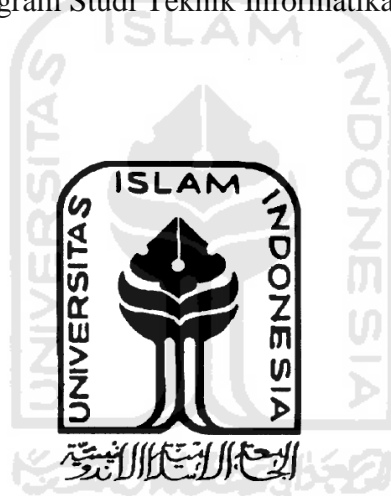


**SISTEM INFORMASI MANAJEMEN KARYASISWA  
DAN PELATIHAN SDM UII MENGGUNAKAN  
FRAMEWORK CODEIGNITER**

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana  
Program Studi Teknik Informatika



Oleh:

Nama : Syauqie Muhammad Marier

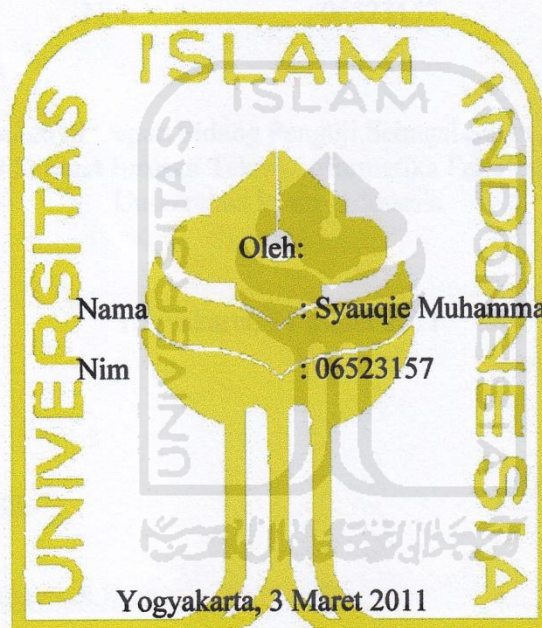
NIM : 06523157

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA  
2011**

**LEMBAR PENGESAHAN**

**SISTEM INFORMASI MANAJEMEN KARYASISWA DAN  
PELATIHAN SDM UII MENGGUNAKAN FRAMEWORK  
CODEIGNITER**

**TUGAS AKHIR**



بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ  
Pembimbing,

Hendrik, ST.,M.Eng

**LEMBAR PENGESAHAN PENGUJI**  
**SISTEM INFORMASI MANAJEMEN KARYASISWA DAN**  
**PELATIHAN SDM UII MENGGUNAKAN FRAMEWORK**  
**CODEIGNITER**

**TUGAS AKHIR**

Oleh:

Nama : Syauqie Muhammad Marier

Nim : 06523157

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat untuk  
Memperoleh Gelar Sarjana Jurusan Teknik Informatika Fakultas Teknologi Industri  
Universitas Islam Indonesia

Yogyakarta, 24 Maret 2011

Tim penguji

Hendrik, S.T., M.Eng.

Ketua

Nur Wijyaning Rahayu, S.Kom

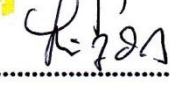
Anggota I

Lizda Iswari, S.T., M.Sc.

Anggota II

()

()

()

Mengetahui

Ketua Jurusan Teknik Informatika

Universitas Islam Indonesia

  
Yudi Prayudi, S.Si., M.Kom

**KUPERSEMBAHKAN TUGAS AKHIR INI KEPADA:**

Kedua orang tua dan keluargaku yang selalu mendoakan kebaikan dan keberhasilan, para guru di ponpes Sunan Pandanaran, asrama mahasiswa dan juga teman-teman asrama.



## MOTTO

قَانَ مَعَ الْعُسْرِ يُسْرًا إِنَّ مَعَ الْعُسْرِ يُسْرًا

Karena sesungguhnya sesudah kesulitan itu ada kemudahan # Sesungguhnya sesudah kesulitan itu ada kemudahan (Q.S. Al-Insyirah: 5-6)



## KATA PENGANTAR

*Alhamdulillah* *rabbilalamin*, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT yang selalu memberikan rahmat dan kekuatan tanpa putus sehingga karena-Nya penulis dapat menyelesaikan tugas akhir. Mudah-mudahan karena rahmat-Nya, penulis dan semua umat islam kelak mendapat rahmat di hari akhir, amin.

*Allahumma sholli ala sayyidina Muhammad wa ala alihi wa dzuriyyati wa ashabi ajmain*. Sholawat serta salam mudah-mudahan selalu terpancar kepada junjungan kita, Nabi Muhammad SAW, beserta keluarga, dzuriyyat dan shahabat beliau. Mudah-mudahan kita semua mendapat syafaatnya kelak di hari akhir, amin.

Laporan tugas akhir ini membahas tentang pembuatan sistem informasi manajemen karyawan dengan menggunakan framework CodeIgniter. Terdapat lima bab pada laporan tugas akhir ini, yaitu: pendahuluan, landasan teori, metodologi, hasil dan pembahasan dan simpulan dan saran.

Semoga laporan ini dapat bermanfaat walaupun hanya sedikit. Khususnya mahasiswa yang memungkinkan mengembangkan lebih lanjut sistem ini dan adik angkatan yang membutuhkan referensi laporan ataupun mencari gambaran judul untuk tugas akhir kedepan.

Penulis menyadari dalam penyusunan laporan tugas akhir ini terdapat kesalahan. Untuk itu penulis tidak menutup kemungkinan kritik serta saran yang bersifat membangun guna kemanfaatan kedepan.

Tak lupa pula penulis mengucapkan terimakasih kepada semua pihak yang secara langsung maupun tidak langsung dalam membantu penulis untuk menyelesaikan tugas akhir ini, yaitu kepada:

1. Para Pendiri dan Badan Wakaf UII.
2. Rektor UII bapak Prof. Dr. Edy Suandi Hamid, M.Ec.
3. Dekan FTI bapak Ir. Gumbolo HS., M.Sc.

4. Ka. Prodi Informatika bapak Yudi Prayudi, S.Si., M.Kom.
5. Bapak Hendrik ST., M.Eng sebagai dosen pembimbing.
6. Kedua Orang Tua.
7. Kiyai Ponpes Pandanaran Asrama Mahasiswa.
8. Teman-teman semua yang tidak dapat disebutkan satu per satu.

Semoga Allah SWT membalas yang lebih baik kepada semua, dan hasil penelitian tugas akhir ini dapat bermanfaat, Amin.



Syauqie Muhammad Marier

## SARI

Materi yang dikerjakan pada pelaksanaan tugas akhir ini adalah pembuatan sistem informasi manajemen karyasiswa UII. Dengan mengembangkan sistem lama yang tidak memiliki aplikasi *reminder* dan cetak laporan sesuai dengan *form* pembayaran dari bank.

Metode analisis yang digunakan adalah metode berorientasi obyek. Dari hasil analisis, didefinisikan sembilan fungsi yang akan dibuat selama proses penelitian. Fungsi tersebut adalah: manajemen karyasiswa, manajemen negara, manajemen perguruan tinggi, manajemen fakultas, manajemen jurusan, manajemen asaldana, manajemen pembayaran, manajemen bank dan manajemen golongan. Pada tahap perancangan perangkat lunak, digambarkan melalui diagram *use case*, *class diagram* dan *activity diagram*.

Kemudian, dari hasil perancangan dilanjutkan dengan implementasi pembuatan program menggunakan framework codeigniter. Secara umum fungsi implementasi perangkat lunak telah sesuai dengan hasil analisis dan perancangan yang telah dilakukan sebelumnya, yaitu adanya aplikasi *reminder* dan cetak laporan pada sistem informasi karyasiswa.

Kata-kata kunci : karyasiswa, UII, CodeIgniter.



## TAKARIR

<i>Living cost</i>	: Biaya kehidupan selama karyasiswa.
<i>Database</i>	: Tempat penyimpanan data.
<i>Form</i>	: Tempat isian data.
<i>Reminder</i>	: Peningat.
<i>Localhost</i>	: Server lokal.
<i>Interface</i>	: Tampilan program.
<i>Framework</i>	: Kumpulan fungsi-fungsi.
<i>Input</i>	: Masukan data.
<i>Ouput</i>	: Hasil keluaran.
<i>Programmer</i>	: Seorang membuat program komputer.
<i>Web disign</i>	: Desain web.
<i>Case by case</i>	: Kasus per kasus.
<i>User guide</i>	: Kumpulan dokumentasi.
<i>Use case diagram</i>	: Diagram fungsional.
<i>Class diagram</i>	: Diagram berdasarkan kelas objek.
<i>Activity diagram</i>	: Diagram berdasarkan aktivitas.
<i>Insert</i>	: Memasukkan data.
<i>Edit</i>	: Melakukan perubahan data.
<i>Delete</i>	: Menghapus data.
<i>Degree</i>	: Tingkatan jenjang studi.

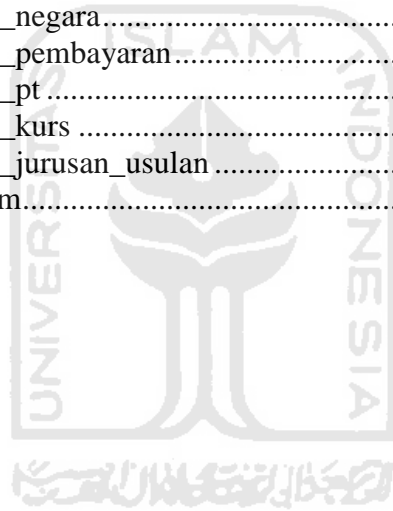
## DAFTAR ISI

Halaman Judul.....	i
Lembar Pengesahan Dosen Pembimbing.....	ii
Lembar Pengesahan Dosen Penguji .....	iii
Halaman Persembahan .....	iv
Halaman Motto.....	v
Kata Pengantar .....	vi
Sari .....	viii
Takarir .....	ix
Daftar Isi.....	x
Daftar Tabel.....	xi
Daftar Gambar.....	xii
BAB I Pendahuluan.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II Landasan teori .....	5
2.1 Sistem Informasi .....	5
2.2 <i>Framework</i> .....	5
2.3 <i>Framework</i> CodeIgniter .....	6
2.3 Struktur Direktori File CodeIgniter.....	8
2.4 Konsep <i>Model-View-Controller</i> (MVC).....	10
2.5 Keuntungan MVC .....	11
2.6 Kelebihan MVC .....	12
BAB III Metodologi .....	13
3.1 Metode Analisis .....	13
3.2 Analisis Kebutuhan.....	13
3.2.1 Masukan .....	13
3.2.2 Keluaran .....	16

3.3	Perancangan Sistem .....	18
3.3.1	UML .....	18
3.3.1.1	<i>Use Case Diagram</i> .....	18
3.3.1.2	<i>Class Diagram</i> .....	20
3.3.1.3	<i>Activity Diagram</i> .....	26
3.3.2	Struktur Tabel .....	31
3.3.3	Rancangan Antarmuka .....	40
3.3.3.1	Halaman Login .....	40
3.3.3.2	Halaman Home .....	40
3.3.3.3	Halaman Tampil dan Edit Data .....	41
BAB IV	Hasil dan pembahasan .....	42
4.1	Batasan Implementasi .....	42
4.2	Implementasi .....	42
4.2.1	Implementasi Rancangan Antarmuka .....	42
4.2.2	Implementasi Program .....	44
4.3	Evaluasi Kinerja Sistem .....	52
4.3.1	Pengujian Normal .....	52
4.3.2	Pengujian Tidak Normal .....	53
4.4	Perbandingan dan Peningkatan Sistem .....	55
4.4.1	Perbandingan Sistem .....	55
4.4.2	Peningkatan Sistem .....	56
BAB V	Simpulan dan Saran .....	57
5.1	Simpulan .....	57
5.2	Saran .....	57
	Daftar Pustaka .....	58

## DAFTAR TABEL

Tabel 3.1 Spesifikasi kebutuhan sistem .....	19
Tabel 3.2 Spesifikasi tabel t_admin .....	31
Tabel 3.3 Spesifikasi tabel t_asaldana.....	31
Tabel 3.4 Spesifikasi tabel t_bank.....	31
Tabel 3.5 Spesifikasi tabel t_degree.....	32
Tabel 3.6 Spesifikasi tabel t_dosen .....	32
Tabel 3.7 Spesifikasi tabel t_fakultas.....	33
Tabel 3.8 Spesifikasi tabel t_golongan .....	34
Tabel 3.9 Spesifikasi tabel t_karyasiswa.....	34
Tabel 3.10 Spesifikasi tabel t_jurusan.....	35
Tabel 3.11 Spesifikasi tabel t_matauang.....	36
Tabel 3.12 Spesifikasi tabel t_negara.....	36
Tabel 3.13 Spesifikasi tabel t_pembayaran.....	37
Tabel 3.14 Spesifikasi tabel t_pt .....	38
Tabel 3.15 Spesifikasi tabel t_kurs .....	39
Tabel 3.16 Spesifikasi tabel t_jurusan_usulan .....	39
Tabel 4.1 Perbandingan sistem.....	55



## DAFTAR GAMBAR

Gambar 1.1 Perkembangan karyawan pertahun .....	1
Gambar 2.1 Gamber direktori codeigniter .....	9
Gambar 2.2 Model-View Controller .....	11
Gambar 3.1 <i>Use case diagram</i> sistem .....	20
Gambar 3.2 <i>Class diagram</i> .....	21
Gambar 3.3 <i>Class diagram</i> aplikasi karyawan .....	22
Gambar 3.4 <i>Activity diagram</i> sistem informasi karyawan .....	27
Gambar 3.5 <i>Activity diagram</i> manajemen bank .....	28
Gambar 3.6 <i>Activity diagram</i> manajemen karyawan .....	29
Gambar 3.7 <i>Activity diagram</i> manajemen biaya .....	30
Gambar 3.8 Rancangan halaman login.....	40
Gambar 3.9 Rancangan halaman home .....	41
Gambar 3.10 Rancangan Halaman Tampil dan Edit Data .....	41
Gambar 3.10 Rancangan Halaman Tampil dan Edit Data .....	41
Gambar 4.1 Halaman Login .....	43
Gambar 4.2 Halaman home .....	43
Gambar 4.3 Halaman tampil dan edit data .....	44
Gambar 4.4 Form tambah data usulan karyawan .....	45
Gambar 4.5 Data usulan karyawan .....	46
Gambar 4.6 Detail karyawan .....	47
Gambar 4.7 Daftar karyawan .....	48
Gambar 4.8 <i>Input</i> nama dosen pada proses <i>living cost</i> .....	48
Gambar 4.9 <i>Form</i> biaya dan riwayat pembayaran .....	49
Gambar 4.10 Daftar penerima <i>living cost</i> .....	49
Gambar 4.11 <i>Reminder</i> .....	50
Gambar 4.12 Hasil cetak laporan .....	50
Gambar 4.13 Klik lulus untuk menjadikan status lulus .....	51
Gambar 4.14 Daftar lulusan .....	52
Gambar 4.15 Pesan input berhasil.....	52
Gambar 4.16 Pesan berhasil update .....	53
Gambar 4.17 Pesan akan menghapus data .....	53
Gambar 4.18 Pesan sukses ganti password .....	53
Gambar 4.19 Pesan data <i>required</i> apabila tidak diisi.....	54
Gambar 4.20 Password lama tidak sesuai .....	54

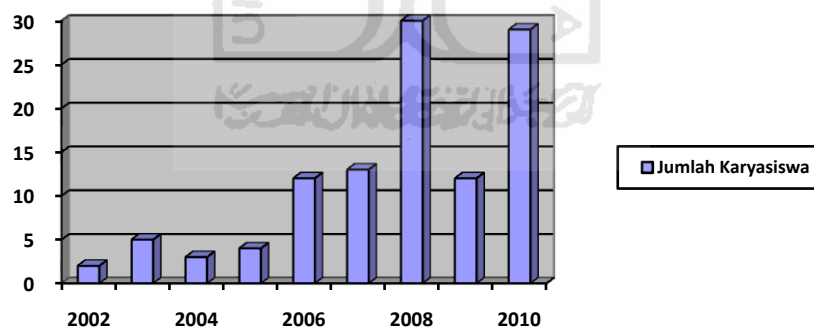
# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam Peraturan Universitas Nomor: 05/PU/Rek/VI/2009 tentang karyasiswa dosen Universitas Islam Indonesia dijelaskan bahwa: “karyasiswa adalah kegiatan atau penyertaan dosen untuk menempuh pendidikan pada jenjang yang lebih tinggi sebagaimana Peraturan yang berlaku di Universitas Islam Indonesia (UII)”.

Data perkembangan karyasiswa dari tahun ke tahun dapat dikatakan meningkat. Berikut jumlah karyasiswa UII pertahun: tahun 2002 berjumlah 2 orang, tahun 2003 berjumlah 5 orang, tahun 2004 berjumlah 3 orang, tahun 2005 berjumlah 4 orang, tahun 2006 berjumlah 12 orang, tahun 2007 berjumlah 13 orang, tahun 2008 berjumlah 30 orang, tahun 2009 berjumlah 12 orang, dan tahun 2010 berjumlah 29 orang. Data perkembangan karyasiswa per tahun dapat dilihat pada gambar 1.1.



Gambar 1.1 Perkembangan Karyasiswa Pertahun

Sistem pengolahan data karyasiswa yang sudah ada, dikembangkan menggunakan dBIII Plus. Dalam implementasi aplikasi, masih terkendala beberapa masalah, seperti tidak adanya program cetak dan *reminder*. Setiap kali akan mencetak data, staf divisi pengembangan sumber daya manusia (SDM) bagian karyasiswa merasa kesulitan karena harus memindahkan data pada dua program yang berbeda, yaitu dBIII Plus dan MsWord. Begitu juga dengan waktu pengiriman biaya bagi dosen yang sedang menempuh pendidikan (karyasiswa),

yaitu kesulitan untuk mengingat kapan dan siapa dosen yang sudah waktunya untuk dikirim biaya pendidikan dan keseharian (*living cost*). Karena untuk mengetahui hal itu harus melihat *database* atau menunggu telpon dari dosen yang bersangkutan.

Oleh karena itu, perlu adanya sistem baru yang dapat mengatasi permasalahan di atas, agar proses pengolahan data karyawan dapat berjalan lebih baik.

## 1.2 Rumusan Masalah

Berdasarkan batasan masalah di atas, didapatkan sebuah rumusan, yaitu: bagaimana membuat aplikasi sistem informasi pengolahan karyawan yang di dalamnya terdapat fitur program cetak dan *reminder*.

## 1.3 Batasan Masalah

Agar tetap fokus dalam pembuatan aplikasi, dibuat beberapa batasan masalah. Batasan masalah tersebut adalah:

1. Sistem untuk sementara berjalan di *localhost* (server lokal).
2. Sistem tidak terintegrasi dengan data karyawan dan kepegawaian.
3. Menu dan *interface* aplikasi berdasarkan sistem yang lama.

## 1.4 Tujuan Penelitian

Tujuan dari pembuatan aplikasi ini adalah untuk mengembangkan sistem informasi karyawan, dan memacu penulis mempelajari lebih lanjut tentang *framework* khususnya CodeIgniter.

## 1.5 Manfaat Penelitian

Aplikasi ini diharapkan dapat dimaksimalkan oleh staf bagian pengembangan SDM dalam mengolah dan memperoleh informasi tentang karyawan di UII.

## 1.6 Sistematika Penulisan

Sistematika penulisan laporan berguna untuk memberikan gambaran umum dari keseluruhan isi laporan. Sistematika penulisan dan garis besar isi laporan ini adalah sebagai berikut:

## **BAB I      PENDAHULUAN**

Bab ini berisi tentang latar belakang penulisan laporan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

## **BAB II     LANDASAN TEORI**

Bab ini berisi tentang teori-teori yang berhubungan dengan penelitian, seperti definisi sistem informasi, *framework*, *framework* CodeIgniter, dan konsep *Model View Controller*(MVC).

## **BAB III    METODOLOGI**

Bab ini berisi penjelasan tentang langkah-langkah penyelesaian masalah dari analisis kebutuhan sistem sampai analisis perancangan sistem.

## **BAB IV    HASIL DAN PEMBAHASAN**

Bab ini berisi tentang batasan implementasi, implementasi dan evaluasi kinerja sistem.

## **BAB V     SIMPULAN DAN SARAN**

Bab ini berisi simpulan-simpulan yang merupakan rangkuman dari hasil analisis kinerja pada bagian sebelumnya. Dan saran berdasar keterbatasan yang ditemukan selama pengembangan sistem.



## BAB II

# LANDASAN TEORI

### 2.1 Sistem Informasi

Sistem informasi terdiri dari dua kata pembentuk, yaitu sistem dan informasi. Sistem adalah kumpulan elemen-elemen yang memproses masukan (*input*) untuk diolah kedalam sistem sampai menghasilkan keluaran (*output*) yang diinginkan [KRI03]. Sedangkan informasi merupakan kumpulan data yang bermanfaat bagi penerima [KRI03]. Jadi, sistem informasi adalah proses pengolahan data masukan (*input*) di dalam sistem untuk menghasilkan keluaran (*output*) berupa data-data yang bermanfaat bagi penerima.

### 2.2 Framework

*Framework* dalam web mempunyai definisi kumpulan modul-modul dalam bentuk *class library* yang digunakan untuk membangun sebuah aplikasi web [WIS10]. Mengapa *framework*? Karena *framework* mempunyai aturan-aturan dalam penulisan *programming* [BAS10]. Banyak kasus *programer* lebih memilih membuat ulang sebuah aplikasi daripada harus mempelajari gaya *coding* *programer* lain ketika dihadapkan masalah menambah atau mengembangkan aplikasi yang sudah ada.

Pada Desember 2010, sedikitnya sudah ada 19 *framework* php yang terdata di website <http://www.phpframeworks.com>. Diantaranya: CodeIgniter, CakePHP, Symphony, Zend, Akelos, Prado dan lain-lain.

### 2.3 Framework CodeIgniter

CodeIgniter adalah sebuah *framework* untuk web yang dibuat dalam format PHP. CodeIgniter merupakan *framework* yang menggunakan konsep *Model-View-Controller* (MVC) yang bersifat *open source*. CodeIgniter

dikembangkan oleh Rick Ellis yang merupakan CEO dari Ellislab, yaitu perusahaan yang mengembangkan CodeIgniter.

Beberapa alasan menggunakan CodeIgniter adalah karena CodeIgniter:

1. *Open source*

CodeIgniter adalah *framework* yang gratis untuk digunakan dan dapat dikembangkan secara legal.

2. Menggunakan konsep *Model-View-Controller* (MVC)

CodeIgniter menggunakan konsep MVC, yaitu memisahkan antara aplikasi logika dengan presentasi pada halaman web sehingga mempermudah *web programmer* dan *web designer* dalam membangun atau mengembangkan aplikasi web.

3. Dokumentasi yang lengkap

CodeIgniter mempunyai dokumentasi yang sangat rapi, lengkap dan jelas. Sehingga membantu para *programmer* pemula yang ingin mempelajari *framework* CodeIgniter.

4. Komunitas

Komunitas pengguna CodeIgniter saat ini berkembang pesat, adakalanya mencari informasi *case by case* yang tidak ditemukan pada *user guide*. Forum CodeIgniter adalah tempat yang tepat, yaitu di <http://codeigniter.com/forums>.

Selain alasan di atas, CodeIgniter mempunyai banyak fitur. Berikut fitur-fitur yang didukung oleh CodeIgniter :

1. Sistem berbasis *Model View Controller*
2. Kompatibel dengan PHP versi 4 dan 5.
3. Ringan dan Cepat.
4. Terdapat dukungan untuk berbagai basis data.
5. Mendukung *Active Record Database*.

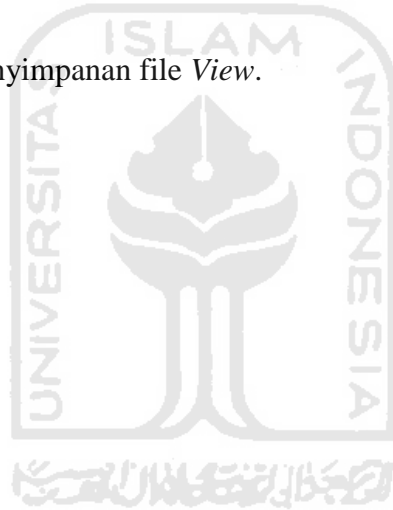
6. Mendukung *form* dan *validasi* data masukan.
7. Keamanan dan *XSS filtering*.
8. Tersedia pengaturan *session*.
9. Tersedia *class* untuk mengirim email.
10. Tersedia *class* untuk manipulasi gambar (*cropping, resizing, dan rotate*).
11. Tersedia *class* untuk *upload* file.
12. Tersedia *class* yang mendukung transfer *via FTP*.
13. Tersedia *class* untuk melakukan *pagination*.
14. Mendukung *enkripsi* data.
15. Mendukung *benchmarking*.
16. Mendukung *caching*.
17. Pencatatan *error* yang terjadi.
18. Tersedia *class* untuk membuat kalender.
19. Tersedia *class* untuk mengetahui *user agent*, misalnya tipe *browser* dan sistem operasi yang digunakan pengunjung.
20. Tersedia *class* untuk pembuatan *template* website.
21. Tersedia *class* untuk membuat *trackback*.
22. Tersedia pustaka untuk bekerja dengan *XMP-RPC*.
23. Menghasilkan *URL* yang bersih.
24. *URI routing* yang fleksibel.
25. Memiliki *helper* yang sangat banyak jumlahnya.

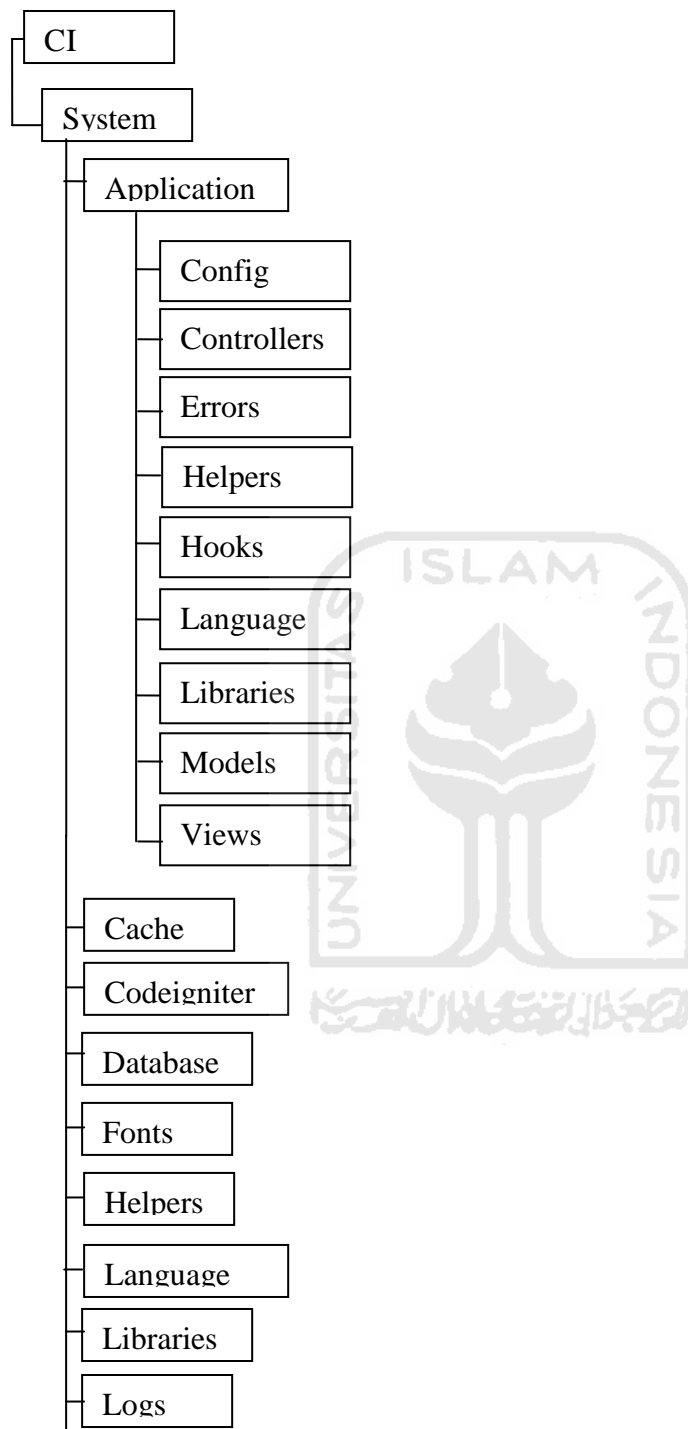
### 2.3 Struktur Direktori File CodeIgniter

CodeIgniter adalah *php framework* yang terdiri dari kumpulan folder dan file. Adapun struktur direktori file CodeIgniter yang digunakan penulis dalam pembuatan aplikasi ini (CodeIgniter versi 1.7.3) dapat dilihat pada gambar 2.1.

Penjelasan gambar: Semua file aplikasi yang dibangun diletakkan di dalam folder application. Berikut beberapa penjelasan fungsi dari folder-folder di CodeIgniter.

- *Config* :Berisi file-file konfigurasi, menyimpan informasi sebuah situs.
- *Controllers* :Penyimpanan file controller.
- *Errors* :Berisi template halaman untuk penampilan *error*.
- *Hooks* :Menyimpan *hooks* yang dibuat, *hooks* mengontrol bagaimana suatu file akan diload.
- *Views* :Penyimpanan file *View*.





**Gambar 2.1** Gambar Direktori Codeigniter Versi 1.7.3

- **Scaffolding** :Penyimpanan data *cache*.
- **CodeIgniter** :File-file dasar untuk sistem.

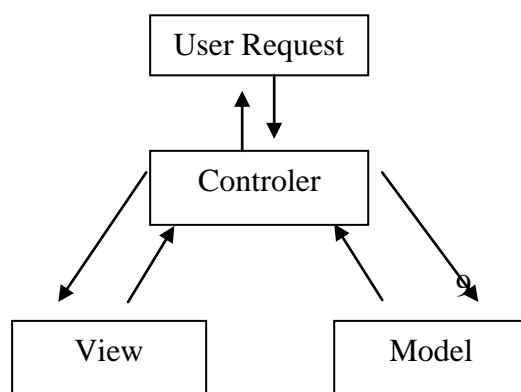
- *Database* :*Library* untuk *class database* CodeIgniter.
- *Fonts* :Untuk font.
- *Helpers* :*Helper* untuk level sistem.
- *Language* :File untuk bahasa.
- *Libraries* :*Library* untuk level sistem.
- *Logs* :Jika *logs* untuk *error* diaktifkan, file-file *logs* akan disimpan secara *default*.
- *Plugins* :Kumpulan blok kode untuk level sistem.
- *Scaffolding* :Bantuan operasi database sederhana.

#### 2.4 Konsep Model-View-Controller(MVC)

MVC adalah sebuah konsep yang memisahkan antara aplikasi logika dengan presentasi pada halaman web.

- **Model** merupakan struktur data, pada *class Model* terdapat fungsi-fungsi yang membantu dalam proses yang berhubungan dengan database, seperti: *insert, edit, delete*.
- **View** merupakan informasi yang ditampilkan ke pengguna melalui *browser*.
- **Controller** digunakan untuk mengontrol aliran data aplikasi antara *Model* dan *View* dan semua sumber yang dibutuhkan untuk memproses permintaan HTTP oleh pengguna.

Adapun gambaran konsep MVC dapat dilihat pada gambar 2.2.



Gambar 2.2 Model-View-Controller

Ketika datang sebuah *user request* (permintaan pengguna), maka permintaan tersebut akan ditangani oleh *Controller*, kemudian *Controller* akan memanggil *Model* jika diperlukan operasi database. Hasil *query* dari *Model* akan dikembalikan ke *Controller*. *Controller* akan memanggil *View* dan mengkombinasikan dengan hasil *query Model*. Hasil akhir dari operasi ini akan ditampilkan ke *browser* yang dilihat oleh pengguna.

## 2.5 Keuntungan MVC

Sekilas, menggunakan konsep MVC sangat merepotkan dan membingungkan. Penulisan kode program harus dipisah menjadi tiga bagian, kode yang berhubungan dengan database di *model*, tampilan di *view* dan *controller* untuk menghubungkan keduanya. Akan tetapi, konsep MVC mempunyai keuntungan-keuntungan.

Keuntungannya adalah kode program yang dibuat lebih *maintainable* karena kita dapat merubah salah satu bagian tanpa harus merubah bagian yang lain. Sebagai contoh, ketika ingin merubah tampilan sebuah aplikasi web, akan banyak melakukan perombakan di mana-mana karena di dalamnya tercampur kode untuk tampilan dan operasi database. Jika menggunakan konsep MVC, kita hanya merubah di bagian viewnya saja.

Keuntungan lainnya, MVC mempermudah *debugging* karena dapat memperkirakan bagian yang bermasalah tanpa harus membongkar kode program.

## 2.6 Kelebihan MVC

Dalam MVC, dikenal tiga komponen pembangun yaitu *Model*, *View*, dan *Controller*. Berikut adalah kelebihan dari tiap-tiap komponen MVC:

### *Model*

- Penempatan detail data dan operasinya pada area yang ditentukan (*Model*).
- Dapat menggunakan kembali aplikasi model yang ada oleh aplikasi yang lain.

***View***

- Dapat memisahkan bagian *design* dan bagian *development* (pengembangan).
- Ketersediaan banyak *interface* dalam aplikasi.

***Controller***

1. Pemisahan antara logika bisnis dan logika presentasi.





## **BAB III**

# **METODOLOGI**

### **3.1 Metode Analisis**

Metode analisis yang digunakan untuk pembuatan perangkat lunak sistem informasi karyawan ini adalah menggunakan analisis berorientasi objek. Metode ini adalah metode yang memeriksa *requirement* (syarat/keperluan yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemukan dalam ruang lingkup permasalahan.

Hasil analisis berorientasi objek adalah gambaran dari sistem secara fungsional dalam bentuk sebuah model konseptual. Hasil analisis ini biasanya akan disajikan sebagai bahan penyelesaian kasus. Tujuan dari analisis berorientasi objek adalah untuk menggambarkan perangkat lunak komputer dan memenuhi persyaratan yang ditentukan pelanggan.

### **3.2 Analisis Kebutuhan**

Menganalisis kebutuhan sistem berupa masukan dan keluaran yang diinginkan. Pada tahapan ini bertujuan untuk proses pengembangan aplikasi yang dibuat.

#### **3.2.1 Masukan**

Masukan pada sistem informasi karyawan terdapat pada beberapa proses penambahan data, seperti: tambah usulan, tambah fakultas, tambah negara, tambah jurusan dan proses penambahan data lainnya. Berikut rincian masukan yang diperoleh setelah tahap analisis kebutuhan masukan.

- a. Proses tambah usulan karyawan

Pada proses ini terdapat dua pengisian data, yaitu masukan data diri dosen dan masukan data usulan karyawan. Masukan data diri dosen terdiri dari:

1. NIP (Nomor Induk Pegawai).
2. Nama.
3. Tempat lahir.
4. Tanggal lahir.
5. Kelamin.
6. Fakultas asal.
7. Jurusan asal.
8. Golongan.
9. Email.
10. Alamat rumah.
11. Nomor telpon.
12. Nomor *handphone*.

Masukan untuk usulan karyasiswa terdiri dari:

1. Negara tujuan.
  2. Perguruan tinggi.
  3. Lokasi perguruan tinggi.
  4. *Degree* (jenjang studi).
  5. Jurusan yang dipilih.
- b. Proses tambah negara
- Masukan pada proses ini adalah nama Negara.
- c. Proses tambah fakultas

Masukan pada proses ini adalah nama fakultas.

- d. Proses tambah jurusan

Masukan pada proses ini adalah nama jurusan.

- e. Proses tambah perguruan tinggi

Masukan pada proses ini terdiri dari:

1. Nama perguruan tinggi.
2. Alamat perguruan tinggi.
3. Kota perguruan tinggi.
4. Nomor telpon perguruan tinggi.

- f. Proses tambah asal dana

Masukan pada proses ini adalah nama lembaga.

- g. Proses tambah bank

Masukan pada proses ini adalah nama bank.

- h. Proses tambah matauang

Masukan pada proses ini terdiri dari:

1. Nama matauang.
2. Alamat singkatan matauang.
3. Negara.

- i. Proses *input* biaya kuliah atau *living cost*

Masukan pada proses ini terdiri dari:

1. Jumlah uang.
2. Nomor rekening.

3. Matauang.
4. Bank.
5. Asal dana.
6. Tanggal kirim
7. Tanggal batas akhir.

### 3.2.2 Keluaran

Berdasarkan masukan dari hasil analisis di atas, keluaran sistem yang diperoleh sesuai dengan masukan yang telah diproses sebelumnya. Berikut hasil keluaran yang didapat.

- a. Keluaran usulan karyasiswa

Hasil tampilan untuk usulan karyasiswa adalah:

1. Nama.
2. Fakultas.
3. Kota tujuan
4. Perguruan tinggi.
5. Beasiswa (jurusan).
6. *Degree* (jenjang studi).
7. Lokasi perguruan tinggi.

- b. Keluaran Negara

Hasil tampilan untuk negara adalah nama negara.

- c. Keluaran fakultas

Hasil tampilan untuk fakultas adalah nama fakultas.

- d. Keluaran jurusan

Hasil tampilan untuk negara adalah nama negara.

e. Keluaran perguruan tinggi

Hasil tampilan untuk perguruan tinggi adalah:

1. Nama perguruan tinggi.
2. Kota tempat perguruan tinggi.

f. Keluaran asal dana

Hasil tampilan untuk asal dana adalah nama lembaga yang membiayai.

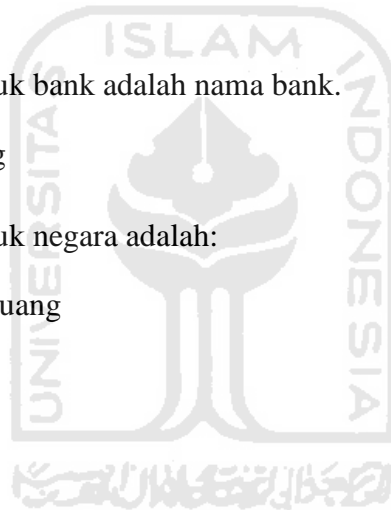
g. Keluaran bank

Hasil tampilan untuk bank adalah nama bank.

h. Keluaran matauang

Hasil tampilan untuk negara adalah:

1. Nama matauang
2. Singkatan
3. Negara.



i. Keluaran pembayaran

Hasil tampilan untuk pembayaran adalah sesuai dengan inputan nip, yaitu:

1. Nama.
2. Jumlah.
3. Bank.
4. Rekening.
5. Keterangan biaya.
6. Tanggal kirim.

7. Tanggal batas.

### 3.3 Perancangan Sistem

#### 3.3.1 UML

*Unified Modelling Language* (UML) adalah sebuah bahasa untuk merancang, mendokumentasikan dan memvisualisasikan sistem perangkat lunak dengan menggambarkan informasi-informasi secara grafis dengan notasi-notasi yang telah ditentukan.

Dalam membangun model perangkat lunak menggunakan UML, digunakan bentuk diagram untuk mempresentasikan elemen-elemen di dalam sistem. Terdapat delapan bentuk diagram yang dapat digunakan untuk mempresentasikan elemen-elemen di dalam sistem, yaitu: *use case diagram*, *class diagram*, *state diagram*, *sequence diagram*, *collaboration diagram*, *activity diagram*, *component diagram* dan *deployment diagram*.

Namun, diagram yang akan digunakan disini adalah *use case diagram*, *class diagram* dan *activity diagram*.

##### 3.3.1.1 Use Case Diagram

*Use case diagram* adalah diagram yang menggambarkan fungsionalitas dari sebuah sistem, dilihat dari perspektif pengguna di luar sistem. *Use case diagram* dapat digunakan untuk analisis *requirement* sistem dan memahami proses apa saja yang dapat dilakukan oleh sistem.

Sebelum membuat *use case diagram*, dilakukan sebuah analisis kebutuhan sistem dengan menjabarkan apa saja *requirement* sistem, siapa aktor dan proses apa saja yang dilakukan sistem.

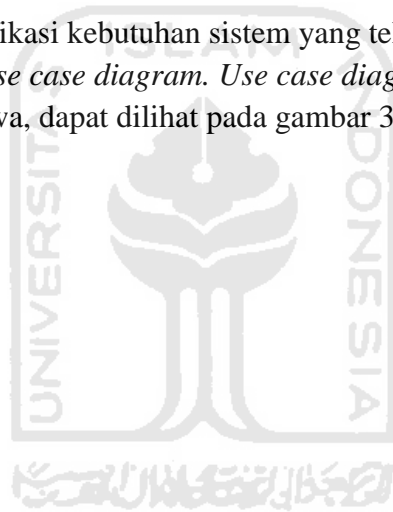
Hasil analisis yang diperoleh, pada sistem ini terdapat satu user yang akan menggunakan aplikasi ini, yaitu admin. Admin dapat melakukan proses manajemen, antara lain: manajemen karyasiswa, manajemen negara, manajemen fakultas, manajemen jurusan, manajemen perguruan tinggi dan lainnya. Untuk lebih detail, spesifikasi kebutuhan sistem dapat dilihat pada tabel 3.1.

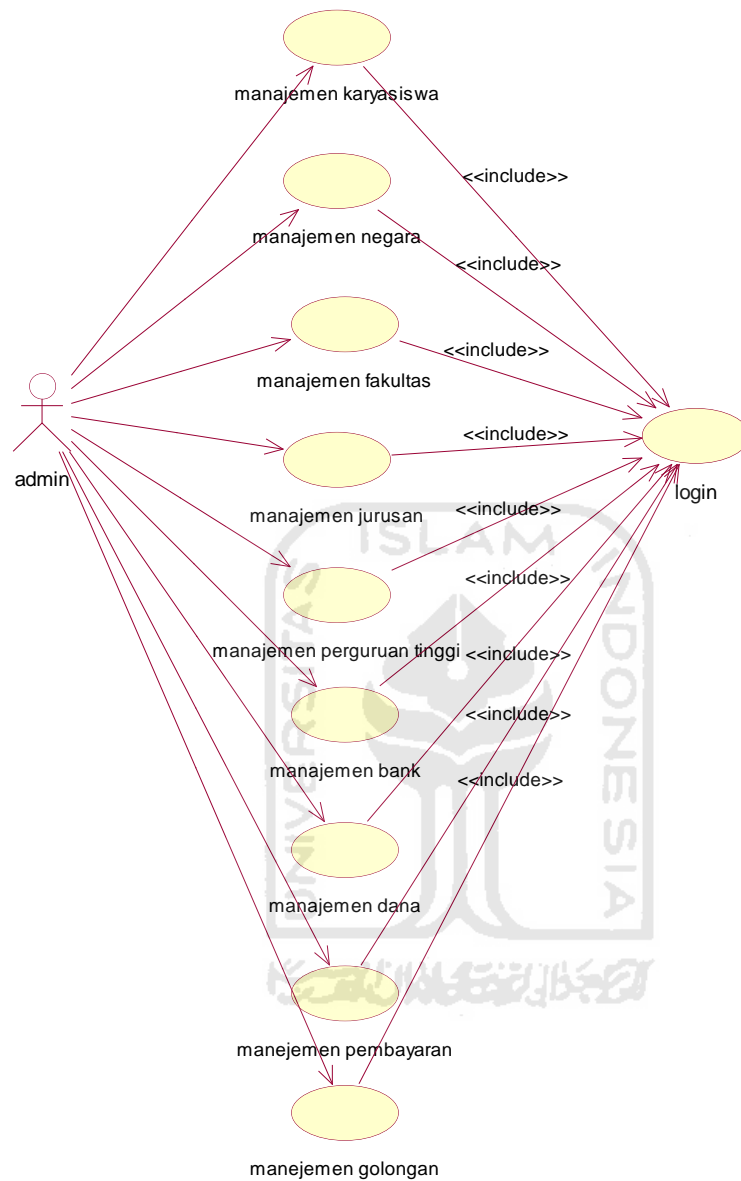
Tabel 3.1 Spesifikasi kebutuhan sistem

<i>Requirement</i>	Aktor	<i>Use case</i>
--------------------	-------	-----------------

<p>Untuk masuk ke dalam halaman web, admin harus login terlebih dahulu. Setelah berhasil login, Admin dapat melakukan proses manajemen, antara lain: manajemen karyasiswa, manajemen negara, manajemen fakultas, manajemen jurusan, manajemen perguruan tinggi, manajemen pembayaran, manajemen asal dana, manajemen bank dan manajemen golongan.</p>	<p>Admin</p>	<p>manajemen karyasiswa, manajemen negara, manajemen fakultas, manajemen jurusan, manajemen perguruan tinggi, manajemen pembayaran, manajemen asal dana, manajemen bank dan manajemen golongan.</p>
---	--------------	---

Berdasarkan spesifikasi kebutuhan sistem yang telah disebutkan pada tabel 3.1, dapat dibuat sebuah *use case diagram*. *Use case diagram* secara umum untuk sistem informasi karyasiswa, dapat dilihat pada gambar 3.1.





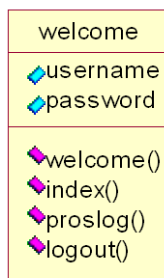
**Gambar 3.1** Use case diagram sistem

### 3.3.1.2 Class Diagram

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan antar *class* di dalam sistem. *Class* adalah sebuah spesifikasi yang jika di-*instansiasi* akan menghasilkan sebuah objek. *Class* menggambarkan keadaan suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*method/fungsi*).

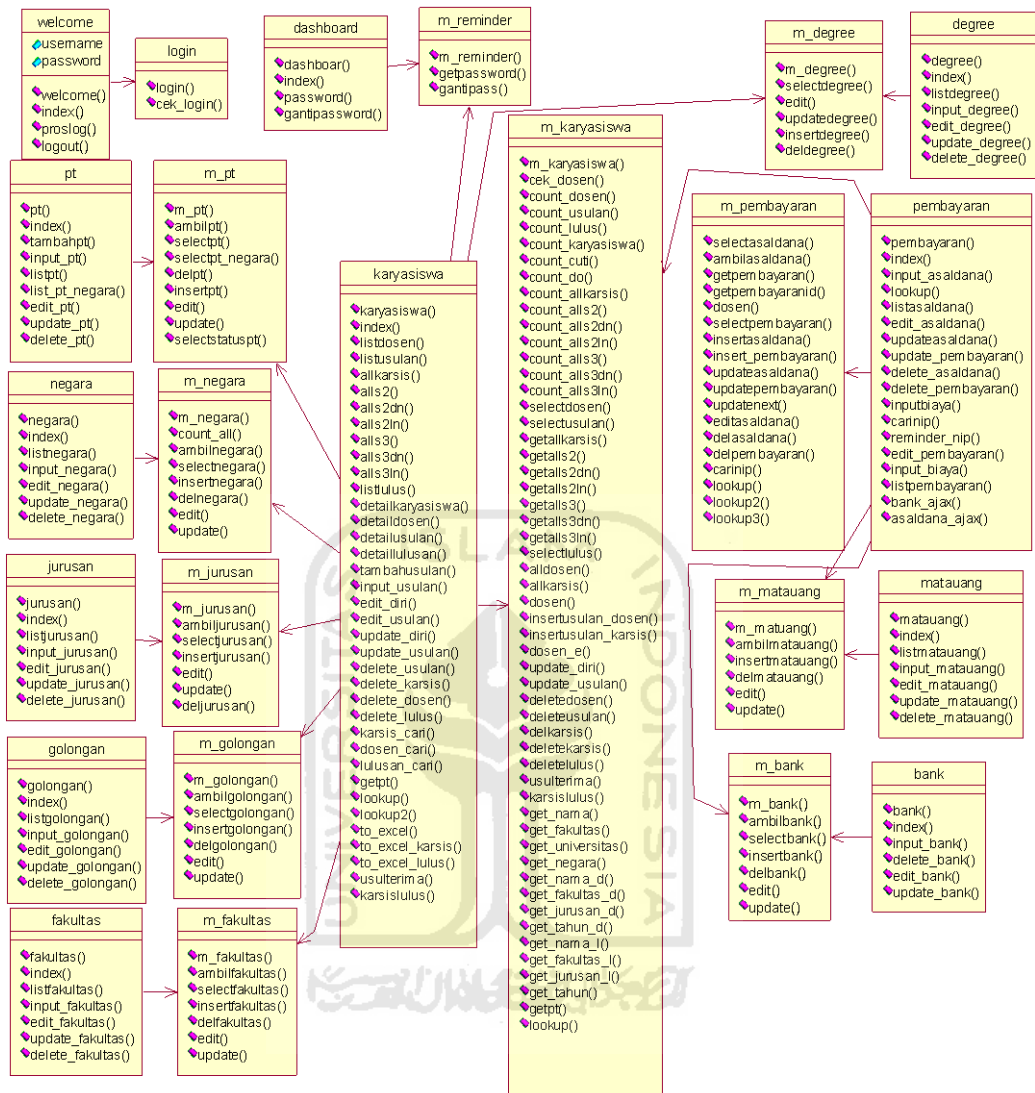


*Class* digambarkan dengan sebuah kotak yang terbagi menjadi tiga bagian. Bagian pertama/ paling atas, diisi nama *class*, bagian tengah/ kedua diisi *variable* / atribut yang dimiliki *class*, dan bagian yang terakhir diisi *method-method* dari *class*. Untuk lebih jelasnya, *class* dapat dilihat pada gambar 3.2.



**Gambar 3.2 Class diagram**

*Class diagram* pada aplikasi ini terdapat dua puluh empat *class*. Sebagian *class* adalah *class* pada *controller* dan sebagian sisanya adalah *class model* yang penamaannya diawali dengan huruf “m”, sebagai contoh m\_reminder, m\_negara, m\_fakultas dan lainnya. Untuk lebih jelasnya, semua *class* dapat dilihat pada gambar 3. 3.



Gambar 3.3 Class diagram aplikasi karyawaniswa

Diagram di atas, menggambarkan *class-class* yang ada di dalam aplikasi dan hubungan asosiasi antar *class* yang digambarkan dengan tanda panah. Di setiap *class* terdapat *method-method* yang menggambarkan perlakuan terhadap *class*.

*Class diagram* di atas terbagi menjadi dua, yaitu *class* pada *class controller* dan *class* pada *class model*. Penjelasan pada *class controller* sebagai berikut:

1. Karyasiswa : *instansiasi* objek dosen yang berisi *method-method* yang berkaitan dengan dosen yang mendapatkan karyasiswa, misalnya: *method insertusulan\_dosen()* yang berfungsi menambah data diri dosen yang mendapatkan karyasiswa, *method insertusulan\_karsis()* yang berfungsi menambah data-data usulan karyasiswa dosen, berupa negara tujuan, jurusan yang diambil, nama perguruan tinggi tujuan dan lainnya, *method to\_excel\_karsis()*, berfungsi membuat daftar karyasiswa bertipe excel.
2. Negara : *instansiasi* objek negara yang berisi *method-method* yang berkaitan dengan negara, misalnya : *method listnegara()*, berfungsi menampilkan daftar negara, *method input\_negara()*, berfungsi input data negara, dan *method delete\_negara()* berfungsi menghapus data negara.
3. Pt : *instansiasi* objek perguruan tinggi yang berisi *method-method* yang berkaitan dengan perguruan tinggi, misalnya : *method listpt ()*, berfungsi menampilkan daftar perguruan tinggi, *method list\_pt\_negara()*, berfungsi menampilkan daftar perguruan tinggi berdasarkan negara, dan *method input\_pt ()* berfungsi menambah data perguruan tinggi.
4. Fakultas : *instansiasi* objek fakultas yang berisi *method-method* yang berkaitan dengan fakultas, misalnya : *method listfakultas()*, berfungsi menampilkan daftar fakultas, *method input\_fakultas()*, berfungsi input data fakultas, dan *method delete\_fakultas()* berfungsi menghapus data fakultas.
5. Jurusan : *instansiasi* objek jurusan yang berisi *method-method* yang berkaitan dengan jurusan, misalnya : *method listjurusan()*, berfungsi menampilkan daftar jurusan, *method input\_jurusan()*, berfungsi input data jurusan, dan *method delete\_jurusan()* berfungsi menghapus data jurusan.
6. Golongan : *instansiasi* objek golongan yang berisi *method-method* yang berkaitan dengan golongan pegawai, misalnya : *method listgolongan()*, berfungsi menampilkan daftar golongan, *method input\_golongan()*,

berfungsi input data golongan, dan *method* `delete_golongan()` berfungsi menghapus data golongan.

7. Bank : *instansiasi* objek bank yang berisi *method-method* yang berkaitan dengan bank, misalnya : *method* `index()`, berfungsi menampilkan daftar bank, *method* `input_bank()`, berfungsi input data bank, dan *method* `delete_bank()` berfungsi menghapus data bank.
8. Matauang : *instansiasi* objek matauang yang berisi *method-method* yang berkaitan dengan matauang, misalnya : *method* `list_matauang()`, berfungsi menampilkan daftar matauang, *method* `input_matauang()`, berfungsi input data matauang, dan *method* `delete_matauang()` berfungsi menghapus data matauang.
9. Pembayaran : *instansiasi* objek pembayaran yang berisi *method-method* yang berkaitan dengan pembayaran karyasiswa, misalnya : *method* `listasaldana()`, berfungsi menampilkan daftar asal pendanaan karyasiswa, *method* `carinip()`, berfungsi menampilkan data riwayat pembayaran berdasarkan nip, dan *method* `inputbiaya()` berfungsi menambahkan pengiriman biaya.
10. Welcome : *instansiasi* objek halaman login web site yang berisi *method-method* yang berkaitan halaman awal login, misalnya : *method* `index()`, berfungsi menampilkan *form* login untuk mengakses aplikasi, *method* `proslog()`, berfungsi memberikan session, dan *method* `logout()` berfungsi menghapus session.
11. Degree : *instansiasi* objek tingkatan karyasiswa yang berisi *method-method* yang berkaitan dengan jenjang studi, misalnya : *method* `list_degree()`, berfungsi menampilkan daftar jenjang studi, *method* `input_degree()`, berfungsi input data jenjang studi, dan *method* `delete_degree()` berfungsi menghapus data jenjang studi.

*Class model* merupakan *class* yang terkait dengan database. Beberapa Penjelasan *class model* :

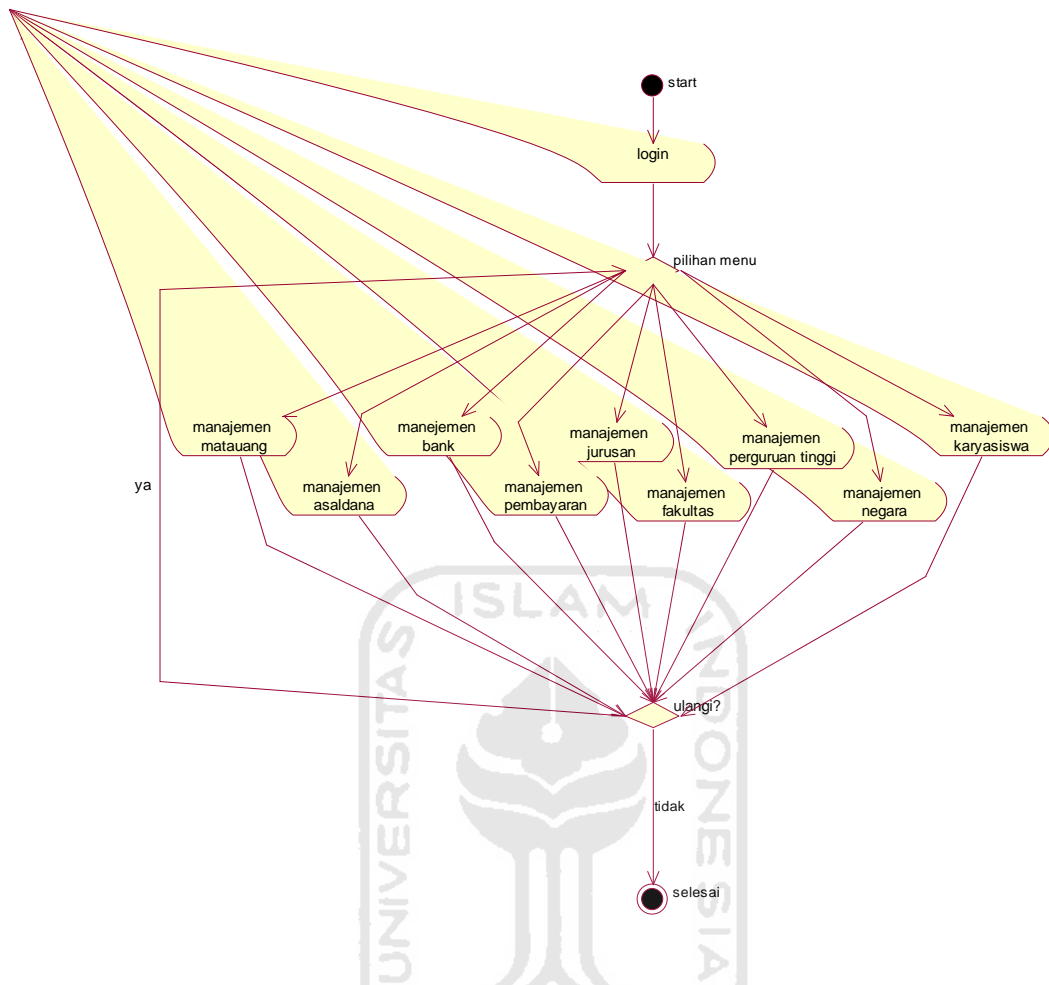
1. M\_karyasiswa : berisi *query* untuk data karyasiswa, misalnya : *method* cek\_dosen(), berfungsi melakukan cek di tabel karyasiswa, apakah nip yang diinputkan pada saat *input* usulan karyasiswa sudah ada atau belum, *method* getalls2(), berfungsi memfilter karyasiswa yang berjenjang s2, *method* count\_dosen(), berfungsi menghitung jumlah dosen.
2. M\_fakultas : berisi *query* untuk data fakultas, misalnya : *method* ambilfakultas(), berfungsi mengambil data fakultas, *method* selectfakultas(), berfungsi mengambil data fakultas dengan tambahan *limit* dan *offset* untuk penggunaan *paging*, *method* delfakultas(), proses menghapus data fakultas.
3. M\_negara : berisi *query* untuk data negara, misalnya : *method* ambilnegara(), berfungsi mengambil semua data negara, *method* insertnegara(), melakukan proses tambah data negara, *method* delnegara(), proses menghapus data negara.
4. M\_jurusan : berisi *query* untuk data jurusan, misalnya : *method* edit(), berfungsi mengambil data jurusan yang ber *id*, *method* insertjurusan(), berfungsi menambah data jurusan, *method* update(), proses *update* data jurusan.
5. M\_bank : berisi *query* untuk data bank, misalnya : *method* ambilbank(), berfungsi mengambil data bank, *method* insertbank(), berfungsi menambah data bank, *method* delbank(), menghapus data bank.
6. M\_pembayaran : berisi *query* untuk data pembayaran karyasiswa, misalnya : *method* lookup(), berfungsi mengambil data nama dosen pada proses auto complete, *method* updatepembayaran(), berfungsi melakukan update pada proses edit pembayaran, *method* getpembayaranid(), melihat riwayat pembayaran berdasarkan id dosen / nip.
7. M\_golongan : berisi *query* untuk data golongan pegawai, misalnya : *method* ambilgolongan(), berfungsi mengambil data golongan, *method* insertgolongan(), berfungsi menambah data golongan, *method* delgolongan(), menghapus data golongan.

8. *M\_pt* : berisi query untuk data perguruan tinggi, misalnya : *method* *ambilpt()*, berfungsi mengambil data perguruan tinggi, *method* *selectpt\_negara()*, berfungsi filter data perguruan tinggi berdasarkan negara, *method* *selectstatuspt()*, memfilter data perguruan tinggi yang berstatus dalam atau luar negeri.
9. *M\_matauang* : berisi query untuk data matauang, misalnya : *method* *ambilmatauang()*, berfungsi mengambil matauang, *method* *insertmatauang()*, berfungsi menambah data matauang, *method* *delmatauang()*, menghapus data matauang.
10. *M\_reminder* : berisi query untuk reminder, misalnya : *m\_reminder()* berisi query untuk proses reminder biaya living cost dan reminder kelulusan, *getpassword()*, berfungsi mengambil password lama di database, digunakan untuk proses validasi ketika admin melakukan ganti password, *gantipass()*, berfungsi mengganti password.
11. *M\_degree* : berisi query untuk data jenjang studi, misalnya : *method* *ambilmatauang()*, berfungsi mengambil data matauang, *method* *insertmatauang()*, berfungsi menambah data matauang, *method* *delmatauang()*, menghapus data matauang.

### 3.3.1.3 Activity Diagram

*Activity diagram* adalah sebuah diagram yang menggambarkan aktivitas-aktivitas pada sistem yang akan dibangun. *Activity diagram* akan menggambarkan aliran aktivitas dalam sistem yang dirancang, bagaimana aliran berawal, keputusan yang mungkin terjadi dan bagaimana aktivitas itu berakhir.

*Activity diagram* keseluruhan aplikasi dapat dilihat pada gambar 3.4.



**Gambar 3. 4** Activity diagram sistem informasi karyawan

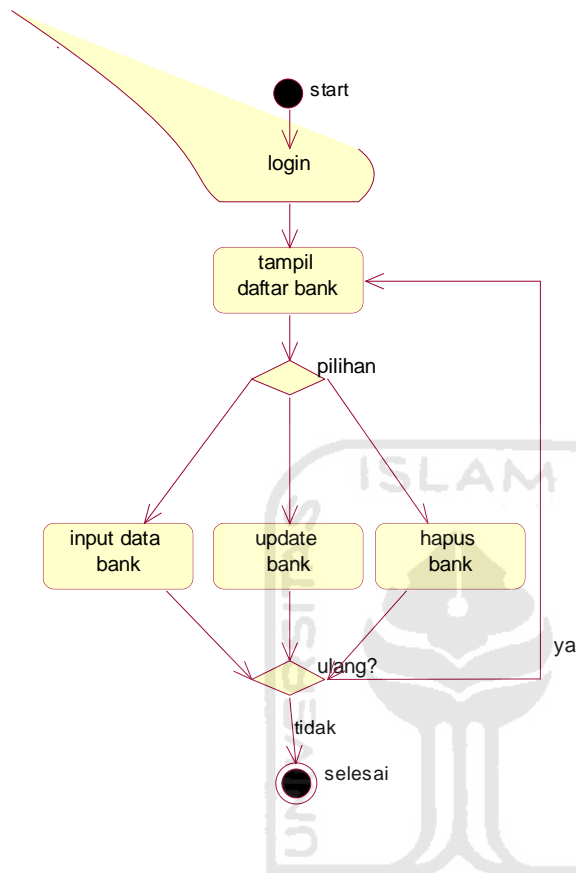
Secara keseluruhan, aktivitas pada aplikasi sistem informasi karyawan dimulai dengan proses login terlebih dahulu. Setelah login, proses selanjutnya adalah memilih aktivitas yang diinginkan seperti manajemen bank, karyawan, input biaya atau aktivitas lainnya. Setelah aktivitas tersebut, user dapat memilih aktivitas lainnya atau logout untuk selesainya aktivitas.

Diagram yang digambarkan di atas, salah satu komponen pembentuknya adalah *activity state* yaitu menggambarkan suatu aktivitas yang masih dapat dipecah kemabali menjadi aktivitas yang lebih kecil lagi sehingga aktivitas tersebut tidak dapat diturunkan kembali. *Activity state* digambarkan dengan notasi bulat lonjong/elips.

Berikut adalah gambaran aktivitas yang telah dipecah menjadi aktivitas yang lebih kecil:

- a. Manajemen bank

Aktivitas manajemen bank dapat diturunkan menjadi aktivitas-aktivitas yang lebih kecil. *Activity diagram* manajemen bank dapat dilihat pada gambar 3. 5.



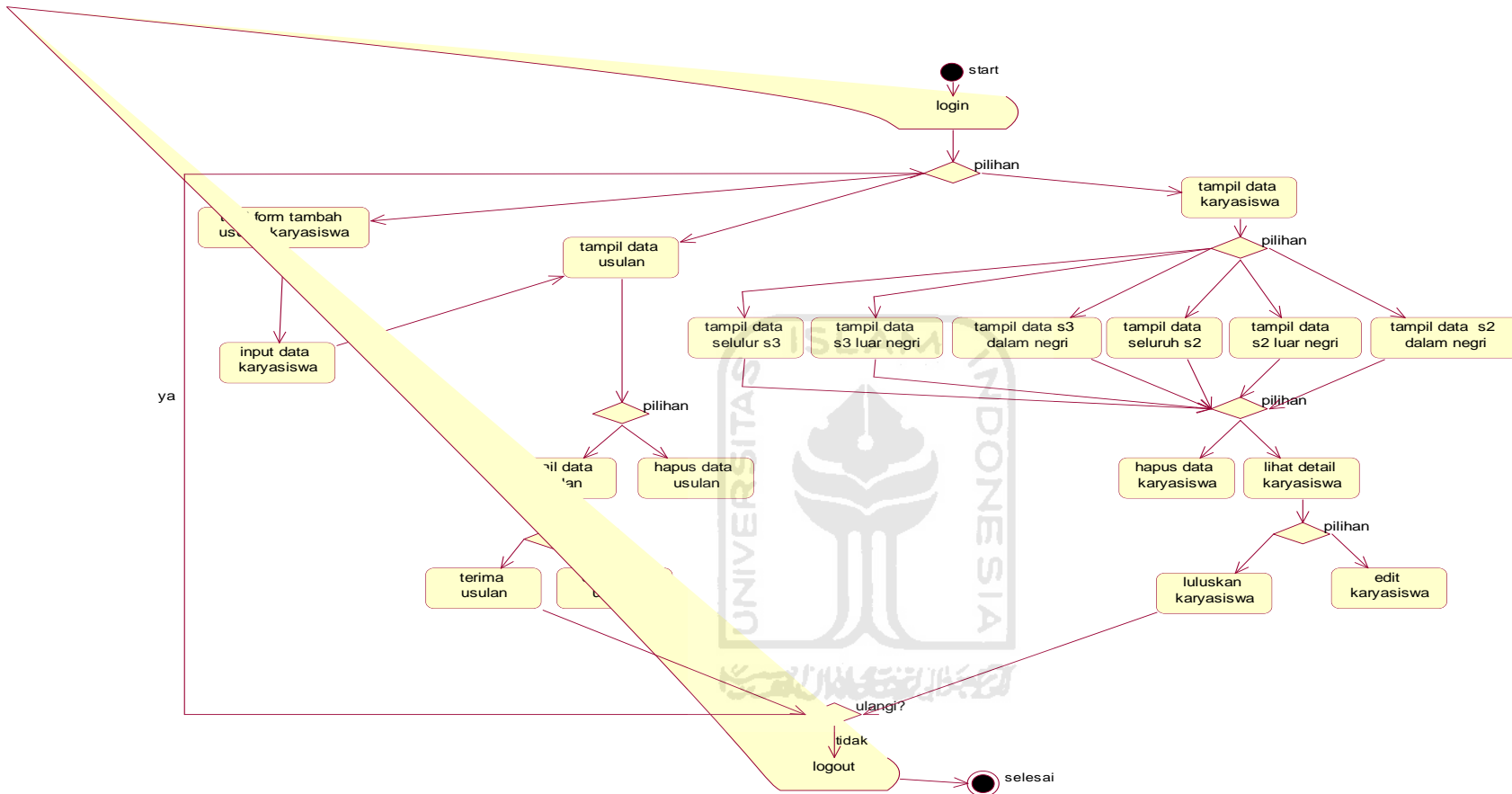
**Gambar 3.5** *Activity diagram* manajemen bank

Untuk aktivitas manajemen lainnya, seperti manajemen fakultas, jurusan, negara, asal dana dan perguruan tinggi, gambar *activity diagram*-nya sama dengan manajemen bank, hanya perubahan kata bank diganti menjadi kata fakultas, jurusan negara, asal dana, dan perguruan tinggi.

b. Manajemen karyasiswa

*Activity diagram* manajemen karyasiswa dapat dilihat secara detail pada gambar 3.6.

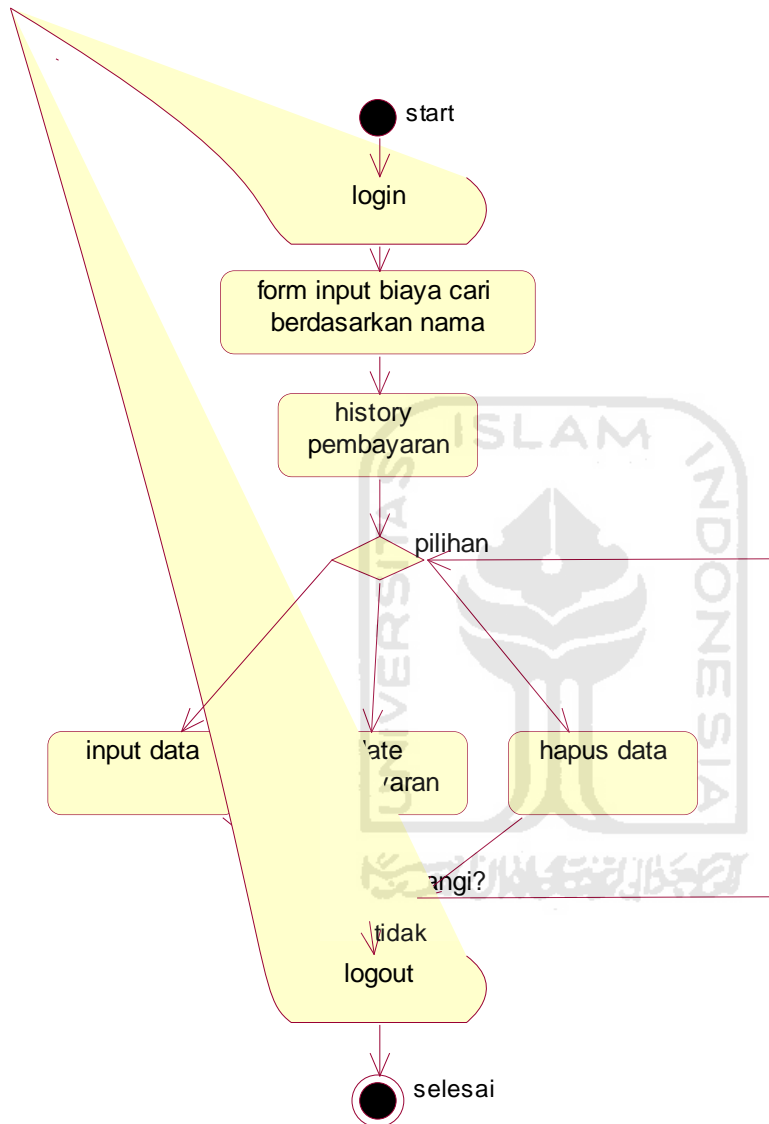




**gambar 3.6 Activity diagram manajemen karyawan**

c. Manajemen biaya

*Activity diagram* manajemen karyawan dapat dilihat secara detail pada gambar 3.7.



Gambar 3.7 *Activity diagram* manajemen biaya

### 3.3.2 Struktur Tabel

Pada sistem informasi mahasiswa, terdapat 16 tabel. Berikut struktur tabel keseluruhan.

### 1. Tabel t\_admin

Tabel ini digunakan untuk menyimpan data admin yang berisi *username* dan *password* admin. Struktur tabel dapat dilihat pada tabel 3.2.

**Tabel 3.2 Spesifikasi tabel t\_admin**

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Default</i>	Ekstra	Deskripsi
<i>Username</i>	<i>Varchar(20)</i>	<i>Primary key</i>			Berisi kumpulan karakter unik
<i>Password</i>	<i>Varchar(100)</i>				Berisi <i>password</i> admin

### 2. Tabel t\_asaldana

Tabel ini digunakan untuk menyimpan data asal dana. Struktur tabel dapat dilihat pada tabel 3.3.

**Tabel 3.3 Spesifikasi tabel t\_asaldana**

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Default</i>	Ekstra	Deskripsi
<i>Id_adaldana</i>	<i>Int(11)</i>	<i>Primary key</i>		<i>Auto_increment</i>	Kode unik asaldana
<i>asaldana</i>	<i>Varchar(20)</i>				Nama asal dana berasal

### 3. Tabel t\_bank

Tabel ini digunakan untuk menyimpan data bank. Struktur tabel dapat dilihat pada tabel 3.4.

**Tabel 3.4 Spesifikasi tabel t\_bank**

<i>Field</i>	<i>Type</i>	<i>Constrai nt</i>	<i>Default</i>	Ekstra	Deskripsi
--------------	-------------	------------------------	----------------	--------	-----------

<i>Id_bank</i>	<i>Int(11)</i>	<i>Primary key</i>		<i>Auto_increment</i>	Kode unik bank
bank	<i>Varchar(20)</i>				Nama bank

#### 4. Tabel t\_degree

Tabel ini digunakan untuk menyimpan data *degree* atau jenjang studi. Struktur tabel dapat dilihat pada tabel 3.5.

**Tabel 3.5 Spesifikasi tabel t\_degree**

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Default</i>	Ekstra	Deskripsi
<i>Id_degree</i>	<i>char(2)</i>	<i>Primary key</i>			Kode unik degree
degree	<i>Varchar(20)</i>				Nama degree/tingkatan, misalnya s2 atau s3

#### 5. Tabel t\_dosen

Tabel ini digunakan untuk menyimpan data dosen. Struktur tabel dapat dilihat pada tabel 3.6.

**Tabel 3.6 Spesifikasi tabel t\_dosen**

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Default</i>	Ekstra	Deskripsi
Nip	<i>Varchar(20)</i>	<i>Primary key</i>			Nomor induk pegawai
Nama	<i>Varchar(30)</i>				Nama dosen
Kelamin	<i>set('lk', 'pr')</i>				Jenis kelamin
Tempat_lahir	<i>Varchar(20)</i>				Kota tempat lahir
Tgl_lahir	<i>Date</i>				Tanggal

					kelahiran
Alamat	<i>Text</i>				Alamat tempat tinggal dosen saat ini
Telpon	<i>Varchar(20)</i>				Nomor telpon rumah
Hp	<i>Varchar(20)</i>				Nomor <i>handphone</i>
Email	<i>Varchar(30)</i>				Alamat email
Rekening	<i>Varchar(20)</i>				Rekening dosen
Jabatan	<i>Varchar(30)</i>				Jabatan dosen di UII
Id_fakultas	<i>char(7)</i>				Kode unik fakultas
Id_jurusan	<i>Int(11)</i>				Kode unik jurusan
Id_golongan	<i>char(5)</i>				Kode unik golongan

#### 6. Tabel t\_fakultas

Tabel ini digunakan untuk menyimpan data fakultas. Struktur tabel dapat dilihat pada tabel 3.7.

**Tabel 3.7 Spesifikasi tabel t\_fakultas**

<i>Field</i>	<i>Tipe</i>	<i>Constraint</i>	<i>Default</i>	<i>Ekstra</i>	<i>Deskripsi</i>
Id_fakultas	<i>char(7)</i>	<i>Primary key</i>			Kode unik fakultas
fakultas	<i>Varchar(20)</i>				Nama fakultas

#### 7. Tabel t\_golongan

Tabel ini digunakan untuk menyimpan data golongan pegawai. Struktur tabel dapat dilihat pada tabel 3.8.

**Tabel 3.8 Spesifikasi tabel t\_golongan**

<i>Field</i>	<i>Tipe</i>	<i>Constraint</i>	<i>Default</i>	<i>Ekstra</i>	<i>Deskripsi</i>
Id_golongan	<i>char(5)</i>	<i>Primary key</i>		<i>Auto_increment</i>	Kode unik golongan
golongan	<i>Varchar(20)</i>				Nama golongan

#### 8. Tabel t\_karyasiswa

Tabel ini digunakan untuk menyimpan data karyasiswa. Struktur tabel dapat dilihat pada tabel 3.9.

**Tabel 3.9 Spesifikasi tabel t\_karyasiswa**

<i>Field</i>	<i>Tipe</i>	<i>Constraint</i>	<i>Default</i>	<i>Ekstra</i>	<i>Deskripsi</i>
Id_karyasiswa	<i>Int(11)</i>	<i>Primary key</i>		<i>Auto_increment</i>	Kode unik karyasiswa
Nip	<i>Varchar(20)</i>				Nomor induk pegawai
No_mhs	<i>Varchar(20)</i>				Nomor mahasiswa di tempat karyasiswa
Tgl_masuk	<i>date</i>				Tanggal masuk karyasiswa
Status_karyasiswa	<i>set('aktif', 'cuti', 'do', 'lulus')</i>				Kode unik status karyasiswa (aktif, cuti atau <i>drop out</i> )

Id_pt	<i>char(20)</i>				Kode unik perguruan tinggi
Id_degree	<i>char(2)</i>				Kode unik degree/tingkatan
Id_negara	<i>char(3)</i>				Kode unik Negara
Status_usulan	<i>set('0', '1')</i>		'0'		Status karyasiswa, 0 untuk usulan, 1 diterima
Id_jurusan	<i>Int(11)</i>				Kode unik jurusan

#### 9. Tabel t\_jurusan

Tabel ini digunakan untuk menyimpan data nama jurusan di lingkungan UII. Struktur tabel dapat dilihat pada tabel 3.10.

**Tabel 3.10 Spesifikasi tabel t\_jurusan**

<i>Field</i>	<i>Tipe</i>	<i>Constraint</i>	<i>Default</i>	<i>Ekstra</i>	<i>Deskripsi</i>
Id_jurusan	<i>Int(11)</i>	<i>Primary key</i>		<i>Auto_increment</i>	Kode unik jurusan
Id_fakultas	<i>char(7)</i>				Kode unik fakultas
Nama_jurusan	<i>Varchar(20)</i>				Nama jurusan di lingkungan UII

#### 10. Tabel t\_matauang

Tabel ini digunakan untuk menyimpan data matauang tiap negara. Struktur tabel dapat dilihat pada tabel 3.11.

Tabel 3.11 Spesifikasi tabel t\_matauang

<i>Field</i>	<i>Tipe</i>	<i>Constraint</i>	<i>Default</i>	Ekstra	Deskripsi
Id_matauang	<i>Char(3)</i>	<i>Primary key</i>			Kode unik matauang
Matauang	<i>Varchar(20)</i>				Nama matauang seperti rupiah, dolar amerika dan lainnya
Id_negara	<i>char(3)</i>				Kode unik negara

## 11. Tabel t\_negara

Tabel ini digunakan untuk menyimpan data negara. Struktur tabel dapat dilihat pada tabel 3.12.

Tabel 3.12 Spesifikasi tabel t\_negara

<i>Field</i>	<i>Tipe</i>	<i>Constraint</i>	<i>Default</i>	Ekstra	Deskripsi
Id_negara	<i>char(3)</i>	<i>Primary key</i>			Kode unik negara
negara	<i>Varchar(20)</i>				Nama negara

## 12. Tabel t\_pembayaran

Tabel ini digunakan untuk menyimpan data pembayaran *living cost*. Struktur tabel dapat dilihat pada tabel 3.13.



Tabel 3.13 Spesifikasi tabel t\_pembayaran

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Default</i>	<i>Ekstra</i>	<i>Deskripsi</i>
Id_pembayaran	<i>Int(11)</i>	<i>Primary key</i>		<i>Auto_increment</i>	Kode unik pembayaran
Nip	<i>Varchar(20)</i>				Nomor induk pegawai
Jumlah	<i>Int(11)</i>				Jumlah kiriman
Rekening	<i>Varchar(30)</i>				Nomor rekening
Id_matauang	<i>char(3)</i>				Kode unik matauang
Id_bank	<i>Int(11)</i>				Kode unik bank
Id_asaldana	<i>Int(11)</i>				Kode unik asaldana
Tgl_kirim	<i>Date</i>				Tanggal pengiriman
Tgl_akhir	<i>Date</i>				Tanggal sampai dengan pengiriman, tanggal ini yang dijadikan dasar dari <i>reminder</i>

## 13. Tabel t\_pt

Tabel ini digunakan untuk menyimpan data perguruan tinggi. Struktur tabel dapat dilihat pada tabel 3.14.

Tabel 3.14 Spesifikasi tabel t\_pt

<i>Field</i>	<i>Tipe</i>	<i>Constraint</i>	<i>Default</i>	Ekstra	Deskripsi
Id_pt	<i>varchar(20)</i>	<i>Primary key</i>			Kode unik perguruan tinggi
Nama_pt	<i>Varchar(50)</i>				Nama perguruan tinggi
Alamat	<i>Text</i>				Alamat perguruan tinggi
Telpon	<i>Varchar(20)</i>				Telpon perguruan tinggi
Kota	<i>Varchar(30)</i>				Kota perguruan tinggi
Status_pt	<i>Char(2)</i>				'dn' untuk perguruan tinggi dalam negeri, dan 'ln' untuk luar negeri

## 14. Tabel t\_kurs

Tabel ini digunakan untuk menyimpan data kurs matauang asing terhadap rupiah. Struktur tabel dapat dilihat pada tabel 3.15.

Tabel 3.15 Spesifikasi tabel t\_kurs

<i>Field</i>	<i>Tipe</i>	<i>Constraint</i>	<i>Default</i>	Ekstra	Deskripsi
Id_kurs	<i>int(11)</i>	<i>Primary key</i>		<i>Auto_increment</i>	Kode unik kurs

Id_matauang	<i>char(3)</i>				Id matauang
Kurs	<i>Int(11)</i>				Kurs matauang asing terhadap rupiah

### 15. Tabel t\_jurusan\_usulan

Tabel ini digunakan untuk menyimpan data jurusan usulan karyasiswa. Struktur tabel dapat dilihat pada tabel 3.16.

**Tabel 3.16** Spesifikasi tabel t\_jurusan\_usulan

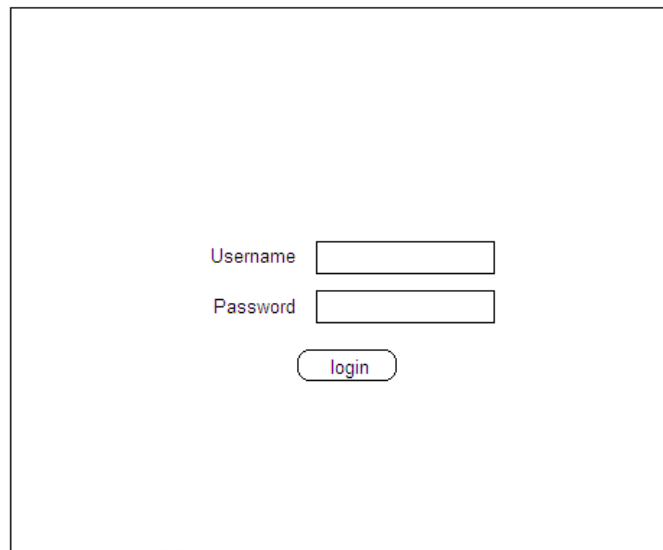
<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Default</i>	<i>Ekstra</i>	<i>Deskripsi</i>
Id_jurusan	<i>int(11)</i>	<i>Primary key</i>		<i>Auto_increment</i>	Kode unik jurusan usulan
jurusan	<i>varchar(50)</i>				Nama jurusan usulan

### 3.3.3 Rancangan Antarmuka

Rancangan antarmuka sistem, secara *template* dibagi menjadi tiga tampilan. Yaitu, halaman login, halaman home dan halaman tampil dan edit data.

#### 3.3.3.1 Halaman Login

Pada halaman ini, *user* akan diminta *password* untuk dapat mengakses halaman admin. Rancangan halaman login dapat dilihat pada gambar 3.8.

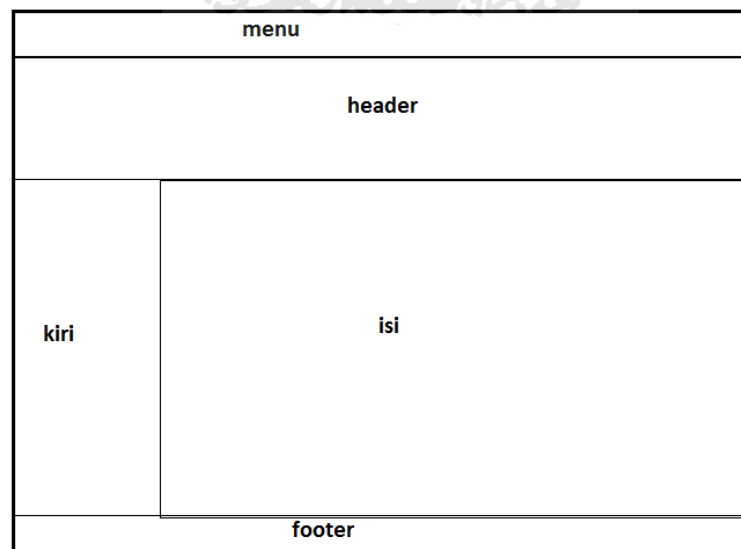


A wireframe of a login page. It features a central area with two input fields: 'Username' and 'Password'. Below these fields is a rounded rectangular button labeled 'login'.

**Gambar 3.8 Rancangan halaman login**

### 3.3.3.2 Halaman Home

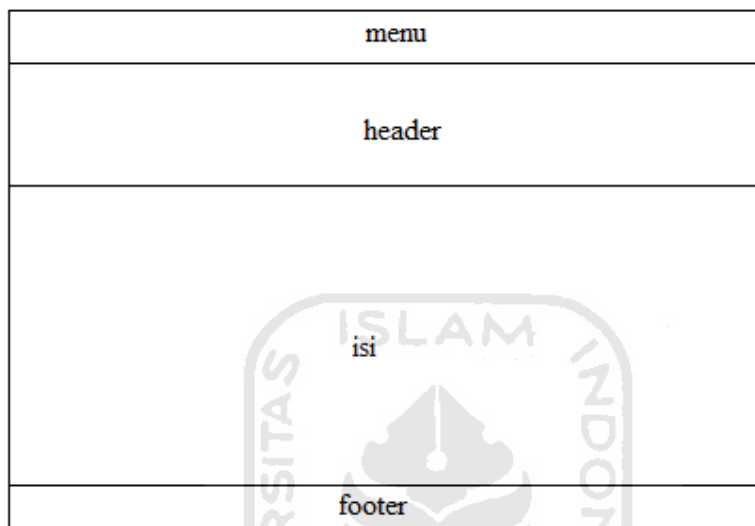
Halaman ini, terbagi menjadi lima bagian. Yang pertama, *header*, menu, isi, kiri dan *footer*. Bagian *header* tempat peletakan gambar, logo universitas dan informasi identitas aplikasi. Bagian menu adalah tempat menu-menu aplikasi, seperti lihat karyasiswa, fakultas dan lain-lain. Bagian isi menampilkan data jumlah karyasiswa. Dan bagian kanan, tempat *reminder* yang berhubungan dengan pembayaran dan kalender. Terakhir, bagian *footer* berisi informasi pengguna aplikasi. Lebih jelasnya dilihat pada gambar 3.9.



**Gambar 3.9 Rancangan halaman home**

### 3.3.3.3 Halaman Tampil dan Edit Data

Halaman ini, terbagi menjadi empat bagian. Sama seperti halaman home, halaman tampil dan edit data tidak menyertakan bagian kiri, dikarenakan untuk memperluas area kerja sehingga data yang disajikan terlihat rapi. Lebih jelasnya dilihat pada gambar 3.10.



Gambar 3.10 Rancangan Halaman Tampil dan Edit Data

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Batasan Implementasi

Untuk menjaga konsistensi aplikasi, dibuat batasan implementasi. Batasan-batasan implementasi tersebut adalah:

1. Pesan *reminder* akan memunculkan nama dosen, satu minggu sebelum tanggal akhir pengiriman biaya *living cost* sebelumnya.
2. Program cetak berdasarkan *form* bank BNI.
3. Aplikasi untuk satu *user* (admin).

#### 4.2 Implementasi

Tahap implementasi sistem merupakan kelanjutan dari tahapan perencanaan. Oleh karena itu, implementasi harus berdasarkan dari perencanaan yang telah dibuat sebelumnya.

#### 4.2.1 Implementasi Rancangan Antarmuka

Hasil implementasi rancangan antarmuka berdasarkan rancangan sebelumnya, yaitu halaman login, halaman admin dan halaman tampil dan edit data. Antarmuka halaman login dapat dilihat pada gambar 4.1.



Gambar 4.1 Halaman login

Dan hasil antarmuka untuk halaman home dapat dilihat pada gambar 4.2.

Gambar 4.2 Halaman home

Pada gambar di atas, sudah sesuai dengan rancangan awal yaitu, terdiri dari lima bagian. Bagian *header*, menu, isi, kiri dan terakhir *footer*.

Hasil implementasi halaman tampil dan edit data dapat dilihat pada gambar 4.3.

Gambar 4.3 Halaman tampil dan edit data

#### 4.2.2 Implementasi Program

Implementasi program dilakukan dengan pengujian aplikasi berdasarkan urutan fungsi aplikasi. Pengujian dimulai dengan memasukkan data usulan

karyasiswa, terima usulan, pembayaran biaya *living cost*, *reminder*, cetak laporan dan terakhir status karyasiswa menjadi lulus.

### 1. *Input* data usulan karyasiswa

Pada proses ini, terdapat dua kelompok data dalam satu *form*. Yang pertama adalah data diri dosen dan yang kedua adalah data untuk usulan karyasiswa. Lebih jelasnya dapat dilihat pada gambar 4.4

The screenshot shows the 'Tambah Usulan Karyasiswa' form. The top navigation bar includes 'HOME', 'KARYASISWA', 'USULAN KARYASISWA', 'PEMBAYARAN', 'DATA MASTER', 'DOSEN', 'LULUSAN', and 'ACCOUNT'. The page title is 'Sistem Informasi Karyasiswa'. The form is titled 'Tambah Usulan Karyasiswa' and is divided into two main sections: 'Data Diri Dosen' and 'Data Usulan Karyasiswa'.

**Data Diri Dosen**

- Nama : faqih
- Nip / Nik : 06622333 (\*)Required
- Tmp\_lahir : yogyakarta
- Tgl\_lahir : 02 maret 1988
- Kelamin :  Laki-Laki  Perempuan
- Fakultas : fakultas teknologi industri (\*)Required
- Jurusan : Teknik Informatika (\*)Required
- Jabatan : dosen
- Golongan : golongan ID (\*)Required
- Email : young.sufi@yahoo.com
- Alamat Rumah : jalan kaliurang km 14.5 nglanjaran sardonoharjo sleman yogyakarta
- Telpon Rumah : 0274-222333
- Handpone : 081222333444
- Rekening : 06523157

**Data Usulan Karyasiswa**

- Negara : thailand (\*)Required
- Perguruan Tinggi : Universitas Sains dan Teknologi Asia (\*)Required
- Degree : s2 (\*)Required
- Jurusan Yang Dipilih : teknik informatika (\*)Required
- Thn\_masuk : 2011
- Dana : DAAD

Submit

Divisi Pengembangan Sumber Daya Manusia (SDM)  
Universitas Islam Indonesia

**Gambar 4.4** Form tambah data usulan karyasiswa



Setelah selesai proses pengisian data, dilanjutkan dengan menekan tombol submit. Data akan ditampilkan pada daftar usulan karyasiswa, lihat gambar 4.5.

NO	NAMA	FAK	NEGERA/KOTA	PT	BEASISWA	JS	ACTION
1	Faqih	FTI	Thailand /	Universitas Sains Dan Teknologi Asia	Teknik Informatika	S2	Detail Hapus

Divisi Pengembangan Sumber Daya Manusia (SDM)  
Universitas Islam Indonesia

**Gambar 4.5 Data usulan karyasiswa**

## 2. Terima usulan karyasiswa

Dilanjutkan proses terima usulan karyasiswa, dengan mengklik detail pada kolom action. Pada halaman detail usulan karyasiswa perhatikan gambar 4.6, terdapat dua pilihan, yaitu tolak usulan yang berarti menghapus data usulan karyasiswa dari daftar usulan dan terima usulan.

UNIVERSITAS ISLAM INDONESIA

HOME | KARYASISWA | USULAN KARYASISWA | PEMBAYARAN | DATA MASTER | DOSEN | LULUSAN | ACCOUNT

Sistem Informasi Karyasiswa

Karyasiswa -> Detail

**Faqih**

Nip : 06622333  
 Kelamin : Lk  
 Alamat : Jalan Kaliurang Km 14.5 Nglangran Sardonoharjo Sleman Yogyakarta  
 TTL : Yogyakarta 1988-03-02  
 Jabatan : Dosen  
 Fakultas : Fakultas Teknologi Industri  
 Jurusan : Teknik Informatika  
 Email : Young.Sufi@Yahoo.Com  
 Telpon : 0274-222333  
 Hp : 081222333444  
 Rekening : 06523157  
[Edit](#)

**Usulan Karyasiswa**

Negara : Thailand  
 Perguruan Tinggi : Universitas Sains Dan Teknologi Asia  
 Kota :  
 Konsentrasi : Teknik Informatika  
 JS : Strata Dua  
 Tahun Masuk : 2011  
 Dana :  
[Edit](#)

[Tolak Usulan](#) [Terima Usulan](#)

Divisi Pengembangan Sumber Daya Manusia (SDM)  
 Universitas Islam Indonesia

Gambar 4.6 Detail karyasiswa

Apabila tombol terima usulan diklik, data otomatis masuk dalam daftar karyasiswa, lihat gambar 4.7.

UNIVERSITAS ISLAM INDONESIA

HOME | KARYASISWA | USULAN KARYASISWA | PEMBAYARAN | DATA MASTER | DOSEN | LULUSAN | ACCOUNT

Sistem Informasi Karyasiswa

Karyasiswa -> Semua Data Karyasiswa

Kata Kunci  Berdasarkan

[Excel](#)

NO	NAMA	FAK	NEGARA / KOTA	PT	BEASISWA	JS	ACTION
1	Aaa	FE	Australia /Melbourne	Victoria University	Ekonomi Pembangunan	S2	<a href="#">Detail</a> <a href="#">Hapus</a>
2	Faqih	FTI	Thailand /	Universitas Sains Dan Teknologi Asia	Teknik Informatika	S2	<a href="#">Detail</a> <a href="#">Hapus</a>

[Jumlah 2 data]

Divisi Pengembangan Sumber Daya Manusia (SDM)  
 Universitas Islam Indonesia

Gambar 4.7 Daftar karyasiswa

### 3. *Input biaya living cost dan buku*

Pada proses *input living cost*, *user admin* akan diminta mengetikkan nama dosen pada sebuah *form*, dan sistem akan membantu memberikan pilihan nama nama dosen berdasarkan karakter awal inputan. Lebih jelasnya dapat dilihat pada gambar 4.8.



UNIVERSITAS ISLAM INDONESIA

HOME | KARYASISWA | USULAN KARYASISWA | PEMBAYARAN | DATA MASTER | DOSEN | LULUSAN | ACCOUNT

Sistem Informasi Karyasiswa

Input Biaya Living Cost

Nama Dosen : f

Nip : faqih

submit

Divisi Pengembangan Sumber Daya Manusia (SDM)  
Universitas Islam Indonesia

**Gambar 4.8** *Input nama dosen pada proses living cost*

Setelah melakukan submit, akan ditampilkan riwayat pembayaran dan juga *form input* biaya, lihat pada gambar 4.9.

UNIVERSITAS ISLAM INDONESIA | HOME | KARYASISWA | USULAN KARYASISWA | PEMBAYARAN | DATA MASTER | DOSEN | LULUSAN | ACCOUNT

Sistem Informasi Karyasiswa

Riwayat Biaya Living Cost :: faqih=> Fakultas Teknologi Industri

NO	TGL KIRIM	SAMPAI BUKU	LC	RUPIAH	BANK	ASALDANA	ACTION
Total 0							

Input Biaya LC dan Buku

**Input Living Cost**

Nip: 06622333

Biaya LC: 2500

Biaya Buku: 400

Matauang: united states of ame (\*)

Bank: Bank Rakyat Indonesia (\*)

Asal Dana: BPPS (\*)

Tanggal Kirim: 2011-02-27

Sampai Tanggal: 2011-03-03

Input

Divisi Pengembangan Sumber Daya Manusia (SDM)  
Universitas Islam Indonesia

**Gambar 4.9 Form biaya dan riwayat pembayaran**

Setelah menginputkan data, data akan ditampilkan dalam daftar pembayaran, lihat gambar 4.10.

UNIVERSITAS ISLAM INDONESIA | HOME | KARYASISWA | USULAN KARYASISWA | PEMBAYARAN | DATA MASTER | DOSEN | LULUSAN | ACCOUNT

Sistem Informasi Karyasiswa

Pembayaran -> Daftar Penerima Living Cost

Excel PDF

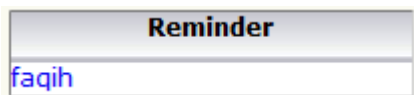
NO	NAMA	FAK	TGL KIRIM	TGL BATAS	BUKU	LC	RUPIAH	BANK	ASALDANA	ACTION	
1	Aaa	FE	26/Feb/2011	09/Mar/2011	500	Usd 254600000	Usd Rp 2342324600000	Bank Negara Indonesi	BPPS	<a href="#">Pdf Edit Hapus</a>	
2	Faqih	FTI	27/Feb/2011	03/Mar/2011	400	Usd 2500	Usd Rp 266800000	Bank Rakyat Indonesi	BPPS	<a href="#">Pdf Edit Hapus</a>	
Total							Rp 2342351280000				

Divisi Pengembangan Sumber Daya Manusia (SDM)  
Universitas Islam Indonesia

**Gambar 4.10 Daftar penerima living cost**

#### 4. Reminder

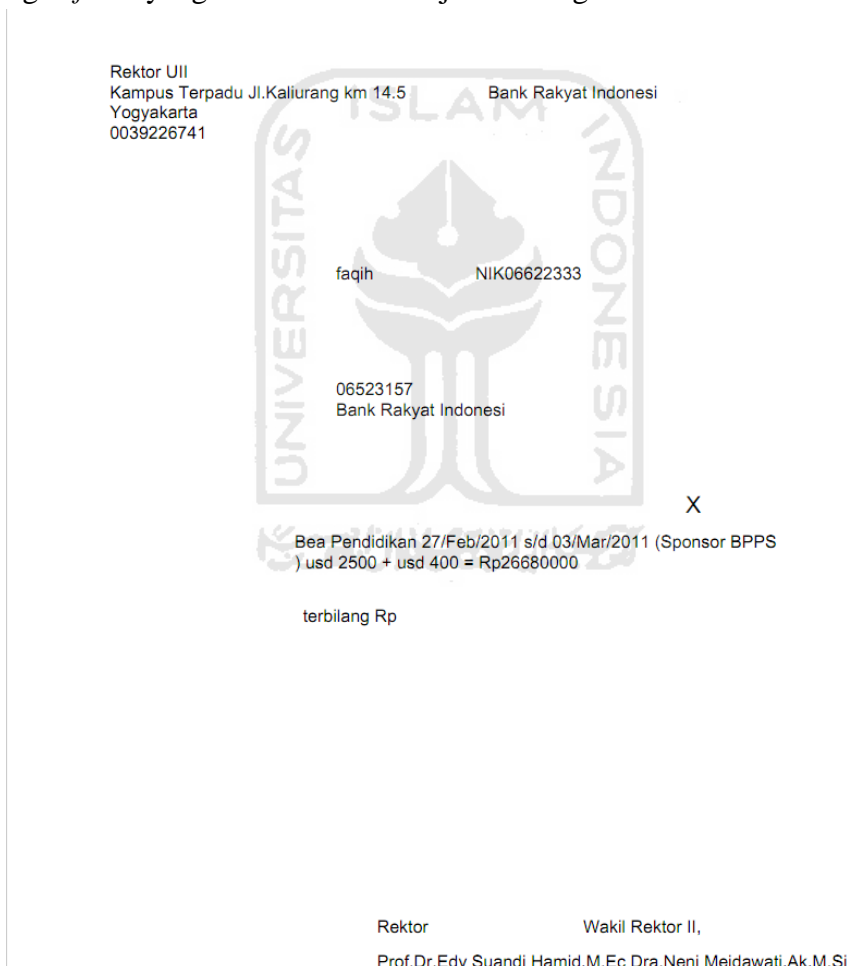
Tujuh hari sebelum batas tanggal akhir, nama dosen yang bersangkutan akan muncul pada kotak reminder, lihat gambar 4.11.



**Gambar 4.11 Reminder**

## 5. Cetak laporan

Pada cetak laporan, data akan ditampilkan dalam format file pdf yang sudah diukur sesuai *form* dari bank, sehingga apabila dilakukan print, data akan tepat dengan *form* yang sudah ada. Lebih jelas lihat gambar 4.12.



**Gambar 4.12 Hasil cetak laporan**

## 6. Karyasiswa lulus

Proses terakhir, apabila dosen telah menyelesaikan studinya, dilakukan perubahan data dengan menjadikan status karyawan dosen berstatus lulus. Menjadikan status lulus dengan cara mengklik detail pada daftar karyawan, lihat gambar 4.7, kemudian akan tampil halaman detail karyawan seperti gambar 4.13.

The screenshot shows the 'Sistem Informasi Karyawan' interface. The header includes the Universitas Islam Indonesia logo and navigation links: HOME, KARYASISWA, USULAN KARYASISWA, PEMBAYARAN, DATA MASTER, DOSEN, LULUSAN, ACCOUNT. The main content area is titled 'Karyawan -> Detail Karyawan' and displays the following information:

Faqih	
Nip	: 06622333
Kelamin	: Lk
Alamat	: Jalan Kaliurang Km 14.5 Nglanjaran Sardonoharjo Sleman Yogyakarta Indonesia Pertiwi Jayaselalu
TTL	: Yogyakarta 1988-03-02
Jabatan	: Dosen
Fakultas	: Fakultas Teknologi Industri
Jurusan	: Teknik Informatika
Email	: Young.Sufi@Yahoo.Com
Telpon	: 0274-222333
Hp	: 081222333444
<a href="#">Edit</a>	
Karyasiswa	
Negara	: Thailand
Perguruan Tinggi	: Universitas Sains Dan Teknologi Asia
Kota	:
Konsentrasi	: Teknik Informatika
JS	: Strata Dua
Tahun Masuk	: 2011
Dana	:

At the bottom of the form, there is a blue button labeled 'Lulus'. Below the button, the text reads: 'Divisi Pengembangan Sumber Daya Manusia (SDM) Universitas Islam Indonesia'.

**Gambar 4.13** Klik lulus untuk menjadikan status lulus

Dosen akan terdaftar dalam halaman lulusan, lihat gambar 4.14.



Gambar 4.14 Daftar lulusan

### 4.3 Evaluasi Kinerja Sistem

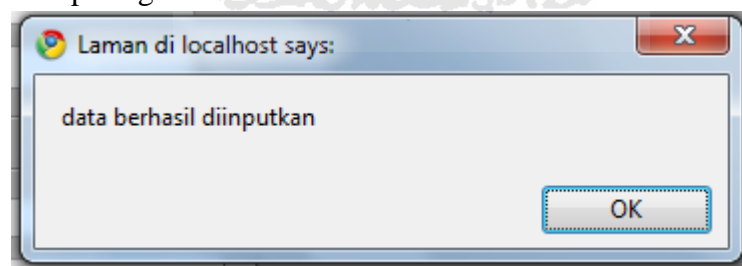
Evaluasi kinerja sistem atau proses pengujian dilakukan untuk mengetahui apakah sistem yang dibuat sudah sesuai dengan yang diharapkan atau belum.

#### 4.3.1 Pengujian Normal

Pengujian normal adalah perlakuan terhadap sistem dengan perlakuan normal, dengan *meninputkan* data sesuai dengan *requirement* sistem.

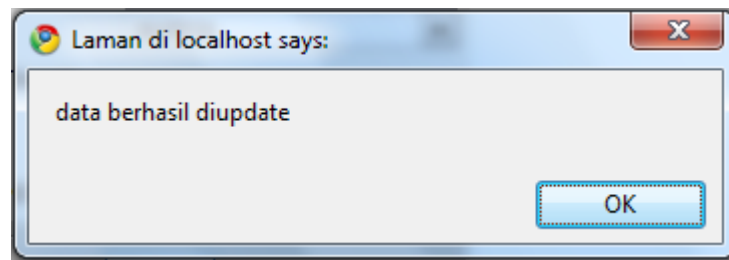
1. Manajemen *input*, *edit* dan *delete*.

Pesan yang ditampilkan ketika *user* *meninputkan* data sesuai *requirement* sistem adalah seperti gambar 4.15.



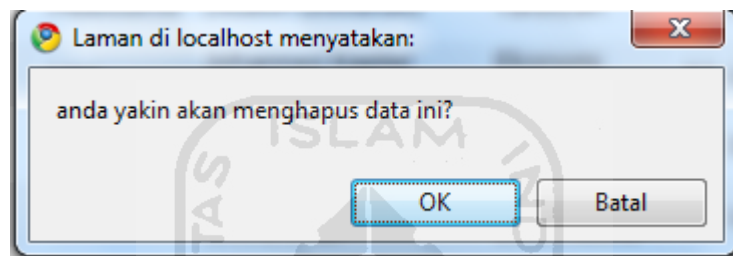
Gambar 4.15 Pesan inputan berhasil

Pesan yang ditampilkan ketika *user* mengedit data sesuai *requirement* sistem adalah seperti gambar 4.16.



Gambar 4.16 Pesan berhasil update

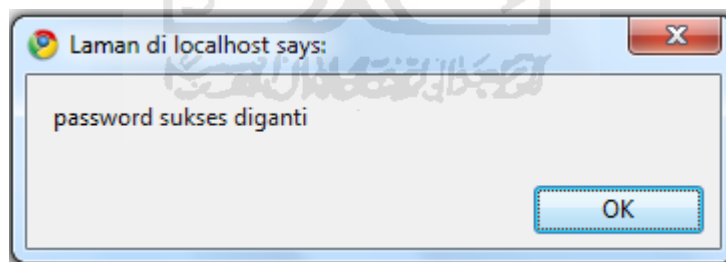
Pesan yang ditampilkan ketika *user* akan menghapus data adalah seperti gambar 4.17.



Gambar 4.17 Pesan akan menghapus data

2. Ganti password *admin*.

Pesan yang ditampilkan ketika *user* akan menghapus data adalah seperti gambar 4.18.



Gambar 4.18 Pesan sukses ganti password

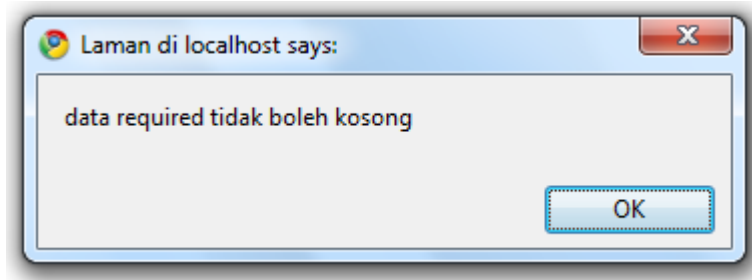
### 4.3.2 Pengujian Tidak Normal

Pengujian normal adalah perlakuan terhadap sistem dengan perlakuan tidak normal, dengan menginputkan data tidak sesuai dengan *requirement* sistem.

1. Data *requirement* pada *form input* data dikosongkan.

Akan muncul pesan tidak boleh kosong, apabila data pada form bertandakan “(\*)” yang berarti data required seperti pada gambar 4.19.

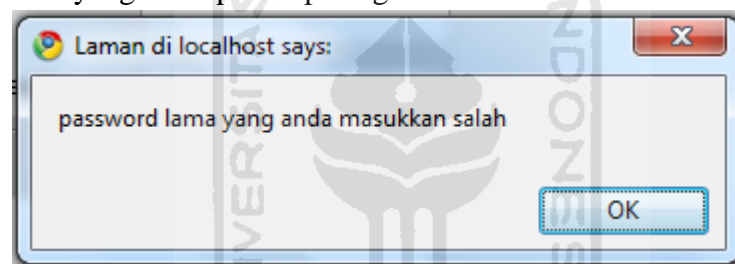




**Gambar 4.19** Pesan data required apabila tidak diisi

## 2. Kesalahan pada manajemen password admin.

Pada menu ganti password admin, akan muncul pesan kesalahan apabila password lama admin yang dimasukkan tidak sesuai dengan password di database. Pesan yang ditampilkan pada gambar 4.20.



**Gambar 4.20** Password lama tidak sesuai

## 4.4 Perbandingan dan Peningkatan Sistem

Merujuk pada latar belakang masalah, disebutkan bahwa aplikasi pengolahan data karyasiswa sudah ada sebelumnya menggunakan dBIII Plus. Dengan dibuatnya aplikasi ini, perlu diketahui apa saja perbandingan dan peningkatan sistem yang didapat, sehingga didapat kesimpulan apakah aplikasi yang dibangun ini menjadi lebih bermanfaat atau tidak.

### 4.4.1 Perbandingan Sistem

Hasil perbandingan sistem diperoleh melalui wawancara dengan bapak Melan selaku *end user*. Data perbandingan sistem dibagi berdasarkan dua kategori, yaitu: fungsional dan tampilan. Lebih rinci, dilihat pada tabel 4.1.

Tabel 4.1 Tabel perbandingan sistem

Kategori	Sistem lama	Sistem baru
Fungsional	<p>2. Lebih lama pada saat melakukan pencarian data, karena harus mengetikkan <i>query database</i>.</p> <p>3. Untuk mengetahui dosen yang akan dikirim biaya <i>living cost</i> harus melihat <i>database</i> satu persatu.</p> <p>4. Saat cetak laporan, harus bekerja pada dua aplikasi. Memindahkan data pada <i>database</i> ke Word.</p>	<p>5. Lebih mudah dan cepat, karena hanya mengetikkan <i>keyword</i> dan kategori, kemudian tekan tombol cari.</p> <p>6. Nama dosen yang akan dikirim biaya <i>living cost</i> otomatis terdapat pada kotak reminder.</p> <p>7. Untuk cetak laporan, hanya mengklik pdf dan data sesuai dengan <i>form</i> dari bank.</p>
Tampilan	<p>8. Tampilan berbasis desktop.</p> <p>9. Tidak <i>user friendly</i>.</p> <p>10. Akan menampilkan data berurut ke bawah sesuai jumlah data.</p>	<p>11. Tampilan berbasi web</p> <p>12. Lebih <i>user friendly</i>.</p> <p>13. Data ditampilkan menggunakan teknik <i>pagging</i>.</p>

#### 4.4.2 Peningkatan Sistem

Berikut adalah peningkatan sistem baru terhadap sistem lama:

1. Pada sistem baru, sudah terdapat fungsi *reminder*. Yaitu, pesan pengingat untuk pengiriman biaya *living cost* dosen. Pesan akan muncul tujuh hari sebelum tanggal batas jatah biaya, berupa nama dosen yang bersangkutan.

2. Efisiensi pada cetak laporan.
3. Informasi didapat lebih cepat dan lebih nyaman dipandang.
4. Untuk *mengexport* data menjadi file excel tidak perlu mengtikkan *query-query*.



## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Simpulan

Dari pembuatan aplikasi sistem informasi manajemen karyawan dan kepegawaian UII menggunakan *framework* CodeIgniter, dapat disimpulkan bahwa:

1. Sistem akan mencatat biaya pengeluaran tiap dosen berupa jumlah kiriman, tanggal kirim dan tanggal akhir. Satu minggu sebelum batas tanggal akhir, sistem akan menampilkan nama dosen yang masa biaya pendidikannya akan habis, sehingga UII akan mempersiapkan pengiriman biaya pendidikan lanjutan.
2. Pada sistem terdapat cetak laporan yang apabila dilakukan *printout* sudah sesuai letaknya dengan *form* dari bank.
3. Sistem yang dibuat sudah dapat memenuhi *requirement* yang telah ditentukan dalam tahap perencanaan. *Requirement* tersebut adalah: manajemen karyawan, manajemen negara, manajemen fakultas, manajemen jurusan, manajemen perguruan tinggi, manajemen pembayaran, manajemen asal dana, manajemen bank dan manajemen golongan.

#### 5.2 Saran

Sistem ini mengolah data karyawan dan tidak terintegrasi dengan data karyawan dan kepegawaian, diharapkan nantinya sistem dapat dikembangkan secara keseluruhan dan dilakukan secara *online*.

## DAFTAR PUSTAKA

- [BAS10] Basuki, Awan Pribadi. *Membangun Web Berbasis PHP Dengan Framework Codeigniter*. Yogyakarta: Loko Media, 2010.
- [KRI03] Kristanto, Andi. *Perancangan Sistem Informasi dan Aplikasinya*. Gava Media: 2008.
- [WIS10] Wiswakarma, Komang. *9 Langkah Menjadi Master Framework Codeigniter*. Yogyakarta: Loko Media, 2010.



## PETUNJUK INSTALASI PROGRAM

1. ekstrak dan install xampplite-win32-1.7.3 pada folder MASTER
2. jalankan xampp, klik start apache dan mysql

## MENJALANKAN APLIKASI

1. copy folder SOURCE\sinkarsis ke xampp\htdocs (sesuai tempat instalasi xampp)
2. copy folder DATA\db\_sinkarsis ke xampp\mysql\data
3. buka browser apa saja
4. ketikkan di URL nya <http://localhost/sinkarsis>
5. username dan passordnya admin

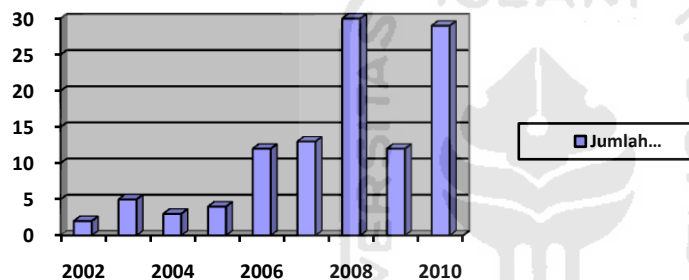


## RESUME

### SISTEM INFORMASI MANAJEMEN KARYASISWA DAN PELATIHAN SDM UII MENGGUNAKAN FRAMEWORK CODEIGNITER

#### Latar belakang

Data perkembangan karyasiswa dari tahun ke tahun dapat dikatakan meningkat. Data perkembangan karyasiswa pertahun dapat dilihat pada gambar 1.1.



Gambar 1.1 Perkembangan Karyasiswa Pertahun

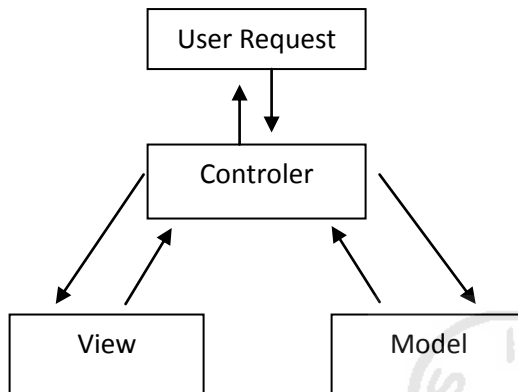
Sistem pengolahan data karyasiswa yang sudah ada, dikembangkan menggunakan dBIII Plus. Dalam implementasi aplikasi, masih terkendala beberapa masalah, seperti tidak adanya program cetak dan *reminder*. Setiap kali akan mencetak data, staf divisi pengembangan sumber daya manusia (SDM) bagian karyasiswa merasa kesulitan karena harus memindahkan data pada dua program yang berbeda, yaitu dBIII Plus dan MsWord. Begitu juga dengan waktu pengiriman biaya bagi dosen yang sedang menempuh pendidikan (karyasiswa), yaitu kesulitan untuk mengingat kapan dan siapa dosen yang sudah waktunya untuk dikirim biaya pendidikan dan keseharian (*living cost*). Karena untuk mengetahui hal itu harus melihat *database* atau menunggu telpon dari dosen yang bersangkutan.

Sistem dikembangkan menggunakan framework CodeIgniter, alasan menggunakan CodeIgniter adalah : *Open source*, Menggunakan konsep *Model-View-Controller* (MVC), Dokumentasi yang lengkap, dan Komunitas.

## Konsep *Model-View-Controller*(MVC)

MVC adalah sebuah konsep yang memisahkan antara aplikasi logika dengan presentasi pada halaman web.

Adapun gambaran konsep MVC dapat dilihat pada gambar 1.2.



**Gambar 1.2 Model-View-Controller**

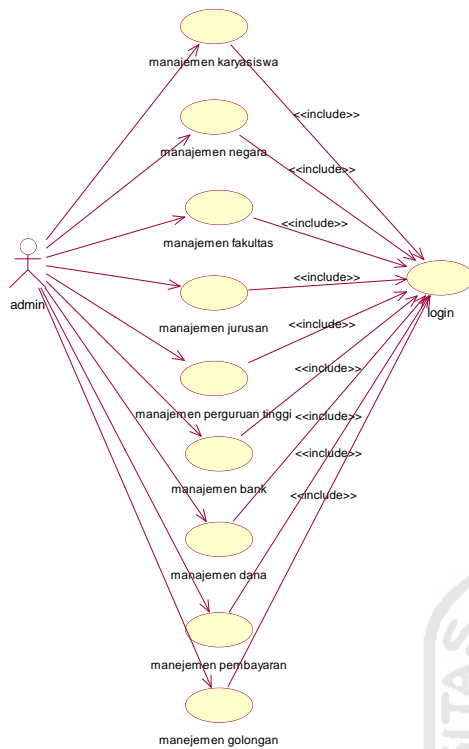
Ketika datang sebuah *user request* (permintaan pengguna), maka permintaan tersebut akan ditangani oleh *Controller*, kemudian *Controller* akan memanggil *Model* jika diperlukan operasi database. Hasil *query* dari *Model* akan dikembalikan ke *Controller*. *Controller* akan memanggil *View* dan mengkombinasikan dengan hasil *query Model*. Hasil akhir dari operasi ini akan ditampilkan ke *browser* yang dilihat oleh pengguna.

## Metodologi

Metode analisis yang digunakan untuk pembuatan perangkat lunak sistem informasi karyasiswa ini adalah menggunakan analisis berorientasi objek. Metode ini adalah metode yang memeriksa *requirement* (syarat/keperluan yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemukan dalam ruang lingkup permasalahan.

*Use case diagram* secara umum untuk sistem informasi karyasiswa, dapat dilihat pada gambar 1.3.

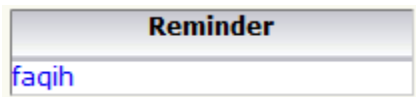




Gambar 1.3 Use case diagram sistem

## Hasil

1. Sistem akan mencatat biaya pengeluaran tiap dosen berupa jumlah kiriman, tanggal kirim dan tanggal akhir. Satu minggu sebelum batas tanggal akhir, sistem akan menampilkan nama dosen yang masa biaya pendidikannya akan habis, sehingga UII akan mempersiapkan pengiriman biaya pendidikan lanjutan.



2. Pada sistem terdapat cetak laporan yang apabila dilakukan *printout* sudah sesuai letaknya dengan *form* dari bank.

Rektor UII  
Kampus Terpadu Jl.Kaliurang km 14.5  
Yogyakarta  
0039226741

Bank Rakyat Indonesia

faqih NIK06622333

06523157  
Bank Rakyat Indonesia

X

Bea Pendidikan 27/Feb/2011 s/d 03/Mar/2011 (Sponsor BPPS  
) usd 2500 + usd 400 = Rp26680000

terbilang Rp

Rektor Wakil Rektor II,  
Prof.Dr.Edy Suandi Hamid,M.Ec Dra.Neni Meidawati,Ak,M.Si

