

Aplikasi Pengolahan Data Akademik Fakultas Ilmu Kesehatan di Sumatera Utara

Juanda Hakim Lubis,ST, M.Kom
Program Studi Teknik Informatika Universitas Medan Area
juandahakim@gmail.com

Abstrak

Di dalam sebuah institusi pendidikan penting adanya pengolahan data akademik yang transparan. Transparan ini dalam arti mahasiswa berhak mengetahui hasil akademik mereka selama proses pembelajaran di suatu institusi pendidikan tertentu. Pengolahan data akademik disini tidak hanya mencakup pengolahan nilai, tetapi juga pengolahan data kehadiran mahasiswa, jadwal kuliah, dan lain-lain. Pengolahan data akademik di Fakultas Ilmu Kesehatan di Sumatera Utara sampai saat ini masih dilakukan secara manual dengan penyimpanan data masih dalam excel, sehingga data yang ada tidak terorganisir dengan baik. Dibutuhkan suatu cara dimana masalah seperti pengolahan data nilai, data kehadiran yang disebutkan di atas dapat ditangani dengan baik dan mahasiswa dapat dengan mudah melihat hasil akademik mereka. Oleh karena itu, aplikasi ini dibangun untuk membantu kegiatan pengolahan data akademik di Fakultas Ilmu Kesehatan di Sumatera Utara. Walaupun terdapat komputer sebagai penyimpan data, pada kenyataannya pengolahan data akademik masih memiliki hambatan. Untuk itu, diperlukan aplikasi yang dapat menangani pengolahan data akademik. Hasil dari aplikasi ini merupakan sebuah aplikasi yang dapat mendukung kegiatan pengolahan data akademik di Fakultas Ilmu Kesehatan di Sumatera Utara. Aplikasi ini dibangun menggunakan model *Waterfall*. Dimplementasikan menggunakan bahasa pemrograman Java dengan menggunakan Database PostgreSQL dan software pendukung lainnya.

Kata kunci: nilai, kehadiran, waterfall, java, postgresQL

1. Pendahuluan

Kegiatan akademik yang ada di Fakultas Ilmu Kesehatan di Sumatera Utara sangat banyak, baik itu yang berhubungan dengan kegiatan kuliah maupun yang di luar kuliah. Perlu adanya suatu sistem informasi yang dapat membantu pengelolaan berbagai kegiatan tersebut secara terkomputerisasi sehingga kegiatan akademik Fakultas Ilmu Kesehatan di Sumatera Utara dapat berjalan dengan baik.

Salah satu kegiatan akademik yang pasti ada di setiap perguruan tinggi adalah kegiatan perkuliahan, dimana dalam kegiatan perkuliahan tersebut pasti ada proses pengolahan data akademik yang salah satunya berhubungan dengan nilai dan kehadiran mahasiswa. Sampai saat ini, pengelolaan nilai mahasiswa di Fakultas

Ilmu Kesehatan di Sumatera Utara masih menggunakan sistem manual. Dosen menyerahkan rekap nilai dan kehadiran mahasiswa kepada bagian Pengajaran dalam bentuk hardcopy atau tulisan tangan, kemudian bagian pengajaran menginputkan data-data tersebut dalam bentuk Microsoft excel. Setelah itu bagian pengajaran mencetak data nilai tersebut dan ditempel di papan pengumuman untuk setiap matakuliah yang sudah ada nilainya. Mahasiswa hanya bisa melihat nilai untuk setiap matakuliah melalui papan pengumuman tersebut. Setelah nilai telah lengkap pada tiap semester, data yang tersimpan dalam bentuk Microsoft excel langsung dicetak dan langsung dibagikan ke setiap mahasiswa pada tiap semester sebagai Kartu Hasil Studi (KHS)

mahasiswa. Hal ini sama halnya seperti pembagian raport waktu SD, SMP, atau SMU.

Sistem pengelolaan ini masih dirasa sangat kurang efisien karena mahasiswa tidak bisa melihat nilai secara keseluruhan, proses perhitungan persentasi absensi juga masih digunakan secara manual, serta proses penyimpanan data nilai juga masih menggunakan excel sehingga data yang ada terpisah-pisah. Selain itu proses penghitungan IP dan IPK setiap semester juga masih dilakukan secara manual oleh mahasiswa sendiri.

2. Dasar Teori

2.1 Rekayasa Perangkat Lunak

Rekayasa mempunyai arti sebagai analisa, desain, konstruksi, verifikasi, dan manajemen kesatuan teknik (sosial). Rekayasa perangkat lunak merupakan suatu teknologi yang meliputi sebuah proses, serangkaian metode, dan aturan dari alat bantu. Teknologi yang dimaksud adalah bagaimana membangun suatu perangkat lunak komputer. Selain itu dapat diartikan juga sebagai sekelompok instruksi (program komputer) yang jika diinginkan akan menghasilkan fungsi dan performansi yang diinginkan[8].

Salah satu strategi yang dipakai sebagai *model rekayasa perangkat lunak* yaitu model *waterfall*. *Waterfall* adalah paradigma yang biasanyadigunakan untuk sistem yang telah terdefinisi dengan baik, mulai dari spesifikasikebutuhan, proses, detail masukan, maupun kebutuhan keluarannya. Paradigmaini juga memungkinkan sistem untuk melakukan pengulangan dalam pengembangannya, namun sebaiknya hal tersebut dihindari.

Model *waterfall* meliputi aktifitas-aktifitas sebagai berikut [8] :

1. System Engineering

Sistem pasti merupakan bagian dari sistem yang besar. System engineering adalah tahap membuat daftar kebutuhan untuk semua elemen sistem dan alokasi beberapa kebutuhan ke software.

2. Analysis

Analysis adalah tahap yang bertujuan untuk mengidentifikasi permasalahan yang ada pada sistem, menjelaskan tentang fungsi-fungsi yang ada pada sistem beserta batasan masalahnya, dan menentukan spesifikasi kebutuhan sistem.

3. Design

Design perangkat lunak merupakan proses-proses yang berfokus pada empat atribut berbeda, yaitu struktur data, arsitektur perangkat lunak, representasi *interface*, dan prosedur rinci. Tahap ini akan menerjemahkan tentang kebutuhan sistem menjadi sebuah presentasi perangkat lunak yang kualitasnya dapat diperkirakan sebelum tahap pengkodean dilakukan.

4. Code

Pengkodean merupakan aktivitas yang menerjemahkan hasil perancangan kedalam suatu bentuk yang dapat dimengerti oleh mesin dengan menggunakan bahasa pemrograman tertentu.

5. Testing

Proses pengujian berfokus pada logika internal perangkat lunak, untuk memastikan bahwa semua pernyataan dan fungsi-fungsi eksternal telah diuji coba. Tujuan dari pengujian ini adalah untuk memperbaiki kesalahan yang terjadi dan menjamin bahwa masukan yang telah didefinisikan akan menghasilkan keluaran sesuai dengan hasil yang dibutuhkan.

6. Maintenance

Pada tahap ini dilakukan pemeliharaan pada aplikasi yang telah berjalan dan jika ternyata ada kerusakan maka dilakukan perbaikan pada aplikasi, sehingga aplikasi tetap berjalan sesuai dengan kebutuhan dan fungsinya. Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi baru), atau karena pelanggan membutuhkan

perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.2 Alat Bantu Pemodelan

Pemodelan sistem merupakan cara untuk memudahkan melakukan perancangan perangkat lunak dengan tujuan memahami perilaku sistem yang akan dirancang dengan mendetail dan menggambarkan sistem secara lengkap dan sederhana. Alat bantu yang digunakan untuk menganalisis dan merancang perangkat lunak adalah:

1. Use Case Diagram

Use case adalah gambaran fungsionalitas suatu sistem, supaya *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Sebuah use case merupakan sekumpulan urutan dari aksi atau langkah-langkah, termasuk variant-nya (skenario lain yang mungkin terjadi), yang dilakukan oleh sistem untuk memberikan hasil yang dapat diamati dan diukur oleh user. Secara grafis use case dilambangkan dengan bentuk elips.

Sebuah use case hendaknya spesifik namun fungsionalitas yang dilakukan juga tidak boleh terlalu kecil. Untuk menamai sebuah use case dapat menggunakan kata kerja aktif yang menggambarkan apa yang dilakukan oleh use case tersebut.

Fungsi Diagram Use Case antara lain [2]:

- Menggambarkan fungsi-fungsi yang disediakan oleh sistem
- Menggambarkan interaksi user dengan sistem menggunakan fungsionalitas yang tersedia
- Berkomunikasi dengan end user dan domain expert
 - Menyediakan buy-in pada tahap awal pengembangan sistem.
 - Memastikan pemahaman yang tepat tentang requirement/kebutuhan sistem
- Mengidentifikasi

- Siapa yang berinteraksi dengan sistem dan apa yang harus dilakukan sistem.
- Interface yang harus dimiliki sistem.
- Memverifikasi
 - Semua requirement yang telah dicapture.
 - Tim pengembang memahami requirement.

2. Sequence Diagram

Sequence diagram menggambarkan interaksi antara sejumlah object dalam urutan waktu. Umumnya sebuah sequence diagram menangkap behavior dari suatu skenario (best case). Diagram ini menunjukkan sejumlah object dan pesan-pesan yang dilewatkan antara object-object ini dalam skenario tersebut.

Komponen-komponen yang terdapat pada sequence diagram, antara lain [2]:

- **Object** didapat dari class diagram
- **Found message** merupakan suatu pesan yang menstimulus terjadinya suatu skenario, found message bisa berasal dari main program atau aktifitas user.
- **Life line** merupakan jalur hidup suatu instance kelas tertentu, selain berfungsi untuk mengetahui kapan suatu instance hidup dan dihapus, juga untuk melinierkan urutan pemanggilan pesan untuk instance yang bersangkutan.
- **Activation bar** dalam setiap life line menunjukkan kapan suatu instance aktif dalam interaksi, activation bar ini juga berhubungan dengan fungsi dari instance yang berada dalam stack.
- **Procedure call** merupakan pemanggilan suatu fungsi milik instance kelas lain, dilambangkan dengan garis tegas dan panah solid, pemilik fungsi yang bersangkutan adalah yang ditunjuk oleh panah.
- **Self call** identik dengan procedure call, hanya saja dipanggil dalam fungsi milik

instance sendiri yang sedang aktif pada saat itu (ditandai dengan activation bar).

- **Return value** merupakan suatu nilai balik yang dihasilkan dari pemanggilan suatu fungsi, return value tidak selalu harus digambar, gambarkan hanya bila menambah informasi tambahan untuk memahami skenario yang terjadi.

3. Class Diagram

Class diagram dapat membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem. Class diagram dapat digunakan untuk menggambarkan kolaborasi sederhana dan memodelkan logical database schema. Class digambarkan dengan segi empat yang dibagi menjadi 3 bagian. Bagian yang pertama merupakan nama dari kelas. Bagian yang tengah merupakan struktur dari class (atribut) dan bagian ketiga merupakan sifat / behavior / operasi dari class tersebut [2].

2.3 Perancangan Basis Data

Perancangan basis data mendefinisikan arsitektur data yang terkait dalam website sehingga data bisa lebih efisien dalam penyimpanan, cepat dalam pengaksesan dan lebih mudah dalam pemanipulasian data.

1. Diagram Entity-Relationship (Diagram ER)

Diagram E-R merupakan suatu model perancangan basis data di mana kelompok-kelompok data dan relasi antar kelompok data tersebut diwujudkan dalam bentuk diagram. Model perancangan basis data ini juga merupakan representasi nyata dari sebuah perancangan. Adapun komponen-komponen model data yang digunakan dalam membuat diagram E-R [4] :

- Entitas, merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain.
- Atribut, mendeskripsikan karakteristik (*property*) dari entitas.
- Relasi, menunjukkan adanya hubungan di antara sejumlah entitas

yang berasal dari himpunan entitas yang berbeda.

- Kardinalitas Relasi, menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

2.4 Pengujian

Pengujian perangkat lunak adalah serangkaian kegiatan yang dilakukan untuk mencari kesalahan dari suatu perangkat lunak. Terdapat beberapa strategi dalam pengujian, salah satunya adalah *validation testing* yang melakukan pengujian berdasarkan kebutuhan pengguna, hasil dari tes ini dapat dicapai dengan melakukan serangkaian *blackbox testing*[8]. *Blackbox testing* adalah jenis pengujian yang berfokus pada pengujian fungsionalitas dari aplikasi[8]. Berikut ini adalah beberapa kelas uji yang sebaiknya dilakukan untuk pengujian aplikasi yaitu:

1) *FunctionalTesting*

Kelas uji ini akan menitikberatkan pengujian terhadap fungsionalitas dari aplikasi yang telah dibangun. Beberapa jenis dari *functional testing* adalah:

a) *Link*

Merupakan suatu pengujian yang memastikan bahwa semua *link* yang terdapat dalam aplikasi dapat berjalan sesuai dengan tujuannya.

b) *Form*

Merupakan pengujian yang meliputi pengecekan terhadap *input form* dan *query database*.

2) *UsabilityTesting*

a) *Graphic*

Menekankan pada skema pewarnaan dari aplikasi agar mudah dimengerti oleh berbagai tipe pengguna.

b) *Accessibility*

Melakukan pengujian untuk mendapatkan kesimpulan apakah *user interface* aplikasi dapat memberikan kemudahan untuk menyediakan sebuah layanan pengaksesan.

3. Analisis Sistem

3.1 Analisis Sistem

Untuk memahami sistem yang akan dibuat, maka salah satu tahapan yang harus dilalui adalah melakukan analisis. Pembahasan berikut merupakan analisis terhadap sistem pengolahan data akademikyng sudah ada dan sistem pengolahan data akademikyng akan dibangun.

3.1.1 Analisis Sistem Sekarang

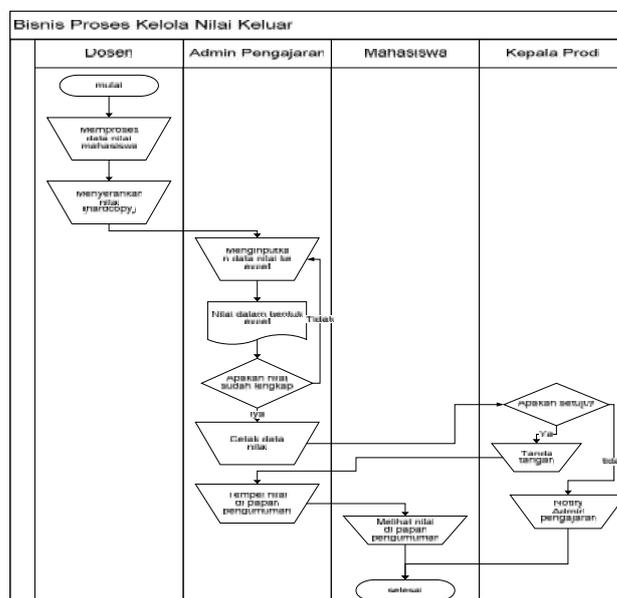
Sistem pengolahan data akademik yang sedang berjalan atau sudah ada di Fakultas Ilmu Kesehatan di Sumatera Utara masih menggunakan sistem manual. Dosen menyerahkan rekap nilai mahasiswa kepada bagian pengajaran dalam bentuk hardcopy atau tulisan tangan, kemudian bagian pengajaran menginputkan data-data tersebut dalam bentuk Microsoft excel. Setelah itu bagian pengajaran mencetak data nilai dan menyerahkan kepada kepala prodi untuk di acc. Setelah data nilai ditanda tangani oleh kepala prodi, selanjutnya data nilai ditempel di papan pengumuman untuk setiap matakuliah. Mahasiswa hanya bisa melihat nilai untuk setiap matakuliah melalui papan pengumuman tersebut. Untuk penjelasan flowchart diagramnya bisa dilihat pada gambar 3.1.

Setelah data nilai telah terisi lengkap untuk tiap semester, maka data tersebut akan dicetak dan dibagikan ke setiap mahasiswa pada tiap semester sebagai Kartu Hasil Studi (KHS) mahasiswa. Hal ini sama halnya seperti pembagian raport waktu SD, SMP, atau SMA. Sistem pengelolaan ini kurang efisien karena mahasiswa tidak bisa melihat nilai secara keseluruhan, proses perhitungan prosentase kehadiran juga masih menggunakan cara manual, dan proses penyimpanan data nilai juga masih menggunakan excel sehingga data terpisah-pisah.

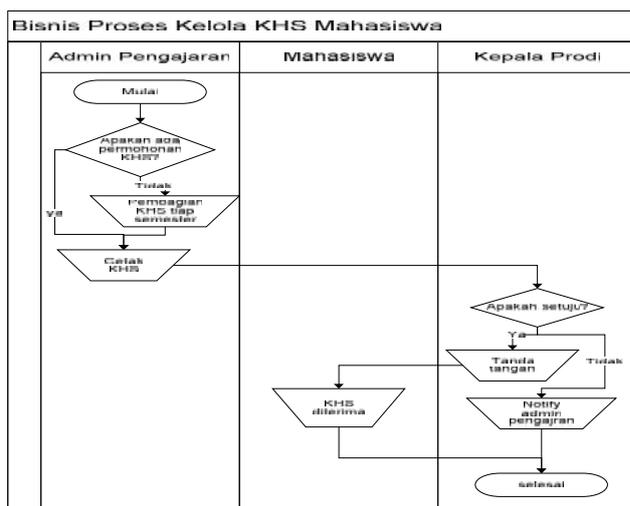
Proses cetak KHS dilakukan setiap akhir semester seperti yang telah dijelaskan pada paragraf sebelumnya. Selain itu proses cetak KHS bisa

dilakukan jika ada permintaan untuk mencetak KHS. KHS yang telah dicetak oleh admin pengajaran, kemudian diserahkan ke kepala prodi untuk ditanda tangani. Setelah kepala prodi menandatangani KHS, kemudian admin pengajaran menyerahkan KHS kepada mahasiswa yang bersangkutan. Untuk penjelasan flowchart diagramnya bisa dilihat pada gambar 3.2.

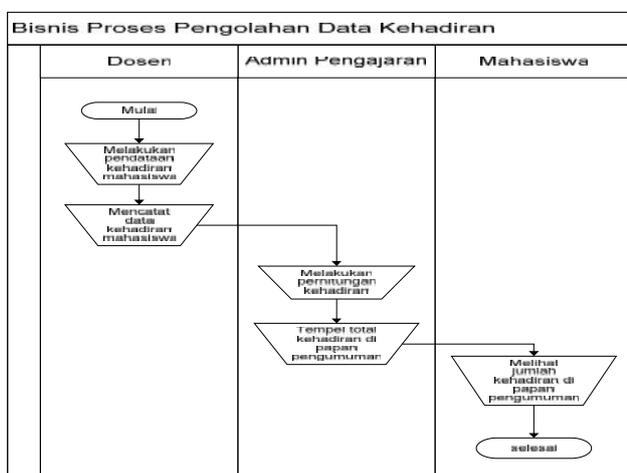
Pengelolaan data kehadiran juga masih dilakukan secara manual. Dosen melakukan pendataan kehadiran mahasiswa dan mencatat data kehadiran mahasiswa. Setelah semua data kehadiran terkumpul, admin pengajaran melakukan perhitungan kehadiran mahasiswa dan total kehadiran yang telah dihitung lalu ditempel di papan pengumuman. Mahasiswa bisa melihat total kehadiran melalui papan pengumuman tersebut. Untuk penjelasan flowchart diagramnya bisa dilihat pada gambar 3.3.



Gambar Error! No text of specified style in document..1: Flowchart Kelola Nilai



Gambar Error! No text of specified style in document..2: Flowchart kelola KHS mahasiswa



Gambar Error! No text of specified style in document..3: Flowchart pengolahan data kehadiran

3.1.2 Analisis Sistem Aplikasi Pengolahan Data Akademik

Setelah menganalisis kekurangan yang terdapat pada sistem lama, maka akan dibangun sebuah Aplikasi Pengolahan Data Akademik Fakultas Ilmu Kesehatan di Sumatera Utara yang diharapkan dapat mengurangi kekurangan pada sistem lama.

Beberapa fungsionalitas berdasarkan user yang mengakses aplikasi ini diantaranya :

a. Admin Pengajaran

- Kelola Data Mahasiswa

Fungsionalitas ini berhubungan dengan pengelolaan data mahasiswa

yang berisi biodata dari mahasiswa. Admin bisa melakukan proses insert, update, view pada fungsionalitas ini.

- Kelola Data Matakuliah

Fungsionalitas ini berhubungan dengan pengelolaan data matakuliah yang ada di Fakultas ini. Pengelolaan data yang bisa dilakukan adalah insert, update, view.

- Kelola Data Dosen

Fungsionalitas ini berhubungan dengan pengelolaan data dosen yang berisi biodata dari dosen. Admin bisa melakukan proses insert, update, view pada fungsionalitas ini.

- Kelola Data Mengajar

Fungsionalitas ini berhubungan dengan pengelolaan data dosen mengajar matakuliah. Pada saat admin menginputkan data mengajar, data jadwal juga otomatis terisi, namun nimnya masih belum ada. Pengelolaan data yang bisa dilakukan adalah insert, update, view.

- Kelola Data Registrasi

Fungsionalitas ini berhubungan dengan pengolahan data mahasiswa mengambil matakuliah atau data registrasi. Pada saat admin menginputkan data registrasi, data nilai juga otomatis terisi, namun nilainya masih null. Nim mahasiswa yang mengambil matakuliah juga otomatis terinput ke data jadwal. Pengolahan data yang bisa dilakukan pada fungsionalitas ini adalah insert, update, view.

- Pengolahan Data Kehadiran

Fungsionalitas ini berhubungan dengan pengolahan data kehadiran mahasiswa. Pengolahan data yang bisa dilakukan pada fungsionalitas ini adalah insert, update, view.

- Validasi Nilai

Fungsionalitas ini digunakan untuk melakukan validasi terhadap nilai yang diinputkan oleh dosen.

- View nilai, view jadwal
- b. Dosen
 - Pengolahan Data Nilai

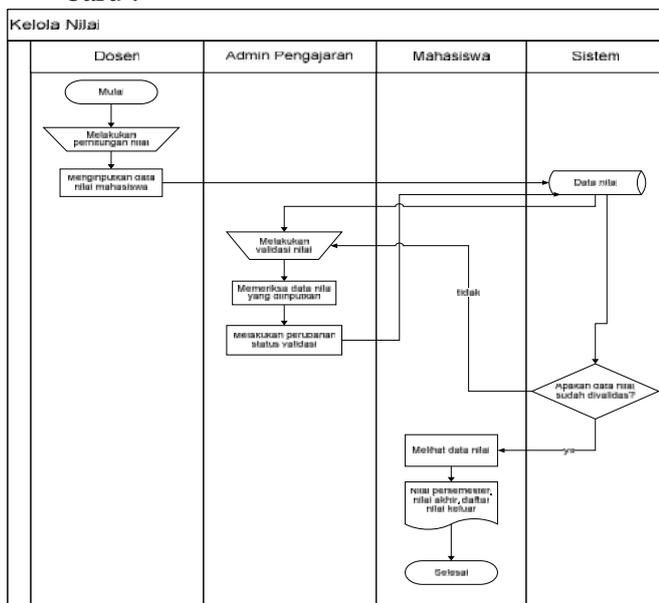
Fungsionalitas ini berhubungan dengan pengolahan data nilai. Pengolahan data yang ada adalah insert, view.
 - View jadwal

Fungsionalitas ini berguna untuk melihat jadwal dosen mengajar.
 - View kehadiran mahasiswa

Fungsionalitas ini berguna untuk melihat kehadiran mahasiswa sesuai dengan dosen yang bersangkutan.
 - Ganti password dosen
- c. Mahasiswa

Fungsionalitas-fungsionalitas yang dimiliki mahasiswa adalah view data mahasiswa, view jadwal, view kehadiran, view nilai yang meliputi nilai per semester (KHS), nilai akhir (Transkrip nilai), daftar nilai keluar dan fungsionalitas ganti password.

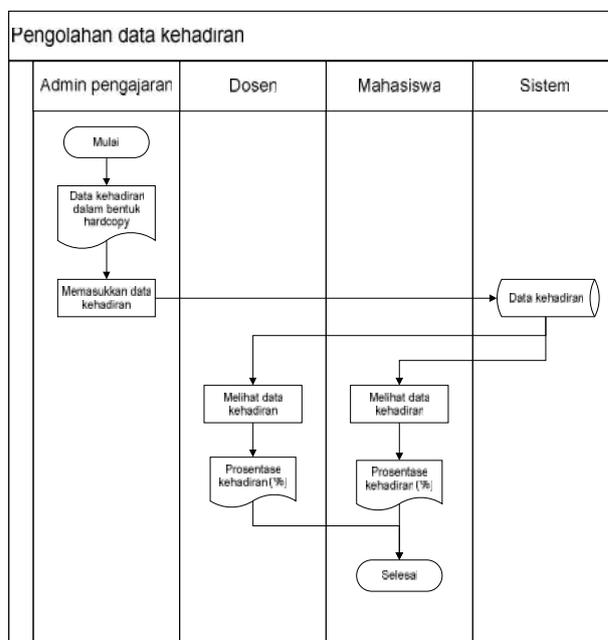
Berikut ini adalah flowchart diagram dari proses yang akan dilakukan saat diterapkannya aplikasi pada sistem baru :



Gambar Error! No text of specified style in document..4: Flowchart kelola nilai

Pengolahan nilai pada sistem baru, dosen melakukan perhitungan nilai mahasiswa dan menginputkan data nilai

tersebut ke dalam aplikasi. Data yang diinputkan akan tersimpan di dalam database nilai. Pada saat dosen menginputkan data nilai, data tersebut tidak bisa langsung dilihat oleh mahasiswa. Namun data nilai harus divalidasi terlebih dahulu oleh admin pengajaran. Admin pengajaran memeriksa apakah data nilai yang diinputkan dosen sudah benar, kemudian admin melakukan proses validasi nilai yaitu mengubah status validasi nilai yang ada pada database. Jika status validasi sama dengan 1, maka mahasiswa bisa melihat data nilai keluar pada aplikasi. Nilai tersebut meliputi nilai per semester, nilai akhir dan daftar nilai keluar. Penjelasan gambar 3.4.



Gambar Error! No text of specified style in document..5: Flowchart pengolahan data kehadiran

Pengolahan data kehadiran pada sistem baru, admin menginputkan data kehadiran ke dalam aplikasi. Data tersebut akan tersimpan dalam database kehadiran. Admin pengajaran menginputkan data kehadiran setiap hari ke dalam aplikasi. Kemudian aplikasi akan mengolah data tersebut dan menampilkan data bentuk prosentase. Dosen dan mahasiswa dapat melihat prosentase kehadiran melalui aplikasi.

4. Implementasi dan Pengujian

4.1 Implementasi Sistem

Tahap implementasi merupakan tahap realisasi dari perancangan yang sebelumnya telah dibuat. Berikut ini merupakan pembahasan mengenai implementasi sistem yang meliputi implementasi pembuatan aplikasi sampai dengan aplikasi digunakan.

4.2 Pengujian Sistem

4.2.1 Persiapan Pengujian

Berdasarkan panduan pengujian yang terdapat pada bab 2, maka berikut ini merupakan hal-hal yang perlu didetailkan untuk melakukan pengujian terhadap aplikasi. Aktifitas yang perlu dilakukan untuk pengujian terdapat pada kolom butir uji. Seluruh pengujian aplikasi menggunakan pengujian validasi karena pengujian dilakukan setelah semua fungsionalitas dibangun. Hasil pengujian validasi dapat dicapai dengan melakukan serangkaian jenis pengujian *blackbox* karena jenis pengujian ini berfokus pada *requirement* dari aplikasi.

4.3 Penggunaan Aplikasi

Sebelum digunakannya aplikasi ini untuk menunjang bisnis proses maka perlu dilakukannya proses migrasi dari sistem yang lama ke sistem yang baru. Proses ini dilakukan agar saat sistem baru yang akan dijalankan tidak menghambat ataupun menyulitkan bisnis proses.

Proses pertama kali yang dilakukan pada tahap migrasi adalah melakukan konversi data yang dahulunya masih dalam file berekstensi .xls ke dalam database. Cara yang dapat dilakukan untuk memindahkan data dari file .xls ke database yaitu dengan menggunakan aplikasi yang dibuat. Proses selanjutnya dalam tahap migrasi adalah dilakukannya pelatihan terhadap pengguna yang akan menggunakan aplikasi ini. Pelatihan ini diberikan untuk memperkenalkan sekaligus membiasakan penggunaan aplikasi sehingga sistem yang baru dapat dijalankan dengan baik.

5. Penutup

5.1 Kesimpulan

Adapun kesimpulan yang dapat diambil dari pembuatan proyek akhir ini, yaitu:

1. Pengolahan data nilai dilakukan oleh aplikasi dan data nilai yang ditampilkan adalah data nilai per semester, nilai akhir dan daftar nilai keluar.
2. Data kehadiran mahasiswa ditampilkan dalam bentuk prosentase.
3. Mahasiswa dapat melihat IP Semester dan IPK melalui aplikasi.
4. Out put pelaporan nilai dapat dibuat dalam format excel.

6. Daftar Pustaka

- [1] Common Laboratory.2009. *Modul Praktikum Basis Data*.Bandung:IT Telkom
- [2] Common Laboratory.2010.*Rational Rose : Modul Praktikum Rekayasa Perangkat Lunak*.Bandung: IT Telkom
- [3] Common Laboratory.2010.*Modul Praktikum Pemrograman Berorientasi Object*. Bandung : IT Telkom
- [4] Fathansyah, Ir. 1999. *Buku Teks Ilmu Komputer Basis Data*. Bandung :Informatika.
- [5] Ramakrishnan, Raghu.2003.Sistem Manajemen Database,edisi3. Yogyakarta: Andy and McGraw-Hill Education.
- [6] Indelarko, Hendi, Riyanto & Suprpto.2008.*Pengembangan Aplikasi Manajemen Database dengan Java 2 (SE/ME/EE)*. Yogyakarta : Gava Media.
- [7] Eckle, Bruce.2000.*Thinking in Java,2nd Edition, Release 11*,Prentice-Hall
- [8] Pressman, Roger S.Ph.D.1997.*Software Engineering: A*

Practitioner's Approach.

http://ce.sharif.ir/courses/84-85/1/ce474/resources/root/Pressman_Software%20Engineering.pdf

tanggal 23 april 2010

- [9] Raharjo, Budi, Heryanto, Imam & Haryanto, Arif.2007. *Mudah Belajar Java*. Informatika Bandung: Bandung

- [10] Schildt, Herbert.2001. *Java™ 2: The Complete References, Fourth Edition*, McGraw-Hill