

PENGEMBANGAN SISTEM INFORMASI PENGOLAHAN DATA PENGECEKAN NILAI MAHASISWA

Ratri Purwaningtyas

Universitas Gunadarma, ratriptyas@staff.gunadarma.ac.id

ABSTRAK

Pemanfaatan Teknologi Informasi dan Komunikasi pada bidang layanan administrasi akademik di perguruan tinggi menjadi suatu kebutuhan untuk meningkatkan pelayanan akademik kepada mahasiswa. Penulis melakukan analisis dari sistem informasi pengecekan nilai akademik yang sedang berjalan di sebuah perguruan tinggi, dimana selama ini melakukan proses secara manual dengan mencatat di buku besar. Tujuan Penelitian ini adalah mengembangkan sebuah sistem informasi untuk pengolahan data nilai mahasiswa, dimana dikhususkan untuk mengolah data nilai mahasiswa yang salah. Kesalahan nilai yang dimaksud seperti dosen atau pihak lab yang salah input nilai sampai kesalahan perolehan range nilai. Pengembangan sistem ini menggunakan metode Waterfall yang terdiri dari Requirements analysis and definition, System and software design, Implementation and unit testing, Integration and system testing, dan Operation and maintenance. Tahap Requirements analysis and definition, identifikasi masalah yang sedang terjadi di dalam organisasi tersebut terutama pada masalah pengumpulan dan penginputan data. Tahap System and software design, desain sistem dibuat supaya memudahkan dalam melakukan perancangan dan membuat tampilan aplikasi yang diinginkan. Tahap Implementation and unit testing, desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman berbasis visual dan pengujian desain dilakukan secara unit. Tahap Integration and system testing, selain menggunakan bahasa pemrograman visual, juga digunakan bahasa pemrograman untuk databasenya. Tahap Operation and maintenance, tahap terakhir dari model waterfall ini adalah tahap aplikasi supaya siap untuk diimplementasikan. Hasil dari penelitian ini adalah mengganti pencatatan yang sebelumnya dilakukan secara manual menjadi terkomputerisasi sehingga memudahkan staff akademik untuk mengolah data pengecekan nilai akademik mahasiswa.

Kata Kunci: Nilai, UML, Pengembangan, Sistem

PENDAHULUAN

Menurut Satoto (2009) sistem informasi akademik adalah perangkat lunak yang digunakan untuk menyajikan informasi dan menata administrasi yang berhubungan dengan kegiatan akademis. Dengan menggunakan perangkat lunak diharapkan kegiatan administrasi akademis dapat dikelola dengan baik dan informasi yang diperlukan dapat diperoleh dengan mudah dan cepat. Beberapa contoh kegiatan yang bersifat administratif adalah Penerimaan Mahasiswa Baru (PMB), penyusunan kurikulum dan jadwal kuliah, mengisi

Kartu Rencana Studi (KRS), mengelola data dosen, karyawan, dan mahasiswa seperti data nilai akademik mahasiswa.

Setiap akhir semester mahasiswa akan menerima nilai akademik dari hasil belajar selama satu semester, nilai yang didapatkan berupa nilai akhir yang sebelumnya telah diolah oleh pihak perguruan tinggi. Mahasiswa tidak mengetahui rincian dari nilai yang telah mereka peroleh sehingga pihak perguruan tinggi membuka pelayanan pengecekan nilai untuk mahasiswa yang ingin mengetahui rincian dari nilai yang telah diperolehnya. Selain untuk mengetahui

rincian nilai, mahasiswa juga dapat mengecek kebenaran dari nilai tersebut. Terkadang terjadi kesalahan dari nilai yang telah mahasiswa peroleh, seperti dosen atau pihak lab yang salah input nilai sampai kesalahan perolehan range nilai. Pengecekan nilai yang dilakukan akan menghasilkan 2 hasil yaitu nilai yang diperoleh benar dan nilai yang diperoleh salah. Nilai yang salah akan diproses lebih lanjut oleh pihak akademik.

Pelayanan yang diberikan pihak akademik terhadap mahasiswa khususnya dalam memproses kesalahan nilai masih belum menerapkan sistem informasi akademik secara optimal. Pengolahan data yang berhubungan dengan pengecekan nilai mahasiswa masih dilakukan secara manual, yaitu pencatatan data mahasiswa yang mengajukan pengecekan nilai dalam buku besar. Hal tersebut menyebabkan ketidakakuratan data atau informasi, antrean mahasiswa menjadi panjang, dan waktu menunggu penyelesaian masalahnya menjadi lama.

Tidak tersedianya sistem terkomputerisasi untuk mengelola data nilai mahasiswa tersebut sehingga dengan dibuatnya aplikasi pengolahan data nilai mahasiswa ini dapat membantu pihak akademik untuk mencatat, menyimpan, mencari, mengubah data mahasiswa menjadi lebih mudah, dan informasi mengenai pengecekan nilai dapat mudah diakses oleh pihak yang berkepentingan.

TINJAUAN PUSTAKA

Pengertian Pengolahan Data dan Sistem

Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan. (Kristanto, 2008)

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul dan bersama-

sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. (Fitzgerald, 1981).

Linear Sequential Model/Waterfal Model

Model ini adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Berikut ini ada dua gambaran dari *waterfall* model. Sekalipun keduanya menggunakan nama-nama fase yang berbeda, namun sama dalam intinya. Kedua gambaran tersebut dapat di lihat pada gambar 1 dan gambar 2 di bawah ini.

Pada gambar 1 di atas adalah model waterfall menurut referensi Pressman sering juga disebut model sekuensial linier (sequential linear) atau alur hidup klasik (classic life cycle). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (support).

Berdasarkan gambar 2 di atas fase-fase dalam waterfall model menurut referensi Sommerville yang terdiri dari Requirements analysis and definition, System and software design, Implementation and unit testing, Integration and system testing, dan Operation and maintenance. Tahap Requirements analysis and definition, Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap. Tahap System and software design, Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap. Tahap Implementation and unit testing, Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah

ditentukan. Program yang dibangun langsung diuji baik secara unit. Tahap Integration and system testing, Penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing). Tahap Operation and maintenance, Mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

Unified Modelling Language (UML)

Menurut Munawar (2005) *Unified Modelling Language* adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek.

Diagram Unified Modeling Language (UML)

1. Use Case Diagram

Use Case Diagram menjelaskan manfaat sistem jika di lihat menurut pandangan orang yang berada di luar system (*actor*). Diagram ini menggambarkan fungsionalitas yang diharapkan dari suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.

2. Class Diagram

Class Diagram membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak digunakan. *Class Diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam perancangan model dari suatu sistem.

3. Activity Diagram

Activity Diagram menggambarkan berbagai alur aktivitas dalam sistem, bagaimana masing-masing alur berawal, keputusan yang mungkin terjadi, dan bagaimana mereka

berakhir. *Activity Diagram* dapat memodelkan alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Juga sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi.

PHPMyAdmin, Microsoft Visual Studio, dan MySQL Connector

Phpmyadmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyadmin, dapat membuat *database*, membuat tabel, manipulasi data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual.

Visual studio 2010 merupakan sebuah *Integrated Development Environment* (IDE) atau lingkungan kerja yang digunakan untuk membangun aplikasi .net (dot net). Microsoft Visual Studio Ultimate 2010 menyediakan berbagai tool yang lengkap bagi para pengembang untuk membangun aplikasi yang berjalan di .net *framework*. MySQL Connector digunakan untuk menghubungkan antara program visual basic dengan MySQL.

METODE PENELITIAN

Tahap Pengumpulan Data

Tahap pengumpulan data dilakukan dengan studi lapangan dan studi pustaka. Studi lapangan yaitu wawancara dengan pihak yang terkait untuk mendapatkan data dan informasi yang diharapkan selain itu melakukan observasi secara langsung sehingga dapat diperoleh gambaran secara jelas, serta untuk mengetahui sejauh mana informasi yang dibutuhkan. Studi pustaka dengan mencari dan menelaah literatur yang berkaitan dengan masalah yang dihadapi.

Tahap Analisis

Setelah data yang diperlukan terkumpul, selanjutnya dilakukan proses analisis terhadap sistem yang akan dikembangkan. Langkah pertama yang dilakukan adalah mengidentifikasi masalah, memahami kerja sistem yang ada, menganalisis sistem yang sedang berjalan, dan menetapkan langkah selanjutnya dalam pengembangan sistem.

Tahap Pengembangan Sistem

Pada penelitian ini penulis melakukan pengembangan sistem dengan menggunakan model waterfall yaitu model yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau urut sedangkan untuk tools yang digunakan yaitu UML (*Unified Modelling Language*).

HASIL DAN PEMBAHASAN

Analisis Kebutuhan

Analisa kebutuhan merupakan spesifikasi kebutuhan dari sistem akademik. Staff akademik akan melakukan pengecekan nilai mahasiswa dari sistem akademik, jika ditemukan ada kesalahan maka akan diproses kesalahan nilai tersebut dengan melakukan input data pada sistem pengolahan data pengecekan nilai akademik, setelah diproses oleh bagian nilai maka nilai mahasiswa yang bersangkutan akan *terupdate* di sistem akademik.

Perancangan

Perancangan Perangkat Lunak

Tahap perancangan perangkat lunak adalah membuat rancangan UML (*Unified Modelling Language*). Rancangan UML ini untuk memahami gambaran aplikasi yang akan dibangun, serta memudahkan dalam melakukan penelusuran kode program. Diagram UML yang digunakan untuk perancangan aplikasi ini yaitu *use case diagram*, *activity diagram* dan *class diagram*.

Use case diagram pada gambar 3 menjelaskan bahwa setelah mahasiswa menyerahkan berkas maka staff akademik dapat melakukan *login* untuk membuka menu sistem informasi pengecekan nilai akademik. Staff akademik dapat memasukkan data pada masing-masing kolom sesuai dengan data yang masuk. Sistem ini terintegrasi dengan laboratorium dan bagian nilai. Setelah data nilai mahasiswa yang salah terinput maka pihak laboratorium dapat memberikan rincian nilai praktikum kemudian staff nilai dapat *update* data nilai mahasiswa.

Activity diagram pada gambar 4 menjelaskan bahwa proses pengecekan nilai yang dilakukan oleh bagian Akademik untuk membantu mahasiswa untuk menerima rincian dari hasil nilai yang telah diperoleh kemudian memproses kesalahan nilai mahasiswa tersebut.

Class diagram pada gambar 5 menjelaskan bahwa terdapat 4 class dalam perancangan ini yaitu class Dosen, class Matakuliah, class Mahasiswa, dan class Pengecekan Nilai. Class pengecekan nilai berelasi dengan class dosen, class matakuliah, dan class mahasiswa sehingga proses pengambilan data lebih mudah. Indikator *multiplicity* yaitu 1 Tepat satu dan 1..* satu atau lebih.

Perancangan Database

Database yang dibutuhkan pada pengembangan sistem ini terdiri dari beberapa tabel di bawah ini.

Tabel mahasiswa terdapat field fakultas dan jurusan. Penentuan Fakultas ditentukan dari digit ketiga NPM mahasiswa dan penentuan Jurusan ditentukan dari 2 digit mulai dari digit kedua kelas. Tabel ini digunakan untuk menyimpan seluruh data mahasiswa, seperti yang terlihat pada Tabel 1.

Tabel 1.
Tabel Mahasiswa

| Nama Field | Tipe Data | Lebar | Keterangan |
|------------|-----------|-------|--------------------|
| NPM | Varchar | 8 | Primary Key, NPM |
| Nama | Varchar | 20 | Nama Mahasiswa |
| Kelas | Varchar | 5 | Kelas Mahasiswa |
| JK | Varchar | 9 | Jenis Kelamin |
| Alamat | Varchar | 20 | Alamat |
| Fakultas | Varchar | 15 | Fakultas Mahasiswa |
| Jurusan | Varchar | 20 | Jurusan |
| Lok_Kuliah | Varchar | 20 | Lokasi Kuliah |

Tabel Mata Kuliah digunakan untuk menyimpan data mata kuliah, seperti yang terlihat pada Tabel 2.

Tabel Dosen digunakan untuk menyimpan data dosen, seperti yang terlihat pada Tabel 3.

Tabel pengecekan nilai terdapat *Rec_ceknil* yang merupakan catatan data pengecekan nilai yang disusun berdasarkan *auto increment*. NPM sebagai *foreign key* yang menghubungkan dengan tabel mahasiswa. *KD_Matkul* sebagai *foreign key* yang menghubungkan dengan tabel matkul, seperti yang terlihat pada Tabel 4.

Perancangan Tampilan Aplikasi

Tahap selanjutnya adalah membuat rancangan tampilan aplikasi. Rancangan tampilan yang dibuat yaitu halaman login dan menu pengolahan data.

Pada gambar 6 di atas Rancangan tampilan dimulai dengan menunjukkan tampilan sistem *login*. *Login* dapat dilakukan oleh semua staf akademik dengan *username* dan *password* yang telah ditentukan.

Gambar 7 di atas menggambarkan rancangan menu pengecekan nilai terdiri dari 3 *input* data, yaitu NPM, kode matakuliah, dan kode dosen. 4 *combobox* dengan berbagai pilihan untuk mengefisiensikan waktu *input* data, yaitu kasus, nilai lama, nilai baru, dan status. Beberapa *field* tidak perlu *diinput*, dikarenakan akan masuk ke tabel secara otomatis. Data nama dan kelas akan muncul setelah NPM *diinput* diambil dari tabel mahasiswa, nama matakuliah akan muncul setelah kode matakuliah *diinput* diambil dari tabel matkul, dan nama dosen akan muncul setelah kode dosen *diinput* diambil dari tabel dosen Terdapat 5 button yaitu *submit*, *search*, *update*, *delete*, dan *cancel*. *Button cancel* untuk kembali ke menu utama. Terdapat data *gridview* untuk menampilkan hasil *input* data.

Implementasi

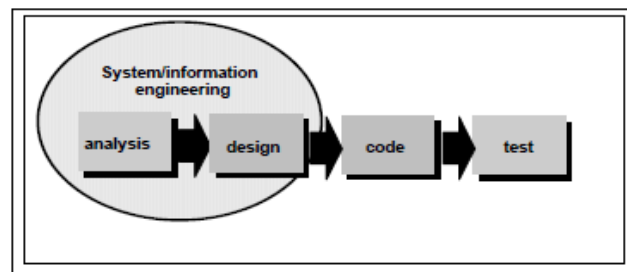
Tampilan aplikasi untuk menunjukkan secara langsung apa yang terjadi di dalam sistem dan dimulai dengan menunjukkan tampilan sistem login, karena sistem login adalah langkah awal dalam memasuki sistem. Aplikasi ini hanya dapat di akses oleh staff akademik dengan menggunakan *username* dan *password* yang sudah ditentukan. Tampilan login ditunjukkan pada gambar 8.

Gambar 9. dibawah merupakan tampilan menu pengecekan nilai, terdapat *button search*, *update*, *delete*. *Button search* untuk mencari data yang diinginkan dengan cara, diklik *search*, maka akan tampil *inputbox* untuk mencari data berdasarkan NPM.

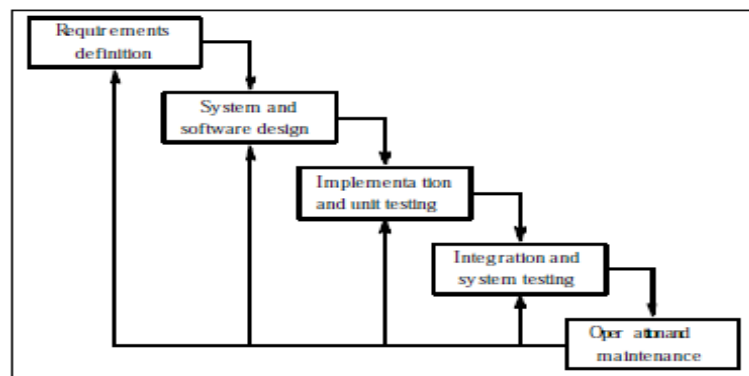
Setelah NPM dimasukkan, maka data yang sesuai dengan NPM tersebut

akan tampil pada *textbox*. Fungsi *search* dapat digunakan untuk *update* data dengan langsung mengubah data pada *textbox* tersebut, tetapi untuk NPM, nama, kelas tidak dapat diupdate karena data tersebut diperoleh dari tabel

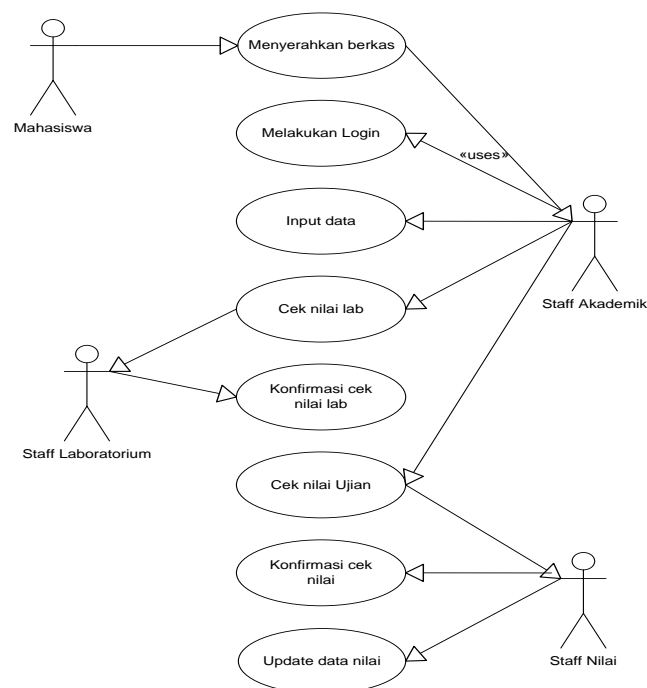
mahasiswa. Selain *update*, fungsi *search* dapat digunakan untuk *delete* data yang ingin dihapus. Pada tampilan menu input pengecekan nilai tersebut data yang telah dimasukkan akan tampil pada tabel dibawah bagian input.



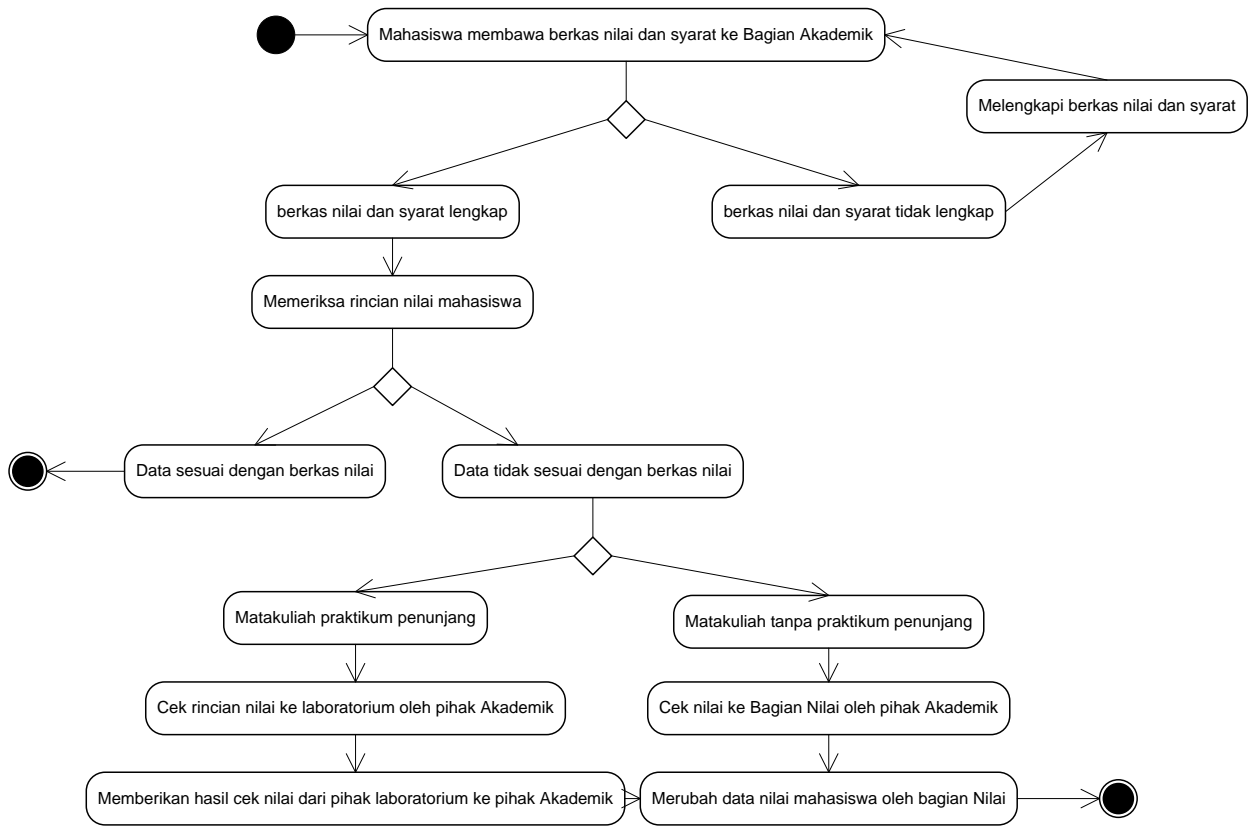
Gambar 1 Fase-fase dalam *Waterfall Model* menurut referensi Pressman



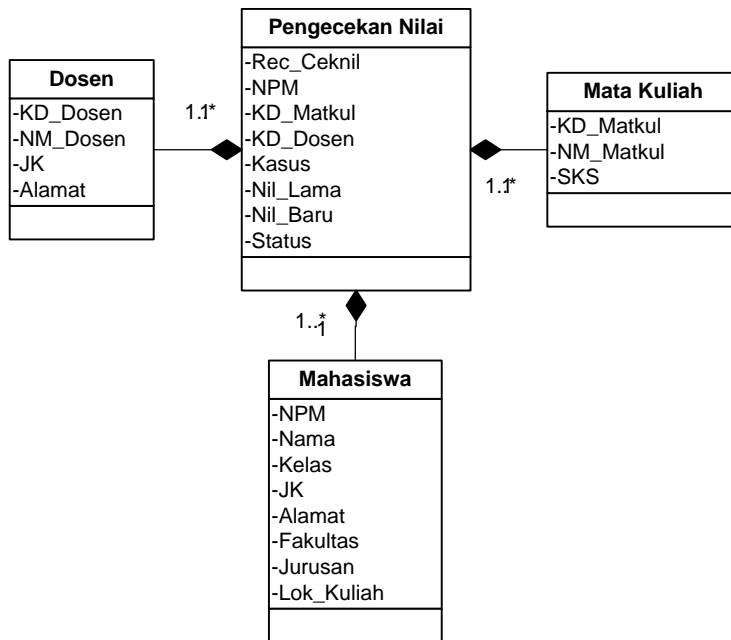
Gambar 2.Fase-fase dalam *Waterfall Model* menurut referensi Sommerville



Gambar 3 Use Case Diagram



Gambar 4 Activity Diagram



Gambar 5 Class Diagram

Tabel 2.
Tabel Mata Kuliah

| Nama Field | Tipe Data | Lebar | Keterangan |
|------------|-----------|-------|------------------------------|
| KD_Matkul | Varchar | 8 | Primary Key, Kode Matakuliah |
| NM_Matkul | Varchar | 20 | Nama MataKuliah |
| SKS | Varchar | 1 | SKS Mata Kuliah |

Tabel 3.
Tabel Dosen

| Nama Field | Tipe Data | Lebar | Keterangan |
|------------|-----------|-------|-------------------------|
| KD_Dosen | Varchar | 4 | Primary Key, Kode Dosen |
| NM_Dosen | Varchar | 20 | Nama Dosen |
| JK | Varchar | 9 | Jenis Kelamin |
| Alamat | Varchar | 20 | Alamat |

Tabel 4.
Tabel Pengecekan Nilai

| Nama Field | Tipe Data | Lebar | Keterangan |
|------------|-----------|-------|---|
| Rec_Ceknil | Int | 11 | Primary Key, Auto Increment, Pengecekan Nilai |
| NPM | Varchar | 8 | Foreign Key, NPM |
| KD_Matkul | Varchar | 8 | Foreign Key, Kode Matakuliah |
| KD_Dosen | Varchar | 20 | Kasus Pengecekan Nilai |
| Kasus | Varchar | 1 | Nilai Lama |
| Nil_Lama | Varchar | 1 | Nilai Baru |
| Nil_Baru | Varchar | 6 | Status Pengecekan Nilai |

SISTEM INFORMASI PENGECEKAN NILAI AKADEMIK

USERNAME

PASSWORD

HARI, BULAN-TANGGAL-TAHUN

Gambar 6 Rancangan Tampilan Login

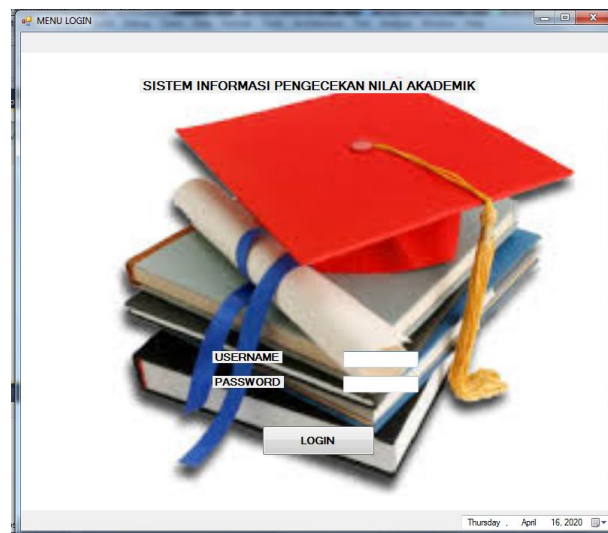
MENU INPUT PERMOHONAN PENGECEKAN NILAI

| | | |
|-----------------|----------------------|---------------------------------------|
| RECORD | <Auto Increment> | |
| NPM | <input type="text"/> | <input type="button" value="SUBMIT"/> |
| NAMA | <XXX> | <input type="button" value="SEARCH"/> |
| KELAS | <XXX> | <input type="button" value="UPDATE"/> |
| KODE MATAKULIAH | <input type="text"/> | <input type="button" value="DELETE"/> |
| MATAKULIAH | <XXX> | <input type="button" value="CANCEL"/> |
| KASUS | <input type="text"/> | |
| NILAI LAMA | <input type="text"/> | |
| NILAI BARU | <input type="text"/> | |
| STATUS | <input type="text"/> | |

<< hasil Input Data >>

HARI, BULAN-TANGGAL-TAHUN

Gambar 7 Rancangan Menu Input Pengecekan Nilai



Gambar 8 Tampilan Login

MENU INPUT PERMOHONAN PENGECEKAN NILAI

| | | |
|-----------------|----------------------|---------------------------------------|
| RECORD | <input type="text"/> | |
| NPM | <input type="text"/> | <input type="button" value="SUBMIT"/> |
| NAMA | <input type="text"/> | <input type="button" value="SEARCH"/> |
| KELAS | <input type="text"/> | <input type="button" value="UPDATE"/> |
| KODE MATAKULIAH | <input type="text"/> | <input type="button" value="DELETE"/> |
| MATAKULIAH | <input type="text"/> | <input type="button" value="CANCEL"/> |
| KASUS | <input type="text"/> | |
| NILAI LAMA | <input type="text"/> | |
| NILAI BARU | <input type="text"/> | |
| STATUS | <input type="text"/> | |

| RECORD | NPM | NAMA | KELAS | KD_MATKUL | NM_MATKUL | KASUS | NIL |
|--------|----------|-----------------|-------|-----------|--------------------|---------------|-----|
| 1 | 16511544 | SAFIRA FITRIANI | 3PA05 | IT051358 | PSIKOLOGI ABNORMAL | CEK NILAI UTS | C |
| * | | | | | | | |

Thursday, April 16, 2020

Gambar 9 Tampilan Menu Input Pengecekan Nilai

KESIMPULAN DAN SARAN

Kesimpulan

Dari hasil penelitian yang penulis lakukan terhadap pengembangan sistem informasi pengecekan nilai akademik mahasiswa maka dapat disimpulkan bahwa pengembangan sistem ini dapat memberikan kemudahan untuk staff akademik dalam bekerja, khususnya pelayanan terhadap mahasiswa. Data yang diolah lebih valid sehingga dapat memberikan informasi secara cepat dan tepat kepada mahasiswa. Mahasiswa menjadi lebih pendek atreannya dan lebih cepat mendapat pelayanannya, sehingga tingkat kepuasan mahasiswa terhadap pelayanan menjadi lebih baik lagi.

Saran

Dari hasil pengembangan sistem ini masih terdapat beberapa kekurangan yang nantinya dapat dijadikan bahan untuk pengembangan selanjutnya, misalnya dapat dikembangkan desain *interface* yang lebih menarik dan *user friendly* agar pelayanan dari staf bisa dilakukan dengan lebih mudah dan lebih efektif. Saran lainnya yang penulis ingin sampaikan adalah masalah keamanan data, mengingat bahwa aplikasi yang dibuat adalah aplikasi mengenai informasi nilai akademik mahasiswa.

DAFTAR PUSTAKA

Djon Irwanto. (2006). *Perancangan Object Oriented Software dengan UML*. Yogyakarta : Andi.

- Fathansyah. (2007). *Basis Data*. Bandung: Informatika.
- Isak Rickyanto, 2003. *Membuat Aplikasi Windows dengan Visual Basic.NET*. Jakarta : Penerbit PT. Elex Media Komputindo.
- Jogiyanto Hartono. (2005). *Analisis dan Disain Sistem Informasi : Pendekatan terstruktur teori dan praktek aplikasi bisnis*. Yogyakarta : PT. Elex Media Komputindo
- Marlon Leong. (2004). *Pemrograman Dasar Microsoft Visual Basic.NET*. Yogyakarta : PT. Elex Media Komputindo
- Pressman, Roger S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi, Edisi Satu*. Yogyakarta : Andi
- Priyanto Hidayatullah. (2012). *Visual Basic.NET Membuat Aplikasi Database dan Program Kreatif* . Bandung : Informatika
- Tedi Kurnia, Dini Destiani, Asep Deddy Supriatna. (2012). *Perancangan Sistem Informasi Akademik Nilai Siswa Berbasis WEB (Studi kasus : SMK Ciledug Al-Musaddadiyah Garut)*. ISSN : 2302-7339 Vol. 09 No. 17.
- Wahana Komputer. (2013). *Membangun Aplikasi Database dengan Visual Basic 2012*. Yogyakarta : Andi.