkan.

Saat ini yii telah mencapai versi 2.0, namun meskipun demikian, hingga tahun 2015 ini, Yii masih akan tetap mendukung varian versi 1.1 untuk memberikan kesempatan bagi para developer sebelum mereka migrasi ke versi 2.0. Hal ini dikarenakan versi 2.0 memiliki perbedaan yang sangat signifikan dibanding dengan versi 1.1, baik dari sisi kode, sebagian fitur, sebagian konsep, dan teknologi yang diadopsi. Dari sisi kode, Yii 2.0 berbeda jauh dengan Yii 1.0, karena memang bukan pengembangan melainkan hasil tulis ulang kode dengan konsep baru, serta mengadopsi teknologi-teknologi terbaru yang telah didukung oleh PHP, seperti composer, PSR, namespace, dsb [11].

Hafidz yang merupakan seorang developer menjabarkan beberapa kelebihan dari Yii 2.0 berdasarkan pengalamannya [11]:

a. Mengadopsi Standard dan Teknologi Terbaru

Setelah versi 2.0, Yii mengadopsi standard dan teknologi terbaru, sebut saja: PHP 5.4 keatas, namespace, composer, PSR, dan bower. Banyak best practice pada framework lain, cms, atau teknologi web lain diadopsi oleh Yii seperti Prado, Sumfony, Ruby dan Rails, joomla, dsb. Developer Yii akan terus menyesuaikan dengan perkembangan teknologi web, dan membuka masukan dari berbagai pihak. Silahkan baca issue di github yii2.

b. Peningkatan keamanan

Keamanan menjadi faktor penting dalam dunia web programming, dan developer yii menyadari akan hal itu. Yii memiliki beberapa fungsi keamanan built-in yang mudah digunakan, antara lain untuk pencegahan SQL injection, XSS attack, CSRF attacks, cookie tampering, dsb. Pada versi 2.0 ini, Yii juga telah direview oleh 2 pakar keamanan web yaitu Tom Worster and Anthony.

c. URL SEO Friendly

Hanya dengan sedikit konfigurasi, kita bisa membuat aplikasi dengan URL yang SEO Friendly. Fitur ini bernama pretty url, yang pada versi 2.0 ini telah disempurnakan.

d. Mudah integrasi dengan Library Lain

Sejak versi 1, Yii terkenal mudah berintegrasi dengan library lain, apalagi setelah mendukung composer. Jadi jangan khawatir, karena library PHP apapun bisa digunakan di Yii 2.0

e. Template Aplikasi

Yii juga menyediakan template aplikasi (basic dan advance) yang akan memudahkan kita dalam memulai mengembangkan aplikasi. Dari sini akan memudahkan kita memahami best practice dari Yii.

2.1.6 *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 [12].

2.1.7 *Black-box Testing*

Black-box testing, bisa juga disebut behavioral testing, berfokus pada *functional requirement* sebuah perangkat lunak [13]

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) [14].

2.2 Penelitian Terkait

Pada tabel 1 penulis mencoba jabarkan penelitian yang terkait dengan penelitian yang sedang penulis kerjakan. Pada tabel dengan *field* yang diberi blok biru atau penelitian dengan nomor (1,2,4,6) merupakan penelitian terkait konsep MVC *framework*, rata-rata semua penelitian tersebut menyimpulkan dengan menggunakan MVC *framework* membuat aplikasi mudah untuk dilakukan perubahan-perubahan, dapat mengurangi tingkat kerumitan apalagi dalam *scope* besar, mudah dimaintain dan diorganisir. Pada *field* yang diberi blok jingga atau penelitian dengan nomor (3 dan 5) merupakan penelitian dengan menggunakan pendekatan *unified process*, pada penelitian tersebut menyimpulkan *unified process* cocok untuk pengembangan aplikasi berbasis objek, dapat diintegrasikan dengan sistem lainnya.

Berdasarkan semua penelitian tersebut semakin menguatkan bahwa aplikasi SIAK-NF sangat perlu untuk menggunakan MVC *framework* karena banyak manfaat yang telah didapatkan oleh peneliti-peneliti sebelumnya, selain itu pendekatan dengan menggunakan *unified process* sangat cocok diterapkan dalam pembangunan aplikasi yang bersifat modular dan dapat diintegrasikan sama seperti proyek yang dilakukan oleh penulis yaitu mengerjakan modul keuangan SIAK STT-NF yang nantinya akan diintegrasikan kedalam sistem SIAK yang lebih besar.

No	Judul Penelitian	Tahun	Kesimpulan
1	Pengamanan Database yang di Implementasikan ke dalam Arsitektur MVC menggunakan Hibernate Framework	2009	Memudahkan perubahan-perubahan pada aplikasi, ini dikarenakan aplikasi ini menggunakan arsitektur MVC yang dimana dalam penggerjaan aplikasinya dibagi menjadi 3 bagian yaitu bagian Model, View dan juga Controller. [15]
2.	Implementasi Framework Spring MVC Untuk Pembuatan Sistem Informasi Manajemen E Commerce	2010	Arsitektur MVC memisahkan antara user interface, model dalam database, dan business logic, sehingga mengurangi tingkat kerumitan suatu aplikasi. Hal ini akan sangat bermanfaat ketika scope aplikasi yang dibuat sangatlah luas, sehingga aplikasi yang dibuat dengan arsitektur MVC akan mudah untuk di-maintain [16].
3	Pengembangan Sistem Informasi Administrasi Pemeriksaan Pasien Di Instalasi Radiologi Rsud Kajen Dengan Unified Process	2011	<ul style="list-style-type: none"> Dengan Unified Process, sistem ini dapat dikembangkan lebih lanjut untuk diintegrasikan dengan system lain seperti sistem administrasi rumah sakit, sistem kasir rumah sakit, dan sistem-sistem lain yang berada di instalasi lain di RSUD Kajen Kabupaten Pekalongan [12].
4.	Eksplorasi Yii Framework Sebagai Pendukung Pembuatan Software Berbasis Web (Studi Kasus Aplikasi Forum)	2012	<ul style="list-style-type: none"> Setelah mengimplementasikan Yii framework membuat tingkat keamanan cukup baik dari berbagai macam ancaman yang sering dialami aplikasi berbasis web, seperti SQL Injection dan XSS attack, selain data di dalam aplikasi akan lebih aman, data juga dapat terorganisir dengan baik karena di dalam framework terdapat MVC yang mampu memisahkan antara penggunaan database (Models), tampilan antarmuka (Views)

			<p>dan Proses atau alur kerja (Controller), dan juga komponen-komponen serta Extension sebagai pendukung di dalam aplikasi berbasis web</p> <ul style="list-style-type: none"> • Konsep CRUD dan MVC, yang terdapat dalam sebuah framework merupakan solusi untuk membantu para developer/programmer website untuk membuat fungsi Create, Read, Update dan Delete. Dan juga untuk mengorganisir dan mengelompokkan bagian-bagiannya sehingga mudah dalam pengelolaannya [5]
5	Analisa Dan Perancangan Aplikasi Dormitory Management Menggunakan Unified process	2013	<ul style="list-style-type: none"> • 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 usecase driven, incremental dan iterative [17]
6.	Kajian Yii Framework Dalam Pengembangan Website Perguruan Tinggi	2014	<ul style="list-style-type: none"> • Dengan adanya framework akan lebih mempermudah memahami mekanisme kerja dari sebuah aplikasi. Ini tentunya akan sangat membantu proses pengembangan sistem yang dilakukan secara tim. • Kedua dengan memakai framework akan menghemat waktu penggerjaan suatu aplikasi, karena setiap anggota sudah memiliki sebuah acuan dalam menyelesaikan modul [6]

Table 1. Penelitian terkait

BAB III

Metodologi Penelitian

Pada 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.

3.1. Tahapan Penelitian

Pada metode *unified process* terdapat 4 tahapan (fase) di dalam pengembangannya, antara lain [18] :

a. *Inception*

Pada tahapan ini kegiatan yang dilakukan adalah proses pendefinisian *scope* dari proyek yang akan dikerjakan dan melakukan pengembangan dari *buisness case* untuk sistem. Pada fase ini *workflow* yang menjadi fokus adalah *requirement*, serta ada sedikit *workflow* analisis.

b. *Elaboration*

Pada tahapan ini kegiatan yang dilakukan adalah melengkapi *functional requirement* maupun *non-functional* dari sistem. Kegiatan utama pada fase ini adalah menciptakan sebuah arsitektur yang nantinya akan digunakan pada seluruh sisa fase dari *unified process*. Dalam proses menciptakan arsitektur, *workflow* yang menjadi fokus adalah proses analisis dan desain.

c. *Construction*

Pada fase ini pembangunan sistem mulai dilakukan, konsentrasi pada fase ini adalah melengkapi analisis dari sistem yang telah dilakukan pada fase sebelumnya dan melakukan pengimplementasian sebagian besar desain yang telah dibuat ke dalam sistem. *Workflow* yang menjadi fokus adalah proses desain, implentasi, dan pengujian

d. *Transition*

Pada fase ini dilakukan proses peralihan sistem ke dalam *user's environment*. Kegiatan utama dalam fase ini adalah proses *deploying* dan *maintenance*. Pada fase ini *workflow* yang menjadi fokus adalah *deploying* dan *testing*. Major milestone pada fase ini adalah produk akhir yang telah lulus uji dan siap rilis.

Pada gambar 3 menunjukkan hubungan antara fase dan workflow pada metode *unified process*

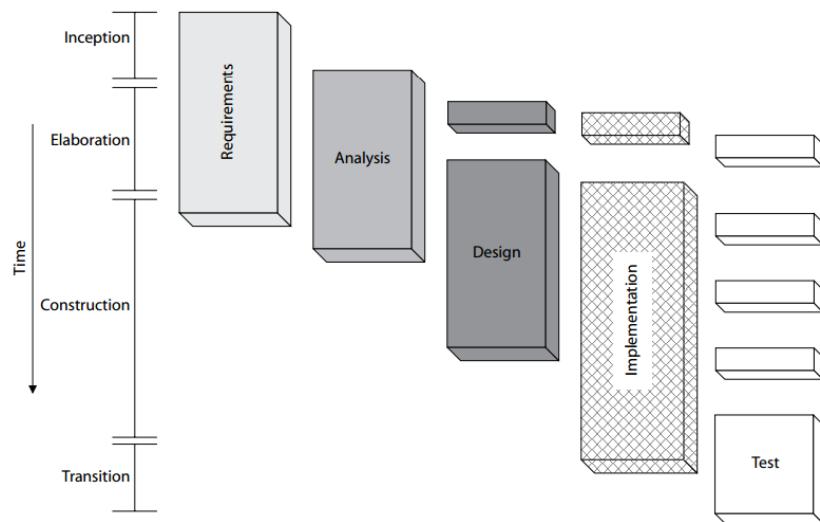


Figure 3.10 Disciplines versus phases.

Gambar 3. Hubungan fase dan workflow

3.2 Rancangan Penelitian

3.2.1 Metode Pengembangan

Pada tahapan penelitian ini penulis menggunakan metode pengembangan perangkat lunak *unified process*. Metode ini dipilih karena sifatnya yang iterative dan bertahap, jadi dapat dilakukan perbaikan berulang-ulang hingga sesuai dengan kebutuhan user. Metode *unified process* ini memiliki tahapan-tahapan utama di setiap fasenya (*core workflow*) antara lain :

1. Requirement

Setelah dilakukan proses *buisness modelling* untuk mengumpulkan data awal, maka dilakukan proses pengumpulan kebutuhan dari *user*, pada tahap ini jika terdapat requirement tambahan akan langsung masuk ke dalam tahap ini. Setelah itu akan masuk ke tahapan selanjutnya yaitu analisis output dari tahap ini berupa *use case model*.

2. Analisis

Setelah proses pengumpulan kebutuhan dari *user* telah selesai, maka dilakukan proses analisa terhadap kebutuhan *user* yang telah dikumpulkan sebelumnya, tujuan dari tahap ini adalah memastikan bahwa tidak terdapat *miss requirement* dari user.

3. Desain

Setelah dilakukan analisa maka selanjutnya dilakukan pendefinisian kebutuhan *user* tersebut ke dalam bentuk desain dengan menggunakan UML antara lain *class diagram*, *sequence diagram*, *activity diagram*, *component diagram*

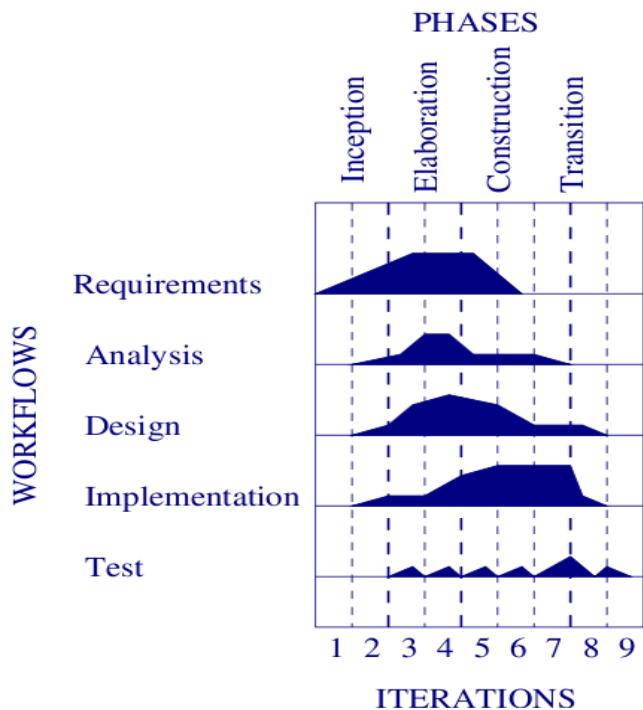
4. Implementation

Setelah dilakukan analisis kebutuhan dan desain maka masuk ke fase implementasi dimana pada fase ini dilakukan proses implementasi dari desain yang telah dibuat ke dalam kode program hingga siap untuk dilakukan testing.

5. Testing

Setelah pembuatan kode program selesai maka dilakukan proses testing. Testing yang dilakukan menggunakan metode *black-box* untuk menguji tiap fungsionalitas menu. Jika modul yang diuji sudah melalui proses *acceptance* maka modul sudah siap untuk di *deploy* dan diintegrasikan.

Digambarkan pada gambar 4 daur hidup dari *unified process* dan hubungannya antara fase dan *workflow*.



Gambar 4. Daur hidup *unified process*

3.2.2 Metode Penelitian

Penelitian ini menggunakan metode deskriptif-kualitatif. Metode deskriptif-kualitatif dipilih karena untuk kasus analisa kebutuhan pengembangan perangkat lunak metode ini lebih relevan. Relevan dalam artian untuk proses analisa kebutuhan penelitian ini dilakukan proses wawancara dan hasil dari wawancara ini berupa data bersifat deskriptif sama seperti pendapat Bogdan dan Taylor yang dikutip Moleong [19] “metodologi kualitatif” sebagai prosedur penelitian yang menghasilkan data deskriptif berupa kata-kata tertulis atau lisan dari orang-orang dan perilaku yang dapat diamati. Metode pengembangan perangkat lunak sendiri menggunakan pendekatan *unified process*. *Unified process* dalam rekayasa perangkat lunak merupakan suatu framework yang menekankan pada pengulangan (iterations) disetiap fasenya.

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

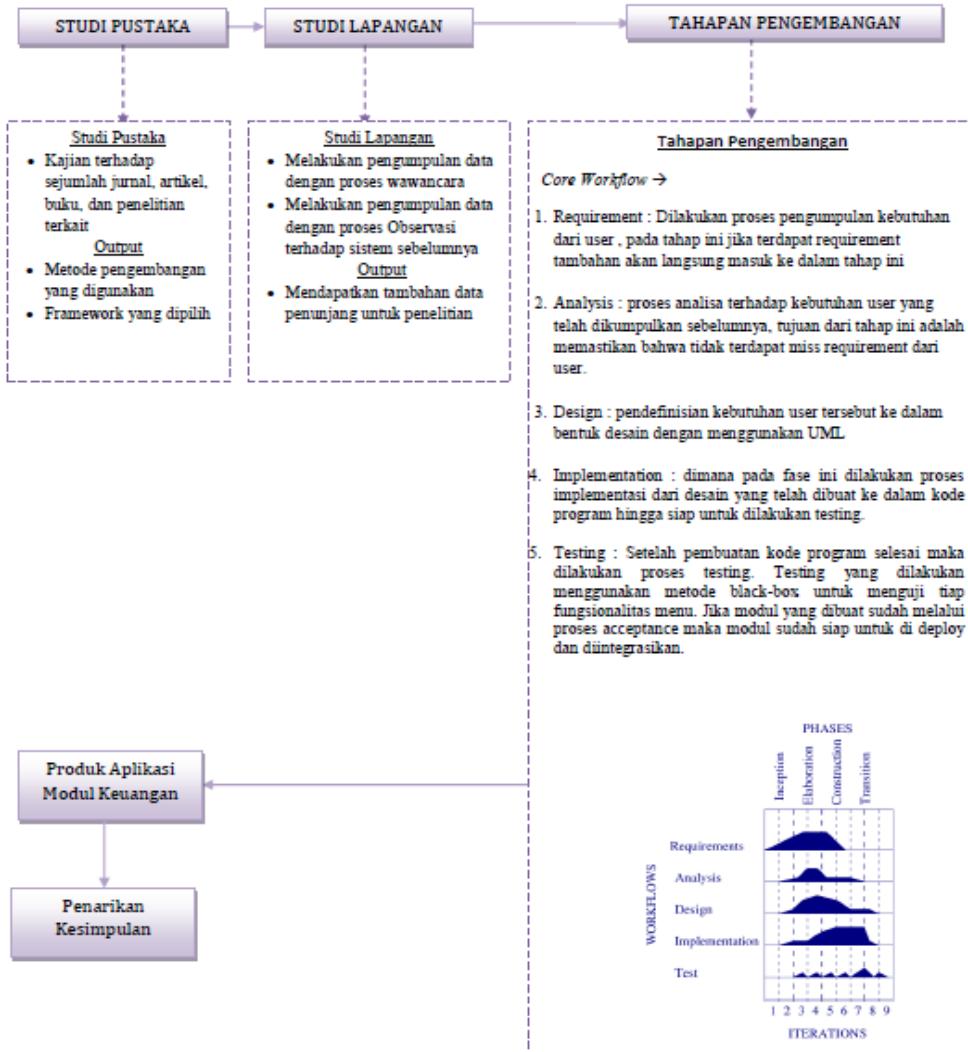
Pada tahapan ini dilakukan proses pengamatan terhadap cara kerja sistem yang sedang berjalan di STT-NF saat ini dan mengamati bagaimana user memanfaatkan aplikasi tersebut.

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 5 digambarkan tahapan penelitian yang dilakukan oleh penulis selama melakukan penelitian, rancangan analisis ini berguna bagi penulis untuk mencapai tujuan yang telah direncanakan sebelumnya.



Gambar 5 . Rancangan Analisis 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

Alat pengembangan 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*

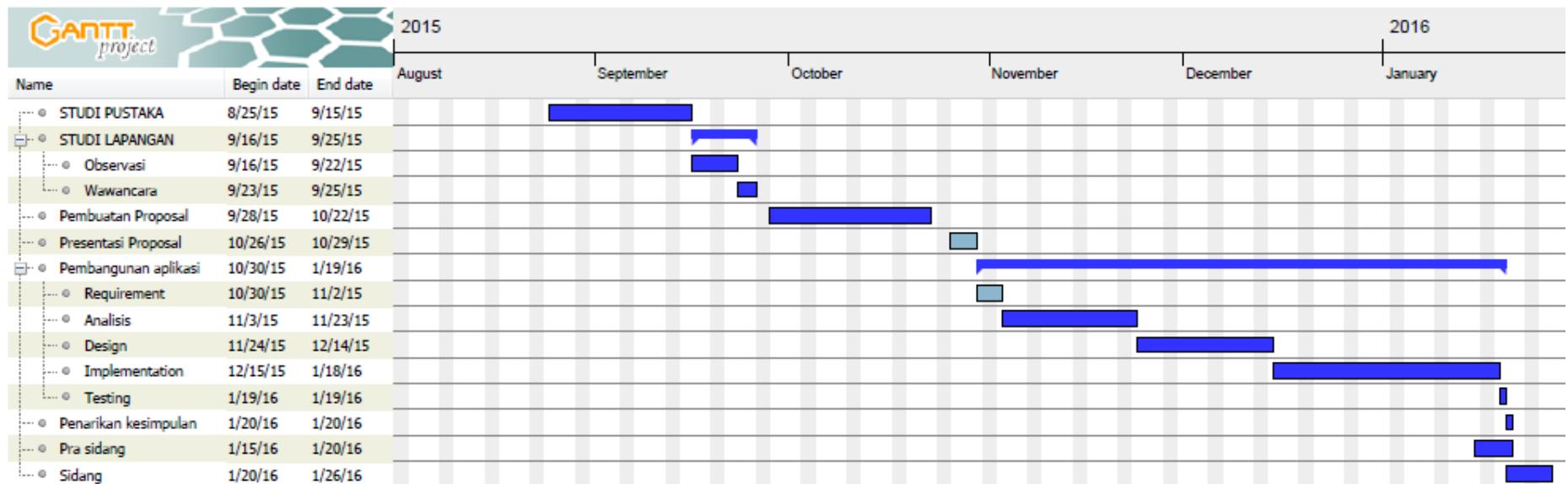
Alat desain yang digunakan :

- a. *Enterprise Architect (UML 2.0)*
- b. *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 administrasi keuangan khususnya *fitur* pembayaran mahasiswa, pelaporan, dan retur. Waktu penelitian ini berlangsung sejak 25 September 2015. Pada gambar 5 merupakan *timeline* dari penelitian yang dilakukan oleh penulis.

Gantt Chart



Gambar 5. Timeline Penelitian

DAFTAR PUSTAKA

- [1] D. Rustandi, "Perancangan Sistem Informasi Akademik Berbasis Web di Akademik Keperawatan Rumah Sakit Hasan Sadikin Bandung," *JBPTUNIKOMPP*, vol. I, no. 1, p. 8, 2009.
- [2] C. Anwar, "SISTEM INFORMASI AKADEMIK ONLINE SEBAGAI PENUNJANG," *Fakultas Teknologi Informasi, Universitas Budi Luhur*, vol. V, no. 1, p. 5, 2010.
- [3] S. M. M. Dahlan, S. A. Latubessy, S. M. L. H. Anggraeni and S. M. Nurkamid, "PENGUJIAN DAN ANALISA KEAMANAN WEBSITE TERHADAP SERANGAN SQL INJECTION (Studi Kasus : Website UMK)," *UNIVERSITAS MURIA KUDUS*, vol. I, no. 1, p. 2, 2014.
- [4] S. and Z. Akib, "PERANCANGAN WEBSITE MENGGUNAKAN RESPONSIVE WEB DESIGN," *SIGMATA*, vol. II, no. 1, p. 1, 2013.
- [5] S. Sabti, "EKSPLORASI YII FRAMEWORK SEBAGAI PENDUKUNG PEMBUATAN SOFTWARE BERBASIS WEB (STUDI KASUS APLIKASI FORUM)," *JBPTUNPASPP*, vol. I, no. 1, pp. 5-1, 2012.
- [6] A. B. Warsito, Y. Muhamad and Y., "KAJIAN YII FRAMEWORK DALAM PENGEMBANGAN WEBSITE PERGURUAN TINGGI," *Jurnal CCIT*, vol. VII, no. 1, p. 3, 2014.
- [7] 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.
- [8] J.E.N.I, "Ekoz06 @ Alumni.IPB.ac.id," 20 may 2008. [Online]. Available: <http://ekoz06.alumni.ipb.ac.id/download/download-materi-java-programming-dari-jeni>. [Accessed 21 octtober 2015].
- [9] A. N. Ismayadi, "PERANCANGAN SISTEM INFORMASI PENDAFTARAN BEASISWA BAGI MAHASISWA UNIVERSITAS RESPATI YOGYAKARTA SECARA ONLINE BERBASIS WEB DENGAN FRAMEWORK CODEIGNITER," Amikom, Yogyakarta, 2014.
- [10] A. R. Rahayu, "PENGEMBANGAN SISTEM INFORMASI PENGARSIPAN PADA WEBSITE RAHARJA.AC.ID MENGGUNAKAN YII FRAMEWORK," AMIK RAHARJA, Tangerang, 2014.

- [11] H. Mukhlasin, "Buku Kupas Tuntas Yii Framework 2.0," 18 january 2015. [Online]. Available: <http://www.hafidmukhlasin.com/kupas-tuntas-yii2/>. [Accessed 19 october 2015].
- [12] 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.
- [13] 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.
- [14] R. M. R. Attamimi, "PEMBANGUNAN SISTEM INFORMASI MANAJEMEN INVENTARISASI BARANG MILIK NEGARA DENGAN METODE AGILE UNIFIED PROCESS (AUP)," IPB, Bogor, 2013.
- [15] T. Wardhana, "Pengamanan Database yang di Implementasikan ke dalam Arsitektur MVC menggunakan Hibernate Framework," Gunadarma, Jakarta, 2009.
- [16] M. Toha, "IMPLEMENTASI FRAMEWORK SPRING MVC UNTUK PEMBUATAN SISTEM INFORMASI MANAJEMEN E COMMERCE," Universitas Sebelas Maret, Surakarta, 2010.
- [17] S. Karouw, "ANALISA DAN PERANCANGAN APLIKASI DORMITORY MANAGEMENT MENGGUNAKAN UNIFIED SOFTWARE DEVELOPMENT PROCESS," *Jurnal Teknik Informatika*, vol. II, no. 1, p. 1, 2013.
- [18] J. Hunt, Guide to the Unified Process featuring UML, Java and Design Patterns, United States of America: Springer, 2003.
- [19] L. J. Moleng, Metode Penelitian kualitatif, Bandung: Remaja Rosdakarya, 2002.