BAB I

PENDAHULUAN

1.1. Latar Belakang Penelitian

Teknologi informasi saat ini terus berkembang dari hari ke hari baik itu informasi dari media cetak maupun elektronik yang menyajikan informasi bentuk tulisan maupun gambar. Perusahaan penyedia jasa informasi berlomba lomba untuk menyajikan informasi terkini yang di butuhkan oleh masyarakat. Sedangkan bagi masyarakat mereka harus pandai -pandai dalam memilih sumber informasi yang dapat dipercaya dengan berita jelas dan dapat di pertanggung jawabkan.

Kemajuan di bidang teknologi informasi yang sangat pesat telah memberikan berbagai kemudahan bagi kehidupan umat manusia, salah satunya adalah kemudahan untuk mendapatkan informasi. Jarak dan waktu tidak lagi menjadi penghalang bagi seseorang / instansi untuk mencari dan mendapatkan informasi yang di butuhkan, dan oleh karena itu maka sebagian orang selalu berusaha untuk mendapatkan teknologi informasi terbaru demi mendapatkan tujuan yang mereka inginkan. Dengan teknologi informasi yang memadai, informasi yang dibutuhkan akan lebih cepat untuk didapatkan dan nilainya pun akan lebih akurat jika dibandingkan dengan informasi yang didapatkan dengan cara manual.

Di jurusan manajemen informatika pada universitas komputer indonesia proses pengolahan data tugas akhir mahasiswa masih dilakukan dengan cara sederhana, di sini mahasiswa harus ke kampus untuk melihat informasi skripsi dan harus mengantri pada saat proses pendaftaran prosal penelitian serta pendaftaran uji program, seminar dan sidang.

Dengan adanya perkembangan teknologi yang semakin pesat pada saat ini, khususnya pada sistem tugas akhir mahasiswa, hal ini menuntut agar lebih bisa menghasilkan informasi yang memenuhi spesifikasi kebutuhan pemakaian informasi. Maka, untuk menghasilkan sebuah informasi diperlukan suatu sistem infomasi yang sesuai dengan bidang serta mampu mengatasi masalah-masalah yang timbul.

Berdasarkan hasil pangamatan dan penelitian dari sistem tugas akhir mahasiswa pada jurusan manajemen infomatika unikom, sistem yang ada belum terstruktur masih menggunakan aplikasi yang masih sederhana sehingga menyebabkan pengolahan datanya kurang akurat.

Oleh karena itu penulis tertarik untuk melakukan penelitian tentang sistem informasi akademik di Jurusan Manajemen Informatika UNIKOM, dan mengambil topik tersebut dalam penyusunan skripsi yang penulis beri judul **"SISTEM INFORMASI TUGAS AKHIR MAHASISWA PADA JURUSAN MANAJEMEN INFORMATIKA UNIVERSITAS KOMPUTER INDONESIA (UNIKOM) BANDUNG"**

Diharapkan dengan adanya sistem informasi ini dapat membantu mengurangi kesalahan yang terjadi dalam proses tugas akhir mahasiswa serta kesalahan dalam penyajian informasi.

1.2. Identifikasi dan Rumusan Masalah

Berdasarkan uraian latar belakang diatas, penulis mengidentifikasikan masalah diantaranya :

1. Pada saat pendaftaran tugas akhir dan pengajuan proposal penelitian, proses pencatatan dan pendataannya masih kurang baik.
2. Mahasiswa harus sering ke kampus untuk melihat informasi skripsi / tugas akhir
3. Pengolahan data mahasiswa tugas akhir pada jurusan manajemen infomatika unikom, sistem yang ada belum terstruktur masih menggunakan aplikasi yang menyebabkan pengolahan datanya kurang akurat.

   Dari identitas masalah diatas, maka bisa didapatkan rumusan masalah sebagai berikut :

   1. Bagaimana sistem informasi tugas akhir mahasiswa yang sedang berjalan pada jurusan manajemen informatika UNIKOM.
   2. Bagaimana perancangan sistem informasi tugas akhir mahasiswa yang akan di buat pada jurusan Manajemen Informatika UNIKOM.

1.3. Maksud dan Tujuan Penelitian

Maksud dari penelitian ini adalah untuk merancang sistem informasi tugas akhir mahasiswa pada jurusan Manajemen Informatika UNIKOM.

Selain maksud diatas, penelitian ini juga mempunyai beberapa tujuan, diantaranya adalah :

1. Untuk memberi kemudahan pada saat proses pendaftaran dan pengajuan proposal penelitian.
2. Membantu proses kerja karyawan sekjur.
3. Membantu mahasiswa dalam pemberian informasi baik informasi skripsi / tugas akhir.

1.4. Kegunaan Penelitian

Dalam penelitian ini terdapat dua kegunaan penelitian, yaitu kegunaan praktis dan kegunaan akademis.

1. Kegunaan Praktis
    a) Memberikan masukan pada jurusan Manajemen Informatika mengenai sistem informasi tugas akhir mahasiswa yang dirancang tersebut nantinya diharapkan akan bisa digunakan oleh pihak jurusan.
    b) Memberikan kemudahan bagi para staf kesekretariatan jurusan yang bertugas dalam proses pendaftaran TA / Skripsi.
    c) Keamanan data mahasiswa lebih terjamin

2. Kegunaan Akademis

   Kegunaan akademis yang dapat diperoleh dari penelitian ini adalah sebagai berikut :

   a) Bagi ilmu pengetahuan : Untuk mengembangkan ilmu di bidang informatika
   b) Bagi peneliti lain : Bisa dijadikan acuan untuk peneliti lain yang mempunyai minat yang sama.
   c) Bagi peneliti : Peneliti bisa mengimplementasikan pengetahuan yang didapat di perkuliahan dengan kenyataan yang sesungguhnya dilapangan.

1.5. Batasan Masalah

Ada beberapa batasan masalah yang dibuat dalam penelitian ini, diantaranya adalah :

1. Sistem informasi yang dibangun hanya meliputi : sistem informasi tugas akhir mahasiswa mulai dari pengajuan proposal penelitian, proses pendaftaran, pembagian jadwal bimbingan seminar dan sidang sampai keluarnya nilai akhir layak tidaknya mahasiswa tersebut lulus atau tidak.
2. Program yang dibangun meliputi pembagian ruangan pada saat proses bimbingan, seminar dan sidang berlangsung.
3. Pada aplikasi yang di buat pemilihan dosen pengguji di random oleh sistem tetapi dapat dipilih kembali secara manual oleh user.

4. Pada aplikasi program pengajuan proposal dan skripsi online yang bertindak sebagai user adalah mahasiswa dan yang bertindak sebagai admin adalah sekretariat jurusan – Manajemen Informatika (Sekjur-MI).

1.6. Lokasi dan Waktu Penelitian

Penelitian akan dilaksanakan di jurusan Manajemen Informatika UNIKOM yang beralamat di Kampus V : Jl. Dipati Ukur 114 Telp. (022) 2506553, 2508412 Bandung 40132.

Sedangkan waktu penelitian dilaksanakan kurang lebih selama lima bulan, untuk lebih jelasnya bisa dilihat pada table dibawah ini :

Tabel 1.1 Estimasi Jadwal Penelitiaan

No	Aktifitas	Tahun 2009 / 2010																			
		Oktober				November				Desember				Januari				Februari			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Analisis Sistem																				
2	Perancangan																				
3	Pengujian Sistem																				
4	Implementasi																				

BAB II

LANDASAN TEORI

Pada bab ini penulis akan menjelaskan teori–teori yang dipakai dalam menyusun laporan skripsi ini.

2.1 Konsep Dasar Sistem

Menurut Jogiyanto (2005:1), suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

Ada beberapa elemen yang membentuk sebuah sistem, diantaranya adalah :

1. Tujuan

   Setiap sistem memiliki tujuan (*Goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi motivasi yang mengarahkan sistem. Tanpa tujuan,

sistem menjadi tidak terarah dan tidak terkendali. Tentu saja, tujuan antara sistem satu dengan sistem yang lain berbeda.

2. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang di proses. Masukan dapat berupa hal-hal yang berwujud (tampak secara fisik) maupun yang tidak tampak.

3. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi keluaran bisa berupa suatu informasi, sasaran, cetakan laporan, dan sebagainya.

4. Batas

Batas (*boundary*) sistem adalah pemisah antara sistem dan daerah diluar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup atau kemampuan sistem.

5. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengenadalian diwujudkan dengan menggunakan umpan balik, yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun keluaran. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

6. Lingkungan

Lingkungan adalah segala sesuatu yang berada di luar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau

menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus dijaga, karena akan memacu terhadap kelangsungan hidup sistem.

2.1.1 Pengertian sistem

Terdapat 2 kelompok pendekatan dalam mendefenisikan sistem yaitu menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang di tekankan pada prosedur menurut Jogiyanto HM(2001:1) mendefenisikan sistem sebagai berikut :

“sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan pada elemen atau komponen nya menurut Jogiyanto HM(2001:2)mendefenisikan sistem sebagai berikut: ”sistem merupakan kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”. Sistem dapat di artikan sebagai kumpulan atau grup dari subsistem/bagian/komponen apapun baik fisik maupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan tertentu. Kumpulan elemen tersebut terdiri dari hardware,software,prosedur,sumber daya dan data-data yang saling

berhubungan dan saling melengkapi satu sama lain untuk mencapai suatu tujuan tertentu.

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu :

1. Komponen sistem (component)

   Suatu system terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan.komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem bagian-bagian dari sistem.setiap subsistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas sistem (boundary)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.Batas sistem ini memungkinkan suatu sistem di pandang sebagai suatu kesatuan.Batas suatu sistem menunjukan ruang lingkup dari sistem tertentu.

3. Lingkungan luar sistem (enviroment)

   Lingkungan luar sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung system (interface)

Penghubung merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya.melalui media penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem yang lain nya.Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan sistem (input)

   Energy yang di masukkan kedalam suatu sistem,masukan ini dapat berupa masukan perawatan(maintanance) dan memasukkan signal.

   Masukan perawatan yaitu energy yang dimasukkan sistem supaya sistem tersebut beroperasi,sedangkan masukkan signal yaitu energy yang diproses untuk mendapatkan keluaran.

6. Keluaran sistem (output)

   Keluaran adalah hasil dari energy yang di olah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolah sistem (proses)

   Suatu sistem dapat mempunyai suatu bagian pengolahan atau sistem itu sendiri sebagai pengolahnya,pengolah akan merubah masukan menjadi keluaran.

8. Sasaran sistem (goal)

Suatu sistem pasti mempunyai tujuan atau sasaran. Sasaran dari suatu sistem sangat menentukan sekali masukan yang di butuhkan dan keluaran yang akan dihasilkan sistem.suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan nya.

2.2 Konsep Dasar Informasi

Seperti halnya sistem, informasi juga merupakan hal yang penting, keduanya saling mendukung dalam membentuk sebuah sistem informasi. Informasi memiliki nilai yang sangat berharga bagi orang yang membutuhkannya, namun sebelum menjadi informasi, data-data yang ada mula-mula harus di olah dengan baik agar informasi yang dihasilkan sesuai dengan apa yang diharapkan.

Menurut Susanto (2004:40), informasi adalah hasil pengolahan data yang memberikan arti dan manfaat.

Jadi informasi dapat dikatakan sebagai hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan data tersebut bisa dijadikan informasi, hasil pengolahan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut. Dari uraian tentang informasi ini ada tiga hal penting yang harus diperhatikan, yaitu:

1. Informasi merupakan hasil pengolahan data
2. Memberikan makna atau arti

3. Berguna atau bermanfaat

2.2.1 Pengertian Informasi

Menurut Jogiyanto HM (2001:8) Informasi dapat di defenisikan sebagai berikut *: "informasi adalah data yang di olah menjadi bentuk yang berguna dan lebih berarti bagi yang menerimanya"* . Sumber dari informasi adalah data.

2.2.1.1 Pengertian Data

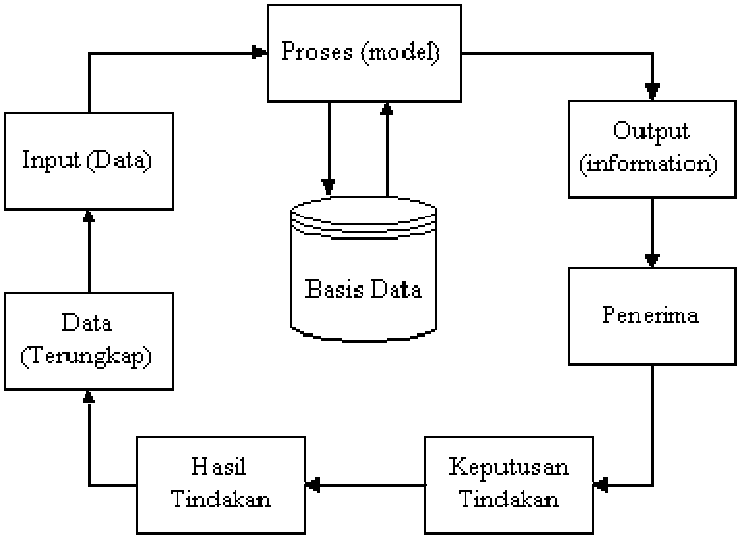
Menurut Azhar Susanto (2000:37) Mendefenisikan data sebagai berikut : "data adalah fakta atau apapun yang dapat digunakan sebagai input untuk menghasilkan informasi". Maka data adalah suatu karangan tertulis mengenai fakta atau kenyataan yang masih berdiri sendiri, belum mempunyai pengertian sebagai kelompok,belum terkoordinasi satu sama lain dan belum di olah sesuai dengan keperluan tertentu menjadi suatu bentuk yang sesuia dengan keperluan pengguna informasi yang bersangkutan.

2.2.2 Siklus Informasi

Informasi merupakan suatu proses perubahan dunia menjadi informasi. Data yang di olah untuk menghasilkan informasi menggunakan suatu model proses tertentu. Data yang diolah melalui suatu model menjadi informasi,penerima kemudian menerima informasi tersebut ,membuat dan suatu keputusan dan

melakukan tindakan yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input,

diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus, adapun siklus informasi sebagai berikut :



Gambar 2.1 Siklus Informasi

Sumber : http://referensisisteminformasi.blogspot.com/2008/07/siklus-informasi.html

2.2.3 Kualitas Informasi

Kualitas suatu informasi dilihat dari 3 hal yaitu :

a. Akurat artinya suatu informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan bagi si pemakai.

b. Tepat waktu artinya suatu informasi harus sesuai dengan keadaan saat itu,keterlambatan suatu informasi bisa berakibat fatal bagi organisasi pemakainya dalam hal ini dikarenakan informasi merupakan landasan dalam pengambilan sebuah keputusan.
c. Relevan artinya suatu informasi harus bermanfaat bagi pengguna nya dalam waktu pendek ataupun dalam waktu panjang.

2.3 Pengertian Sistem Informasi

Sistem informasi adalah kumpulan dari sub sub sistem baik fisik maupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan yaitu mengolah data menjadi informasi yang berguna .komponen dasar sistem informasi meliputi:

1. Blok masukan (*input blok*)

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan di masukkan ,dapat berupa dokumen –dokumen dasar.

2. Blok model (*model block*)

   Blok ini terdiri dari kombinasi prosedur,logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang di inginkan.

3. Blok teknologi (*technology block*)

Teknologi digunakan untuk menerima input,menjalankan model,menyimpan dan mengakses data,menghasilkan dan mengirimkan keluaran dan membantu pengendalian sistem secara keseluruhan.

4. Blok keluaran (*blok output*)

   Keluaran merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

5. Blok basis data (*database block*)

   Merupakan kumpulan data yang saling berhubungan satu sama lain,tersimpan di perangkat keras computer dan digunakan perangkat lunak untuk memanipulasi nya.

   Data perlu disimpan dalam basis data untuk keperluaan penyediaan informasi lebih lanjut.data di dalam basis data perlu di organisasikan sedemikian rupa,supaya informasi yang di hasilkan berkualitas.

6. Blok kendali (*controls block*)

   Beberapa pengendalian perlu di rancang dan di terapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat di cegah ataupun bila terlanjur terjadi kesalahan langsung cepat diatasi.

   Tujuan dilakukan nya tahap ini sebagai berikut:

a) Membuat keputusan apabila sistem saat ini mempunyai masalah atau sudah tidak berfungsi secara baik dan hasil analisisnya digunakan sebagai dasar untuk memperbaiki sistem
b) Mengetahui ruang lingkup pekerjaan yang akan ditangani.
c) Memahami sistem yang sedang berjalan saat ini,mengidentifikasikan dan mencari solusi.

2.4 Sistem basis data

Menurut Fathansyah,Ir (2002:9) Basis data adalah :

Sistem basis data merupakan sistem yang terdiri dari atas sekumpulan tabel yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program (DBMS) yang menungkinkan beberapa pemakai lain untuk mengakses dan memanipulasi table-tabel tersebut.

Sistem Basis data pada dasarnya di buat untuk tujuan Sebagai berikut: (fatansyah 2002:5)

1. Kecepatan dan kemudahan

   Dengan sistem basis data diharapkan pengolahan data dapat lebih cepat dan lebih mudah di bandingkan dengan sistem manual.

2. Efisiensi ruang penyimpanan

   Dengan basis data,efisiensi /optimalisasi penggunaan ruang penyimpan dapat dilakukan,karena kita dapat melakukan penekanan jumlah redudansi data.

3. Keakuratan

   Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan atau batasan tipe,data,domain data ,keunikan data dan sebagainya ,yg secara ketat dapat diterapkan dalam sebuah basis data ,sangat berguna untuk menekan ketidak akuratan masukan ataupun penyimpanan data.

4. Ketersediaan

   Dengan pertumbuhan data yang semakin banyak,maka di perlukan tempat penyimpanan yang sangat besar,karena itu kita perlu memilah data dalam bentuk kategori-kategori tertentu sehingga data yang tidak terlalu penting dapat di hapus.

5. Kelengkapan

   Untuk mengakomodasi kebutuhan kelengkapan data yang semaki berkembang maka kita tidak hanya dapat menambah record-record data tetapi juga melakukan.

   perubahan struktur dalam basis data,baik dalam bentuk penambahan objek baru (tabel)atau dengan penambahan field-field baru pada suatu tabel.

6. Keamanan

   Dengan sistem basis data kita dapat memproteksi semua objek yang ada dengan berbagai tingkatan keamanan.

7. Kebersamaan Pemakaian

   Pemakai basis data biasanya terdiri dari banyak pemakai,sehingga hendaknya basis data harus bisa di akses oleh banyak pemakai dalam waktu yang bersamaan.

Dalam sebuah Basis Data secara lengkap akan terdapat komponen-komponen utama yaitu

1. Hardware
2. Operating system
3. Database
4. Sistem(aplikasi/perangkat lunak)pengelola basis data(DBMS)
5. User

Keuntungan Database adalah :

a. Mereduksi redudansi
b. Data dapat di share antar aplikasi
c. Standarisasi data dapat di lakukan
d. Batasan security dapat diterapkan
e. Mengelola integritas(keterjaminan akurasi)data
f. Menyeimbangkan kebutuhan yang saling konflik
g. Independesi data (obyektif DBS)

Struktur sistem Database :

a. File manager : mengelola space dan struktur data.
b. Database manager : menyediakan antar muka dengan data fisik
c. Query processor : menterjemahkan query ke instruksi yang dimengerti database manager
d. DML precompiler : mengkonversi perintah DML menjadi calls normal.
e. DDL compiler : menkonversi perintah DDL menjadi meta data.

2.5 Alat-Alat (Tools) Yang Digunakan

2.5.1 UML (*Unified Modeling Language*)

Menurut Bambang Hariyanto. Ir.,MT. (2004:17)

UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi. Pemaduan beberapa notasi di beragam metodologi usaha bersama dari banyak pihak, di dukung oleh kakas- kakas yang di integrasikan lewat XML . Standar UML di kelola oleh OMG(*Object Management Group*).

2.5.1.1 Pengenalan UML

UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem.

1. Dari dalam sistem intensive process metode diterapkan sebagai proses untuk menurunkan atau mengevolusikan sistem.
2. Sebagai bahasa UML digunakan untuk komunikasi yaitu alat untuk menangkap pengetahuan mengenal suatu subyek dan mengekspresikan pengetahuan (sintaks) yang mempedulikan subyek. Untuk maksud berkomunikasi subyek adalah sistem yang di bahas.
3. Begitu diterapkan untuk memvisualisasikan sistem UML digunakan untuk menjelaskan sistem secara visual. Sebelum direalisasikan.
4. Begitu di terapkan untuk membangun sistem UML dapat digunakan untuk memandu realisasi sistem serupa dengan *"blue print"*.
5. Begitu diterapkan untuk mendokumentasikan sistem, UML dapat di gunakan untuk menangkap pengetahuan sistem pada seluruh siklus hidup.

UML Bukanlah

1. Bahasa pemprograman visual.
2. Spesifikasi kakas, tapi spesifikasi pemodelan
3. Proses, tapi memungkinkan proses-proses.

Tujuan UML

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti

3. Mendukung spesifikasi independent bahasa pemprograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar kakas berorientasi objek.
6. mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen kolaborasi framework dan pattern.

2.5.1.2 Bagian-Bagian Dari UML

Bagian-bagian dari UML dibagi menjadi 2 kategori, yakni view dan diagram View digunakan untuk melihat sistem yang di modelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis view dalam UML antara lain *usecase,component view,deployment view.*

View

A. *Usecase* View

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang di inginkan external *actors*. *Actor* yang berinteraksi dengan sistem berupa user atau sistem lain nya. View ini di gambarkan dalam usecase diagram dan kadang-kadang dengan *activity* diagram. View ini digunakan terutama untuk pelanggan perancangan (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

B. Logical View

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class,object* dan *relationship*) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu. View ini di gambarkan dalam *class diagram* untuk struktur statis dan dalam state, sequence, collaboration, dan *activity* diagram untuk model dinamisnya view ini digunakan untuk perancangan (*designer*) dan pengembang (*developer*).

C. *Component View*

Mendeskripsikan implementasi dan ketergantungan modul komponen yang merupakan tipe lainnya dari code module, diperlihatkan dengan struktur dan ketergantungan juga alokasi sumber daya komponen dan informasiadministrative lain nya. View ini di gambar kan dalam component view dan digunakan untuk pengembang.

D. *Concurrency View*

Membagi sistem kedalam proses dan processor. View ini digambarkan dalam diagram dinamis (*state, sequence, collaboration, dan activity diagram*) dan diagram implementasi (*Component Development Diagram*) serta digunakan untuk pengembang *(developer), pengintegrasi (Integrator*), dan penguji (*tester*).

E. *Deployment View*

Mendeskripsikan fisik dari sistem seperti computer dan perangkat (nodes) dan bagaimana hubungan nya dengan yang lain nya.

Diagram

Diagram bentuk grafik yang menunjukan symbol elemen model yang di susun untuk mengilustrasikan bagian atau aspek tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu, adapun jenis diagram antara lain :

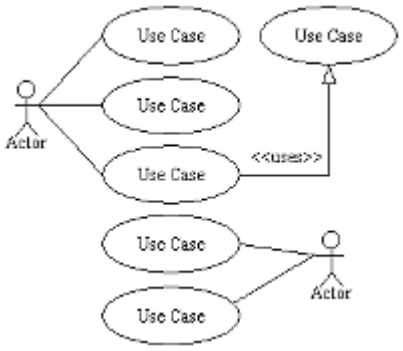
A. *Usecase Diagram*.

Usecase diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *usecase* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Usecase* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Usecase diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua *feature* yang ada pada sistem. Sebuah *usecase* dapat meng-include fungsionalitas *usecase* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *usecase* yang di-include akan dipanggil setiap kali *usecase* yang meng-include dieksekusi secara normal. Sebuah *usecase* dapat di-include oleh lebih dari satu *usecase* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah *usecase* juga dapat meng-extend *usecase* lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi

antar *usecase* menunjukkan bahwa *usecase* yang satu merupakan spesialisasi dari yang lain.

Contoh *usecase diagram* :



Gambar 2.2 *Usecase* Diagram

Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

B. Class Diagram

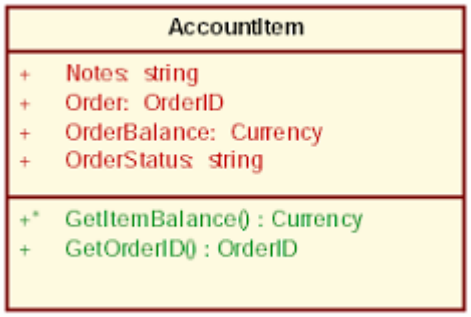
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class, package dan objek* beserta hubungan satu sama lain seperti *containment*, pewarisan, *asosiasi*, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a) *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
b) *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
c) *Public*, dapat dipanggil oleh siapa saja

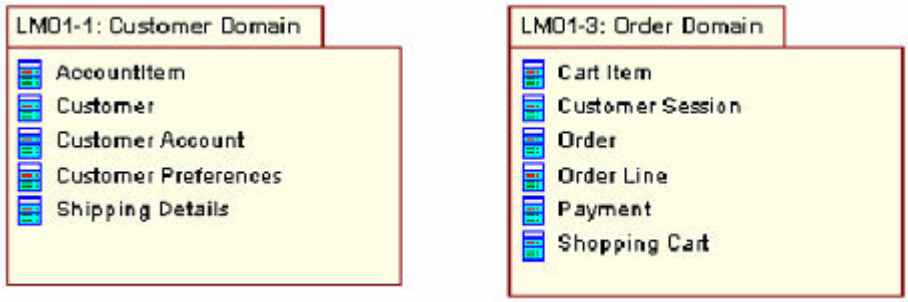


Gambar 2.3 *Class Ikon diagram*

Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

Class dapat merupakan implementasi dari sebuah *interface*, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time.

Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package.



Gambar 2.4 *Relation Class Diagram*

Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

Hubungan Antar Class

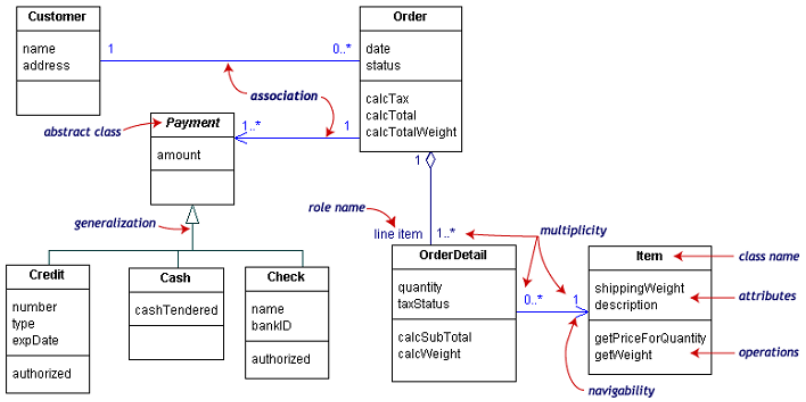
1. *Asosiasi*, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class.

2. *Agregasi*, yaitu hubungan yang menyatakan bagian ("terdiri atas..").

3. Pewarisan, yaitu hubungan hirarkis antar class. *Class* dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4. Hubungan dinamis, yaitu rangkaian pesan (message) yang di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.

Contoh class diagram:



Gambar 2.5 *Class Diagram*

Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

C. *Statechart Diagram*

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statech diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statech diagram). Dalam UML, state digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar state umumnya

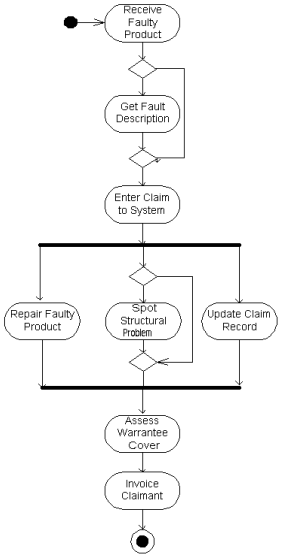
memiliki kondisi guard yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. Action yang dilakukan sebagai akibat dari event tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

D. *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram merupakan state diagram khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *usecase* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *usecase* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan

behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Contoh *activity diagram tanpa swimlane*:



Gambar 2.6 *Diagram Activity*

Sumber :

http://www.idemployee.id.tue.nl/p.markopoulos/downloadablePapers/TUPIS2000/P_MarkopoulosTupis00.html

E. *Sequence Diagram*

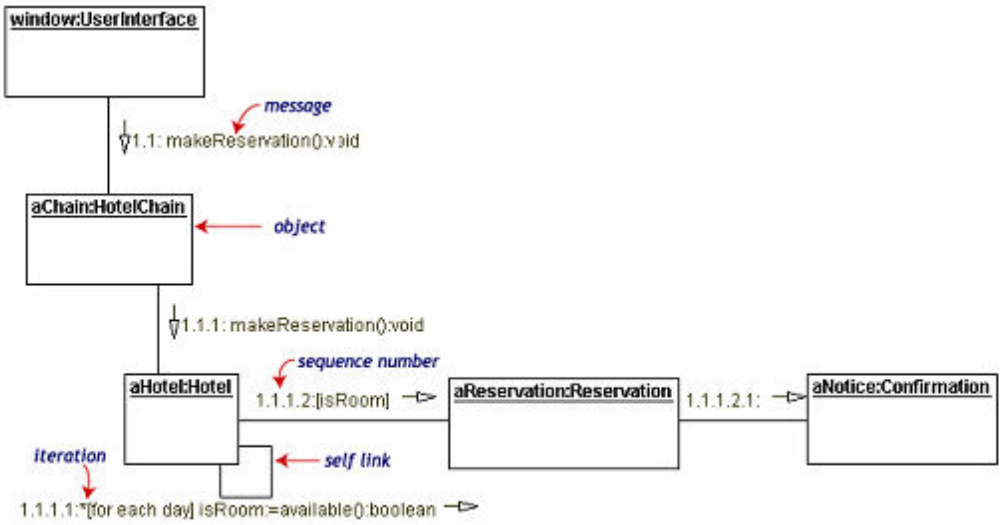
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untukobjek boundary, controller dan persistent entity.

F.*Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing

objek dan bukan pada waktu penyampaian message. Setiap message memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1.

Messages dari level yang sama memiliki prefiks yang sama.



Gambar 2.7 *Collaboration Diagram*

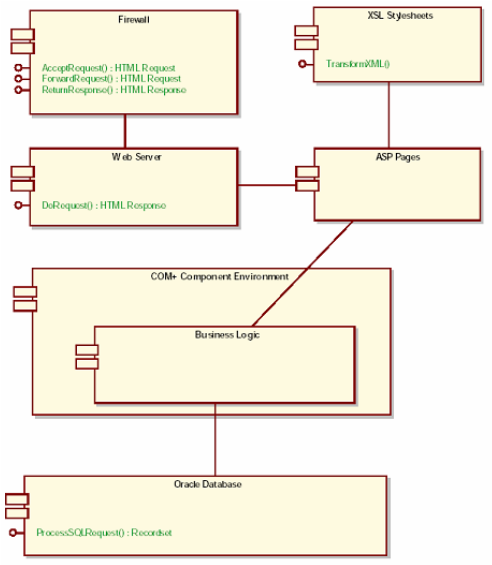
Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

F. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time. Umumnya

komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

Contoh *component diagram*:



Gambar 2.8 *Component Diagram*

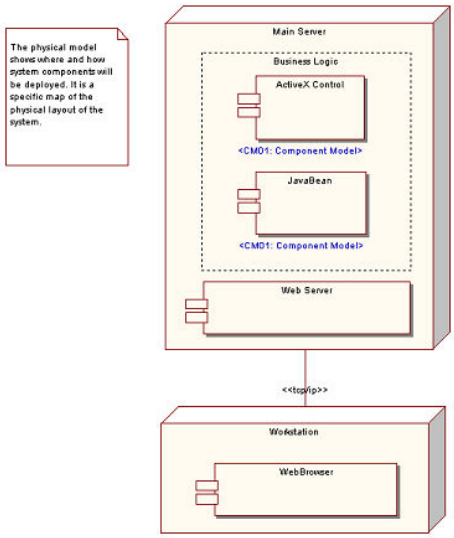
Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

H. *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan

jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal . Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

Contoh *deployment diagram* :



Gambar 2.9 *Deployment Diagram*

Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

Tool Yang Mendukung UML

Saat ini banyak sekali tool pendesainan yang mendukung UML, baik itu tool komersial maupun opensource. Beberapa diantaranya adalah:

1. Rational Rose (www.rational.com)
2. Together (www.togethersoft.com)
3. Object Domain (www.objectdomain.com)
4. Jvision (www.object-insight.com)
5. Objecteering (www.objecteering.com)
6. MagicDraw (www.nomagic.com/magicdrawuml)
7. Visual Object Modeller (www.visualobject.com)

2.6 Bahasa Pemprograman Yang di Gunakan

Bahasa pemprograman yang digunakan dalam penulisan skripsi ini adalah PHP, HTML, Mysql.

2.6.1 PHP (*personal home page*)

PHP merupakan bahasa pemprograman server side yang di desain khusus untuk aplikasi web yang di tambahkan kedalam HTML . Sifat server side berarti pengerjaan skrip akan dilakukan di server ,baru kemudian hasilnya dikirim ke browser.

Kelebihan PHP dari bahasa pemrograman lain

(http://id.wikipedia.org/wiki/*php*/ 29 maret 2009)

a) Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilas dalam penggunaanya.

b) Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttp, hingga Xitami dengan configurasi yang relatif mudah.
c) Dalam sisi pengembangan lebih mudah, karena banyaknya millis dan developer yang siap membantu dalam pengembangan.
d) Dalam sisi pemahamanan, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
e) PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.
f) Skrip tidak dapat di intip dengan menggunakan fasilitas view HTML source.

PHP dirancang sedemikian khusus untuk membentuk web dinamis artinya dapat membentuk suatu tampilan berdasarkan permintaan terkini.

Pada saat ini PHP cukup popular sebagai peranti pemprograman web karena dapat berfungsi pada server-server yang berbasis unix,linux,windows dan mancitosh dan untuk mencoba kita tidak perlu menggunakan computer berkelas server ,cukup dengan menggunakan computer biasa kita bisa mempelajari dan mempraktekkan PHP.

9 Tipe data yang di miliki PHP :

1. Integer
2. Double
3. Boolean
4. String
5. Object
6. Array
7. Null
8. Nill
9. Resource

Program HELLO WORLD yang dituliskan PHP adalah sebagai berikut :

```
<?php
echo "<b>Hello World</b>" ;
echo "<b>Halo Dunia</b>" ;
?>
```

2.6.2 HTML (*Hypertext Markup Languange*) Dan CSS (*Cascading Style Sheet*)

(http://id.wikipedia.org/wiki/*HTML* / 29 maret 2009)

HyperText Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser Internet. Bermula dari sebuah bahasa yang

sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML *(Standard Generalized Markup Language),* HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C).

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan browser web seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: **TAMPIL TEBAL**, maka penulisannya dilakukan dengan cara: TAMPIL TEBAL. Tanda digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda untuk menonaktifkan cetak tebal tersebut.

Secara garis besar, terdapat 4 jenis elemen dari HTML:

1. *Structural*. tanda yang menentukan level atau tingkatan dari sebuah teks (contoh, <h1>Golf</h1> akan memerintahkan browser untuk

menampilkan "Golf" sebagai teks tebal besar yang menunjukkan sebagai Heading 1.

2. *Presentational*. tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, boldface akan menampilkan **bold**. Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks.
3. *hypertext*. tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain (contoh,Wikipedia akan menampilkan wikipedia sebagai sebuah hyperlink ke URLtertentu).
4. Elemen *widget* yang membuat objek-objek lain seperti tombol (<button>), list (<li>), dan garis horizontal (<hr>).

Selain markup *presentational* , markup yang lin tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan tag HTML untuk menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheets.*

Contoh dokumen HTML sederhana

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Halo HTML</title>
</head>
<body>
<p>Halo Dunia!</p>
</body>
</html>
```

Yang bisa dilakukan dengan HTML yaitu :

1. Mengontrol tampilan dari web dan content nya
2. Mempublikasikan dokumen secara online sehingga bisa di akses dari seluruh dunia.
3. Membuat online form yang bisa digunakan untuk menangani pendaftaran,transaksi secara online.
4. Menambahkan objek2 seperti image,audi,video dan juga java applet dalam dokumen dan HTML.

Sedangkan CSS merupakan feature yang sangat penting dalam dynamic HTML. Meskipun bukan merupakan suatu keharusan dalam membuat web,akan tetapi penggunaan style sheet merupakan kelebihan tersendiri.

Suatu style sheet merupakan tempat dimana dapat mengontrol dan memanage style-style yang ada. Style sheet mendeskripsikan bagaimana tampilan document HTML yang menggunakannya,juga bisa membuat efek-efek spesial di web anda dengan

menggunakan style sheet. Sebagai contoh anda bisa membuat style sheet yang mendefenisikan style untuk <H1> dengan style bold dan italic dan berwarna biru. Atau pada tag <P> yang akan di tampilkan dengan warna biru dan menggunakan font gothic dan masih banyak lagi yang bisa dilakukan dengan style sheet. Secara teoritis bisa menggunakan style sheet technologi denga HTML. Akan tetapi pada prakteknya hanya Cascading Style Sheet (CSS) tehnology yang support pada hampir semua web browser .karena css telah di standarakan oleh World Wide Web consortium(w3c) untuk digunakan di web browser.

2.6.3 Java Script

(http://id.wikipedia.org/wiki/*JavaScript* / 29 maret 2009)

JavaScript adalah nama implementasi Netcape Communication Corporation untuk ECMAScript standar, suatu bahasa skripsi yang didasarkan pada konsep pemrograman berbasis prototipe. Bahasa ini terutama terkenal karena penggunaannya di situs web (sebagai JavaScript sisi klien) dan juga digunakan untuk menyediakan akses skrip untuk objek yang dibenamkan (*embedded*) di aplikasi lain.

Walaupun memiliki nama serupa, JavaScript hanya sedikit berhubungan dengan bahasa pemrograman Java, dengan kesamaan utamanya adalah penggunaan sintaks C. Secara semantik, JavaScript memiliki lebih banyak kesamaan dengan bahasa pemrograman Self.

Skrip JavaScript yang dimasukkan di dalam berkas HTML ataupun XHTML harus dimasukkan di antara tag <script>...</script>. Berikut ini adalah contohnya (yang akan menampilkan sebuah *dialog box* berisi Halo Dunia! ketika sebuah tombol diklik oleh pengguna) :

```
<input type="button" value="Tekan di sini" onclick="halo();">
<script type="text/javascript">
 function halo() {
 alert( "Halo Dunia!" );
  }
</script>
```

2.6.4 Ajax

Asynchronous JavaScript And XML, atau disingkat AJAX, adalah suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web interaktif. Tujuannya adalah untuk memindahkan sebagian besar interaksi pada komputer web surfer, melakukan pertukaran data dengan server di belakang layar, sehingga halaman web tidak harus dibaca ulang secara keseluruhan setiap kali seorang pengguna melakukan perubahan. Hal ini akan meningkatkan interaktivitas, kecepatan, dan usability. Ajax merupakan kombinasi dari:

1. XHTML (atau HTML) dan CSS untuk bahasa mark up dan tampilan.

2. DOM yang diakses dengan client side scripting language, khususnya implementasi ECMAScript JavaScript dan JScript, untuk menampilkan secara dinamis dan berinteraksi dengan informasi yang ditampilkan seperti
3. Objek XMLHttpRequest untuk melakukan pertukaran data asinkronus dengan web server. Pada beberapa framework dan kasus Ajax, objek IFrame lebih dipilih daripada XMLHttpRequest untuk melakukan pertukaran data dengan web server.
4. XML umumnya digunakan sebagai format untuk pengiriman data, walaupun format lain juga memungkinkan, seperti HTML, plain text, JSON dan EBML. Seperti halnya DHTML, LAMP, atau SPA, Ajax bukanlah teknologi spesifik, melainkan merupakan gabungan dari teknologi yang dipakai bersamaan. Bahkan, teknologi turunan/komposit yang berdasarkan Ajax, seperti AFLAX sudah mulai bermunculan.

2.7 Perangkat Lunak Yang Digunakan

Database MySQL

http://id.wikipedia.org/wiki/*MySQL*/ 29 maret 2009

MySQL adalah sebuah perangkat lunak sistem manajemen bass data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread,

multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson dan Michael "Monty" Widenius.

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat

oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

1. Portability

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Open Source

   MySQL didistribusikan secara open source (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. MultiuserMySQL

   Dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Performance tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Column types

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Command dan functions

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam query.

7. Security

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

8. Scalability dan limits

MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Connectivity

MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

10. Localisation

    MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. Interface

    MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. Clients dan tools

    MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

13. Struktur tabel

    MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

2.8 Definisi Online

Secara umum, sesuatu dikatakan online adalah biala ia terkoneksi/terhubung dengan suatu jaringan ataupun sistem yang lebih besar. Beberapa arti kata lainnya yang lebih spesifik adalah sebagai berikut :

1. Dalam percakapan umum, jaringan/*network* yang lebih besar dalam konteks ini biasanya lebih mengarah pada *internet*, sehingga '*online*' menjelaskan status bahwa ia dapat diakses melalui *internet*.
2. Secara lebih spesifik dalam sebuah sistem yang terkait pada ukuran dalam satu aktifitas tertentu, sebuah elemen dari sistem tersebut dikatakan *online* jika elemen tersebut beroperasional. Sebagai contoh, sebuah instalasi pembangkit listrik dikatakan *online* jika ia dapat menyediakan listrik pada jaringan elekrik.
3. Dalam telekomunikasi, istilah *online* memiliki arti lain yang lebih spesifik. Suatu alat diasosiasikan dalam sebuah sistem yang lebih besar dkatakan *online* bila berada dalam kontrol langsung dari sistem tersebut. Dalam arti jika ia bersedia saat akan digunakan oleh sistem (*on-demand*), tanpa membutuhkan intervensi manusia, namun tidak bisa beroperasi secara mandiri di luar sistem tersebut.

2.9 Internet

http://id.wikipedia.org/wiki/*Internet*/ Internet 16 Novemver 2009

Singkatan dari Interconnection Networking. The network of the networks. Diartikan sebagai a global network of computer networks atau sebuah jaringan komputer dalam skala global/mendunia. Jaringan komputer ini berskala internasional yang dapat membuat masing-masing komputer saling berkomunikasi. Network ini membentuk jaringan inter-koneksi (Inter-connected network) yang terhubung melalui protokol TCP/IP. Dikembangkan dan diuji

coba pertama kali pada tahun 1969 oleh US Department of Defense dalam proyek ARPAnet. dalam arti akan membentuk suatu.

2.10 WWW (*Word Wide Web*)

http://id.wikipedia.org/wiki/*www*/ www 16 Novemver 2009

WWW (World Wide Web) adalah sebuah bagian dari internet yang sangat dikenal dalam dunia internet, dengan adanya WWW seorang dapat menampilkan sebuah halaman virtual yang disebut website.

Jika dilihat dari proses kerjanya WWW dapat dibagi menjadi beberapa komponen yaitu:

1. Protocol
2. Address
3. HTML (Hypertext Markup Language)
4. HTTP (Hypertext Transfer Protocol)
5. FTP (File Transfer Protocol)
7. Mailto
8. News
9. Telnet (Tele Networking)

Secara teknis, web adalah sebuah sistem dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah *internet webserver* dipresentasikan dalam bentuk *hypertext*. Informasi di web dalam bentuk teks umumnya ditulis dalam format HTML (*Hypertext Markup Language*). Informasi lainnya disajikan dalam bentuk *grafis* (dalam format GIF, JPG, PNG), *suara* (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World). Web dapat diakses

oleh perangkat lunak *web client* yang secara populer disebut sebagai *browser*. *Browser* membaca halaman-halaman web yang tersimpan dalam *webserver* melalui protokol yang disebut HTTP (*Hypertext Transfer Protocol*). Dewasa ini, tersedia beragam perangkat lunak *browser*. Beberapa diantaranya cukup populer dan digunakan secara meluas, contohnya seperti Microsoft Internet Explorer, Netscape Navigator, maupun Opera, namun ada juga beberapa produk browser yang kurang dikenal dan hanya digunakan di lingkungan yang terbatas. Sebagai dokumen *hypertext*, dokumen-dokumen di web dapat memiliki *link (sambungan)* dengan dokumen lain, baik yang tersimpan dalam *webserver* yang sama maupun di *webserver* lainnya. Link memudahkan para pengakses web berpindah dari satu halaman ke halaman lainnya, dan "berkelana" dari satu server ke server lain. Kegiatan penelusuran halaman web ini biasa diistilahkan sebagai *browsing*, ada juga yang menyebutnya sebagai *surfing* (berselancar).

2.11 *Apache* Server

Apache ini dibuat versi pertamanya oleh *Robert Mc Cool* — yang terlibat di NCSA– pada tahun 1996. Ditulis dalam bahasa C, perkembangannya dilakukan bersama rekan-rekan melalui email. Dia mengerjakan proyek itu bersama *Apache* groupnya : *Brian Behlendorf, Roy T. Fielding, Rob Hartill, David Robinson, Cliff Skolnick, Randy Terbush, Robert S. Thau, Andrew Wilson, Eric Hagberg, Frank Peters and Nicolas Pioch*. Kenapa diberi nama *Apache*? Kata yang mendirikan karena pertama mereka ingin menghargai penduduk asli Amerika Indian *Apache*

yang dikenal ketahanan dan skilnya saat perang, dan kedua karena akar proyek si apche ini merupakan sebuah *'a patchy server'*. Alasan kedua sebenarnya dengan hoki ditemukan. *Apach*e dikembangkan oleh komunitas terbuka yang di bawahi oleh *Apache Software Foundation*. Aplikasinya dapat digunakan untuk OS yang beragam(tidak bergantung pada vendor tertentu (*cross platform*)), sebut saja *UNIX, FreeBSD, Linux, Novell netware, MacOS X, Windows*. *Sumber (http://www.apache.org/ The Apache Software Foundation 31 desember 2009)*

2.12 Topologi Jaringan

Seperti yang kita ketahui topologi jaringan yang sering kita baca terbagi atas 3 jenis yaitu topologi bus, ring, dan star. Berikut penjelasan dari masing-masing topologi ;

a) Topologi Bus

Pada topologi Bus, kedua unjung jaringan harus diakhiri dengan sebuah terminator. Barel connector dapat digunakan untuk memperluasnya. Jaringan hanya terdiri dari satu saluran kabel yang menggunakan kabel BNC. Komputer yang ingin terhubung ke jaringan dapat mengkaitkan dirinya dengan mentap Ethernetnya sepanjang kabel. Linear Bus: Layout ini termasuk layout yang umum. Satu kabel utama menghubungkan tiap simpul, ke saluran tunggal komputer yang mengaksesnya ujung dengan ujung. Masing-masing simpul dihubungkan ke dua simpul lainnya, kecuali mesin di salah satu ujung

kabel, yang masing-masing hanya terhubung ke satu simpul lainnya. Topologi ini seringkali dijumpai pada sistem client/server, dimana salah satu mesin pada jaringan tersebut difungsikan sebagai File Server, yang berarti bahwa mesin tersebut dikhususkan hanya untuk pendistribusian data dan biasanya tidak digunakan untuk pemrosesan informasi. Instalasi jaringan Bus sangat sederhana, murah dan maksimal terdiri atas 5-7 komputer. Kesulitan yang sering dihadapi adalah kemungkinan terjadinya tabrakan data karena mekanisme jaringan relatif sederhana dan jika salah satu node putus maka akan mengganggu kinerja dan trafik seluruh jaringan. Keunggulan topologi Bus adalah pengembangan jaringan atau penambahan workstation baru dapat dilakukan dengan mudah tanpa mengganggu workstation lain. Kelemahan dari topologi ini adalah bila terdapat gangguan di sepanjang kabel pusat maka keseluruhan jaringan akan mengalami gangguan. Topologi linear bus merupakan topologi yang banyak dipergunakan pada masa penggunaan kabel Coaxial. Dengan menggunakan T-Connector (dengan terminator 50ohm pada ujung network), maka komputer atau perangkat jaringan lainnya bisa dengan mudah dihubungkan satu sama lain. Kesulitan utama dari penggunaan kabel coaxial adalah sulit untuk mengukur apakah kabel coaxial yang dipergunakan benar-benar matching atau tidak. Karena kalau tidak sungguh-sungguh diukur secara benar akan merusak NIC (network interface card) yang dipergunakan dan kinerja jaringan menjadi terhambat, tidak mencapai kemampuan maksimalnya. Topologi ini juga sering digunakan pada jaringan dengan basis

fiber optic (yang kemudian digabungkan dengan topologi star untuk menghubungkan dengan client atau node).

b) Topologi Ring

Topologi cincin adalah topologi jaringan dimana setiap titik terkoneksi ke dua titik lainnya, membentuk jalur melingkar membentuk cincin. Pada topologi cincin, komunikasi data dapat terganggu jika satu titik mengalami gangguan. Jaringan FDDI mengantisipasi kelemahan ini dengan mengirim data searah jarum jam dan berlawanan dengan arah jarum jam secara bersamaan.

c) Topologi Star

Topologi bintang merupakan bentuk topologi jaringan yang berupa konvergensi dari node tengah ke setiap node atau pengguna. Topologi jaringan bintang termasuk topologi jaringan dengan biaya menengah. Kelebihan kerusakan pada satu saluran hanya akan mempengaruhi jaringan pada saluran tersebut dan station yang terpaut, tingkat keamanan termasuk tinggi, tahan terhadap lalu lintas jaringan yang sibuk, Penambahan dan pengurangan station dapat dilakukan dengan mudah. Kekurangan jika node tengah mengalami kerusakan, maka seluruh jaringan akan terhenti. *Sumber Wahana Komputer (2003:6-8).*

BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Pada penelitian ini yang menjadi objek penelitian adalah sistem informasi tugas akhir mahasiswa pada jurusan Manajemen Informatika UNIKOM yang beralamat di kampus IV : Jln. Dipati Ukur 114 Bandung.

3.1.1 Sejarah Perusahaan

Pada tahun 1994, Dr.Ir. Eddy Suryanto Soegoto, M.Sc mendirikan Lembaga Pendidikan Komputer Indonesia Jerman (LPKIG) untuk program pendidikan 1 tahun salah satu program studinya adalah Sistem informasi dan Perpajakan. Program pendidikan ini berbasis pada pendidikan komputer dengan sistem praktek langsung dimana di semua Program Studi para siswanya diberikan pengetahuan perangkat lunak komputer terbaru saat itu, menempati kampus Jl. Dipati Ukur No. 102. Di tahun kedua dibuka program pendidikan 3 tahun dan 1 tahun untuk Program Studi Sistem informasi dan Perpajakan selain menempati kampus Jl. Dipati Ukur No. 102 juga menempati kampus di Jl. Dipati Ukur No. 118. Di tahun ketiga, nama Lembaga Pendidikan Komputer Indonesia Jerman (LPKIG) diubah menjadi Indonesian Germany Institute (IGI).

Pada tanggal 24 Desember 1998 dibentuk Yayasan Science dan Teknologi yang dilanjutkan dengan Pengajuan ke Kopertis IV Jabar untuk pendirian STIE IGI dan STMIK IGI.

Pada bulan juli 1999 keluarlah SK Mendiknas No. 119/D/O/1999 atas STIE IGI dengan 5 Program Studi yaitu Sistem informasi S1, Manajemen S1, Manajemen Pemasaran D3 & D1, Keuangan Perbankan D3 & D1, Sistem informasi D3 dan D1 dan hanya berselang 1 bulan, yakni dibulan Agustus 1999 menyusul keluarnya SK Mendiknas No. 143/D/O/1999 atas STIMK IGI juga dengan 5 Program Studi yaitu Teknik Informatika S1, Sistem informasi D3 & D1, Teknik Komputer D3 & D1, Teknik Informatika D3 & D1, Komputerisasi Sistem informasi D3 & D1.

Pada saat STMIK IGI dan STIE IGI berdiri, Institusi telah menempati kampus di Jl. Dipati Ukur No. 102 (Kampus III sekarang) dan Jl. Dipati Ukur No. 116 (Kampus II sekarang). Mengantisipasi pesatnya perkembangan Teknologi Informasi, IPTEK, Era Globaliasi dan Milenium ke-3 serta untuk memberikan yang terbaik bagi masa depan mahasiswa/i nya, Yayasan Science dan Teknologi kemudian mengajukan usulan ke DIKTI melalui Kopertis Wilayah IV Jabar untuk melakukan merger atas kedua Sekolah Tinggi diatas untuk menjadi Universitas Komputer Indonesia.

Pada hari selasa, tanggal 8 Agustus 2000 keluar SK Mendiknas No. 126/D/O/2000 atas nama Akreditasi Universitas Komputer Indonesia, disingkat UNIKOM. Akreditasi Mendiknas atas UNIKOM disertai dengan dibukanya 12

Program Studi baru yakni : Teknik Komputer (S1), Sistem informasi (S1), Teknik Industri (S1),Teknik Arsitektur (S1), Perencanaan Wilayah dan Kota (S1), Ilmu Hukum (S1), Ilmu Komunikasi (S1), Ilmu Pemerintahan (S1), Desain Interior (S1), Desain Komunikasi Visual (S1,D3&D1). Berkaitan dengan pendirian Universitas Komputer Indonesia, Yayasan Science dan Teknologi berturut-turut pada tahun 1998 membangun Kampus Baru The Miracle (Kampus I sekarang/Rektorat) sebanyak 5 tingkat dan pada tahun 1999 dilanjutkan dengan membangun kampus IV sebanyak 7 tingkat berlokasi di Jl. Dipati Ukur No. 112-114 Bandung. Setelah itu pada tahun 2004, UNIKOM merampungkan pembangunan kampus V dengan jumlah lantai sebanyak 6 lantai. Pada kampus V inilah sekretariat, ruang dosen, dan ruang ketua jurusan manajemen informatika bertempat. Dengan pembangunan ketiga kampus baru tersebut aktivitas belajar mengajar dilingkungan UNIKOM semakin semarak dan dinamis. Lokasi kampus yang strategis, terletak di Pusat Pendidikan kota Bandung dan mudah dijangkau dari berbagai arah memudahkan mahasiswa/i UNIKOM berinteraksi dan melaksanakan aktivitas perkuliahannya dengan lancar.

Universitas Komputer Indonesia memasuki tahun kedelapan dan akan melaksanakan Dies Nataliesnya pada hari Jum’at 8 Agustus 2008. Jajaran Pimpinan UNIKOM telah bertekad sesuai dengan Visi, Misi, dan Tujuan Universitas untuk memberikan pendidikan yang terbaik bagi masa depan mahasiswa/i nya dengan berbasis pada keunggulan teknologi informasi, komputer dan *software* mutakhir disamping keunggulan pada bidang studinya masing-masing.

3.1.2 Visi Dan Misi

Setiap organisasi pasti memiliki visi dan misi. Program Studi Manajemen Informatika UNIKOM juga merupakan sebuah organisasi yang tentunya memiliki visi dan misi.

a. Visi Program Studi Manajemen Informatika UNIKOM

Menjadi Program Studi unggulan di bidang sistem informasi dan penerapan teknologi informasi serta menjadi program studi yang dapat diterima sebagai panutan dalam pendidikan dan pengembangan teknologi sistem informasi di Indonesia.

b. Misi Program Studi Manajemen Informatika UNIKOM

Menyelenggarakan pendidikan bermutu tinggi di bidang sistem informasi bagi masyarakat sebagai sarana untuk menghasilkan lulusan yang berkualitas, profesional, kompeten dan sesuai dengan kebutuhan saat ini dan akan datang.

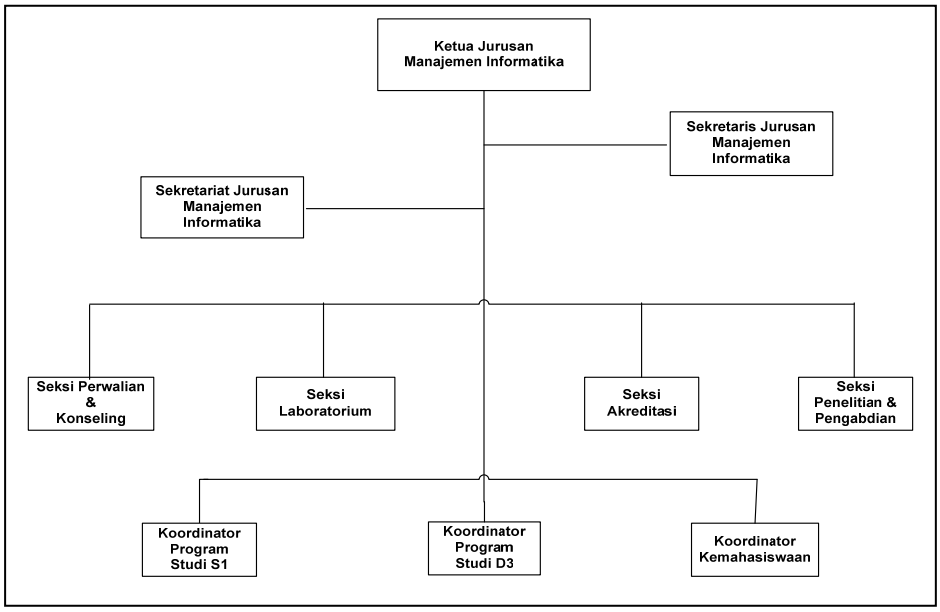
c. Tujuan Program Studi Manajemen Informatika UNIKOM

Sesuai dengan visi dan misi tersebut, tujuan pendidikan Program Studi Sistem informasi adalah membekali lulusan sarjana yang dihasilkan dengan kualifikasi : Mempunyai sikap profesional dan etika profesi yang tinggi berdasarkan ketaqwaan tehadap Tuhan YME dan kecintaan terhadap tanah air.

Mempunyai wawasan dan menguasai pengetahuan teknologi informasi dan penggunaannya untuk membantu pelaksanaan manajemen dan proses bisnis. Mampu melakukan rancang bangun sistem informasi berbasis komputer dan mengimplementasikannya dengan menggunakan metode, teknik, dan alat bantu tertentu berikut dokumentasinya. Mempunyai kemampuan untuk mengembangkan bidang ilmu / teknologi atau melanjutkan studinya.

3.1.3 Struktur Organisasi

Struktur organisasi merupakan hal yang penting bagi sebuah organisasi, termasuk organisasi Program Studi Manajemen Informatika UNIKOM. Berikut ini adalah struktur organisasi Program Studi Manajemen Informatika UNIKOM.



Gambar 3.1 Struktur Organisasi Jurusan Manajamen Informatika UNIKOM

[Sumber : mi.unikom.ac.id]

3.1.4 *Job Description*

Seperti yang telah diuraikan pada bagian sebelumnya, program studi sistem informasi ini dalam menjalankan manajemen kegiatannya dipimpin oleh Ketua Jurusan, dibantu oleh Sekretaris Jurusan dan Dosen serta Administrasi Jurusan (sekretariat jurusan). Di bawah ini adalah deskripsi kerja semua bagian yang terkait :

1. Ketua Jurusan

Tugas dari ketua jurusan manajemen informatika sebagai berikut :

a. Pimpinan akademik dan administratif di tingkat jurusan

b. Melakukan bimbingan bagi mahasiswa yang berada dalam perwalian secara akademik dan administratif
c. Mengkoordinasikan seluruh wewenang akademik dan administratif di Program Studi

2. Sekretaris Jurusan

Tugas dari sekretaris jurusan manajemen informatika sebagai berikut :

a. Menentukan, menyusun dan melakukan sosisalisasi mengenai seluruh kegiatan akademik secara operasional
b. Menentukan, menyusun dan melakukan sosialisasi sistem perkuliahan secara garis besar kepada dosen dan mahasiswa
c. Mengkoordinasikan pembagian jadwal perkuliah pada setiap semester

3. Sekretariat Jurusan

Tugas dari administrasi jurusan (sekretariat) sebagai berikut :

a. Pengolahan data perwalian.
b. Pengolahan data registrasi.
c. Pembuatan KHS.
d. Melayani mahasiswa untuk masalah perkuliahan.

4. Seksi Perwalian dan Konseling (Dosen Wali)

Tugas dari seksi perwalian dan konseling / dosen wali adalah sebagai berikut:

a. Mengarahkan dan membantu mahasiswa dalam menentukan perkuliahan yang akan diambil.
b. Mengikuti perkembangan studi setiap mahasiswa perwaliannya.
c. Memberikan konsultasi kepada mahasiswa yang berada dalam perwaliannya.

5. Koordinator Program Studi S-1 dan Program Studi D3

Tugas dari koordinator program studi S-1 dan koordinator program studi D3 adalah sebagai berikut :

a. Melaksanakan revisi kurikulum secara berkala
b. Mengamati perkembangan kurikulum yang ada dan menyesuaikan dengan perkembangan yang ada pada ketetapan kurikulum inti.
c. Studi banding kurikulum dengan perguruan tinggi lainnya.

6. Koordinator Kemahasiswaan

Tugas dari dosen pendamping kemahasiswaan jurusan manajemen informatika adalah sebagai berikut :

a. Melakukan sosialisasi dan koordinasi dengan unit himpunan atau kelompok minat yang berada dibawah jurusan atas seluruh kegiatan kurikulum dan ekstakurikuler.
b. Menjadi pelindung dan penasehat langsung bagi kegiatan-kegiatan ekstrakulikuler mahasiswa.

3.2 Metode Penelitian

Metode yang di gunakan penulis dalam pengembangan sistem adalah metode prototipe. Metode prototipe merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sebuah program dengan cepat dan bertahap sehingga dapat segera di evaluasi oleh pemakai (user). Berdasarkan pengertian metode prototipe diatas penulis mempunyai beberapa alasan mengapa penulis menggunakan metode pengembangan sistem prototipe yaitu karena penulis akan lebih mudah dalam merancang sistem yang di inginkan perusahaan dan dapat di terima oleh user sebagai pengguna sistem, hal lain nya adalah penulis menginginkan perancangan sistem yang telah dihasilkan kemudian di presentasikan kepada user dan user di berikan kesempatan untuk memberikan masukan ataupun komentar sehingga sistem informasi yang dihasilkan benar-benar sesuai dengan keinginan dan kebutuhan user terutama bagi perusahaan sendiri. Perubahan dan presentasi prototipe ini dapat dilakukan berkali-kali sampai di capai kesepakatan dari bentuk sistem informasi yang akan di implementasikan.

3.2.1 Desain Penelitian

Desain penelitian yang Penulis pakai disini adalah Penelitian Deskriptif yaitu suatu metode dalam meneliti status sekelompok manusia, suatu objek, suatu set kondisi, suatu sistem pemikiran ataupun suatu kelas peristiwa pada masa sekarang.

3.2.2 Jenis Dan Metode Pengumpulan Data

Jenis data yang digunakan dalam penelitian ini adalah data primer dan data sekunder.

3.2.2.1 Sumber Data Primer

Data primer adalah data yang diperoleh secara langsung dari unit pengamatan / responden penelitian. Metode pengumpulan data yang akan dilaksanakan adalah wawancara dan observasi.

1. Observasi

   Mengadakan pengamatan langusng ke lapangan, dalam hal ini Program Studi Manajemen Informatika UNIKOM.

2. Penelitian Lapangan

   Penelitian lapangan adalah melakukan penelitian ke obyek masalah yang di bahas untuk mendapatkan data yang akurat dan dapat dipertanggungjawabkan karena bersumber langsung dari pihak yang berhubungan dengan permasalahan yang diteliti.

3.2.2.2 Sumber Data Sekunder

Data sekunder adalah data yang digunakan untuk mendukung data primer, yaitu merupakan jenis data yang sudah diolah terlebih dahulu oleh pihak pertama.

Metode pengumpulan data yang akan dilaksanakan adalah dokumentasi, yaitu dengan cara mempelajari data-data jurusan yang dapat dijadikan pemicu untuk memahami persoalan yang muncul dalam sistem informasi tugas akhir mahasiswa yang sedang berjalan di Program Studi Manajemen Informatika UNIKOM, dan membaca buku-buku yang berkaitan dengan sistem informasi yang di rancang.

3.2.3 Metode Pendekatan Dan Pengembangan Sistem

Metode pendekatan sistem yang digunakan pada penelitian ini adalah pendekatan dengan Object Oriented yang menggunakan AOO(Analisis Object Oriented) dan DOO yang di visualisasikan dengan UML, sedangkan pengembangan sistemnya menggunakan metode pendekatan prototype.

3.2.3.1 Metode Pendekatan Sistem

Metode pendekatan sistem yang di gunakan adalah pendekatan dengan Object Oriented yang menggunakan AOO(Analisis Object Oriented) dan DOO yang di visualisasikan dengan UML dan di antara nya adalah sebagai berikut :

Use Case, , Sequence Diagram, Class Diagram, Collaboration Diagram, Component Diagram dan Deployment Diagram.

3.2.3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode pendekatan prototype.

Prototyping merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan. Prototype dibangun untuk mendefinisikan kebutuhan. Prototype akan dihilangkan sebagian atau seluruhnya dan perangkat lunak aktual direkayasa dengan kualitas dan implementasi yang sudah ditentukan .

Metode prototipe di rancang agar dapat menerima perubahan-perubahan dalam rangka menyempurnakan prototipe yang di buat dan perubahan-perubahan yang terjadi di anggap dapat merupakan sebagian dari proses pengembangan itu sendiri.

Berikut adalah langkah-langkah dalam merancang sistem yang penulis gunakan dalam mekanisme pengembangan sistem dengan prototipe , langkah-langkah tersebut adalah :

1. Mengidentifikasikan kebutuhan user agar bisa merancang sistem yang akan di bangun sesuai dengan harapan user . Sebelum pada tahap perancangan, tahap awal yang harus dilakukan adalah menganalisis sistem denga cara melakukan pengumpulan data, obervasi, dan wawancara serta membaca literatur yang berhubungan dengan sistem yang akan di bangun, baik dalam model interface, teknik, prosedural, maupun teknik yang di gunakan.
2. Pada tahap kedua akan di buat prototipe sistem tersebut untuk memperlihatkan kepada pemakai model sistem yang akan di rancang.

3. Pada tahap ketiga yaitu pengujian prototipe sistem yang telah di rancang untuk memastikan bahwa sistem tersebut dapat di gunakan dengan baik dan benar sesuai kebutuhan pemakai.
4. Pada tahap keempat yaitu mengembangkan versi produksi, penulis akan merampungkan sesuai dengan masukan terakhir dari pemakai dan memberkan gambaran bagaimana penggunaan sistem tersebut kepada pemakai setelah sistem tersebut di setujui.

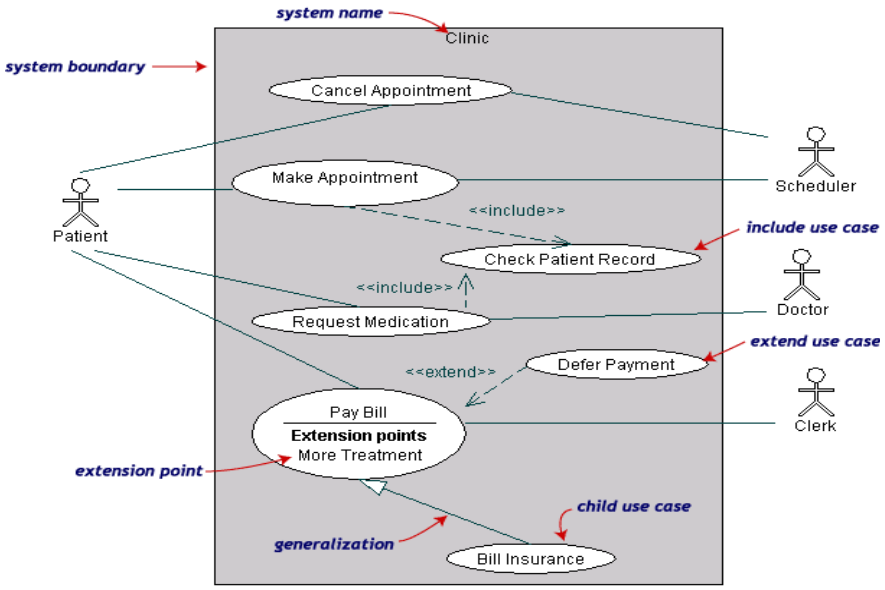
3.2.3.3 Alat Bantu Analisis Dan Perancangan.

Alat-alat bantu yang digunakan dalam suatu metodologi umumnya berupa suatu gambar, diagram atau grafik. Selain berbentuk gambar, alat-alat yang digunakan juga ada yang tidak berupa gambar atau grafik *(nongraphical tools)*. Alat-alat yang digunakan pada metodologi pendekatan prototype diantaranya adalah sebagai berikut :

1. Usecase Diagram.

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah usecase merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.



Gambar 3.2 contoh *Use-case diagram*

2. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)

2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private, tidak dapat dipanggil dari luar class yang bersangkutan
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. Public, dapat dipanggil oleh siapa saja

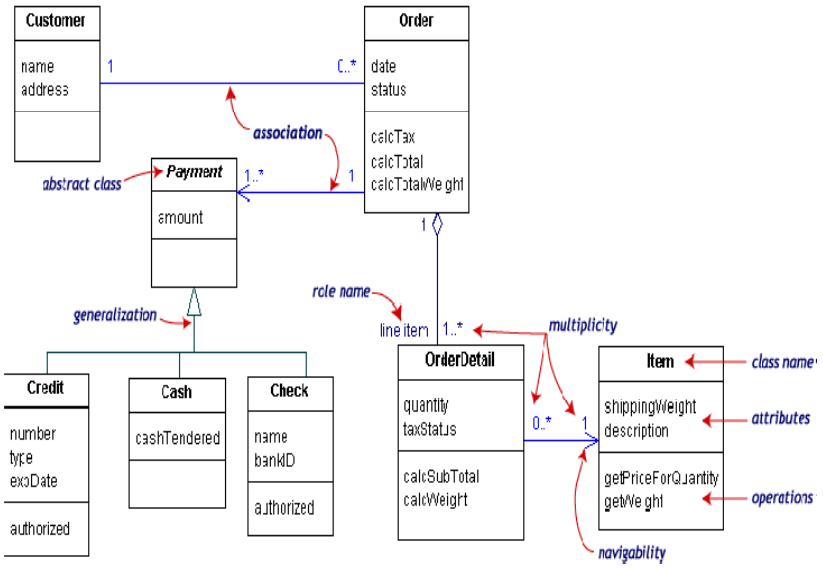
   Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time.

Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package.

Hubungan Antar Class

1) Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class.

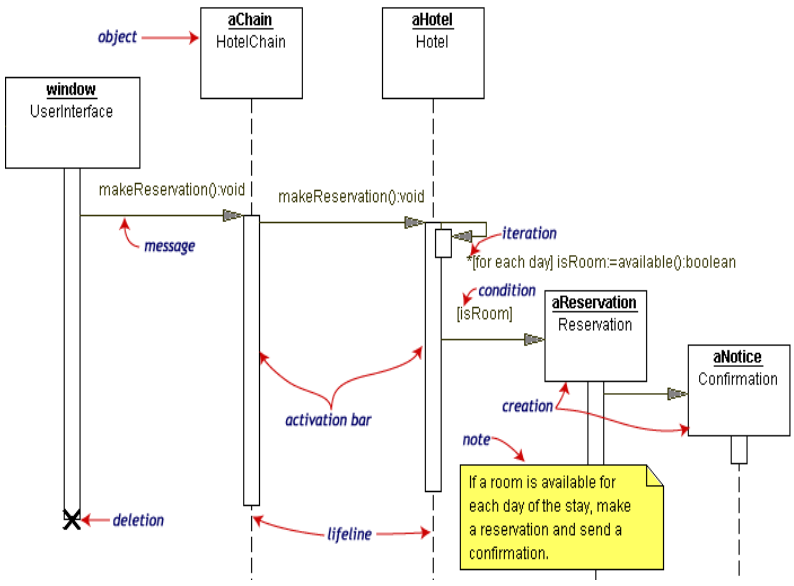
2) Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3) Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4) Hubungan dinamis, yaitu rangkaian pesan (message) yang di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.



Gambar 3.3 contoh *Class diagram*

3. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untukobjek boundary, controller dan persistent entity.

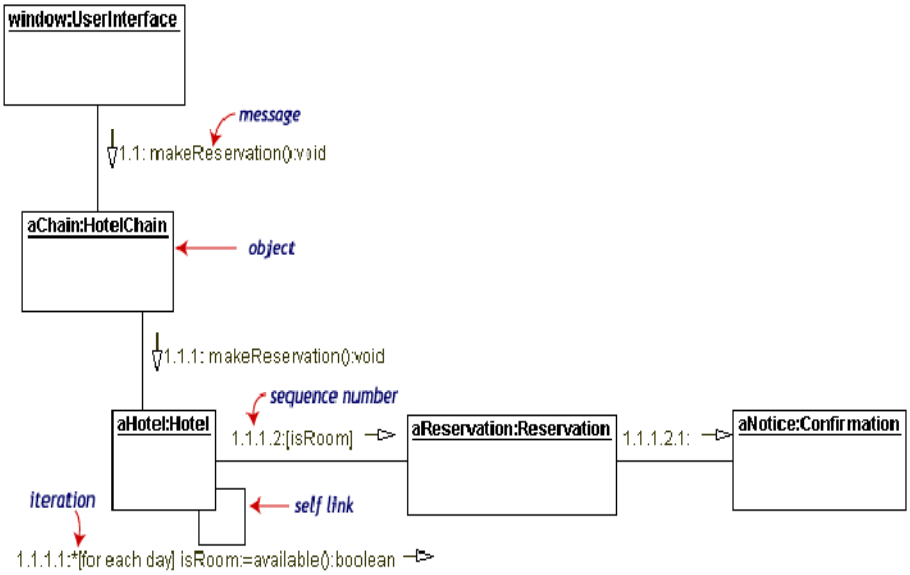


Gambar 3.4 Contoh *Sequence Diagram*

4.*Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1.

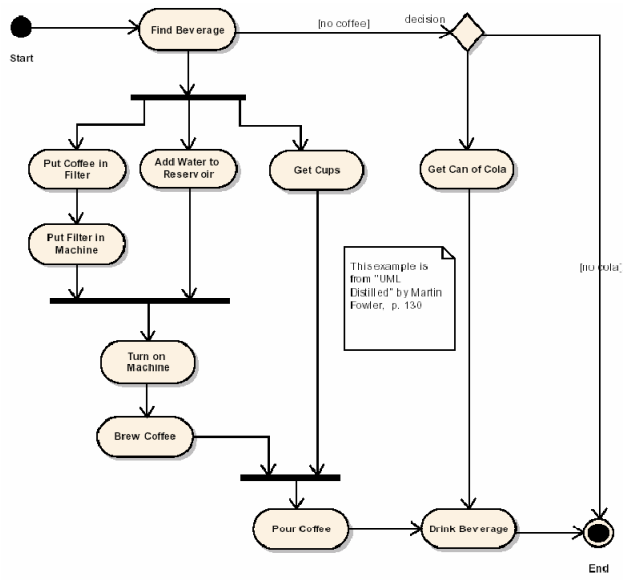
Messages dari level yang sama memiliki prefiks yang sama.



Gambar 3.5 *Contoh Collaboration Diagram*

5. Activity Diagram

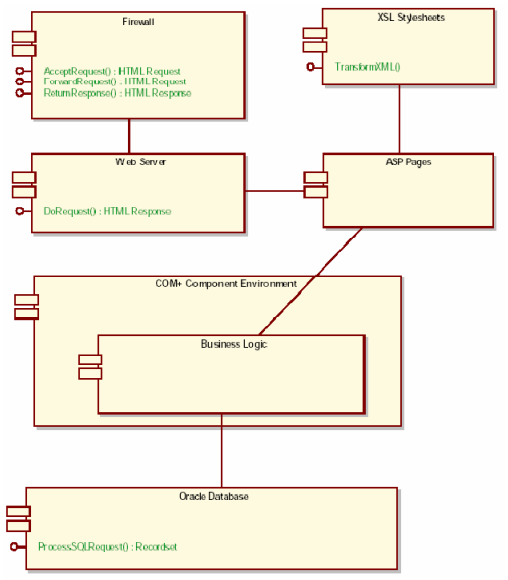
Sebuah *Activity Diagram* menunjukkan suatu alur kegiatan secara berurutan. *Activity Diagram* digunakan untuk mendiskripsikan kegiatan-kegiatan dalam sebuah operasi meskipun juga dapat digunakan untuk mendeskripsikan alur kegiatan yang lainnya seperti *use case* atau suatu interaksi. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.



Gambar 3.6 Contoh *Activity Diagram*

6. Component Diagram

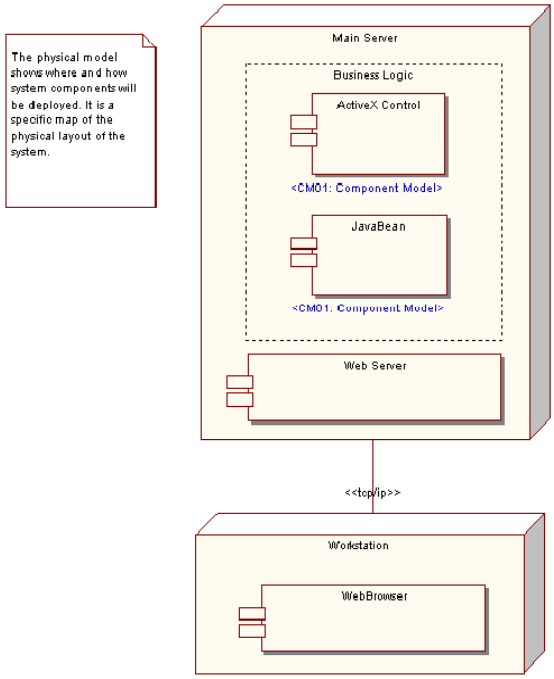
Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.



Gambar 3.7 *Contoh Component diagram*

7 . *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal . Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.



Gambar 3.8 Contoh *Deployment Diagram*

3.2.4 Pengujian Software

Pengujian software adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji dan dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan.

Dalam pengujian perangkat lunak ada dua cara pengujian yang bisa dipakai, yaitu pengujia *White Box* dan *Black Box.*

1. *White Box Testing*

   Adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

2. *Black Box Testing*

   Adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *White Box* merupakan metode perancangan data uji yang di dasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

   Dan pada penelitian ini akan digunakan teknik atau cara pengujian *Black Box / Black Box Testing*.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1. Analis Sistem Yang Berjalan

Analisis sistem yang berjalan meliputi analisis dokumen dan analisis prosedur.

4.1.1. Analisis Dokumen

Analisis dokumen ini di lakukan guna untuk mengetahui dokumen apa saja yang menjadi input, proses, dan output dari sistem ini. Dokumen input adalah dokumen yang akan diproses oleh sistem yang biasanya dilakukan oleh entitas luar system (mahasiswa yang mengambil tugas akhir), Dokumen proses adalah dokumen yang diperlukan oleh sistem dalam melakukan kegiatan pemprosesan data, sedangkan yang menjadi Dokumen output adalah dokumen yang dihasilkan oleh proses olahan sistem, yang berasal dari dalam sistem ke entitas luar (buku angkatan).

Dokumen yang di gunakan dalam system ini adalah :

1. Dokumen pengajuan proposal

   Nama Dokumen : Pengajuan Proposal

   Deskripsi : Formulir Pengajuan Proposal

Fungsi : sebagai form pengajuan proposal

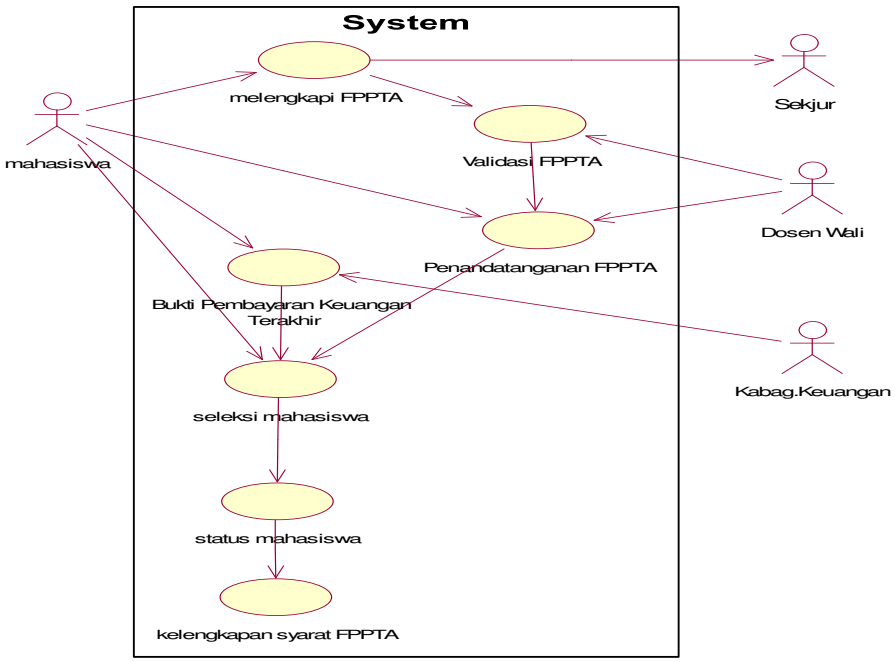
Sumber : Sekretariatan Jurusan Manajemen Informatika

Atribut : nama_lengkap, nim, alamat, no_tlp, judul_proposal, pembimbing.

2. Dokumen outputnya (Proposal Penelitian)

4.1.2. Analisis Prosedur yang sedang berjalan

4.1.1.1 *Usecase*



Keterangan:

1. FPPTA : Form Pengajuan Proposal Tugas Akhir

2. Status mahasiswa : mahasiswa reguler dan mahasiswa pending

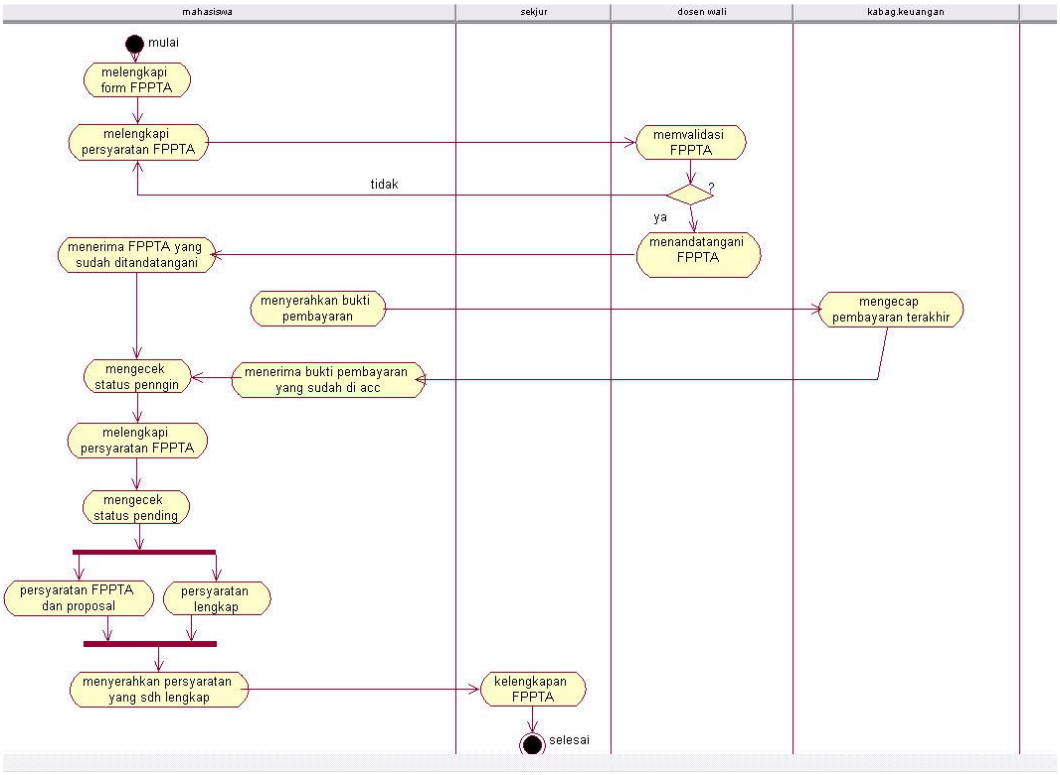
4. kabag.keuangan: kepala bagian keuangan

Gambar 4.1 *Usecase* Analisis Yang Sedang Berjalan

Ketika seorang mahasiswa yang hendak mengajukan Tugas akhir / Skripsi dia harus melengkapi persyaratan yang di sediakan panitia pelaksana tugas akhir dan skripsi dan mengumpulkan proposal penelitian bagi mahasiswa yang baru mengambil TA / Skripsi sedang bagi mahasiswa pending hanya mengumpulkan persyaratan lengkap yang sudah di tandatangani oleh dosen wali.

Dari use case di atas tampak empat actor yang berperan di dalam nya, yaitu: mahasiswa, sekjur, dosen wali, dan kabag. keuangan. Actor yang berperan utama di dalam system ini adalah panitian pelaksana tugas akhir / skripsi dan sekjur sedangkan mahasiswa merupakan entitas luar.

4.1.2.2 *Diagram Activity*



Keterangan:

1. FPPTA : Form Pengajuan Proposal Tugas Akhir

2. Status mahasiswa : mahasiswa reguler dan mahasiswa pending

4. kabag.keuangan: kepala bagian keuangan

Gambar 4.2 *Diagram Activity* Analisis Yang Sedang Berjalan

Visualisasi dari activity yang digambarkan disini merupakan bentuk Swimlane (partisi), dimana didalamnya digambarkan serta dijelaskan proses bisnis yang memasukkan beberapa Use Case sekaligus. Kenapa pada activity diagram yang disini tidak di gambar kan secara satu persatu dari masing use case seperti yang terdapat pada gambar Use case Diagram diatas, karena untuk lebih memudahkan pemvisualisasikan dari proses (pendaftaran dan pengajuan proposal) itu sendiri.

Walaupun dipisahkan (dibuatkan activity berdasarkan jumlah usecase), maka tampak membingungkan dikarenakan beberapa usecase hanya mempunyai 1 (satu) proses didalamnya (contoh: usecase / class kabag.keuangan),,dimana pada usecase tersebut hanya melakukan pencapan bukti pembayaran terakhir sebagi syarat untuk diserhakan bersam dengan persyaratan yang lain dan proposal kepada sekjur. Jadi, itulah mengapa pada activity diagramnya usecase tersebut digambarkan menjadi 1 dalam diagram activity.

Prosedur pengajuan proposal

1. Mahasiswa mengambil form pengajuan proposal TA (FPPTA) pada sekjur
2. Mahasiswa melangkapi seluruh data dan persyaratan pengajuan proposal
3. Dosen wali akan menandatangani form pengajuan proposal bila mahasiswa yang bersangakutan letah memenuhi syarat, yaitu:
    - S1: memiliki nilai Kerja Praktek (KP), telah mengikuti perkuliahan minimal 127 sks (LULUS), IPK minimal 2,00 (Tanpa nilai D > 10 sks, dan Nilai E)
    - D3: memiliki nilai Kerja Praktek (KP), telah mengikuti perkuliahan minimal 91 sks (LULUS), IPK minimal 2,00 (Tanpa nilai D > 10 sks, dan Nilai E)
    - Nilai kuliah wajib di validasi Jurusan MI (melalui email KAJUR / SEKJUR)

4. Mahasiwa wajib mengumpulkan bukti pembayaran terakhir (ACC.Kabag. Keuangan)
5. Mahasiswa wajib menyerahkan surat dan bukti pembayaran cuti bagi yang pernah cuti
6. Menyerahkan seluruh persyaratan diatas beserta proposal TA / Skripsi kepada Sekretarian MI (Sekjur).

4.1.3. Evaluasi Sistem yang sedang Berjalan

Setelah dilakukan analisis terhadap sistem yang sedang berjalan saat ini, maka dapat disimpulkan bahwa secara garis besar sistem pengajuan proposal pada jurusan manajemen informatika cukup baik walaupun masih memiliki kekurangan dan kelemahan, diharapkan dengan adanya Sistem Informasi Tugas Akhir Mahasiswa dapat membantu jurusan Manajemen Informatika dalam pengolahan data tugas akhir mahasiswa.

4.2 Perancangan Sistem

Perancangan sistem adalah gambaran, perancangan dan pembuatan skema atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan mempunyai fungsi dan tujuan. Elemen-elemen sistem informasi dirancang dengan tujuan untuk dikomunikasikan kepada *user*. Dalam pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang ada, baik secara keseluruhan

maupun memperbaiki sistem yang telah ada. Proses pengembangan sistem melewati beberapa tahapan, mulai dari sistem diterapkan, dioperasikan dan dipelihara.Bila sistem yang sudah dikembangkan masih timbul kembali permasalahan-permasalahan yang kritis serta dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali suatu sistem untuk mengatasinya dan proses ini kembali ketahap yang pertama yaitu perencanaan sistem. Siklus ini disebut juga dengan siklus hidup suatu sistem.

4.2.1. Tujuan Perancangan

Tujuan utama dari perancangan sistem adalah untuk memberikan gambaran secara umum kepada pemakai mengenai sistem informasi yang baru. Adapun tujuan dari perancangan sistem yang diusulkan yaitu :

1. Memperbaiki sistem yang lama dengan yang baru.
2. Meningkatkan sumber daya manusia dan peralatan yang ada untuk mendapatkan hasil yang lebih baik.
3. Mengelola dan memberikan informasi yang cepat kepada yang membutuhkan.

4.2.2. Gambaran Umum Sistem yang Diusulkan

Perancangan Sistem Informasi Tugas Akhir Mahasiswa ini menggunakan bahasa pemodelan UML *(Unified Modeling Language).* Mulai dari pembuatan rancangan *Usecase* nya hingga *Deployment Diagramnya*.

Secara umum, proses tersebut dimulai dari penentuan arsitektur utama dari sistem yang ingin dirancang dan dibuat diagram alur proses pendaftaran sebagai *Admin* (*Administrator/*Operator) dan dilanjutkan alur proses pendaftaran TA / Skripsi *online* oleh mahasiswa sebagai *Actor*.

a. Rancangan Alur Akses User ke Server Web

Perancangan ini, di fungsikan sebagai model koneksi alur akses antar User ke server Web. Prosesnya dimulai dari akses *client* dengan melakukan browse ke Server web, yang kemudian *request* dari user/client tersebut dikirim ke Server *database*. Setelah itu Server menjawab/merespon segera permintaan dari Client oleh Server web tadi. Dan setelah menerima respon dari server *database* yang berisi konten/informasi yang terdapat dalam *database*, kemudian server web langsung menjawab/merespon request dari *user/client* tadi.

b. Rancang-bangun Sistem Informasi Pengolahan Data Tugas Akhir Mahasiswa

Untuk alur kerja sistem pengolahan data tugas akhir mahasiswa juga sama halnya seperti pada Rancangan alur kerja antara User ke Server, hanya saja pada rancang-bangun ini lebih detail, dan visualisasi terhadap siapa saja yang menggunakan sistem pengolahan data tugas akhir mahasiswa ini dan siapa saja yang berperan didalam menjalankan sistem tersebut. Pada prakteknya, *Actor* yang mengakses websitenya :

1. Dengan mengetikkan URL / alamat

2. Kemudian URL yang dialamatkan tadi, masuk ke website pengajuan proposal dan skripsi online yang terhubung dengan server, begitu juga dengan Admin, namun yang berbeda adalah Admin ini memilki akses bebas pada website dan mempunyai username serta password untuk mengakses sistem tersebut. Hal ini demi menjaga keamanan dari sistem yang di jalankan.

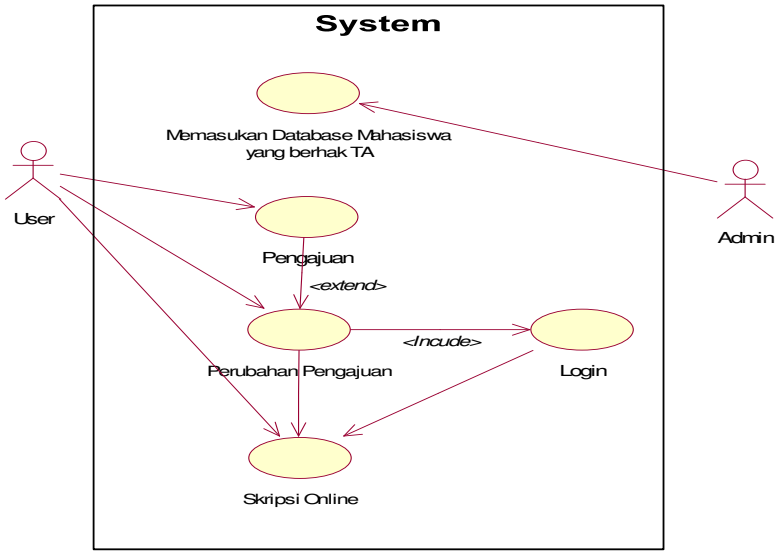
4.2.3 Perancangan Prosedur Yang Di Usulkan

Pada tahapan perancangan proses ini akan dibuat suatu *Use case, Diagram Sequence, Collaboration Diagram, Class Diagram* . Tahapan perancangan ini akan memberikan gambaran mengenai aliran informasi dan proses yang berjalan dalam sistem. Pada website yang dibangun yang yang bisa di akses atau yang bersifat *onlin*e adalah informasi TA dan informasi skripsi *online* berupa *output* sedangkan yang bersifat *offline* adalah data-data pengajuan proposal, dan perubahan pengajuan proposal, dan perubahan pengajuan proposal karena selain mahasiswaa mendaftar secara *online* mahasiswa juga tetap ke kampus untuk menyerahkan berkas atau data pengajuan proposal atau perubahan pengajuan proposal.

4.2.3.1 Use Case Sistem Perancangan Di Usulkan

Usecase diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan

bukan “bagaimana”. Sebuah *Usecase* mempresentasikan sebuah interaksi antara *Actor* dengan sistem. Berikut adalah perancangan *Usecase* nya:



Gambar 4.3 *Usecase* Usulan Perancangan

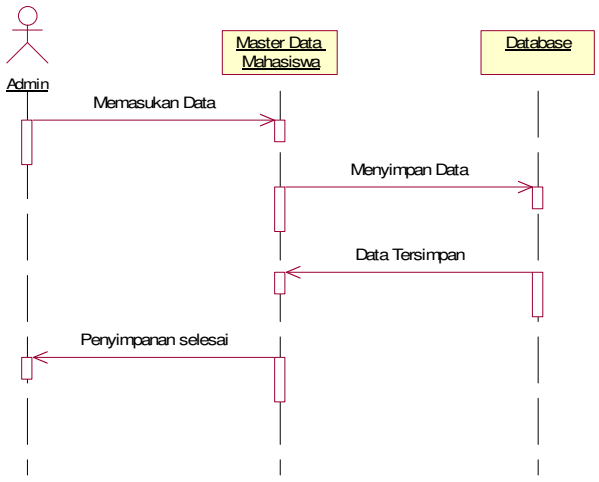
Model usecase diagram merupakan gambaran aktivitas sistem yang dilihat dari kebutuhan *actor.* Pada tahapan ini menggambarkan hubungan antara actor dengan sistem. Usecase diagram diatas menggambarkan dimana user mahasiswa sebelum memasuki page pendaftaran tugas akhir harus melakukan konfirmasi terlebih dahulu, untuk konfirmasi ini di butuhkan validasi berupa NIM user. Setelah itu user akan mengisi data-data yg tersedia beserta ID dan password yang akan di gunakan untuk masuk ke page - page pendaftaran tugas akhir, Lalu sistem akan masuk ke dalam halaman utama pendaftaran tugas akhir.

4.2.3.2 Sequence Diagram

Sequence diagram di gunakan untuk menggambarkan perilaku pada sebuah *skenario*. Diagram ini menunjukan sejumlah contoh obyek dan message yang di letakkan di antara obyek-obyek di dalam use case. Komponen utama sequence diagram terdiri dari obyek yang di tuliskan dengan kotak segi empat bernama. Message di wakili oleh garis dengan tanda panah dan waktu yang di tunjukkan dengan progress vertical.

4.2.3.2.1 Sequence Diagram Memasukan Data Mahasiswa yang Berhak Mengambil TA / Skripsi

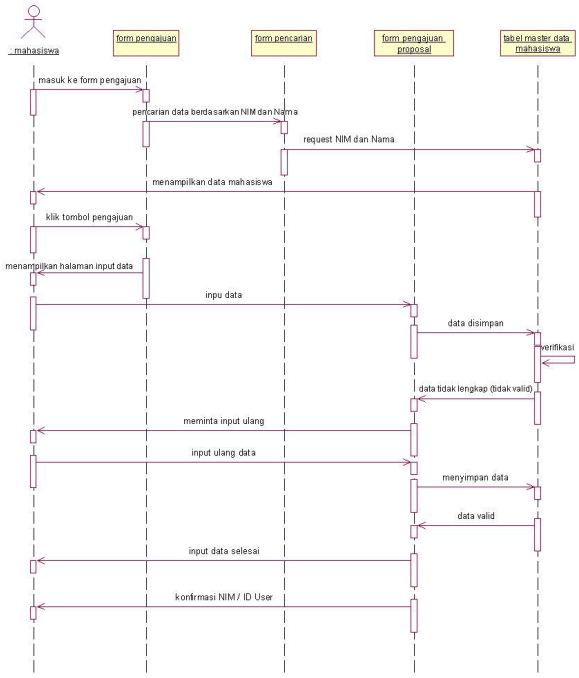
Proses memasukkan data mahasiswa yang berhak mengambil ta / skripsi ke dalam database dilakukan oleh admin, untuk lebih jelas dapat dilihat pada gambar sequence diagram di bawah ini :



Gambar 4.4 *Sequence Diagram* Diagram Memasukan Data Mahasiswa yang Berhak Mengambil TA / Skripsi

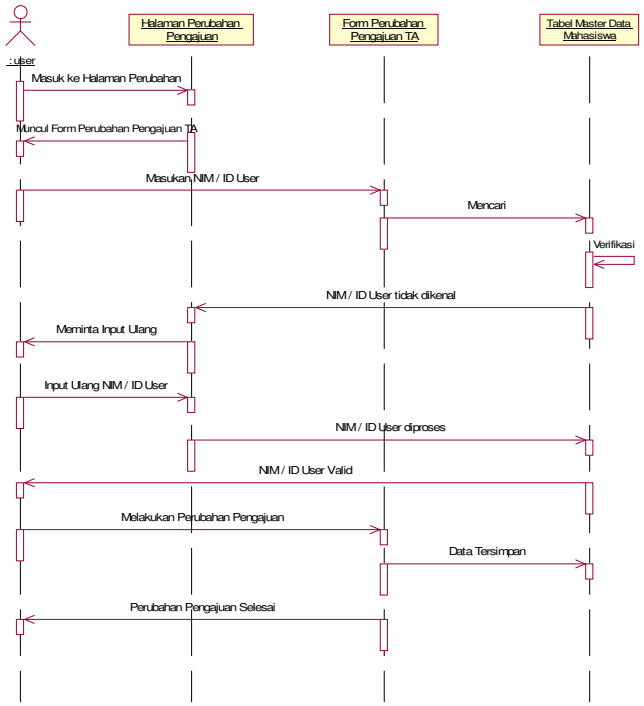
4.2.3.2.2 Sequence Pengajuan

Gambar di bawa ini merupakan *sequence diagram* pengajuan, yang didalamnya terjadi proses pengajuan berdasarkan data yang di ambil dari data master mahasiswa yang telah di saring dan di nyatakan layak untuk mengambil TA / Skripsi, setelah proses penyajuan di nyatakan berhasil maka mahasiswa Tugas Akhir akan memperoleh ID User / No. Registrasi yang akan di gunakan melakukan perubahan pengajuan dan untuk masuk ke proses skripsi online . Untuk lebih jelasnya dapat di lihat pada gambar berikut :



Gambar 4.5 *Sequence Diagram* Pengajuan

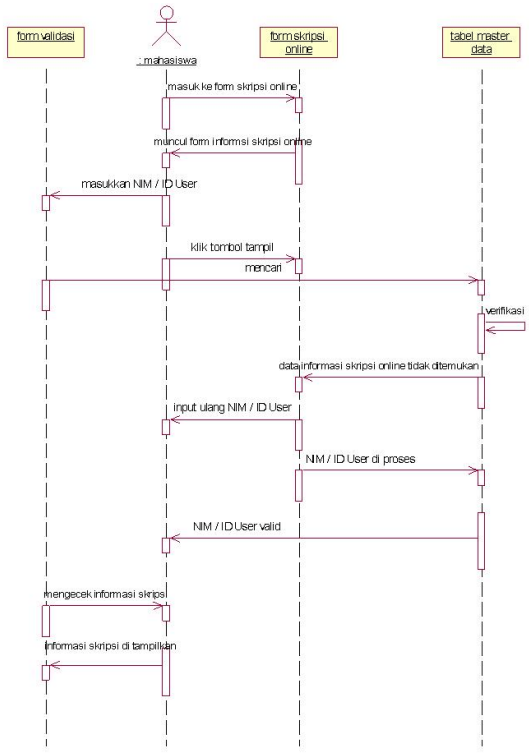
4.2.3.2.3 Sequence Perubahan Pengajuan TA



Gambar 4.6 *Sequence Diagram* Perubahan Pengajuan

Gambar di atas merupakan *sequence diagram* perubahan pengajuan, dalam hal ini adalah perubahan pengajuan TA / Skripsi yang harus menginput NIM dan ID User yang di dapat setelah proses pengajuan berhasil. Dan data ini secara otomatis akan tersimpan di database master mahasiswa yang mengambil TA / skripsi.

4.2.3.2.4 Sequence Skripsi Online



Gambar 4.7 *Sequence Diagram* Skripsi Online

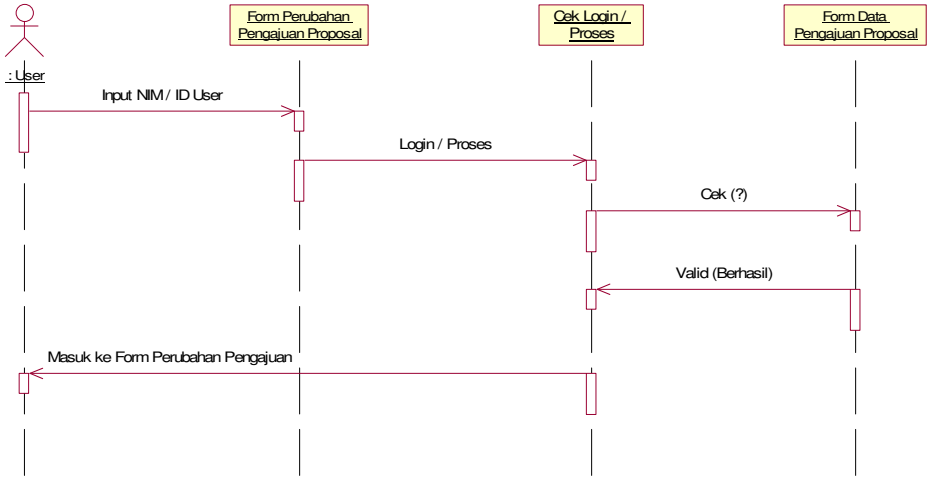
Gambar di atas merupakan *sequence diagram* skripsi online yang menjelaskan sebelum masuk ke dalam proses skripsi online harus menginput NIM dan ID User sebagai validasi, setelah itu user dapat melihat jadwal bimbingan, seminar dan sidang serta nilai akhir yang di peroleh .

4.2.3.2.5 Sequence Login Perubahan Pengajuan TA / Skripsi

Gambar di bawah ini adalah sequence diagram login, yang terdiri dari logi admin dan login user. Login admin Prosesnya dimulai dengan admin masuk kedalam *form Login* terlebih dahulu, kemudian

selanjutnya admin dapat melakukan login dengan memasukkan *Username* dan *Password* untuk dapat masuk kedalam sistem. Proses *Login* itu sendiri diikuti juga validasi yang dilakukan oleh sistem dimana terdapat proses verifikasi *Username* dan *Password* apakah sesuai atau tidak. Jika berhasil *Admin* tersebut dapat masuk dan menggunakan sistem. Sedangkan login user di lakukan pada saat proses pengajuan selesai yang kemudian user akan memperoleh NIM dan ID User yang di gunakan untuk masuk proses perubahan pengajuan dan skripsi online.

Untuk lebih jelasnya dapat di lihat pada gambar berikut :

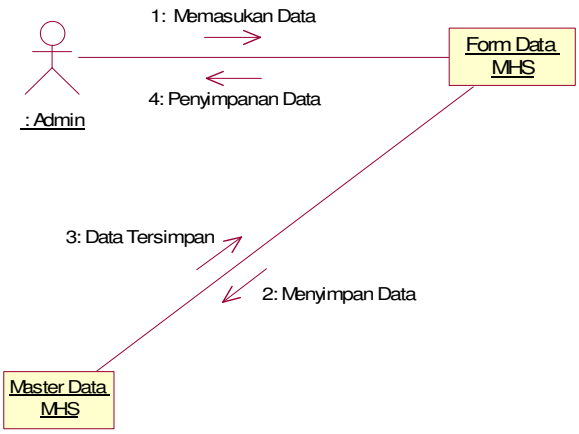


Gambar 4.8 *Sequence Diagram* Login Perubahan Pengajuan TA / Skripsi

4.2.3.3 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

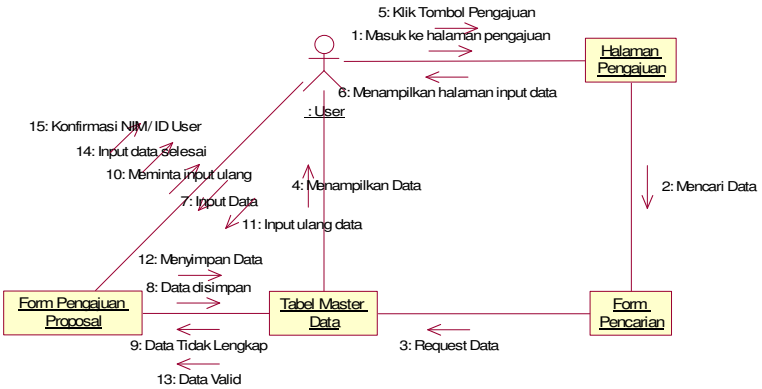
4.2.3.3.1 Collaboration Diagram Memasukan Data Mahasiswa yang Berhak Mengambil TA / Skripsi



Gambar 4.9 *Collaboration Diagram* Memasukan Data Mahasiswa yang Berhak Mengambil TA / Skripsi

Dari gambar diatas dapat di jelaskan, admin memasukkan data mahasiswa ke dalam master data.

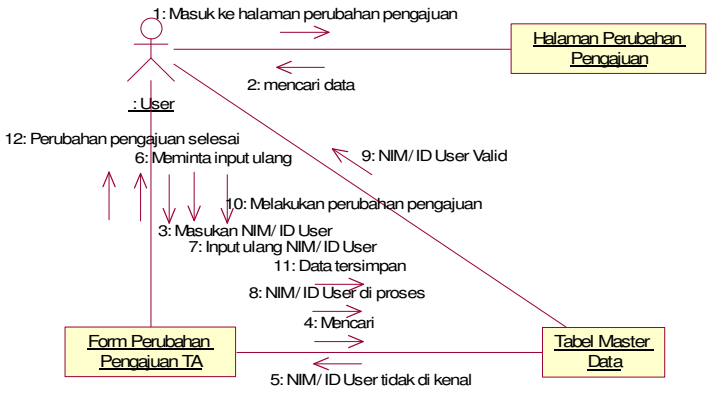
4.2.3.3.2 Collaboration Diagram Pengajuan



Gambar 4.10 *Collaboration Diagram* Pengajuan

Pada gambar collaboration diagram pengajuan di atas menjelaskan bahwa sebelum melakukan pengajuan user harus masuk ke halaman pengajuan dan melakukan proses pencarian data pada table master data mahasiswa jika ada datanya maka proses bisa di lanjutkan.

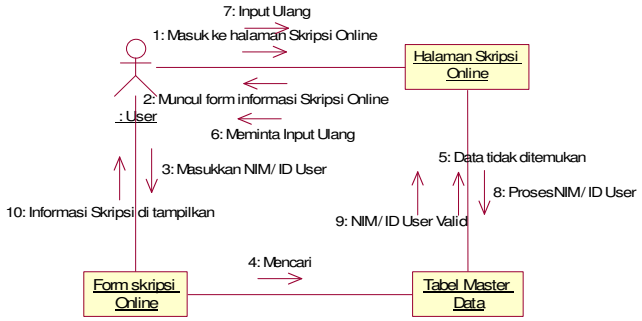
4.2.3.3.3 Collaboration Diagram Perubahan Pengajuan TA



Gambar 4.11 *Collaboration Diagram* Perubahan Pengajuan TA

Untuk melakukan perubahan pengajuan TA user wajid melewati tahap validasi yaitu dengan memasukan NIM dan ID User terlebih dahulu.

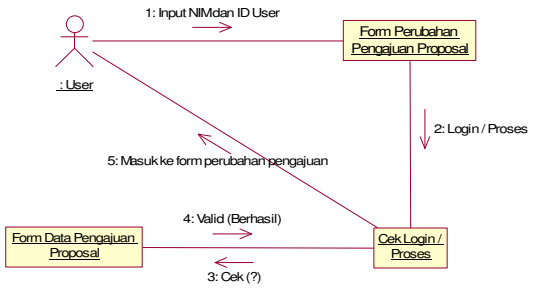
4.2.3.3.4 Collaboration Diagram Skripsi Online



Gambar 4.12 *Collaboration Diagram* skripsi Online

Sebelum masuk ke halaman skripsi online user wajid melewati tahap validasi yaitu dengan memasukan NIM dan ID User terlebih dahulu.

4.2.3.3.5 Collaboration Diagram Login Perubahan Pengajuan Proposal

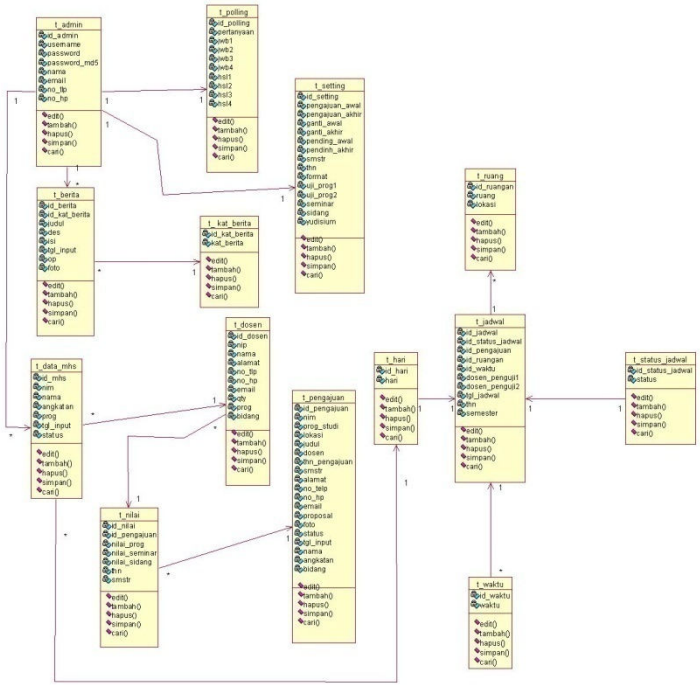


Gambar 4.13 *Collaboration Diagram* Login Perubahan Pengajuan Proposal

Pada gambar *collaboration diagram* di atas di jelaskan bahwa sebelum user melakukan proses login, maka user harus menginputkan NIM dan *ID User* terlebih dahulu untuk masuk kedalam sistem.

4.2.3.4 Class Diagram

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.



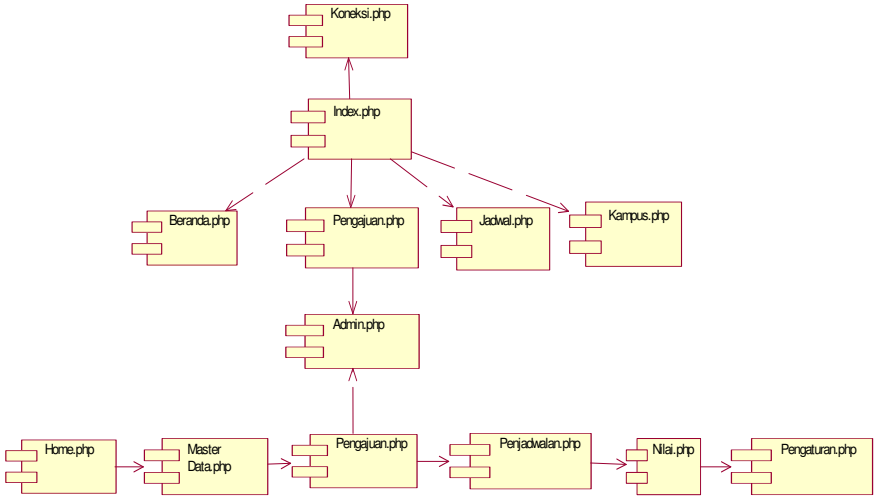
Gambar 4.14 *Class Diagram* Yg Diusulkan

4.2.3.1 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time.

Umumnya komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga

berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.



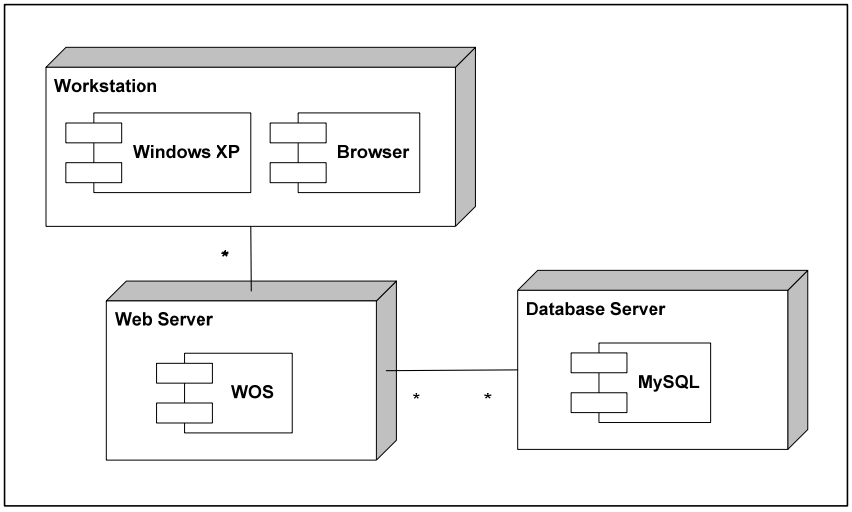
Gambar 4.15 *Component Diagram*

4.2.3.1 Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.

Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya.

Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.



Gambar 4.16 Deployment Diagram

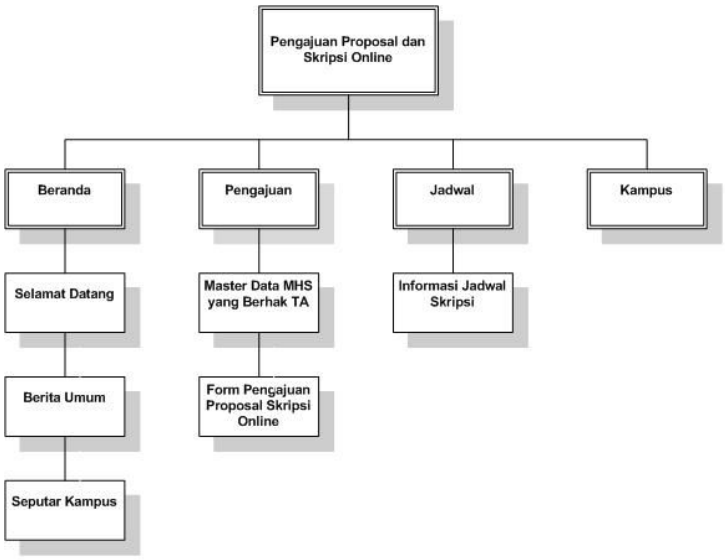
4.2.4 Perancangan Antar Muka

4.2.4.1 Struktur Menu

Perancangan menu di gunakan untuk memudahkan dalam penelusuran program yang di buat. Struktur menu di bagi menjadi dua yaitu struktur menu admin dan struktur menu user, berikut adalah gambar perancangan nya :

1) Struktur Menu Site

Perancangan menu di gunakan untuk memudahkan dalam penelusuran program yang di buat. Struktur menu di bagi menjadi dua yaitu struktur menu admin dan struktur menu user, berikut adalah gambar perancangan nya :



Gambar 4.17 Struktur Menu

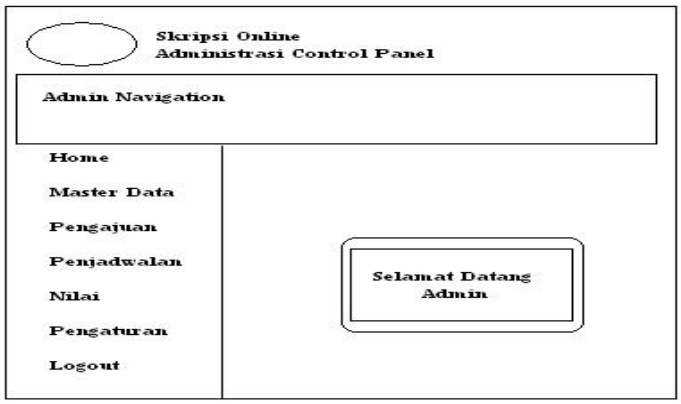
Berikut adalah gambar perancangan struktur menu interface user dan admin :

2) Perancangan Struktur Menu User



Gambar 4.18 Struktur Menu User

3) Struktur Menu Admin

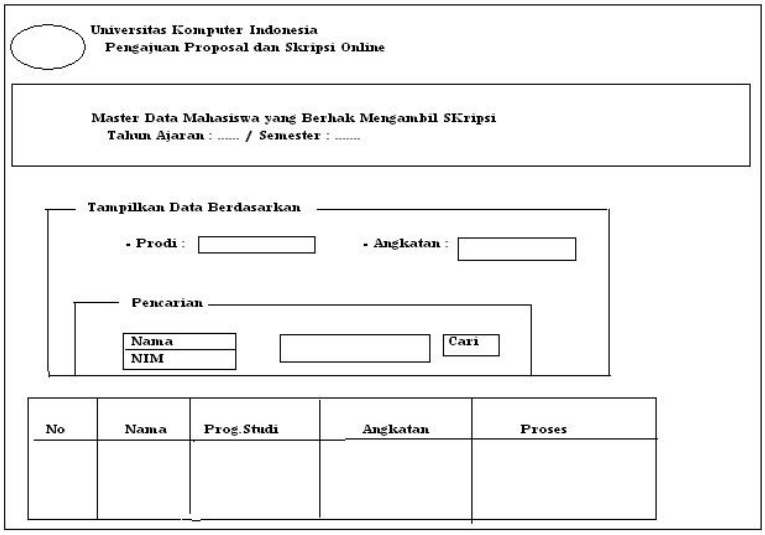


Gambar 4.19 Struktur Menu Admin

4.2.4.2 Perancangan Input

Perancangan input merupakan gambaran interface atau antarmuka tempat memasukan data-data kedalam sistem. Berikut ini form-form utama untuk input data :

1. **Form Master Data Mahasiswa yang Berhak TA**



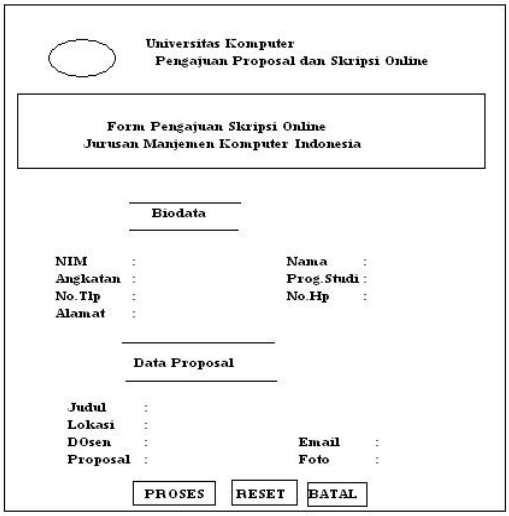
Gambar 4.20 Form Master Data Mahasiswa yang Berhak TA

2. Login



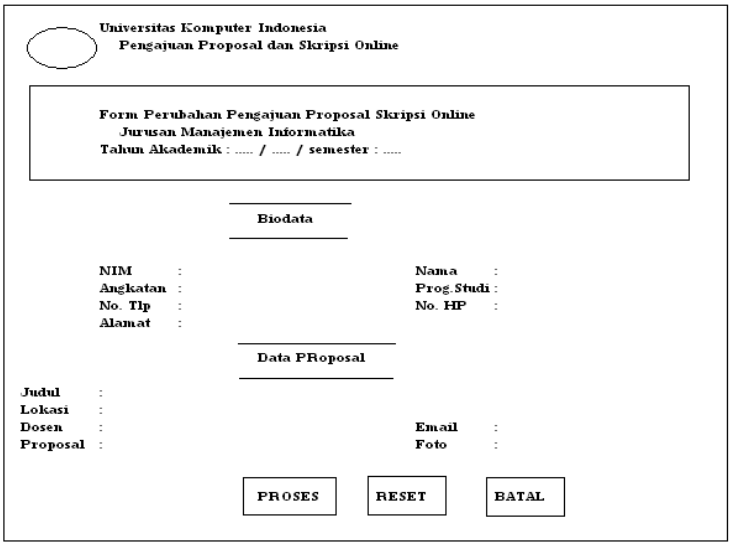
Gambar 4.21 Login

3. Form Pengajuan Proposal



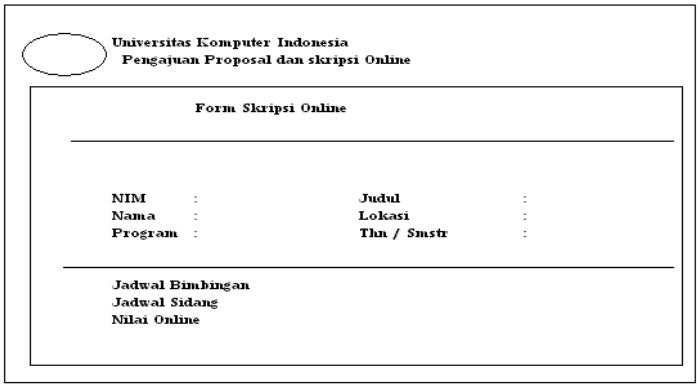
Gambar 4.22 Form Pengajuan Proposal

4. Form Perubahan Pengajuan Proposal



Gambar 4.23 Form Perubahan Pengajuan Proposal

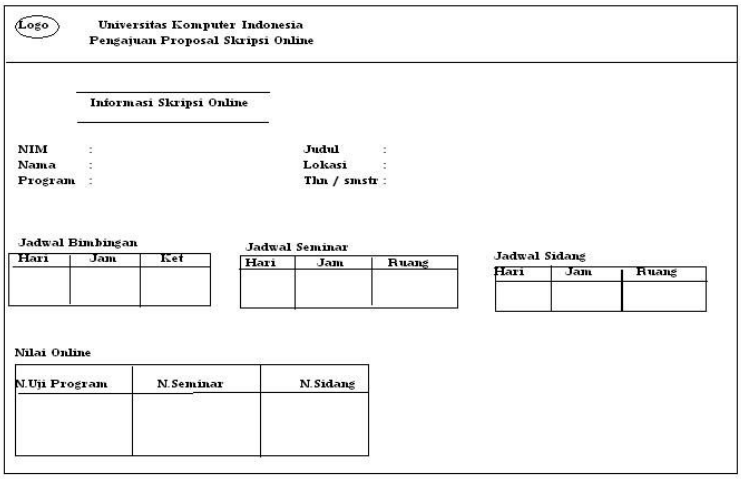
5. Skripsi Online



Gambar 4.24 Form Skripsi Online

4.2.4.3 Perancangan Output

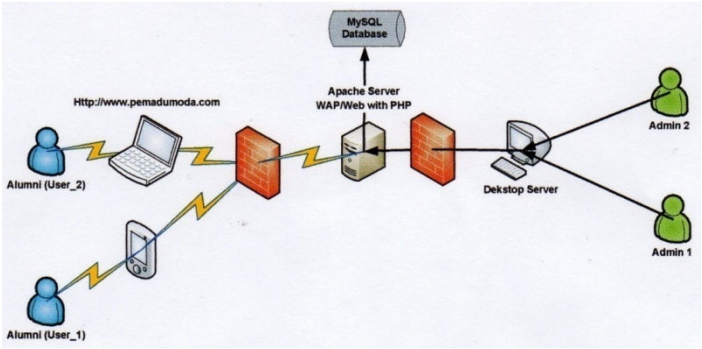
Perancangan output ini merupakan rancangan informasi yang dihasilkan dan tetapi tidak dapat dicetak sebagai output sistem.



Gambar 4.25 Form Perancangan Output

4.1.1 Perancangang Arsitektur yang Diusulkan

Perancangan arsitektur jaringan adalah pola keterhubungan antara PC (*Personal Computer*) sebagai *Client* dengan *Server* yang juga terkoneksi ke ISP (*Internet Service Provider*).



Gambar 4.26 Perancangan Arsitektur Jaringan

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. Implementasi

Implementasi merupakan bagian dimana hasil dari setiap tahapan analisis dan perancangan akan diterapkan dalam sebuah program. Berikut ini akan dipaparkan mengenai, implementasi perangkat lunak, implementasi perangkat keras, implementasi basis data, implementasi antar muka, implementasi program dan penggunaan program.

5.1.1. Implementasi Perangkat Lunak

Berikut ini adalah implementasi perangkat lunak yang digunakan untuk mendukung pembuatan program aplikasi ini, sebagai berikut :

a) Sistem Operasi yang digunakan adalah Microsoft Windows Xp
b) Database server yang digunakan adalah WampServer
c) Perangkat lunak sebagai Bahasa pemrograman yang digunakan adalah PHP.
d) Design interpress menggunakan Macromedia Dreamweaver 8
e) Browser yang digunakan Internet Explorer, Mozilla Firefox

5.1.2. Implementasi Perangkat Keras

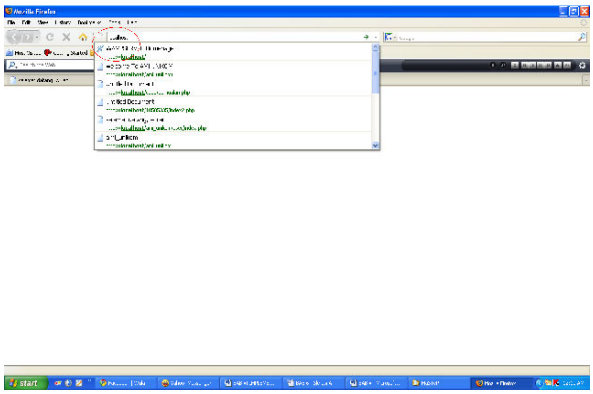
Pada Implementasi perangkat keras, kebutuhan perangkat keras minimum yang dibutuhkan adalah, sebagai berikut :

1. Server
    a) Processor Intel Pentium IV atau setara
    b) Harddisk 80 GB.
    c) RAM 1 GB.
    d) *VGA* 128 MB.
    e) *Mouse* , *Keyboard* dan *monitor*
2. *Client*
    a) Processor Intel Pentium IV atau setara
    b) Harddisk 80 GB.
    c) RAM 1 GB.
    d) *VGA* 128 MB.
    e) *Mouse* , *Keyboard* dan *monitor*
    f) Kabel jaringan

5.1.3. Implementasi Basis Data

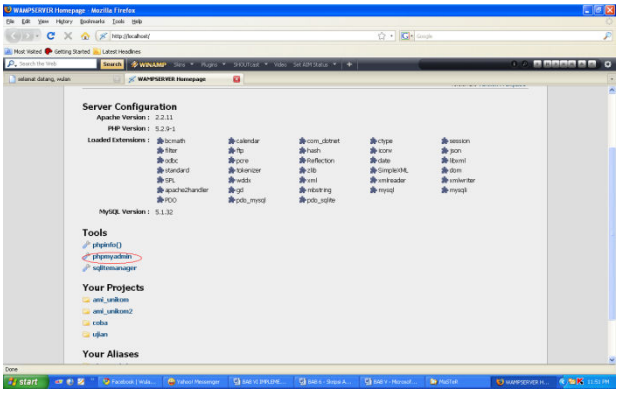
Pembuatan basis data dilakukan dengan menggunakan fasilitas WampServer2.0g-1. Untuk membuat database nya, silahkan ikuti langkah – langkah berikut :

1. Buka lah web browser anda lalu ketik di addressnya ”http ://localhost/”



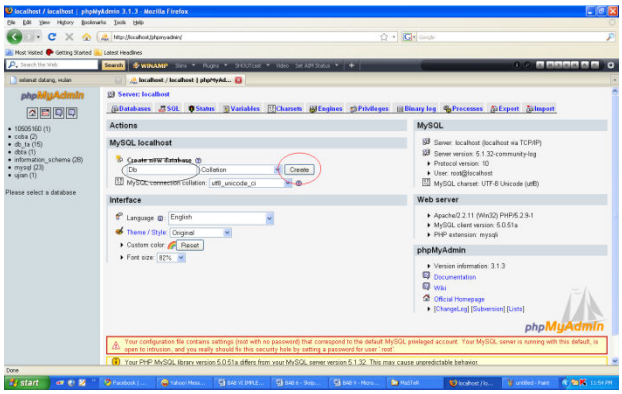
Gambar 5.1 Implementasi Basis Data (a)

2. Pilih phpmyadmin



Gambar 5.2 Implementasi Basis Data (b)

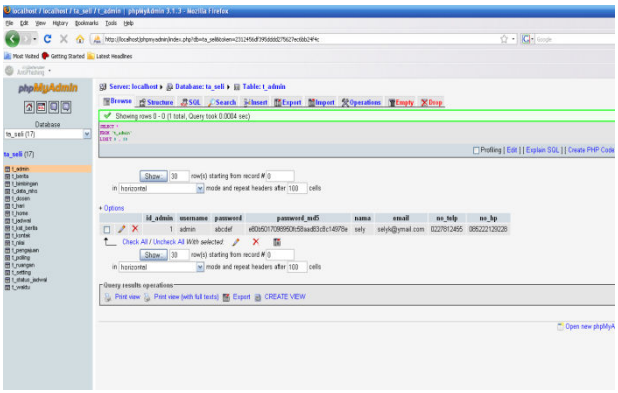
3. Lalu tulis nama database di create database and then klik ”create”



Gambar 5.3 Implementasi Basis Data (c)

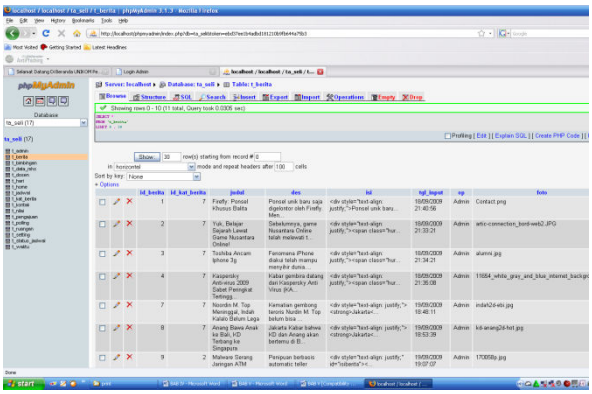
Adapun hasil dari implementasi basis data adalah sebagai berikut :

1. Tabel Admin



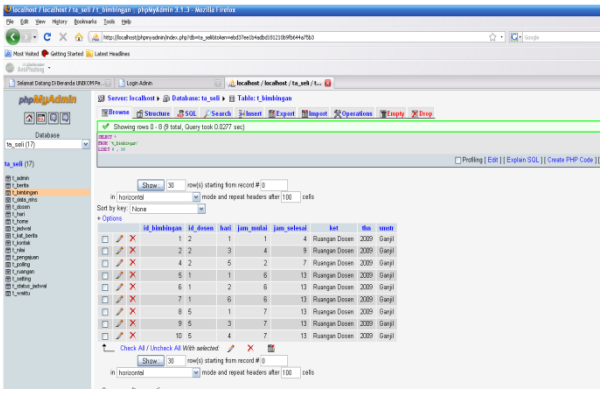
Gambar 5.1 Implementasi Basis Data Tabel Admin

2. Tabel Berita



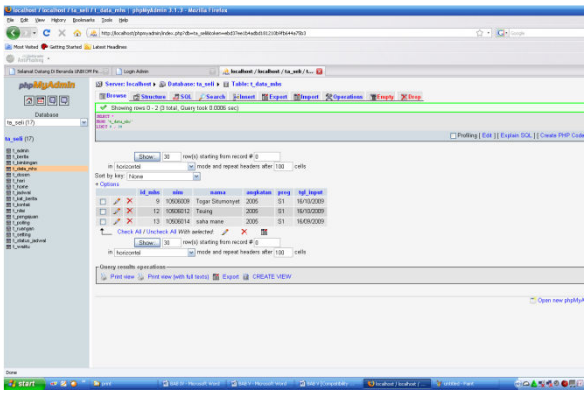
Gambar 5.2 Implementasi Basis Data Tabel Berita

3. Tabel Bimbingan



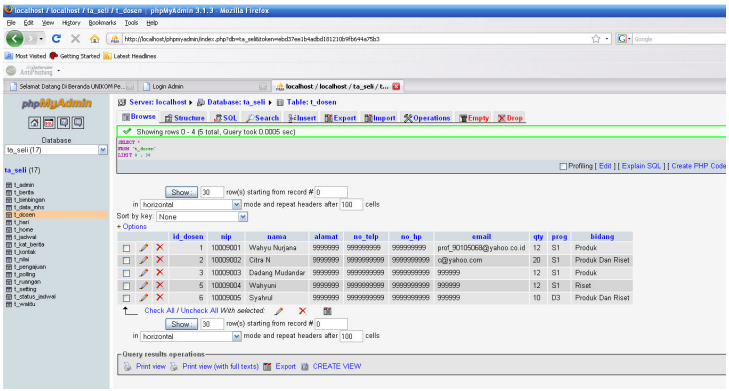
Gambar 5.6 Implementasi Basis Data Tabel Bimbingan

4. Tabel Data Mahasiswa



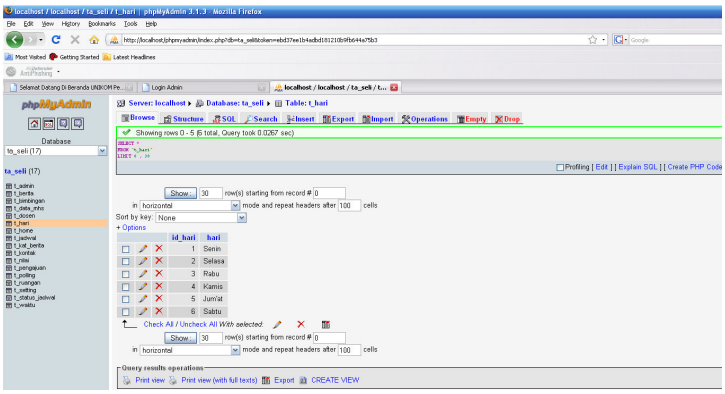
Gambar 5.3 Implementasi Basis Data Tabel Data Mahasiswa

5. Tabel Dosen



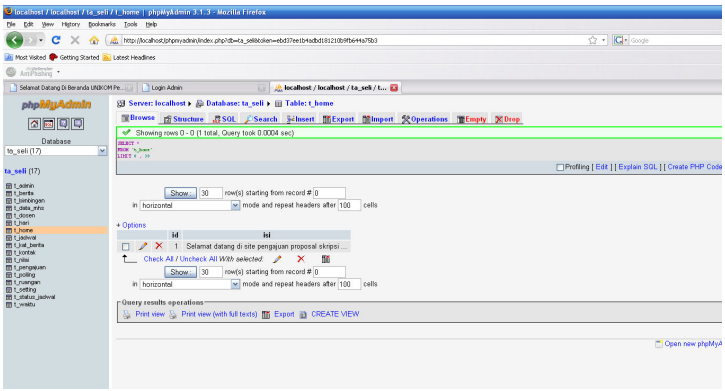
Gambar 5.4 Implementasi Basis Data Tabel Dosen

6. Tabel Hari



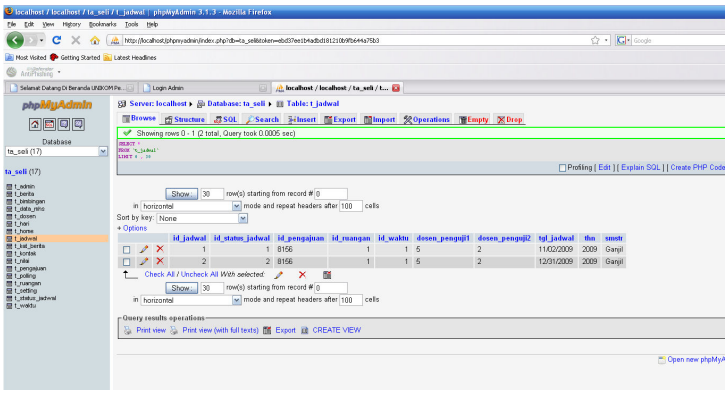
Gambar 5.5 Implementasi Basis Data Tabel Hari

7. Tabel Home



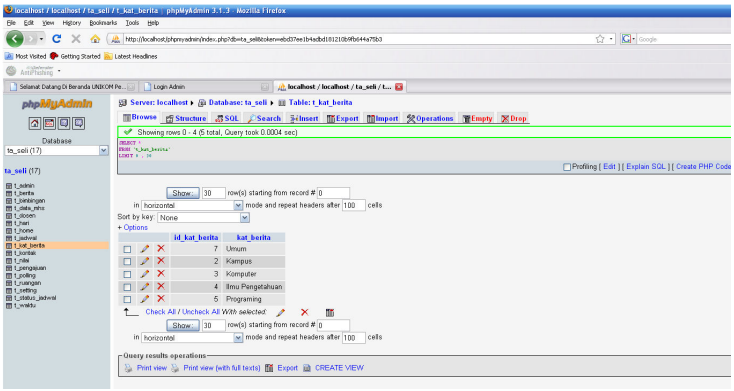
Gambar 5.6 Implementasi Basis Data Tabel Home

8. Tabel Jadwal



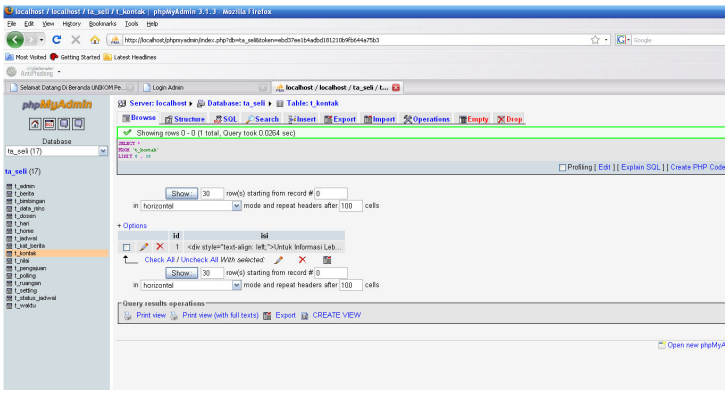
Gambar 5.7 Implementasi Basis Data Tabel Jadwal

9. Tabel Kategori Berita



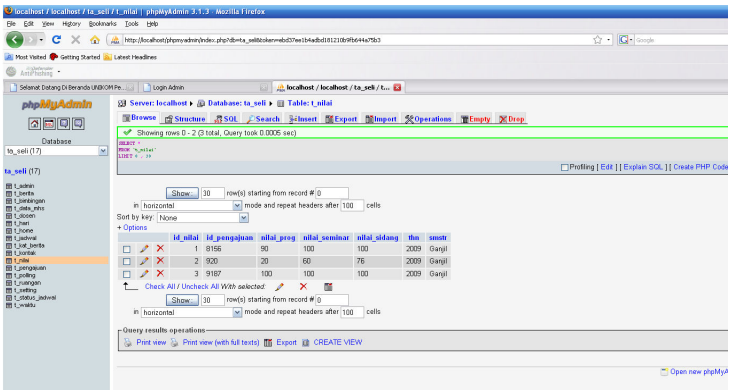
Gambar 5.8 Implementasi Basis Data Tabel Berita

10. Tabel Kontak



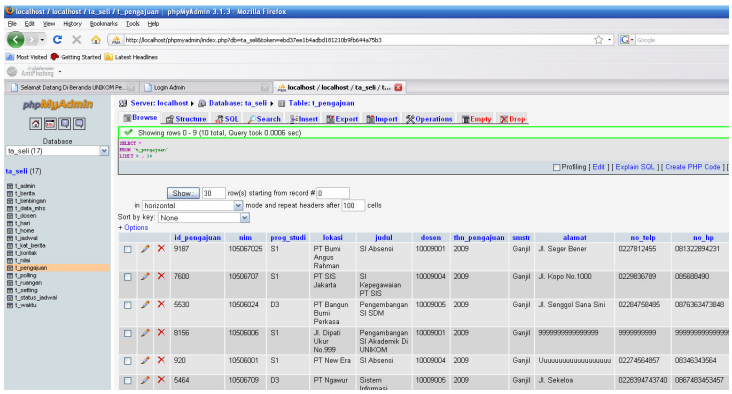
Gambar 5.9 Implementasi Basis Data Tabel Kontak

11. Tabel Nilai



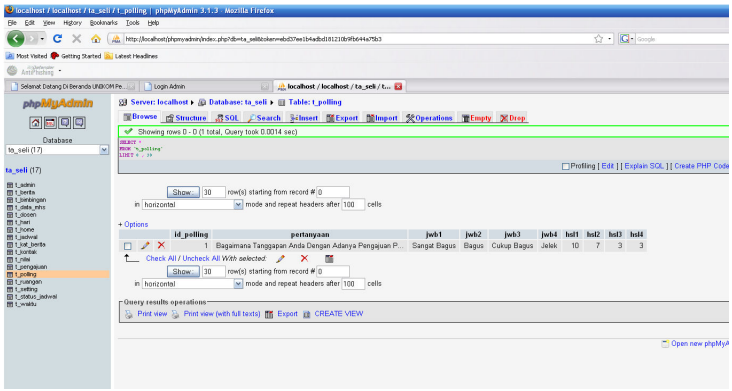
Gambar 5.10 Implementasi Basis Data Tabel Nilai

12. Tabel Pengajuan



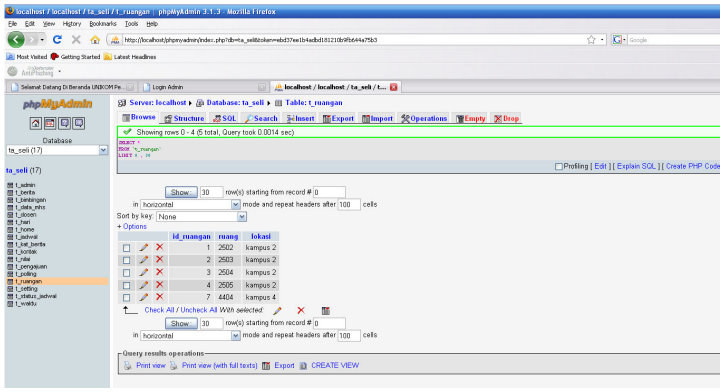
Gambar 5.11 Implementasi Basis Data Tabel Pengajuan

13. Tabel Polling



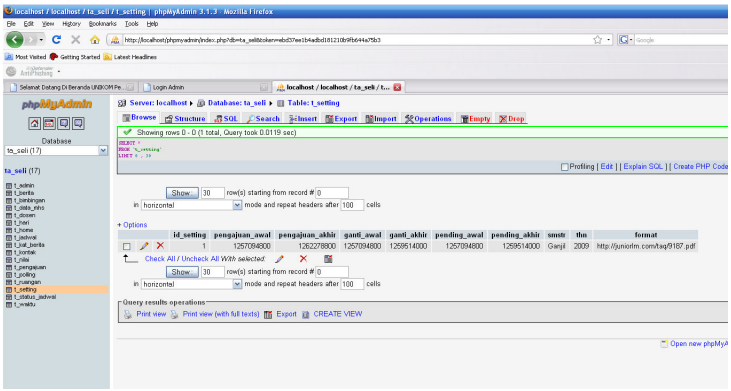
Gambar 5.12 Implementasi Basis Data Tabel Polling

14. Tabel Ruangan



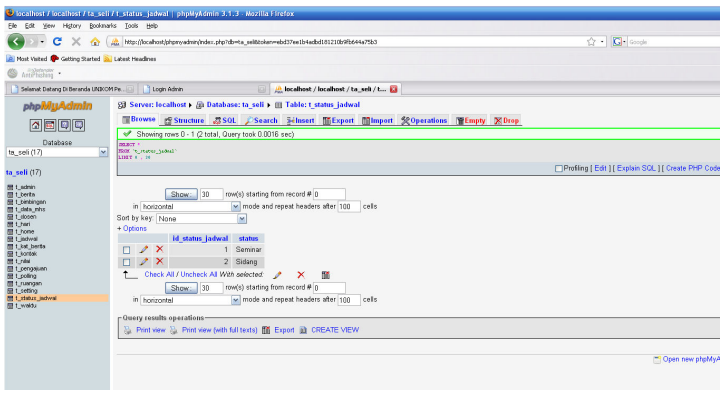
Gambar 5.13 Implementasi Basis Data Tabel Ruangan

15. Tabel Setting



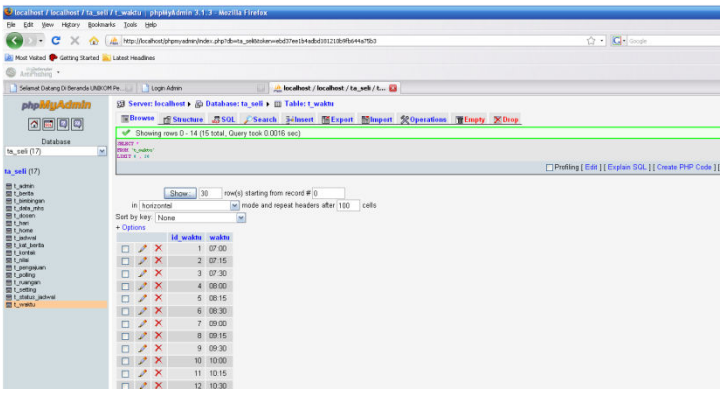
Gambar 5.14 Implementasi Basis Data Tabel Setting

16. Tabel Status Jadwal



Gambar 5.15 Implementasi Basis Data Tabel Status Jadwal

17. Tabel Waktu



Gambar 5.16 Implementasi Basis Data Tabel Waktu

5.1.4 Implementasi Antar Muka

Berikut ini akan dijelaskan tampilan antarmuka (*interface*) pada aplikasi yang dirancang dari setiap halaman yang dibuat dan dibedakan berdasarkan user dan admin.

5.1.4.1 Implementasi Halaman Utama

Implementasi halaman yang terdapat pada web yang dibuat bentuk file program dengan berekstensi PHP dapat dilihat, sebagai berikut :

Tabel 5.17 Implementasi Halaman Utama

MENU	DESKRIPSI	NAMA FILE
Beranda	File program untuk menangani halaman utama.	index.php
Pengajuan	File program yang berisi halaman pengajuan proposal skripsi	pengajuan.php
Jadwal	File program yang berisi informasi jadwal pengajuan skripsi	jadwal.php
Kampus	File program untuk menuju ke website kampus dan website mahasiswa	kampus.php

5.1.4.2 Implementasi Halaman User

Implementasi halaman yang terdapat pada agen yang dibuat bentuk file program dengan berekstensi PHP dapat dilihat, sebagai berikut :

Tabel 5.18 Implementasi Halaman User

MENU	DESKRIPSI	NAMA FILE
Beranda	File program untuk menangani halaman utama.	index.php
Pengajuan	File program yang berisi nama	pengajuan.php

	mahasiswa yang berhak mengambil skripsi dan halaman pengajuan proposal skripsi	
Perubahan Pengajuan	File program yang berisi perubahan pengajuan proposal TA / Skripsi	perubahan.php
Jadwal	File program yang berisi informasi jadwal pengajuan skripsi	jadwal.php
Kampus	File program untuk menuju ke website kampus dan website mahasiswa	kampus.php
Beranda	File program untuk menangani halaman utama.	index.php
Kontak	File program yang menampilkan informasi kontak skripsi online	kontak.php

5.1.4.3 Implementasi Halaman Admin

Implementasi halaman yang terdapat pada admin yang dibuat bentuk file program dengan berekstensi PHP dapat dilihat, sebagai berikut :

Tabel 5.19 Implementasi Halaman Admin

MENU	DESKRIPSI	NAMA FILE
Login	File program untuk login.	login.php
Home	Halaman Utama Admin	index.php

Master Data	Halaman Master data	master_data.php
Pengajuan	File program yang berisi halaman pengajuan proposal skripsi	pengajuan.php
Penjadwalan	File program yang berisi informasi jadwal pengajuan skripsi	jadwal.php
Nilai	File program yang berisi nilai mahasiswa	nilai.php
Pengaturan	File program untuk mengatur waktu pengajuan, akun, kategori berita, dan ruangan	pengaturan.php
Logout	File program untuk keluar dari halaman agen.	logout.php

5.1.5 Implementasi Instalasi Program

Implementasi dari prototipe perancangan Sistem Informasi Pengolahan Data Tugas Akhir Mahasiswa ini dilakukan menggunakan bahasa pemograman PHP, dengan basisdata yang digunakan adalah WampServer. Aplikasi PHP tersebut dijalankan pada berbagai *platform* sistem operasi dan perangkat keras, tetapi implementasi dan pengujian sepenuhnya hanya dilakukan diperangkat keras PC (*Perrsonal Computer*) dengan sistem operasi Microsoft Windows XP SP2.

5.1.6 Penggunaan Program

Berikut ini adalah langkah-langkah penggunaan program pada Sistem Informasi Pengolahan Data Tugas Akhir Mahasiswa.

A. Halaman *User* (Mahasiswa)

1. Form Utama

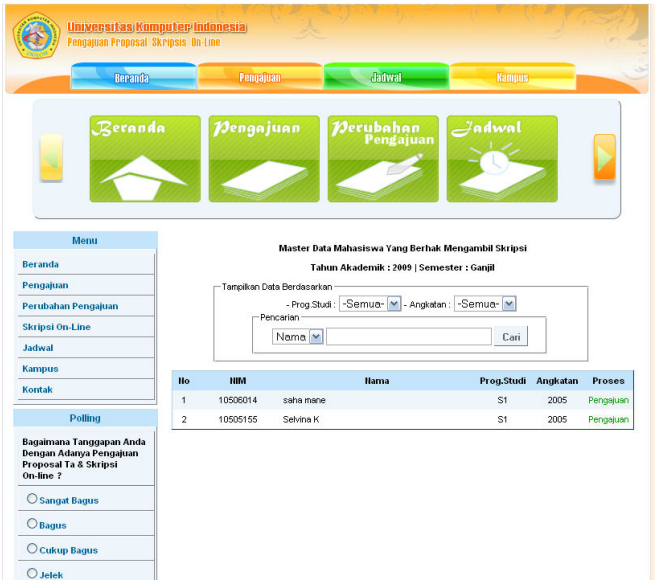
Form ini muncul sebagai tampilan utama website.



Gambar 5.4 Halaman Utama

2. Pengajuan

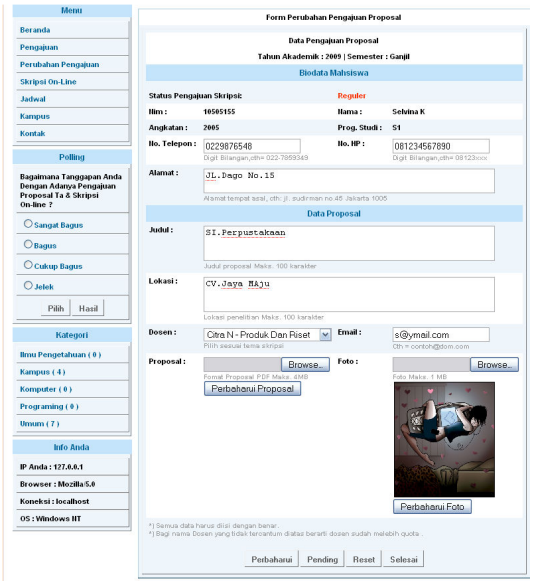
Di halaman ini mahasiswa yang telah memenuhi syarat TA berhak mengajukan proposal penelitian .



Gambar 5.5 Pengajuan

3. Perubahan Pengajuan

Pada halaman ini mahasiswa dapat melakukan perubahan pengajuan dengan validasi berupa ID user dan NIM.



Gambar 5.6 Perubahan Pengajuan

4. Skripsi Online

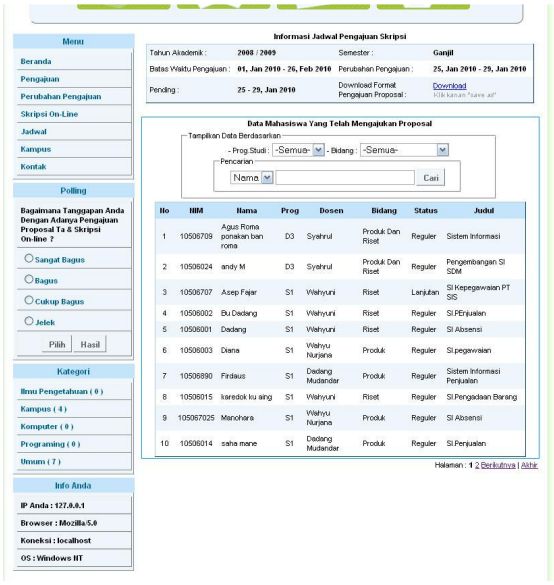
Merupakan output atau informasi skripsi online yang menampilkan informasi online,sebelum masuk ke form skripsi online harus melakukan validasi berupa ID User dan NIM.



Gambar 5.7 Skripsi Online

5. Jadwal

Form ini merupakan form yang menampilkan informasi jadwal pengajuan skripsi / TA berisi NIM, nama, prodi, dosen pembimbing, bidang, status dan judul.

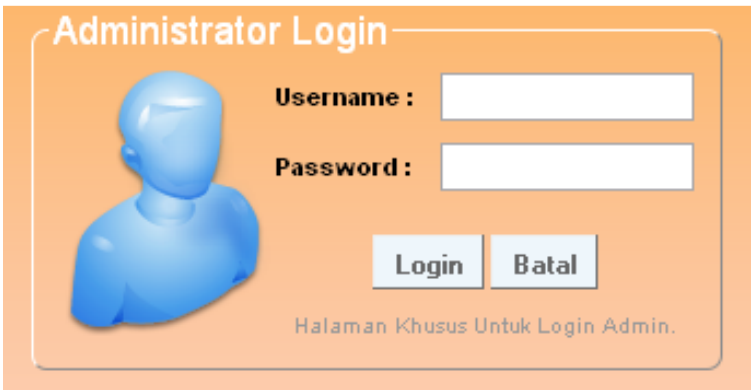


Gambar 5.8 Jadwal

B. Form Admin

1. Form Login Admin

Untuk melakukan login, Isi data login dan klik tombol login. Jika berhasil maka Form utama akan memunculkan menu bar sesuai hak akses yang telah ditentukan.



Gambar 5.9 Form Login Admin

2. Form Utama Admin

   Merupakan halaman depan admin



Gambar 5.10 Form Utama Admin

3. Form Master Data Mahasiswa

   Merupakan halaman yang berisi daftar mahasiswa/i yang berhak mengambil Skripsi / TA yang sudah mendapat persetujuan dari dosen wali.



Gambar 5.11 Form Master Data Mahasiswa

4. Form Master Dosen

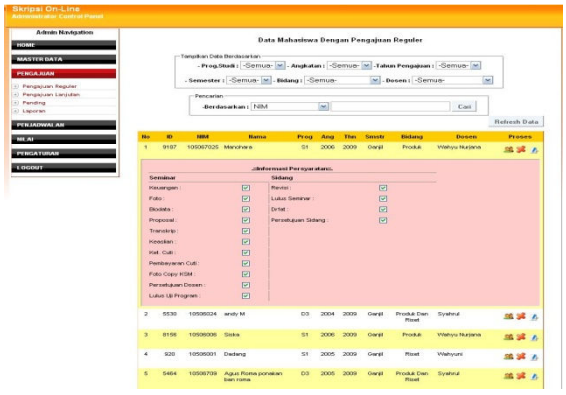
   Form master dosen ini berisi master data dosen yang menjdi dosen pembimbing skripsi / TA.



Gambar 5.12 Form Master Mahasiswa

5. Form Pengajuan

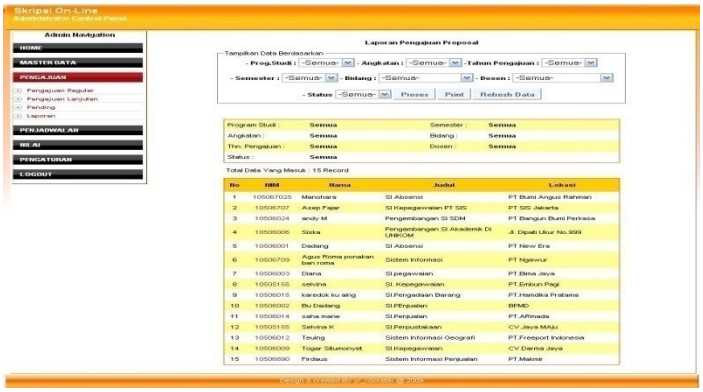
   Pada form ini berisi data mahasiswa dengan pengajuan reguler, lanjutan, pending.



Gambar 5.13 Form Pengajuan

6. Form Laporan Pengajuan

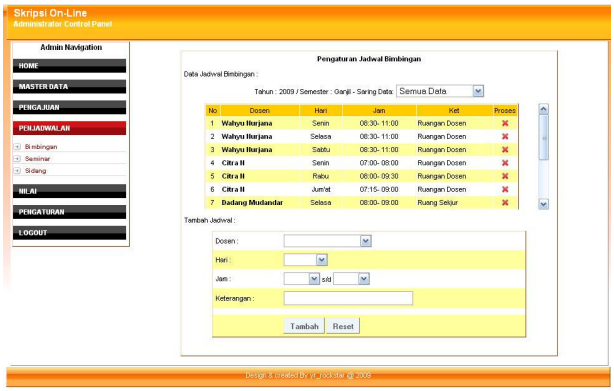
   Pada form ini berisi semua laporan mahasiswa yang sudah melakukan pengajuan proposal



Gambar 5.14 Form Laporan Pengajuan

7. Form Penjadwalan

   Merupakan form yang berisi pengaturan jadwal pembimbing, jadwal seminar dan jadwal sidang.



Gambar 5.15 Form Penjadwalan

8. Form Nilai

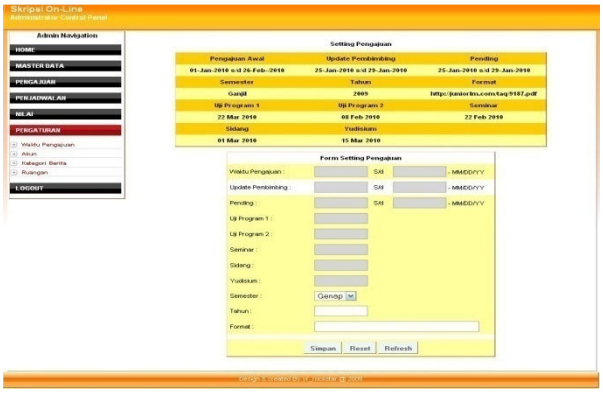
   Pada form ini berisi pengaturan nilai yang terdiri dari nilai uji program 20 %, seminar 30% an sidang 50 %.



Gambar 5.16 Form Nilai

9. Form Pengaturan

   Merupakan form yang berisi settingan waktu pengajuan, akun, kategori berita, ruangan.



Gambar 5.17 Form Pengaturan

10. Form LogOut

Merupakan form untuk keluar dari menu admin dengan cara klik logout kemudian tekan enter.



Gambar 5.18 Form Logout

5.2 Pengujian

Pada pengujian perangkat lunak ini, menggunakan metode pengujian *black box.* Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini di gunakan untuk mengetahui apakan perangkat lunak berfungsi dengan benar.

5.2.1 Rencana Pengujian

Pengujian perangkat lunak sistem informasi pengolahan data tugas akhir mahasiswa ini menggunakan data uji input dari pengguna perangkat lunak yang telah di buat .

Adapun hal-hal yang akan diuji melalui teknik pengujian *Black Box* adalah sebagai berikut :

Tabel 5.20 Rencana Pengujian

Requirement Yang Di Uji	Butir Uji
Pengajuan	Melakukan pengajuan proposal
Perubahan Pengajuan	Melakukan perubahan pengajuan proposal
Login Form Perubahan Pengajuan Proposal	Pengecekan login untuk melakukan perubahan pengajuan proposal
Login Informasi Skripsi Online	Pengecekan login untuk melihat informasi skripsi online
Login Admin	Melakukan Login Admin

5.2.2 Kasus dan Hasil Pengujian Program

Pengujian program ini di lakukan dengan melakukan validasi data – data yang akan dimasukkan kedalam sistem. Dengan adanya pengujian ini maka data yang di masukkan akan sesuai kondisi yang telah di buat. Berikut adalah beberapa pengujian yang telah di lakukan, yaitu :

Tabel 5.21 Implementasi Pengujian Antar Muka Halaman Admin dan Halaman Utama

Requirement	**Skenario Uji**	**Hasil yang Diharapkan**	**Hasil Pengujian**
Pengajuan	11. Input biodata mahasiswa, data proposal (Data Benar)	Muncul Pesan ”Trimakasih Atas Kerjasama Anda”	Sesuai
	12. Input biodata mahasiswa, data proposal (Data Salah)	Muncul Pesan ”Maaf Data yang di Input Belum Valid”	Sesuai
Perubahan Pengajuan	1. Input biodata mahasiswa, data proposal (Data Benar)	Muncul pesan ”Data telah di perbaharui”	Sesuai
	2 Input biodata mahasiswa, data	Muncul pesan	Sesuai

	proposal (Data Salah)	”data yang Anda input belum valid atau belum lengkap”	
Login Form Perubahan Pengajuan Proposal	1. Input NIM dan ID User / No. Registrasi (Data Benar)	Masuk ke form perubahan pengajuan proposal	Sesuai
	2. Input NIM dan ID User / No. Registrasi (Data Salah)	Muncul pesan ”Maaf NIM atau ID User / No.Registrasi belum valid”	Sesuai
Login Informasi Skripsi Online	1. Input NIM dan ID User / No. Registrasi (Data Benar)	Masuk ke form informasi skripsi online	Sesuai
	2. Input NIM dan ID User / No. Registrasi	Muncul pesan ”Maaf NIM	Sesuai

	(Data Salah)	atau ID User / No.Registrasi belum valid”	
Login Admin	1. Input Username dan Password (Data Benar)	Maka muncul halaman utam admin dan pesan ”Selamat Datang Admin”	Sesuai
	2. Input Username dan Password (Data Salah)	Muncul pesan ”Username atau Password tidak dikenali”	Sesuai

5.2.3 Kesimpulan Dan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji *sample* di atas dapat ditarik kesimpulan bahwa perangkat lunak secara fungsional mengeluarkan hasil sesuai dengan yang diharapkan.

BAB VI

KESIMPULAN DAN SARAN

Berdasarkan hasil pengembangan sistem yang penulis lakukan, maka penulis mencoba melakukan kesimpulan, dan mengajukan beberapa saran-saran yang berhubungan dengan pembahasan yang telah dikemukakan dibab-bab sebelumnya.

6.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari pembahasan dan penerapan sistem terhadap permasalahan Sistem Informasi Tugas Akhir Mahasiswa adalah sebagai berikut :

1. Dengan adanya aplikasi Pengajuan Proposal dan Skripsi Online ini dapat mempermudah mahasiswa/i yang mengambil TA / Skripsi dalam pengajuan Proposal dan skripsi yang bersifat online.
2. Membantu SEKJUR dalam proses pengolahan data tugas akhir mahasiswa.
3. Berdasarkan hasil penelitian terhadap *user*, aplikasi Pengajuan Proposal dan Skripsi Online ditinjau dari pendekatan *usability*-nya adalah dapat diterima dengan baik oleh *user*.

6.2 Saran

Berikut ini beberapa hal dan saran yang kiranya dapat menjadi bahan pertimbangan dalam upaya pengembangan yang lebih baik dimasa yang akan datang :

1. Selanjutnya untuk tahap yang akan datang sistem informasi ini, juga akan di kembangkan menjadi aplikasi web yang berbasis WAP. Hal tersebut supaya sistem ini dapat di akses oleh siapapun yang memiliki perangkat mobile (telepone genggam).
2. Jika nanti di bangun pengembangan aplikasi pengajuan proposal dan skripsi online , maka di sarankan untuk membangun page forum yang lebih fleksibel.