# APLIKASI CI-SMART SEBAGAI SISTEM INFORMASI UNTUK MENGELOLA HONOR DOSEN DENGAN PENDEKATAN BERORIENTASI OBJEK STUDI KASUS: STMIK AMIKOM YOGYAKARTA

**Nila Feby Puspitasari**

*Teknik Informatika STMIK AMIKOM Yogyakarta*
*Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55283*
*Email : nilafeby@amikom.ac.id*

**Abstrak**

Aplikasi Ci-Smart sistem informasi untuk mengelola honor dosen ini difungsikan untuk mengelola honor dosen dengan pendekatan berorientasi objek. Aplikasi ini dibuat berbasis web yaitu menggunakan bahasa pemrograman PHP yang dikombinasikan dengan *Code Ignitier* (CI) dan *Smarty*. CI sebagai *framework* tentunya menghasilkan aplikasi yang ringan dan memiliki akses yang cepat, sedangkan *Smarty* adalah *template engine* yang handal dan membuat web lebih tertata rapi dan menarik, dan *database* yang digunakan adalah MySQL. Metode yang digunakan dalam pembuatan aplikasi ini adalah berorientasi objek, perancangannya dibuat dengan menggunakan bahasa pemodelan Unified Modeling Language (UML).

***Kata kunci: Ci-Smart, Framework, Code Ignitier, Smarty.***

## 1. Pendahuluan

### 1.1 Latar belakang

Pemanfaatan Teknologi Informasi dan Komputer (TIK) memberikan angin segar terhadap kemudahan layanan akses informasi, sehingga penggunaan komputer sangat diminati oleh pengguna layanan informasi, karena memberikan pengaruh yang cukup besar terhadap perkembangan gaya hidup dan kebiasaan manusia di segala aspek kehidupan baik dunia bisnis, institusi pendidikan, instansi pemerintahan serta perkembangan ilmu pengetahuan yang semakin maju dan dinamis.

Untuk dapat mengakses informasi menggunakan teknologi komputer dibutuhkan Sumber Daya Manusia (SDM) yang mampu mengoperasikan komputer secara baik. STMIK AMIKOM Yogyakarta merupakan institusi pendidikan yang berbasis TIK yang akan menghasilkan SDM yang memiliki wawasan luas mengenai teknologi informasi, oleh karena itu untuk melayani kebutuhan mahasiswa, dosen, karyawan maupun civitas akademik lainnya harus dilakukan secara optimal yaitu dengan menyediakan dan mengembangkan Sistem Informasi yang terintegrasi di seluruh bagian yang berada di lingkungan civitas akademik guna mempermudah layanan informasi terkait.

Akan tetapi masih ditemui masalah dalam pengelolaan penghonoran dosen mengajar masih dilakukan secara *semi computerize* dalam proses kegiatannya sehingga memungkinkan tingkat kesalahan yang cukup *significan* dalam penghonoran dosen mengajar khususnya dalam pembuatan laporan.

Berdasarkan permasalahan yang telah diuraikan pada paragraf sebelumnya., maka peneliti merasa perlu untuk melakukan penelitian dengan merancang dan membuat sebuah aplikasi yang mengelola honor dosen mengajar dan honor ujian untuk memberikan solusi yang lebih baik untuk kedepannya.

### 1.2 Rumusan Masalah

Dari latar belakang yang telah dijelaskan, penulis merumuskan bahwa permasalahan yang di kaji adalah:

a. Bagaimana merancang dan membuat aplikasi ci-smart untuk mengelola honor dosen mengajar dan honor ujian/koreksi berbasis web di lingkungan STMIK AMIKOM Yogyakarta?
b. Bagaimana aplikasi pegelolaan honor dosen dapat menyediakan dan memberikan informasi yang cepat dan akurat untuk mengelola honor dosen mengajar dan honor ujian/koreksi di lingkungan STMIK AMIKOM Yogyakarta?

### 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

a. Aplikasi ini hanya mengelola honor dosen mengajar dan honor ujian/koreksi di lingkungan STMIK AMIKOM Yogyakarta.
b. Aplikasi ini dibuat berbasis web menggunakan bahasa pemrograman PHP, *Code Ignitier* sebagai *framework*, dan *Smarty* untuk *template web enginenya.*

### 1.4 Tujuan Penelitian

a. Memberikan solusi yang lebih baik dalam menangani permasalahan yang sering terjadi dalam mengelola honorarium dosen dilingkungan STMIk AMIKOM Yogyakarta

yang saat ini masih menggunakan semi *computerize*.

b. Membantu beban kerja khususnya SDM bagian terkait dengan pengelolaan honorarium dosen di lingkungan STMIK AMIKOM Yogyakarta.
c. .Menyediakan dan memberikan informasi yang cepat dan akurat dalam pengelolaan honorarium dosen di lingkungan STMIK AMIKOM Yogyakarta.

**1.4 Metodologi Penelitian**

Untuk menyelesaikan permasalahan yang mengarah pada penelitian ini, maka metode penelitian yang digunakan adalah :

**1.4.1 Metode Pengumpulan data**

**1.4.1.1.Studi literatur**

Dengan mempelajari informasi dan pemahaman literatur melalui berbagai media antara lain internet, buku bacaan, artikel, jurnal ilmiah, situs-situs yang terkait dengan penelitian ini.

**1.4.1.2 Wawancara**

Melakukan tanya jawab secara langsung kepada bagian terkait untuk mendapatkan kelengkapan data yang lebih detail.

**1.4.1.3 Observasi**

Mengadakan pengamatan secara langsung sehingga diperoleh gambaran secara jelas, serta untuk mengetahui sejauh mana informasi yang dibutuhkan terkait penelitian ini.

**1.4.2 Metode Rekayasa Perangkat Lunak**

Metodologi yang digunakan menggunakan pendekatan berorientasi objek yang akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Konsep ini dapat diterapkan pada tahapan analisis, perancangan, pemrograman dan pengujian perangkat lunak.

**1.4.3 Pemodelan Sistem dengan UML *(Unified Modelling Language)***

UML adalah sebuah bahasa standar yang digunakan untuk memodelkan dan mengembangkan sebuah perangkat lunak yang berupa grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan mendokumentasikan sebuah sistem berorientasi objek.

**1.5 Tinjauan Pustaka**

Then dkk (2009) mengadakan penelitian tentang analisis, perancangan dan implementasi perangkat lunak aplikasi penggajian berbasis web. Metodologi pengembangannya berbasis *System Development Life Cycle* (SDLC). Analisis dan desain secara rinci membahas mengenai fungsi dari perangkat lunak yang diimplementasikan berbasis open source antara lain XAMPP, PHP, MySQL dll.

Abdillah (2006) mengadakan penelitian tentang perancangan basis data sistem informasi penggajian. Metode perancangan basis datanya yang digunakan adalah secara konseptual dan *logical* dan secara fisik, fokus penelitiannya pada mekanisme penggajian dosen. perangkat lunak yang digunakan pada sistem informasi adalah Borland Delphi 6.0.

Mahdiana (2011) mengadakan penelitian tentang analisa dan rancangan sistem informasi pengadaan barang dengan metodologi berorientasi objek. Studi kasus nya adalah PT. Liga Indonesia. Adapun desain sistemnya menggunakan diagram yang terkandung pada *Unified Modellling Language* (UML).

Sihombing (2011) melakukan penelitian tentang pembangunan aplikasi Sistem Informasi Dosen Politeknik Telkom pada SmartPhone berbasis Android. Sistem Informasi ini menyajikan kehadiran dosen, jadwal mengajar, informasi gaji,. Aplikasi diimplementasikan pada hardware yang lebih portable, serta aplikasi lebih memiliki *stlye* yang sedang *in* di saat sekarang dengan basis androidnya.

**1.6 Landasan Teori**

**1.6.1 Pengertian Sistem Informasi**

Sistem Informasi merupakan kumpulan komponen yang saling berhubungan untuk mengolah *input* (data) menjadi *output* (informasi) sehingga dapat memenuhi kebutuhan pemakai[10]. Komponen-komponen utama dalam suatu sistem informasi berbasiskan komputer terdiri dari: 1) *Database*, 2) *Database software*, 3) Aplikasi *software*, 4) *Hardware* komputer termasuk media penyimpanan, dan 5) Personal yang menggunakan dan mengembangkan system[10].

**1.6.2 Metode** Berorientasi **Objek**

Menurut Mahyuzir[11] pendekatan berorientasi objek ini memiliki keunggulan dan mekanisme yang handal, yakni membuat penulisan pemrograman menjadi lebih sederhana, lebih kompak, lebih fleksibel tetapi dengan kemampuan yang jauh lebih baik.

Fokus utama metodologi ini pada objek, dengan melihat suatu sistem terdiri dari objek yang saling berhubungan. Objek dapat digambarkan sebagai benda, orang, tempat dan sebagainya yang mempunyai atribut dan metode. Metodologi terdiri dari pembuatan model dan domain aplikasi, kemudian menambahkan rincian implementasi pada saat pembuatan desain dari suatu sistem. Tahap-tahap metodologi berdasarkan *Sistem Development Life Cycle* (*SDLC*) digunakan dengan memperhatikan karakteristik khusus berorientasi objek[1] yaitu Analisis, Desain dan Implementasi.

Menurut Gora yang dipublikasikan oleh Abdillah [10], bahwa semua metodologi analisis dan perancangan berorientasi objek mempunyai kesamaan dalam hal objek, kelas, *inheritance*, dan *relationship*.
Menurut Yordan [12], ada 3 (Tiga) karakteristik kunci pendekatan berorientasi objek untuk pengembangan sistem yaitu;

1. Abstraksi yaitu mekanisme yang menyederhanakan realitas yang kompleks menjadi model yang lebih sederhana.

2. Enkapsulasi yaitu mekanisme yang menyembunyikan implementasi objek sehingga komponen lain tidak akan menyadari bagaimana data itu disimpan pada objek.
3. *Inheritance* yaitu mekanisme menurunkan atribut dan fungsi dari suatu superkelas ke subkelas dalam suatu hirarki.

### 1.6.3 Pemodelan Perangkat Lunak

#### 1.6.3.1 *Unified Modelling Language (UML)*

UML *( Unified Modelling Language )* adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat *blue print* atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain[3].

UML menyediakan cukup banyak diagram yang dapat membantu mendefinisikan sebuah aplikasi, diagram-diagram tersebut antara lain:

***a. Use Case***

Sebuah use case menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase *requirements*, model *use case* mengambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari input user dan respon-respon sistem.

**b. Aktor**

Sebuah aktor mencirikan suatu bagian *outside* user atau susunan yang berkaitan dengan user yang berinteraksi dengan sistem. Dalam model use case, aktor merupakan satu-satunya kesatuan eksternal yang berinteraksi dengan sistem.

***c. Activity Diagram***

*Activity diagram* adalah teknik untuk mendiskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya flowchart, akan tetapi perbedaanya dengan flowchart adalah activity diagram bisa mendukung periaku parallel sedangkan flowchart tidak bisa.

***d. Sequence Diagram***

*Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukan sejumlah contoh objek dan message (pesan) yang diletakan diantara objek-objek ini didalam *use case*. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama. Message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukan dengan *progress vertical.*

**e. *Class Diagram***

Class, dalam notasi UML digambarkan dengan kotak. Nama class menggunakan huruf besar diawal kalimatnya dan diletakan diatas kotak. Bila class mempunyai nama yang terdiri dari dua suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar.

#### 1.6.3.2 *Code Ignitier*

*Code Igniter* (CI) adalah salah satu framework php yang tangguh dan popular yang awalnya ditulis oleh Rick Ellis, pendiri dan CEO EllisLab.com, perusahaan yang mengembangkan *code igniter*. Saat ini, *code igniter* dikembangkan oleh komunitas dan disebarkan keseluruh dunia dengan lisensi bebas. *Code Igniter* tergolong *framework* dengan ukuran kecil dan cukup mudah di kuasai. CI juga memiliki manual yang tergolong lengkap. *Code Igniter* adalah sebuah *framework* PHP. *Framework* itu sendiri adalah suatu kerangka kerja yang berupa sekumpulan folder yang memuat file-file php yang menyediakan class libraries, helpers, plugins dan lainnya. *Framework* menyediakan konfigurasi dan teknik coding tertentu. Untuk menggunakan codeigniter dengan maksimal, diperlukan[13]:

- PHP 4.3.2 atau lebih baru.
- Web server, saya rekomendasikan Apache.
- MySQL, MySQLi, MS SQL, Postgre, Oracle, SQLite atau ODBC
  Struktur file CI

Struktur CI merupakan sebuah php *framework* yang berupa kumpulan folder dan file php, java script,css,txt dan file berbasis web lainnya dengan setting tertentu untuk menggunakannya dan menyediakan library dan helper yang dapat di manfaatkan di dalam pemrograman php. CI di jalankan under web dan harus dengan web server[13].

#### *1.6.3.3* Smarty

Smarty adalah mesin template untuk PHP. Lebih khusus, ia memfasilitasi cara yang bisa diatur untuk memisahkan logika aplikasi dan konten dari penampilannya. Ini jauh lebih baik dijelaskan dalam situasi di mana pemrogram aplikasi dan desainer template memainkan aturan yang berbeda, atau secara umum bukan orang yang sama. Salah satu aspek unik mengenai Smarty adalah kompilasi template. Ini berarti Smarty membaca file template dan membuat naskah PHP. Setiap template dapat memanfaatkan solusi *cache* kompilator PHP seperti eAccelerator, ionCube mmCache atau Zend Accelerator adalah beberapa diantaranya[14].

Beberapa fitur Smarty antara lain :

- Sangat cepat.
- Efisien karena pengurai PHP yang mengerjakan pekerjaan beratnya.
- Tidak ada kelebihan penguraian template, hanya sekali mengompilasi.

- Pintar mengenai rekompilasi hanya file template yang telah diubah.

## 2. Pembahasan

### 2.1 Analisis dan perancangan sistem

#### 2.1.1 Analisis Sistem yang sedang berjalan

Pengelolaan penghonoran dosen mengajar di STMIK AMIKOM Yogyakarta masih dilakukan secara *semi computerize*. Kegiatan penghonoran ini hanya dikelola oleh 2 (dua) orang saja yang masing-masing mempunyai beban kerja yang cukup padat. Hal ini memungkinkan tingkat kesalahan yang cukup *significan* dalam penghonoran dosen mengajar khususnya dalam pembuatan laporan (slip gaji). Dalam pengelolaan honor dosen, ada beberapa kegiatan yang dilakukan:

- Pengelolaan data dosen mengajar setiap semester.
- Menyediakan fasilitas pengaturan (setting) terhadap penggajian maupun penghonoran dosen mengajar dan koreksi ujian.
- Pengelolaan honor dosen mengajar dan honor ujian/koreksi.
- Pencetakan slip gaji dan honor dosen mengajar dan honor ujian/koreksi.

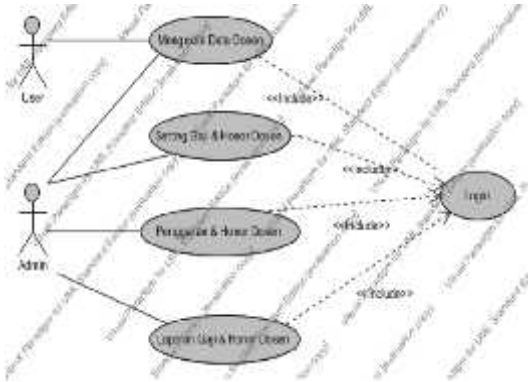
Dari deskripsi yang telah diuraikan sebelumnya, diharapkan dapat diminimalisir tingkat kesalahan dalam pengelolaan penghonoran dosen yang ada di STMIK AMIKOM Yogyakarta. Untuk itu peneliti membuat aplikasi pengelolaan honor dosen (simgaji) menggunakan *framework* Ci-Smart.

#### 2.1.2 Perancangan Sistem

Berdasarkan analisa sistem yang berjalan, maka penulis mengusulkan sistem yang dapat memecahkan masalah pengelolaan honor dosen dan akan ditampilkan dalam bentuk *use case diagram, activity diagram, sequence diagram dan class diagram.*

**a. Skenario *Use Case* sistem yang diusulkan**

Gambar menunjukkan Skenario Use Case sistem yang diusulkan, aktor yang terlibat adalah admin dan user. Proses yang terjadi adalah pengelolaan data dosen yang dilakukan oleh user, sedangkan setting gaji dan honor dosen, penggajian dan honor dosen serta laporan gaji dan honor dosen dilakukan oleh admin.




**Gambar 2.1** Skenario *Use Case* sistem yang diusulkan

**b. Skenario aplikasi yang diusulkan**

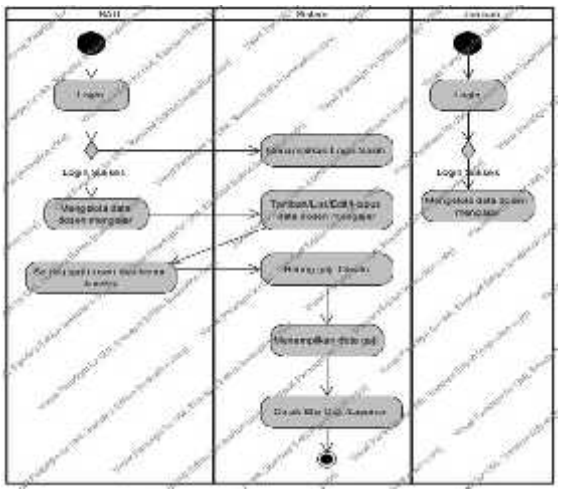
Skenario digunakan untuk memudahkan dalam menganalisa sebuah permasalahan yang akan kita gunakan pada tahapan selanjutnya dengan melakukan penilaian terhadap skenario yang ada. Tabel 2.1 menunjukkan skenario sistem yang diusulkan.

**Tabel 2.1** *Skenario* sistem *yang diusulkan*

| No | Aktor | Sistem |
|---|---|---|
| 1. | Pihak Jurusan mengelola data dosen mengajar. | Menyimpan data dosen mengajar |
| 2. | Bagian keuangan mensetting gaji dosen | Menyimpan hasil setting penggajian dosen |
| 3. | Bagian keuangan melakukan mekanisme penggajian. | |
| 4. | | Mencetak Slip gaji |
| 5. | Dosen menerima slip gaji | |

***c. Activity Diagram***

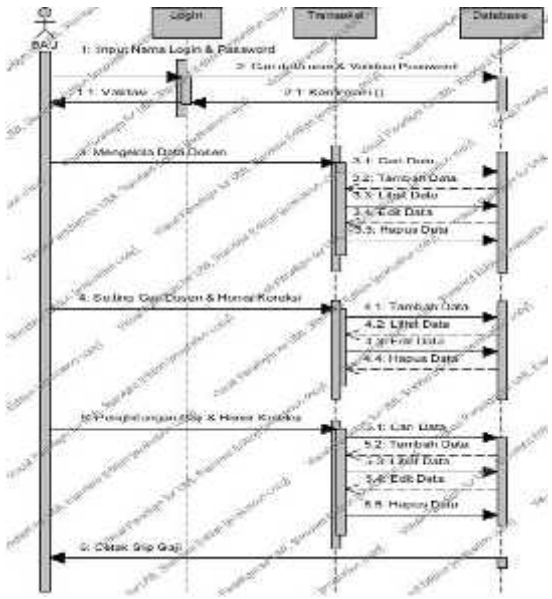
Pada bagian ini akan digambarkan alur fungsionalitas terhadap sistem penghonoran dosen yang akan diusulkan.



**Gambar 2.2** Skenario Activity Diagram sistem yang diusulkan

***d. Sequence Diagram***

*Sequence diagram* digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Gambar menunjukkan interaksi antar objek pada sequence diagram sistem yang diusulkan.




**Gambar 2.3** *Sequence Diagram* sistem yang diusulkan

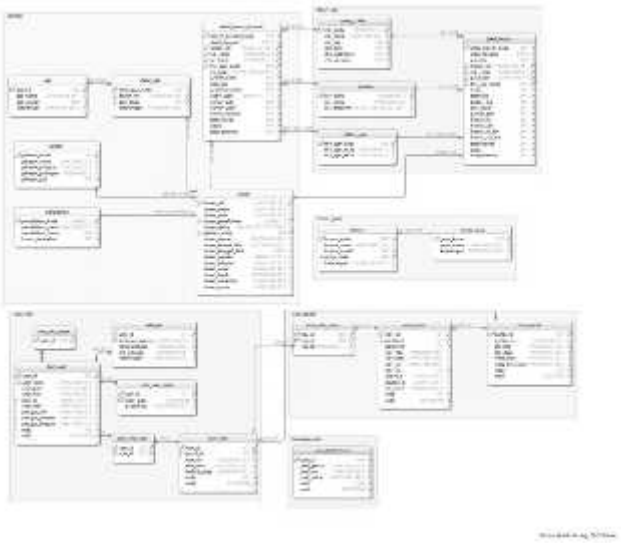
***e. Class Diagram***

Pada Gambar 2.4 menunjukkan keadaan (objek/atribut) pada sistem yang akan diusulkan.

**Gambar 2.4** Class Diagram sistem yang diusulkan

**2.1.3 Perancangan Database**

Semua data yang digunakan maupun yang dihasilkan oleh setiap operasi akan disimpan dalam tabel-tabel *database*. Untuk menjaga keefektifan dan efisiensi pemakai tabel serta menghindari *abnormaly* dalam proses manipulasi data, diperlukan struktur tabel yang baik. Pada gambar ada 22 (duapuluhdua) tabel yang saling berelasi antara lain: 1) Tabel dosen, 2) Tabel Jabatan. 3) Tabel Matakuliah, 4) Tabel tahun_ajar, 5) Tabel jurusan, 6) Tabel pendidikan, 7) Tabel gaji, 8) Tabel detail_gaji, 9) Tabel honor, 10) Tabel detail_honor, 11) Tabel honor_grub, 12) Tabel honor_koreksi, 13) Tabel petugas, 14) Tabel com_user_super, 15) Tabel com_user, 16) Tabel com_user_login, 17) Tabel com_role_user, 18) Tabel com_role, 19) Tabel com_menu, 20) Tabel com_role_menu, 21) Tabel com_portal, 22) Tabel com_preferences.



**Gambar 2.5** Relasi antar Tabel

**2.2 Implementasi**

**2.2.1 Perangkat Keras**

Pembuatan Aplikasi Ci-Smart sebagai Sistem Informasi Penghonoran Dosen ini menggunakan spesifikasi perangkat keras :

- NoteBook ASUS EeePC 1015 PX
- Processor Intel Atom CPU N570 @1.66 Ghz
- Memory 2 GB
- Harddisk 320 GB

**2.2.2 Perangkat Lunak**

Perangkat lunak yang digunakan untuk membuat aplikasi Ci-Smart sebagai sistem informasi penghonoran dosen adalah:

- XAMPP meliputi :
  - Bahasa Pemrograman : PHP 5.0
  - Database Server : MySQL 5.0
  - Web Service : Apache

**2.2.3 Design Interface :**

- Code Ignitier
- Smarty
- Axure RP Pro 5.6
- Visual Paradigm for UML 10 Trial

**2.3 Rancangan Antarmuka Aplikasi**

Pada aplikasi Ci-Smart yang dibuat terdapat 65 (enampuluhlima) rancangan antar muka aplikasi sistem yang diusulkan antara lain :

a. Rancangan Halaman Login
b. Rancangan Halaman Menu Utama
c. Rancangan Halaman Master Data Dosen
   - List Data Dosen
   - Tambah Data Dosen
   - Edit Data Dosen
   - Hapus Data Dosen
d. Rancangan Halaman Jabatan
   - List Data Jabatan
   - Tambah Data Jabatan
   - Edit Data Jabatan
   - Hapus Data Jabatan
e. Rancangan Halaman Matakuliah
   - List Data Matakuliah
   - Tambah Data Matakuliah
   - Edit Data Matakuliah
   - Hapus Data Matakuliah
f. Rancangan Halaman Tahun Ajaran
   - List Data Tahun Ajaran
   - Tambah Data Tahun Ajaran
   - Edit Data Tahun Ajaran
   - Hapus Data Tahun Ajaran
g. Rancangan Halaman Jurusan
   - List Data Jurusan
   - Tambah Data Jurusan
   - Edit Data Jurusan
   - Hapus Data Jurusan
h. Rancangan Halaman Pendidikan
   - List Data Tahun Pendidikan
   - Tambah Data Pendidikan
   - Edit Data Pendidikan
   - Hapus Data Pendidikan
i. Rancangan Halaman Setting Gaji
   - List Data Setting Gaji
   - Tambah Data Gaji
   - Edit Data Gaji
   - Hapus Data Gaji
j. Rancangan Halaman Setting Honor
   - List Data Setting Honor
   - Tambah Data Honor
   - Edit Data Honor
   - Hapus Data Honor
k. Rancangan Halaman Setting Honor Grup
   - List Data Setting Honor Group
   - Tambah Data Honor Group
   - Edit Data Honor Group
   - Hapus Data Honor Group
l. Rancangan Halaman Penggajian
   - List Data Gaji Pokok
   - View Data Gaji Pokok
   - Tambah Data Gaji Pokok
   - Edit Data Gaji Pokok
   - Hapus Data Gaji Pokok
m. Rancangan Halaman Honor Mengajar
   - List Data Honor Mengajar
   - View Data Honor Mengajar
   - Tambah Data Honor Mengajar
   - Edit Data Honor Mengajar
   - Hapus Data Honor Mengajar
n. Rancangan Halaman Honor Koreksi
   - List Data Honor Koreksi
   - View Data Honor Koreksi
   - Tambah Data Honor Koreksi
   - Edit Data Honor Koreksi
   - Hapus Data Honor Koreksi

**2.3.1 Rancangan halaman login**

Gambar 2.6 menampilkan rancangan halaman login yang merupakan halaman utama sebelum masuk ke menu utama. User memasukkan username dan password. Apabila terjadi kesalahan maka akan muncul validasi bahwa username dan password tidak terdaftar.

**Gambar 2.6** Rancangan Halaman Login

**2.3.2 Rancangan Halaman Menu Utama**

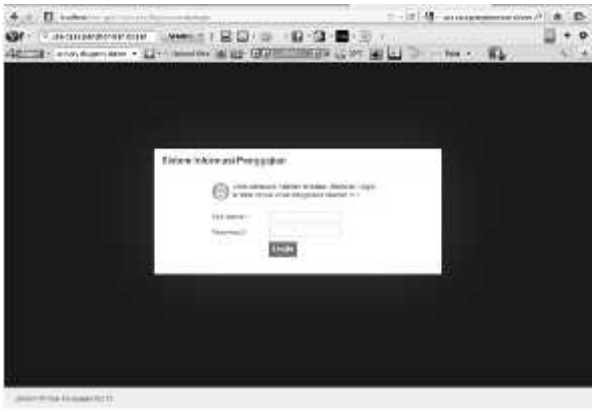
Setelah user memasukkan username dan password secara benar, maka akan tampil halaman menu utama sistem penggajian dosen yang dapat dilihat pada Gambar 2.7.

**Gambar 2.7.** Rancangan Halaman Menu Utama

Untuk rancangan halaman sub menu penggajian, setting penggajian dan master data tidak dapat penulis tampilkan.

### 2.4 Antar Muka Aplikasi

#### 2.4.1 Antar Muka Halaman Login



**Gambar 2.8** Antar Muka Halaman Login

#### 2.4.2 Antar Muka Halaman Menu Utama

**Gambar 2.9** Antar Muka Halaman Menu Utama

Untuk antar muka halaman sub menu penggajian, setting penggajian dan master data tidak dapat penulis tampilkan.

### 3. Kesimpulan

Dengan selesainya pembuatan aplikasi Ci-Smart sebagai Sistem Informasi untuk mengelola honor dosen (simgaji) yang berbasis web dengan metode pendekatan berorientasi objek yang dilakukan oleh peneliti diharapkan mampu memberikan solusi yang terbaik dan dapat membantu pihak keuangan maupun jurusan untuk menangani kegiatan pengelolaan data dosen mengajar, pengelolaan honor dan penggajian dosen mengajar dan koreksi ujian. Aplikasi ini tentunya menyediakan dan memberikan informasi yang cepat dan akurat dalam pengelolaan honorarium dosen di lingkungan STMIK AMIKOM Yogyakarta.

### Daftar Pustaka

[1] Sutopo, H.A. *"Analisis dan Desain Berorientasi Objek"* , J & J Learning. Yogyakarta, 2002.
[2] Yourdan, Edward., *"Object-Oriented System Design An Integrated Approach"*, Prentice Hall Inc. New Jersey, 1994.
[3] Munawar, Pemodelan Visual dengan UML,Penerbit GRAHA ILMU:Yogyakarta, 2005.
[4] Jogiyanto H.M, Sistem Informasi Berbasis Komputer Edisi 2, Penerbit Graha Ilmu,Yogyakarta, 1997.
[5] Witarto, Pendekatan Praktis Rekayasa Sistem Informasi di Sekitar kita, Penerbit Informatika, Bandung, 1999.
[6] Linda Marlinda, Konsep Basis Data, Penerbit Andi Publisher,Yogyakarta, 2006.
[7] Burch, J.G.., *"System, Analysis, Design, and Implementation"*, Boyd & Fraser Publishing Company, 1992.
[8] Jogiyanto, Analisis dan Disain Sistem Informasi, ANDI OFFSET Yogyakarta, 1990.
[9] John G. Burch, Jr, Felix R. Strater, Gary Grudnistski, "*Information Systems Theory and Practice, Second Edition*", John Wiley & Sons, 1979.
[10] LA Abdillah, *Jurnal Ilmiah MATRIK Vol.8 No.2, Agustus 2006:135-152.*
[11] Tauri D, Mahyuzir, "Pengantar Analisis dan Perancangan Perangkat Lunak", PT. Elex Media Komputindo Jakarta, 1991.
[12] Yourdan, *"Object-Oriented System Design An Integrated Approach"*, Prentice Hall Inc. New Jersey, 1994.
[13] EllisLab, *"Code Igniter User Guide"*, http://www.codeigniter.com, (diakses 1 Desember 2013).
[14] New Digital Group, *"Manual Smarty"*, http://www.smarty.net/docsv2/id/ (diakses 1Desember 2013).

### Biodata Penulis

***Nila Feby Puspitasari***, memperoleh gelar Ahli Madya (Amd) Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2003. Memperoleh gelar Sarjana (S.Kom) Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2005. Saat ini penulis sedang menempuh program Pasca Sarjana *Magister Computer Science* (MCs) di Universitas Gadjah Mada.. Saat ini menjadi Dosen di STMIK AMIKOM Yogyakarta.

***ISSN : 2302-3805***