
Penerapan LINQ (*Language Intergrated Query*) untuk Implementasi Sistem Pendukung Keputusan Evaluasi Kinerja Dosen (Studi kasus : FTI-UKSW Salatiga)

¹⁾M. A. Ineke Pakereng, ²⁾ Yos. Richard Beeh, ³⁾Anderson Hattu

Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60, Salatiga 50711, Indonesia
E-mail: ¹⁾inekep200472@yahoo.com, ²⁾yos.fti.uksw@gmail,
³⁾comahattu@yahoo.co.id

Abstract

Teacher evaluation is a very important evaluation method to determine whether a teacher is carrying out his/her responsibilities well or not. Information Technology Faculty of Satya Wacana Christian University Salatiga is one of faculties in Satya Wacana Christian University Salatiga which applies the teacher evaluation system. This research applies analytic hierarchy process method to evaluate teachers' work performance in Information Technology Faculty of Satya Wacana Christian University Salatiga. The result of this research is an application evaluating teachers' work performance which will become a parameter to determine whether or not a teacher is qualified.

Keywords : Analytic Hierachy Process, Language Intergrated Query

1. Pendahuluan

Evaluasi kinerja dosen yang dilakukan pada setiap akhir semester bertujuan untuk menilai kualitas seorang dosen dalam menjalankan tugasnya di Fakultas Teknologi Informasi Universitas Kristen Satya Wacana Salatiga (FTI-UKSW). Evaluasi kinerja dosen perlu dilakukan karena hasil dari evaluasi ini dapat digunakan untuk bahan pertimbangan dalam jabatan seorang dosen. Untuk mencapai jabatan tertentu biasanya seorang dosen harus memenuhi semua kriteria penilaian yang ada pada tri darma perguruan tinggi. Biasanya hasil akhir dari evaluasi kinerja dosen adalah nilai yang mempresentasikan kinerja kerja dari masing-masing dosen FTI-UKSW. Proses evaluasi kinerja dosen FTI-UKSW biasanya dilakukan dengan menggunakan perhitungan secara manual sesuai dengan kriteria yang telah ditetapkan. Selain menyita banyak waktu dan tenaga, sistem manual ini dinilai kurang efisien karena kemungkinan besar terjadi kesalahan dalam melakukan perhitungan hasil

akhir evaluasi. Selain itu juga dalam proses penyimpanan data tidak efektif karena data yang tersimpan di dalam *file* kemungkinan dapat terhapus. Selain kemungkinan terhapusnya dokumen evaluasi, dalam pencarian dokumen evaluasi juga memerlukan waktu yang cukup lama karena *user* harus mengecek setiap dokumen. Berdasarkan kelemahan yang ada pada sistem ini, maka dirasa perlu untuk membuat sebuah aplikasi sistem pendukung keputusan untuk membantu pihak FTI-UKSW dalam mengevaluasi kinerja kerja dosen. Penerapan LINQ merupakan pilihan yang tepat dalam menangani masalah pengelolaan data-data yang ada. Adanya aplikasi yang terkomputerisasi, maka diharapkan dapat membantu pihak FTI-UKSW dalam mengatur dan mengelola dokumen atau hasil evaluasi dengan baik. Dengan demikian penerapan LINQ mampu meningkatkan performa dari sistem itu sendiri. Selain dengan LINQ, penerapan SPK dapat membantu *user* untuk mengolah dan menganalisa hasil *survey* yang dilakukan oleh pihak FTI-UKSW. Penerapan metode AHP diharapkan membuat hasil evaluasi kinerja dosen dapat digunakan sesuai dengan kebutuhan yang ada. Selain itu, AHP sendiri merupakan salah satu metode SPK yang sangat cocok karena penilaian yang dilakukan tidak bersifat subyektif. Penggunaan komputer dalam menyelesaikan masalah tersebut dinilai sangat mempengaruhi tingkat kesalahan dalam memperoleh informasi sesuai dengan data yang diberikan.

2. Kajian Pustaka

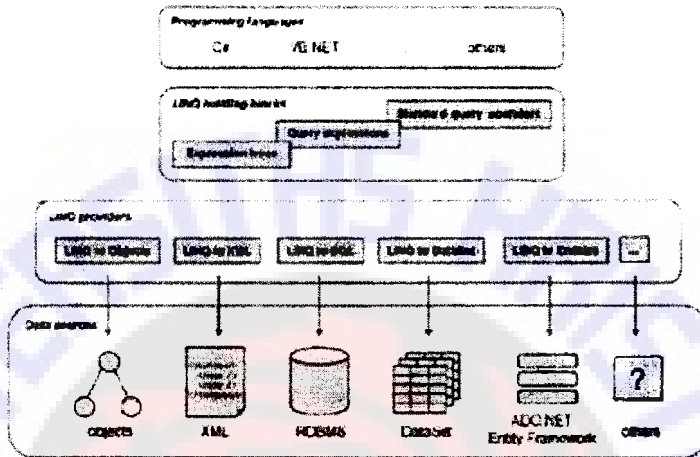
Sistem basisdata berisi kumpulan dari semua data bisnis yang dimiliki perusahaan, baik yang berasal dari transaksi sehari-hari, maupun data dasar (*master file*). Untuk keperluan DSS, diperlukan data yang relevan dengan permasalahan yang hendak dipecahkan melalui simulasi. Komponen kedua adalah Model Base atau suatu model yang merepresentasikan permasalahan ke dalam format kuantitatif (model matematika sebagai contohnya) sebagai dasar simulasi atau pengambilan keputusan, termasuk di dalamnya tujuan dari permasalahan (objektif), komponen-komponen terkait, batasan-batasan yang ada (*constraints*), dan hal-hal terkait lainnya. Kedua komponen tersebut untuk selanjutnya disatukan dalam komponen ketiga (*software system*), setelah sebelumnya direpresentasikan dalam bentuk model yang "dimengerti" komputer. Contohnya adalah penggunaan teknik RDBMS (*Relational Database Management System*), OODBMS (*Obyek Oriented Basis Data Managemen Sistem*) untuk memodelkan struktur data. Sedangkan MBMS (*Model Base Management System*) dipergunakan untuk merepresentasikan masalah yang ingin dicari pemecahannya. Entity lain yang terdapat pada produk DSS baru adalah DGMS (*Dialog Generation and Management System*), yang merupakan suatu sistem untuk memungkinkan terjadinya "dialog" interaktif antara komputer dan manusia (*user*) sebagai pengambilan keputusan. Rancangan tersebut diatas harus diimplementasikan dengan dukungan DBMS yang mendukung *query* bersarang, seperti SQL server dan *interbase* [4]. Informasi-informasi yang berhubungan dengan karyawan maupun jabatan dapat disimpan dalam suatu basis data, sehingga jika suatu saat diperlukan untuk proses *profile matching* dapat dilakukan dengan mudah lebih mudah dari pada dengan bentuk *hardcopy* [5]. Dalam pengolahan aplikasi

yang mendukung Sistem Pendukung Keputusan dibutuhkan sarana dan prasarana khususnya *software* yang dapat membantu dalam mengolah dan memberikan informasi seperti yang diharapkan. Beberapa dari definisi Sistem Pendukung Keputusan adalah suatu kumpulan model dasar dari prosedur- prosedur pengelolaan data dan penilaian untuk membantu seorang manajer dalam membuat keputusan [1]. SPK sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi yakni sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen SPK lain), sistem pengetahuan (*repository* pengetahuan domain masalah yang ada pada SPK sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan). Konsep-konsep yang diberikan oleh definisi tersebut sangat penting untuk memahami hubungan antara SPK dan pengetahuan[2]. Sistem Pendukung Keputusan merupakan suatu sistem informasi berbasis komputer (CBIS) yang interaktif, mampu beradaptasi dan secara fleksibel saling mempengaruhi, dimana sistem ini menggunakan aturan keputusan, model dan penggabungan model dasar dengan meliputi basisdata dan pengambilan keputusan berada didalamnya menuju pada suatu hasil tertentu, yang merupakan keputusan yang diterapkan dalam menyelesaikan masalah. Sistem Pendukung Keputusan mampu mendukung pengambilan keputusan yang kompleks dan menambah efektifitas[3].

Maksud dan tujuan dari adanya Sistem Pendukung Keputusan yaitu untuk mendukung pengambilan keputusan memilih alternatif pengambilan keputusan yang merupakan hasil pengelolaan informasi-informasi yang diperoleh/tersedia dengan menggunakan model-model pengambilan keputusan serta untuk menyelesaikan masalah-masalah bersifat terstruktur, semi terstruktur, dan tidak terstruktur. AHP menerapkan prinsip komparasi yang berarti membuat penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu yang dalam kaitannya dengan tingkat diatasnya. Penilaian ini merupakan inti dari AHP, karena akan berpengaruh terhadap prioritas beberapa elemen. Hasil dari penilaian ini akan tampak lebih baik bila disajikan dalam bentuk matriks yang dinamakan matriks *pairwise comparison* [6].

Hirarki fungsional dari AHP dapat memecahkan masalah kompleks yang mengambil kriteria cukup banyak, sehingga dapat dimanfaatkan sebagai model dalam sistem pendukung keputusan seleksi penerimaan karyawan yang mengambil banyak kriteria seleksi dan alternatif pelamar yang dicalonkan untuk diterima. Keputusan untuk menentukan calon pelamar mana yang akan diterima sebagai karyawan perusahaan menentukan kelangsungan hidup perusahaan itu sendiri sehingga diperlukan keputusan yang tepat dalam pemilihan, agar tujuan perusahaan dapat tercapai [7]. AHP mampu memberikan solusi yang tepat dalam pengambilan keputusan hirarki seperti yang dihadapi oleh kepala kantor. Keputusan yang diambil oleh kepala kantor dapat dipertanggungjawabkan dengan dukungan dari perhitungan yang dilakukan dengan AHP sebagai model dalam sistem pendukung keputusan [8]. Dalam pengolahan aplikasi yang mendukung SPK, dibutuhkan sarana dan prasarana khususnya *software* yang dapat membantu dalam mengolah dan memberikan informasi seperti yang diharapkan. Teknologi LINQ ini sendiri diperkenalkan pertama kali oleh Anders Hejlsberg dalam *Microsoft Profesional*

Developers Conference (PDC) tahun 2005, tujuannya adalah membuat standarisasi dan memudahkan *pattern* proses pengaksesan data. Dengan adanya *standart* pengaksesan data maka para *developer* atau *programmer* dapat melakukan cara yg sama dalam memproses beberapa format data (Basis Data, *XML* ataupun *Collection* data objek). Berikut diagram arsitektur LINQ secara global dalam proses pengaksesan data, seperti yang terlihat pada Gambar 1.



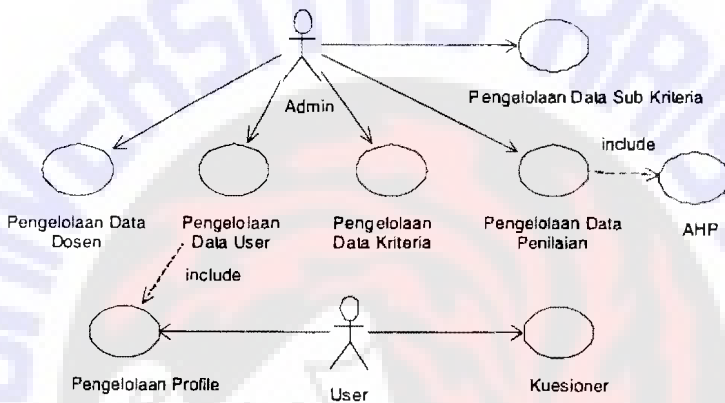
Gambar 1 Arsitektur LINQ

LINQ adalah fitur baru dalam bahasa pemrograman di .NET Framework. Pernyataan LINQ dituliskan menggunakan sintaks bahasa pemrograman dengan notasi yang mirip dengan pengaksesan objek menggunakan *class collection*. LINQ terbagi menjadi tiga bagian utama yakni 1) LINQ to Object adalah sekumpulan pustaka pemrograman yang mengandung sejumlah *Standard Query Operators* (SQO). SQO membantu mekanisme pengambilan data dalam memori melalui mekanisme antarmuka dengan *generic* pada antarmuka *IEnumerable*; 2) LINQ to ADO.NET merupakan sekumpulan pustaka pemrograman SQO yang memungkinkan komunikasi dengan basis data relasional. LINQ to ADO.NET terbagi lagi menjadi tiga bagian berdasar akses basis data yang diakses yakni, LINQ to SQL untuk pengaksesan ke SQL Server, LINQ to DataSet melakukan *query* terhadap Datasets yang kompatibel dengan semua pustaka ADO.NET, dan LINQ to Entities yang memungkinkan akses LINQ dengan menggunakan obyek bisnis; 3) LINQ to XML adalah sekumpulan pustaka pemrograman yang memungkinkan SQO bekerja dengan berkas XML. LINQ bekerja di atas bahasa pemrograman, atau dengan kata lain pernyataan-pernyataan LINQ akan dikompilasi oleh *compiler* bahasa pemrograman yang mendukungnya. Hasil kompilasi pernyataan LINQ berupa *Intermediate Language* (IL) yang nantinya diterjemahkan oleh *Common Language Runtime* (CLR) untuk mengakses sumber data yang diinginkan.

3. Metode Penelitian

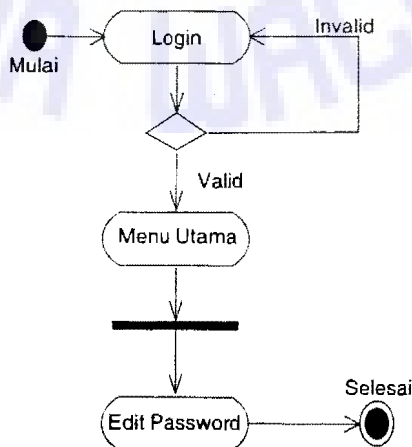
Metode penelitian yang dikembangkan dalam membuat sistem informasi adalah

prototyping model yang dimulai dengan pengumpulan bahan dan dalam hal ini penulis melakukan studi kepustakaan mengenai hal-hal yang berhubungan dengan *Analytic Hierarchy Process* dan *Language Integrated Query*, kemudian dilanjutkan dengan perancangan aplikasi, dan evaluasi *prototyping* sebagai langkah akhir dari *prototyping model*. Aplikasi evaluasi dosen ini dibangun dengan menggunakan bahasa pemrograman *Visual Studio 2008*. Berikut ini adalah spesifikasi dari pembuatan sistem evaluasi dosen, yakni Sistem Operasi *Windows XP Profesional SP dua*, *Visual Studio 2008* dengan bahasa pemrograman *VB.NET*, Basis Data *SQL Server 2008*. Sistem ini dibuat untuk dua pemakai dengan hak akses yang berbeda-beda yakni *User* dan *Admin*. Adapun masing-masing hak akses dari tiap pemakai adalah sebagai berikut



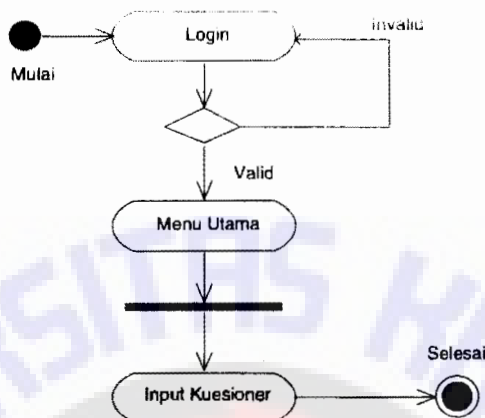
Gambar 2 Perancangan Use Case Diagram Sistem

Pemakai yang pertama adalah *user*. *User* dalam aplikasi ini mempunyai hak akses untuk mengisi formulir atau kuesioner penilaian. Hasil dari kuesioner ini nantinya akan digunakan sebagai nilai dalam kriteria yang akan dipakai sebagai bahan penilaian. Selain mengisi kuesioner, *user* juga dapat melakukan pengolahan data *login* yakni perubahan *password* yang dia miliki. Berikut ini adalah *activity diagram user*.



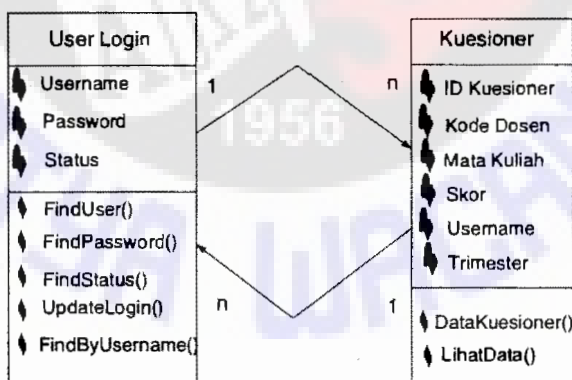
Gambar 3 Activity Diagram Pengelolaan Data Profile User

Gambar 3 menunjukkan *activity diagram* yang dapat dilakukan oleh seorang *user* dalam mengelola data *login*. Langkah pertama yang dilakukan adalah *login*. Apabila *login valid*, maka *user* dapat melakukan proses perubahan *password login*.



Gambar 4 *Activity Diagram Kuesioner*

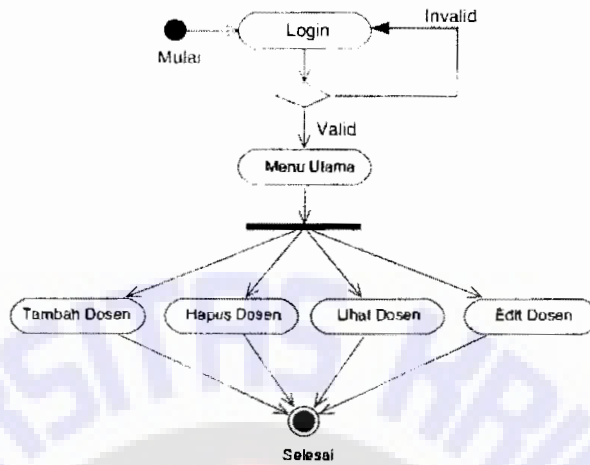
Gambar 4 menjelaskan bagaimana proses seorang *user* dapat melakukan proses pengisian kuesioner. Proses ini diawali dengan proses *login*. Apabila proses *login valid* maka *user* dapat mengakses menu kuesioner. *User* dapat memberikan penilaian kepada beberapa dosen yang akan dievaluasi sesuai dengan *form* kuesioner yang telah tersedia.



Gambar 5 *Class Diagram User*

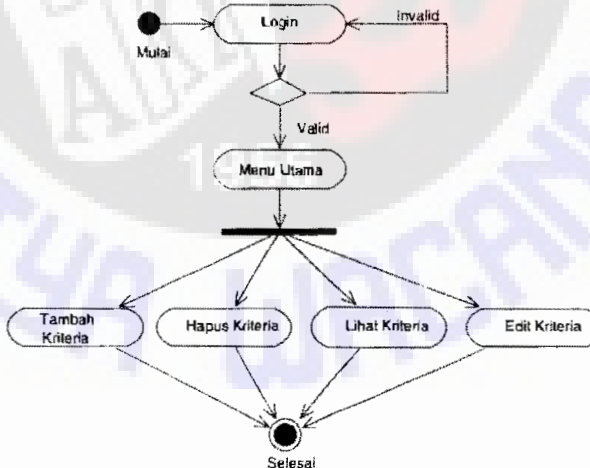
Gambar 5 menunjukkan beberapa *class diagram user* yang akan digunakan dalam pembuatan aplikasi. *Class userlogin* dan *kuesioner* berisi penjelasan tentang atribut dan operasi yang akan pada masing-masing class dalam aplikasi. Pemakai yang kedua adalah admin. Admin mempunyai hak akses penuh atas aplikasi evaluasi dosen yang dibuat. Seorang admin mempunyai tugas dalam melakukan pengolahan

data *user* maupun pengolahan data yang berhubungan dengan penilaian atau evaluasi dosen.



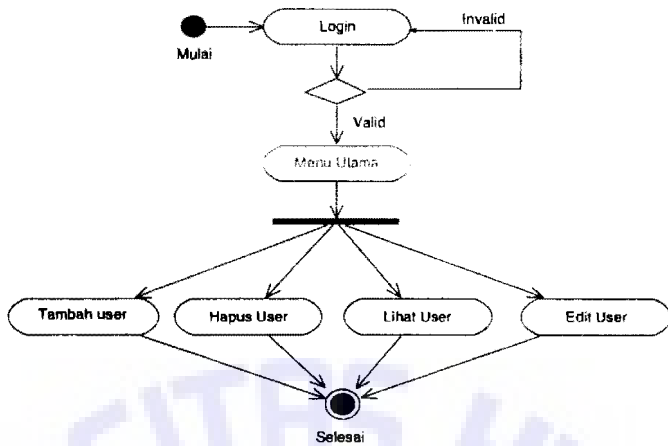
Gambar 6 Activity Diagram Pengelolaan Data Dosen

Gambar 6 adalah *activity diagram* admin yang berisi penjelasan tentang urutan aktifitas dalam pengolahan data dosen. Untuk dapat mengakses menu pengolahan data dosen, admin diwajibkan untuk melakukan proses *login*. Apabila *login* admin *valid* maka admin dapat mengakses menu pengolahan dosen. Menu ini berisi fasilitas untuk menambah, menghapus, mencari, dan mengubah data dosen.



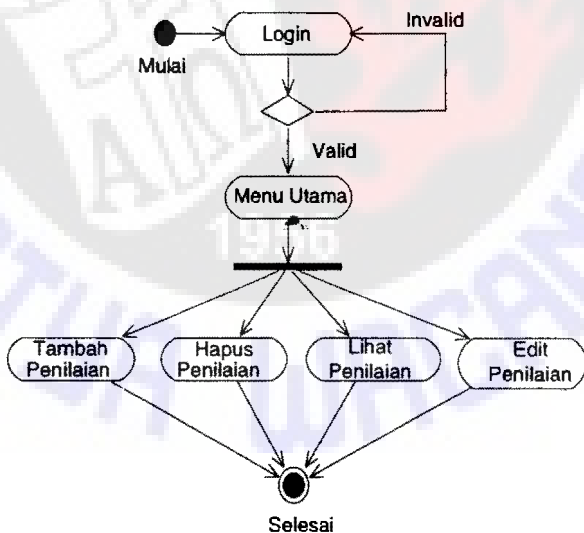
Gambar 7 Activity Diagram Pengelolaan Data Kriteria

Gambar 7 berisi penjelasan tentang urutan aktifitas dalam pengolahan data kriteria. Untuk dapat mengakses menu pengolahan data kriteria, admin diwajibkan untuk melakukan proses *login*. Apabila *login* admin *valid* maka admin dapat mengakses menu pengolahan kriteria. Menu ini berfungsi untuk mengolah kriteria yang nantinya akan dipakai sebagai acuan dalam penilaian.



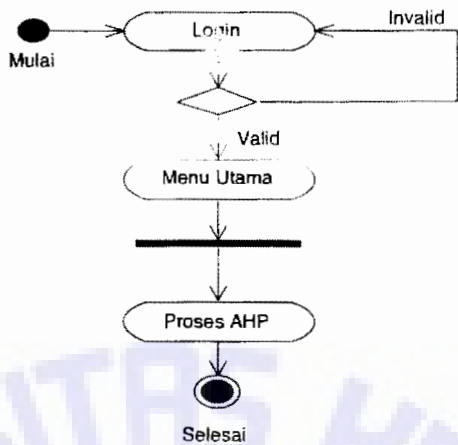
Gambar 8 Activity Diagram Pengelolaan Data User

Gambar 8 berisi penjelasan tentang *activity diagram* dari pengolahan data *user*. Aktivitas ini sama dengan aktivitas *user*. Namun, *user* hanya dapat mengakses menu ubah *profile*, sedangkan untuk admin dapat melakukan proses penambahan dan penghapusan data *user*. Untuk memulai proses ini, seorang admin diwajibkan untuk melakukan proses *login*. *Login* dinyatakan *valid* apabila *username* dan *password* yang dimasukan sama dengan yang ada di dalam basisdata.

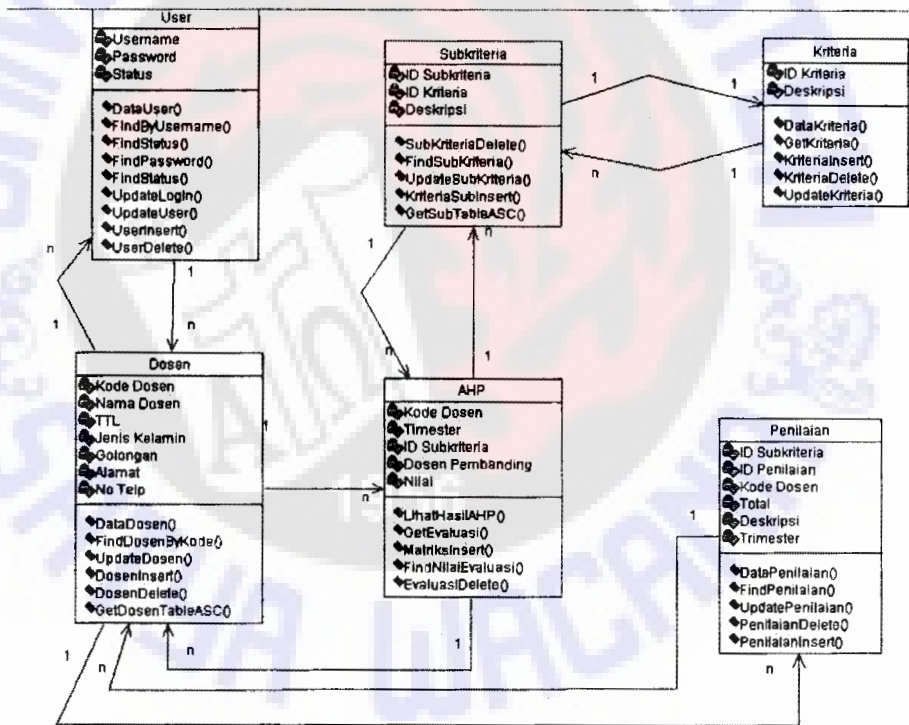


Gambar 9 Activity Diagram Pengelolaan Data Penilaian

Gambar 9 berisi penjelasan tentang *activity diagram* dari pengolahan data penilaian. Pengolahan penilaian bertujuan untuk memasukan, menghapus, maupun mengubah penilaian yang telah dimasukan oleh admin. Hasil masukan dari pengolahan data penilaian ini nantinya akan diolah sehingga menghasilkan informasi seperti yang diinginkan oleh admin.



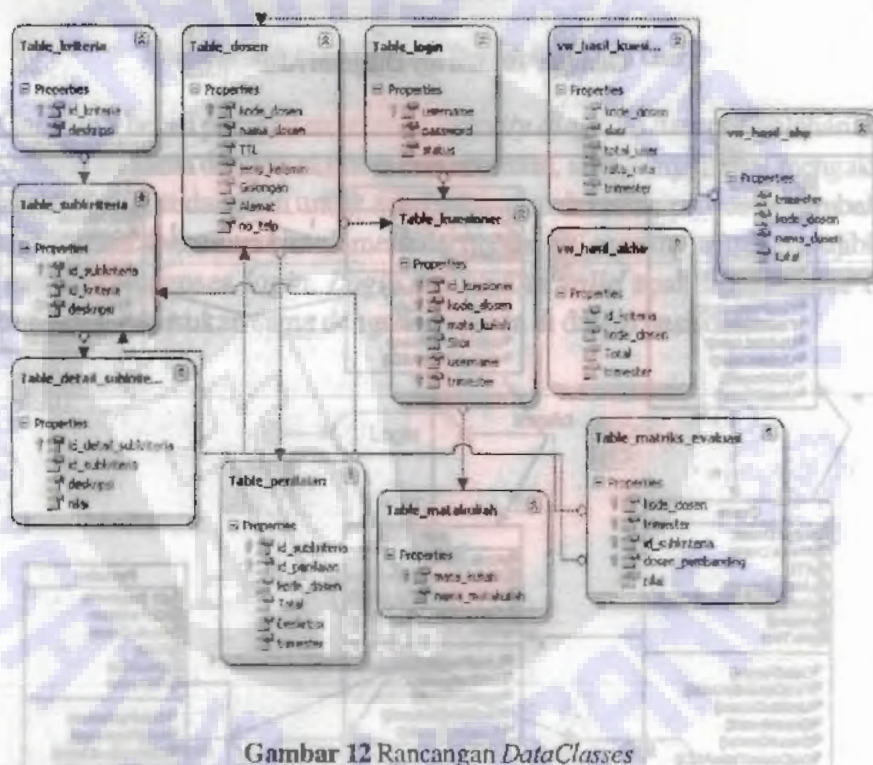
Gambar 10 Activity Diagram AHP



Gambar 11 Class Diagram Admin

Gambar 10 merupakan gambaran dari proses AHP yang digunakan sebagai acuan proses dalam melakukan pengolahan data penilaian seperti yang telah dijelaskan pada Gambar 9. Proses AHP sendiri merupakan salah satu metode SPK yang dipakai dalam aplikasi ini. Proses ini hanya menampilkan hasil evaluasi berupa nilai akhir dari masing-masing dosen. Dalam penerapan AHP, ada beberapa *view* tambahan yang dibuat guna menerapkan metode tersebut ke dalam aplikasi. *View* yang dimaksud adalah *view matriks* pasangan, *view weighted sum vector*, *view jumlah*

kolom, *view consistency vector*, *view consistency ratio*, *view* hasil akhir. Gambar 11 merupakan beberapa *class* yang dipakai dalam aplikasi evaluasi dosen untuk seorang admin. Gambar 11 jelas terlihat bagaimana relasi antar satu *class* dengan *class* yang lain. Selain itu kita juga dapat melihat atribut dan operasi yang digunakan pada masing-masing *class*. Proses koneksi adalah awal dari semua proses yang terjadi pada sistem evaluasi ini. Proses dimulai dengan mengakses SQL Server 2008 menggunakan Visual Basic 2008, yang mana Visual Basic 2008 menyediakan *framework* yang akan dipakai dalam proses koneksi. Dengan menggunakan LINQ to SQL Classes, tabel-tabel yang akan dipakai untuk merancang aplikasi ini dapat dibentuk. Setelah tabel-tabel dapat dibentuk, relasi antar tabel bisa dibentuk. Proses koneksi dapat dilihat seperti Gambar 11.



Gambar 12 Rancangan DataClasses

Gambar 12 menunjukkan bahwa Rancangan *dataclasses* yang telah dibentuk untuk proses koneksi pada aplikasi sudah bisa digunakan. Selanjutnya membuat *class* yang mana *class* tersebut akan memanggil setiap tabel yang ada pada *dataclasses*.

4. Implementasi dan Pengujian Sistem

Uji coba aplikasi dilakukan dengan melakukan pemasukkan data pada *form user* maupun *form administrator*. Untuk dapat mengakses sistem ini, baik admin maupun *user* harus melakukan *login* seperti yang terlihat pada Gambar 13. Selanjutnya apabila *login* berhasil, tampilan menu utama untuk masing-masing *user* ditunjukkan pada Gambar 14

Gambar 13 Halaman Login

Gambar 14 menunjukkan bahwa halaman menu utama akses *user* yang menyediakan beberapa menu pilihan bagi *user* dalam melakukan proses evaluasi, diantaranya adalah menu kuesioner, menu laporan untuk data kuesioner yang sudah diproses, menu ganti *password*, dan menu keluar.

Gambar 14 Tampilan Menu Utama Akses *User*

Gambar 15 Tampilan Menu Utama Akses *Admin*

Selanjutnya tampilan menu utama untuk Admin adalah seperti yang terlihat pada Gambar 15, yang jelas terlihat bahwa seorang *admin* mempunyai hak akses

yang cukup luas dalam aplikasi evaluasi dosen ini. Hak akses tersebut telah dijelaskan sebelumnya yakni Pengolahan Data *User*, Pengolahan Data Dosen, Pengolahan Data Kriteria, Pengolahan Data Sub Kriteria, Pengolahan Data Penilaian, Pengolahan Lihat Data, Keluar. Salah satu proses yang paling penting dalam aplikasi ini adalah proses pengisian kuesioner oleh *user*. Hasil dari kuesioner ini nantinya akan digabungkan dengan hasil dari kriteria yang lain untuk kemudian diolah sesuai dengan metode yang diterapkan. Gambar 16 merupakan *form* evaluasi yang mana berisi delapan kumpulan soal evaluasi yang akan diisi oleh *user* sebagai salah satu dari kriteria penilaian dosen. *User* cukup memberikan jawaban dari pertanyaan dengan memberikan penilaian sesuai dengan aturan yang berlaku. Menu Pengolahan data penilaian merupakan *form* yang digunakan untuk memasukan data penilaian dari dosen-dosen yang akan dievaluasi. *Form* Pengolahan data penilaian dapat dilihat pada Gambar 16.

Gambar 16 *Form* Kuesioner

Gambar 17 *Form* Evaluasi Kinerja Dosen

Selain itu *form* ini juga menyediakan sebuah tautan ke *form* hasil evaluasi yang

mana *form* tersebut bertujuan untuk menampilkan hasil evaluasi dari dosen-dosen sesuai dengan nilai yang diisi oleh admin. *Form* hasil evaluasi dapat dilihat pada Gambar 18 yang menampilkan hasil akhir penilaian berupa nilai dari masing-masing dosen. Dosen dengan nilai yang paling tinggi merupakan dosen dengan prestasi kerja yang baik. Kita dapat melihat hasil penilaian dosen berdasarkan Semester yang diinginkan.

Semester	NIDN	Nama Dosen	Prestasi Kerja	Nilai
1/2008-2009	67001	Indra Pakarung	0.340545555...	
1/2008-2009	67002	Henry Hattu	0.816955555...	
1/2008-2009	67003	Angen	0.800000000...	
1/2008-2009	67004	Joko Susilo	0.899555555...	
1/2008-2009	67008	Susi	0.423333333...	

Gambar 18 *Form* Hasil SPK

5. Simpulan

Dalam pembuatan aplikasi sistem pendukung keputusan untuk evaluasi kinerja dosen FTI-UKSW Salatiga, diperlukan *framework* yang tepat untuk mendukung pembuatan aplikasi ini. *Framework* yang disediakan oleh *Microsoft Visual Studio* 2008, dinilai mampu untuk mengatasi masalah ini. *Framework* yang ditawarkan adalah LINQ yang merupakan alat bantu dalam proses evaluasi, dimana LINQ dipakai untuk pemetaan tabel-tabel dari data evaluasi didalam basisdata di *SQL* server 2008. Selain LINQ, SPK merupakan pilihan yang tepat untuk menghitung representasi nilai dari masing-masing dosen sesuai dengan kualitas yang akan dinilai.

6. Daftar Pustaka

- [1] Widyatama. Sistem Pendukung Keputusan Pemberian Kredit PT. ASRA. <http://google.com>. Diakses tanggal 2 Juni 2009.
- [2] Budiantoro. A. Pusat Bahan Ajar. <http://google.com>. Diakses tanggal 3 Juni 2009.
- [3] Turban Efraim, Aronson Jay & Liang Ting-Peng. *Decision Support Systems and Intelligent Systems*. Yogyakarta : Penerbit Andi.