

Sistem Informasi Manajemen Layanan SMS Center Universitas Muhammadiyah Malang

Rifky Pujiyansyah Suropaty^{*1}, Maskur², Evi Dwi Wahyuni³

^{1,2,3}Universitas Muhammadiyah Malang

rifky@webmail.umm.ac.id^{*1}, maskur@umm.ac.id², evidwi@umm.ac.id³

Abstrak

Universitas Muhammadiyah Malang memiliki sebuah legacy system, yakni layanan SMS gateway. Selama sistem ini digunakan muncul beberapa kebutuhan baru, yakni kebutuhan akan fungsional dan non fungsional, diantaranya adalah automatic reply yang diintegrasikan dengan data request, automatic broadcast dan interface yang responsif. Munculnya berbagai kebutuhan baru ini menandakan bahwa institusi berharap performa dan fungsionalitas dari legacy system bertambah. Ketika perlu untuk memperbarui software dan berharap performa, keandalan, dan fungsionalitasnya bertambah, tanpa mempengaruhi fungsi utamanya, software re-engineering dapat mengatasi masalah seperti ini. Software re-engineering dapat diimplementasikan pada legacy system (SMS gateway) untuk ditransformasikan kedalam sistem informasi manajemen layanan SMS center, sehingga beberapa kebutuhan baru seperti automatic reply yang diintegrasikan dengan data request, automatic broadcast, dan interface yang responsif dapat direalisasikan. Pada software re-engineering terdapat model proses yang harus dilakukan dalam proses implementasi. sehingga proses transformasi dari legacy system ke sistem informasi manajemen layanan SMS center serta proses penambahan beberapa kebutuhan baru dapat berjalan dengan baik, sehingga apa yang diharapkan institusi dapat direalisasikan.

Kata kunci: Software Re-engineering, Legacy System, SMS Gateway, Automatic Reply, Automatic Broadcast

Abstract

University of Muhammadiyah Malang is currently utilizing a legacy system namely SMS gateway service. During its utilization, additional functional and non-functional needs emerge, such as automatic reply integrated with data request, automatic broadcast, and responsive interface. These new additional needs indicate that the institution is expected to improve the performance and functionality of the existing legacy system. If we desire to update the software in order to improve its performance, capacity, and functionality without altering its main function, software re-engineering could be used to handle the issue. Software re-engineering is installed to legacy system (SMS gateway) which then transformed into management information system of SMS center. Hence, the additional needs such as automatic reply integrated with data request, automatic broadcast, and responsive interface would be solved. In software re-engineering, there are some steps to follow during the implementation to make the transformation of legacy system to management information system of SMS center running well; therefore, the institution could afford the expectation.

Keywords: Software Re-engineering, Legacy System, SMS Gateway, Automatic Reply, Automatic Broadcast

1. Pendahuluan

SMS (Short Message Service) merupakan layanan pengiriman pesan singkat berbasis teks. SMS pertama kali dikirimkan pada 3 Desember 1992 oleh Neil Papworth ke sebuah ponsel Orbitel 901 milik Richard Jarvis [1]. Semenjak itu, SMS berkembang sehingga para perusahaan telekomunikasi menjadikan SMS sebagai sebuah layanan utama yang memberikan profit kepada mereka. 23 tahun berlalu SMS memiliki banyak perkembangan sebagai media pemberitahuan utama bagi sebagian perusahaan atau instansi, seperti perusahaan telekomunikasi Telkomsel yang memberikan informasi kepada customer melalui SMS. Selain itu

perusahaan maupun instansi lain memanfaatkan SMS sebagai media pemberitahuan kepada karyawannya.

Legacy system atau disebut dengan sistem warisan adalah sistem lama ataupun tua yang masih digunakan dan menjadi bagian penting dan bernilai bagi sebuah perusahaan atau institusi dalam menjalani aktivitas mereka. Permasalahan yang timbul pada *legacy system* di antaranya adalah pengembang sekaligus pembuat sistem sudah tidak ada (dalam keadaan tertentu), metode yang digunakan kemungkinan sudah kadaluwarsa, pencarian dan modifikasi mungkin akan dilakukan, dokumentasi yang mungkin tidak tersedia, sudah hilang atau tidak berlaku lagi [1].

Re-engineering atau rekayasa ulang sistem merupakan sebuah proses membangun kembali sistem, di mana produk yang dihasilkan diharapkan bertambah fungsionalitasnya, semakin baik performa dan keandalannya, serta meningkatkan kemampuan *maintainability*-nya [2]. Mengapa suatu sistem harus dilakukan *re-engineering* atau rekayasa ulang, alasannya adalah munculnya kebutuhan baru selama sistem tersebut digunakan, berubahnya lingkungan bisnis, adanya *error* yang harus segera diperbaiki, penambahan *hardware* terhadap sistem, dan peningkatan performa dan keandalan yang harus ditingkatkan [3].

Ketika perlu untuk memperbarui *software* dan berharap performa, keandalan, serta fungsionalitasnya bertambah, tanpa mempengaruhi fungsi utamanya, hal ini disebut *software re-engineering*. Ini adalah semua proses di mana desain sistem dari perangkat lunak berubah dan program ditulis ulang. *Legacy system* tidak bisa terus-menerus menggunakan teknologi baru yang tersedia. Tidak seperti *hardware* yang menjadi usang, *software* akan menua setiap waktu tetapi fungsinya tidak. Keunggulan dalam melakukan *software re-engineering*, yaitu dengan dilakukannya *re-engineering* dapat mengurangi biaya jika dilakukan pengembangan ulang terhadap sistem. Dengan menggunakan cara ini, waktu dan dana yang dibutuhkan akan lebih sedikit karena cukup mencari alternatif pada hal yang akan diubah atau ditambahkan [4].

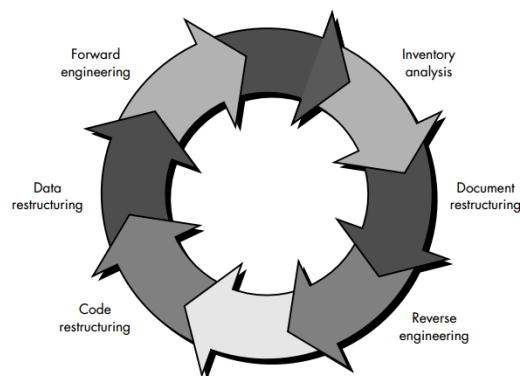
Universitas Muhammadiyah Malang memiliki sebuah *legacy system*, yakni layanan SMS *gateway*, dimana dapat melakukan *broadcast message* ke dosen dan karyawan serta pengiriman *single SMS*. Layanan ini berbasis *web* yang dapat diakses secara lokal pada domain SMS.umm.ac.id. Secara fungsional sistem ini dapat melakukan pengiriman *single SMS* dan *broadcast message* ke grup. Grup dibagi menjadi dua kategori, karyawan dan dosen. Pada kategori karyawan, sistem dapat melakukan *broadcast* ke semua karyawan, berdasarkan unit kerja, dan grup yang dapat dibuat manual. Sedangkan pada kategori dosen, sistem dapat melakukan *broadcast* ke semua dosen, berdasarkan fakultas, berdasarkan jurusan dan grup yang dapat dibuat manual. Secara non fungsional layanan ini belum memiliki *interface* yang responsif.

Selama sistem ini digunakan muncul beberapa kebutuhan baru, yakni kebutuhan akan fungsionalitas diantaranya saldo pulsa yang nantinya digunakan untuk membayar biaya penggunaan *broadcast message* di setiap fakultas, jurusan, dan unit kerja. Kemudian *automatic reply* yang diintegrasikan dengan data *request* sebagai media pengambilan informasi terkait akademik dan keuangan mahasiswa, *automatic broadcast* yang memberikan informasi secara terjadwal, serta memiliki *interface* yang responsif. Selain itu, kebutuhan akan *single SMS* dihapus karena dianggap tidak sesuai dengan kebutuhan saat ini. Dengan latar belakang masalah di atas, maka implementasi *software re-engineering* dapat membantu memudahkan permintaan dan pengiriman data atau informasi kepada mahasiswa, staf (dosen dan karyawan), dan orang tua mahasiswa di Universitas Muhammadiyah Malang melalui SMS.

2. Metode Penelitian

Metodologi pada penelitian ini, dengan melakukan studi pustaka, pengumpulan data, analisis kebutuhan, desain sistem, implementasi, dan pengujian sistem. Pada studi pustaka peneliti melakukan pengumpulan dokumen, jurnal, buku, dan literatur ilmiah lainnya yang berhubungan dengan *software re-engineering*. Pada penelitian ini peneliti menggunakan model proses dari *software re-engineering* dalam melakukan transformasi sistem. Berikut model proses dari *software re-engineering* atau rekayasa ulang perangkat lunak.

Gambar 1 merupakan sebuah paradigma *cyclical model* atau berulang. Berarti setiap kegiatan merupakan bagian dari paradigma yang mungkin dapat bertemu kembali karena memiliki sifat *cyclical*. Sehingga dapat menentukan ditahap mana proses dapat berhenti jika telah dianggap memenuhi atau selesai. Di bawah ini terdapat penjelasan proses-proses dari *software re-engineering*.



Gambar 1. Model Proses Software Re-engineering [3]

A. Inventory Analysis

Tahap melakukan analisis terhadap data inventarisasi dari aplikasi, mengenai pembuat aplikasi, proses bisnis, *size*, umur aplikasi dan lain-lain.

B. Document Restructuring

Tahap pembentukan ulang dokumentasi dari aplikasi yang akan direkayasa ulang dan akan terus di-*update* jika terdapat perubahan pada proses bisnis dari sistem, kebutuhan yang baru, dan lain-lain.

C. Reverse Engineering

Proses penemuan prinsip-prinsip teknologi dari suatu perangkat, objek, atau sistem melalui analisis strukturnya, fungsinya, dan cara kerjanya [5]. Dalam rekayasa ulang perangkat lunak proses yang dilakukan pada tahap *reverse engineering*, yakni melakukan abstraksi dari bentuk kode ke dalam bentuk konseptual yang biasanya berisi proses bisnis, variabel, dan fungsional dari aplikasi.

D. Code Restructuring

Tahap pembentukan ulang kode dari *legacy system* ke bentuk yang lebih baru dengan teknologi yang lebih baru, bahasa pemrograman yang lebih modern ataupun proses bisnis yang lebih baik, sehingga kode yang dibentuk mudah untuk dipahami, dilakukan pengujian, dan *maintenance*.

E. Data Restructuring

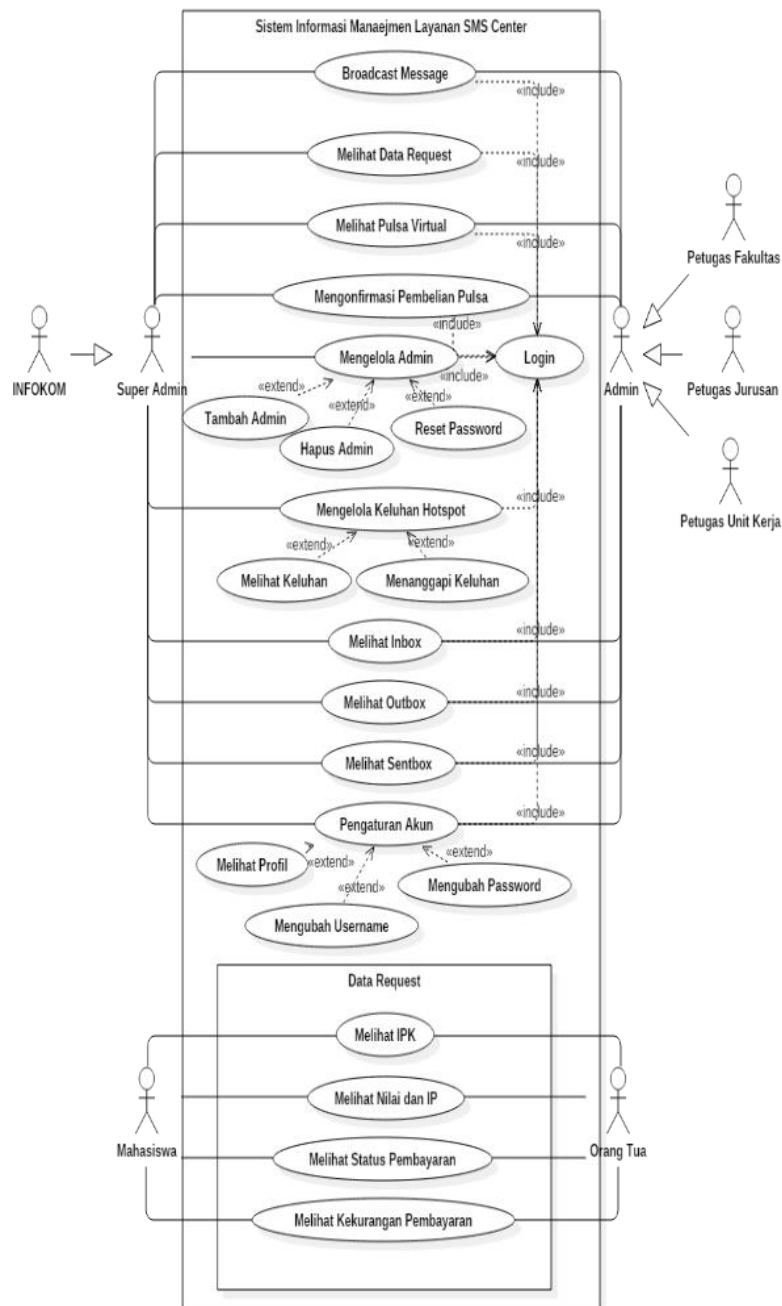
Tahap dilakukannya pembedahan data, data yang digunakan untuk dibedah dan dilakukannya restrukturisasi sehingga arsitektur, model, dan struktur data dapat jauh lebih baik dan mendukung untuk kelangsungan hidup sistem dalam jangka yang lebih panjang.

F. Forward Engineering

Proses abstraksi dari level yang lebih tinggi ke level yang lebih rendah. Ini adalah tahap dimana sistem direalisasikan atau diimplementasikan menjadi bentuk baru, yang sebelumnya sudah dilakukan teknik *reverse engineering* dan *restructuring*.

Setelah melakukan studi pustaka terhadap *software re-engineering*, tahap selanjutnya pada metodologi penelitian ini, yakni melakukan pengumpulan data yang digunakan untuk kepentingan penelitian. Data yang dibutuhkan, seperti data akademik dan keuangan mahasiswa. Peneliti melakukan pengambilan data di tempat studi kasus, yaitu Lembaga Informasi dan Komunikasi di Universitas Muhammadiyah Malang.

Tahap selanjutnya adalah tahap analisis kebutuhan dari *legacy system*, yakni SMS gateway dan kebutuhan-kebutuhan baru, sehingga dapat menyimpulkan seluruh kebutuhan dari sistem informasi manajemen layanan SMS center. Pada tahap analisis, peneliti mengimplementasikan model proses dari *software re-engineering*, yaitu tahap *reverse engineering*. *Reverse engineering* adalah proses penemuan prinsip-prinsip teknologi dari suatu perangkat, objek, atau sistem melalui analisis strukturnya, fungsinya, dan cara kerjanya [5]. Pada proses *reverse engineering* peneliti mengambil fitur atau fungsi yang terdapat pada *legacy system*, kemudian merangkum berbagai kebutuhan baru terkait kebutuhan fungsional, serta dilakukan pemilahan dan menyimpulkan kebutuhan untuk sistem. Sehingga dapat disimpulkan kebutuhan fungsional dari sistem informasi manajemen layanan SMS center seperti Gambar 2.

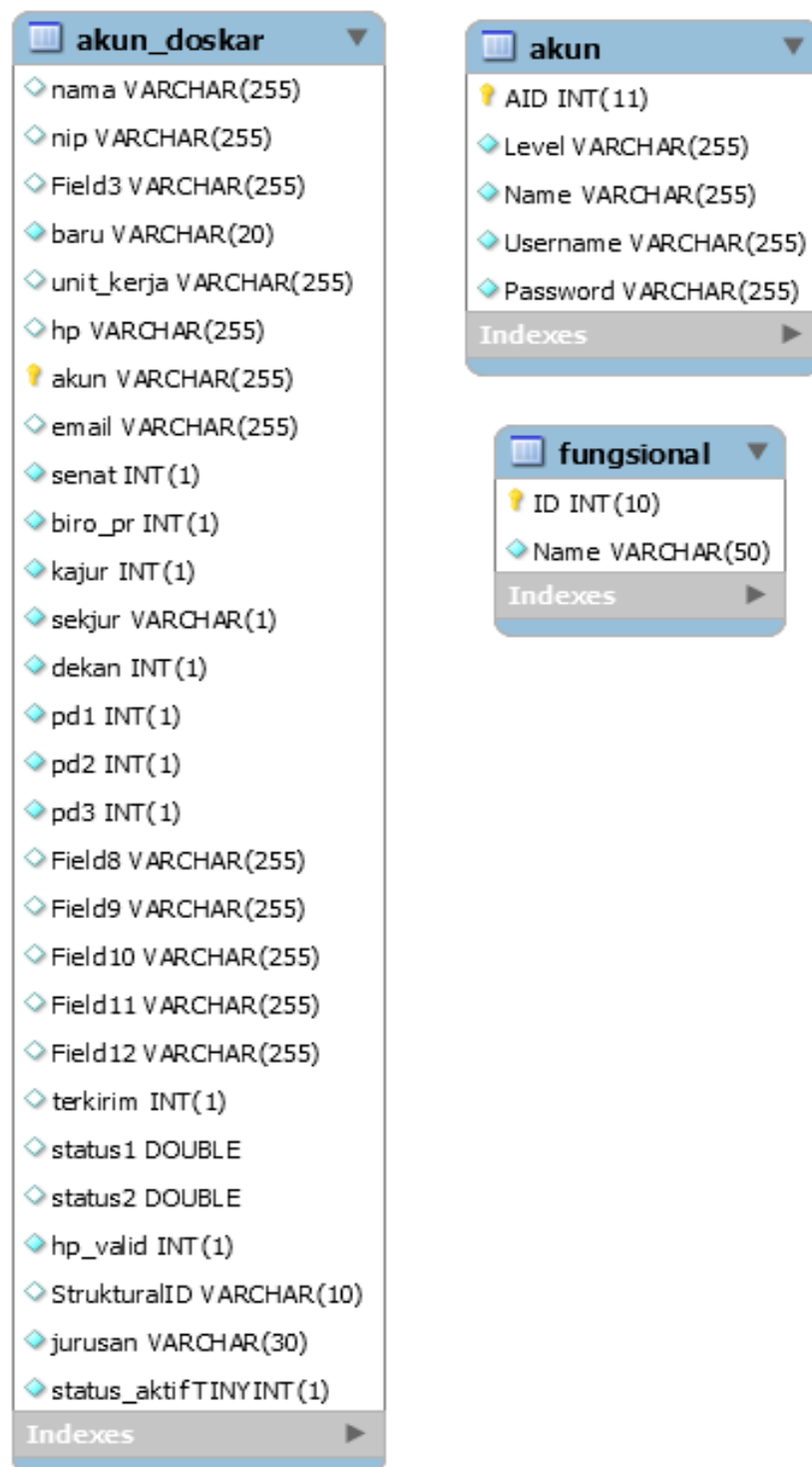


Gambar 2. Use Case Diagram SIM Layanan SMS Center

Setelah mengetahui kebutuhan dari sistem informasi manajemen layanan SMS center, tahap selanjutnya adalah mendesain sistem, pada tahap ini peneliti melanjutkan proses dari *software re-engineering*, yakni tahap *code restructuring* dan *data restructuring*. Pada tahap *code restructuring*, peneliti akan membentuk ulang kode program dari *legacy system* menjadi bentuk yang lebih baru dengan teknologi yang lebih baru. Sebagai contoh *legacy system* masih menggunakan teknologi lama untuk *markup language* atau HTML, sehingga *interface* dari *legacy system* belum responsif. Kemudian pada bagian *server side legacy system* masih menggunakan versi PHP yang lama dan memiliki beberapa *bug* untuk fungsionalitasnya, sehingga peneliti akan mentransformasikannya menggunakan versi PHP yang lebih baru dan menggunakan *framework* CodeIgniter serta menerapkan konsep MVC.

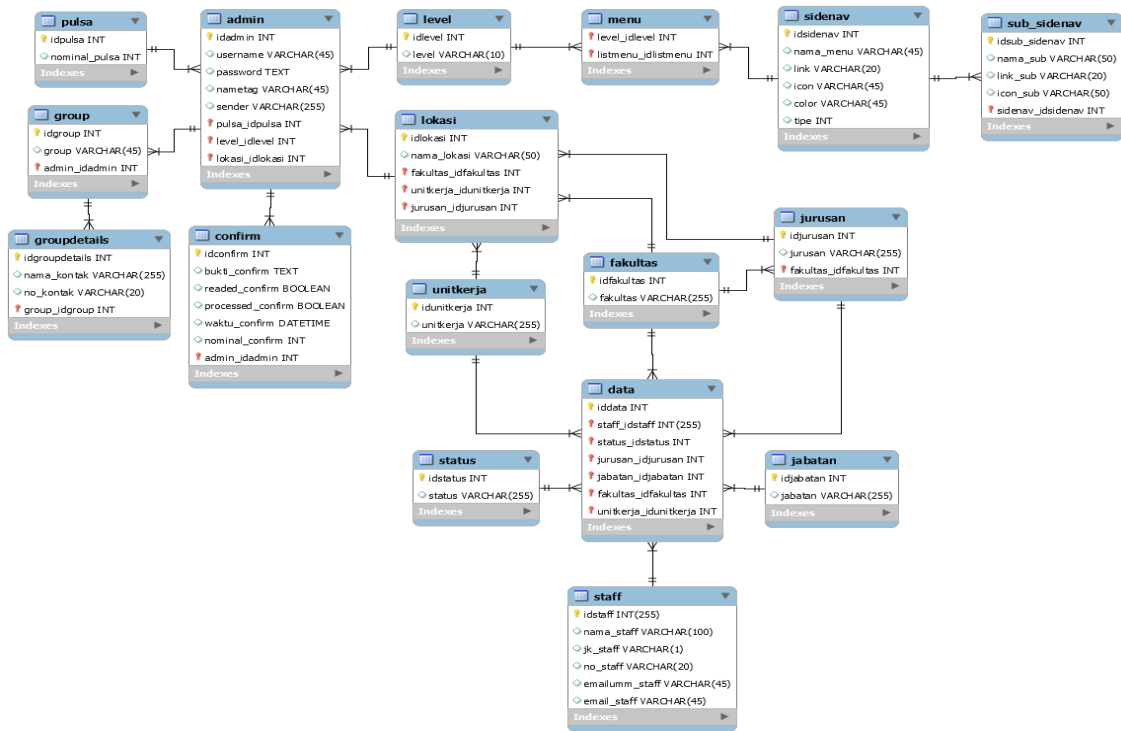
Setelah dilakukannya tahap *code restructuring*, peneliti melanjutkan ke proses *data restructuring* pada desain *database*. Pada *legacy system* memiliki arsitektur, model, dan struktur data yang tidak mendukung untuk kehidupan jangka panjang dari sistem. Sebagai contoh memiliki *field* yang tidak jelas namanya, menggunakan *engine* lama, yakni MyISAM

yang tidak mendukung relasi antar tabel seperti InnoDB. Desain *database legacy system* dapat dilihat seperti Gambar 3.



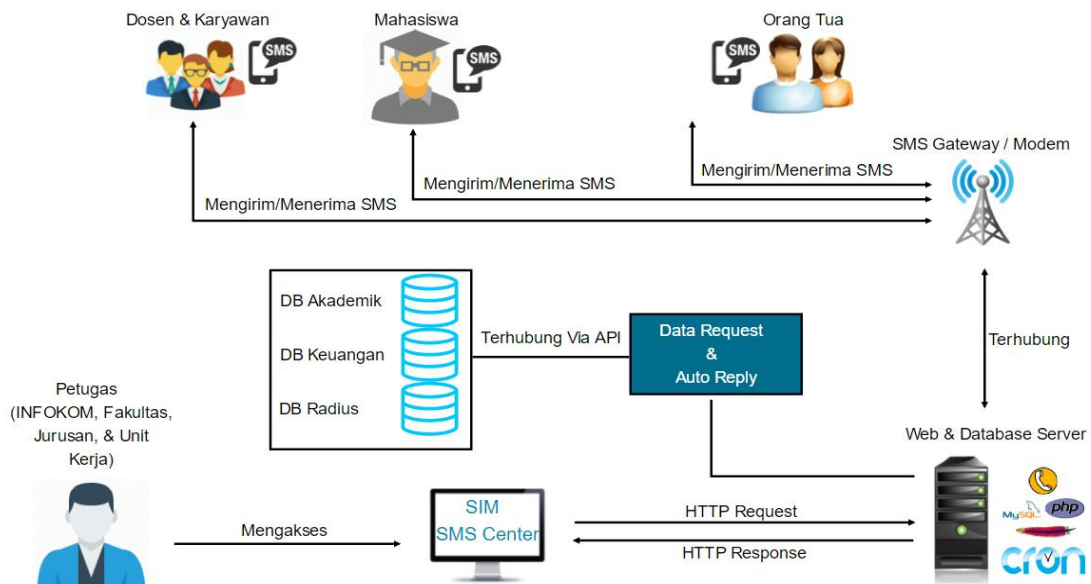
Gambar 3. Desain Database dari Legacy System

Dari desain *database legacy system* Gambar 3, maka akan dilakukan pembentukan ulang data, sehingga desain *database* pada sistem informasi manajemen layanan SMS center sebagai media penyimpanan data dari sistem akan diuraikan dengan entitas yang jelas dan seluruh relasi. Berikut desain *database* untuk sistem informasi manajemen layanan SMS center.



Gambar 4. Desain Database SIM Layanan SMS Center

Pada sistem informasi manajemen layanan SMS center memiliki arsitektur sistem yang dapat dijelaskan sebagai berikut:

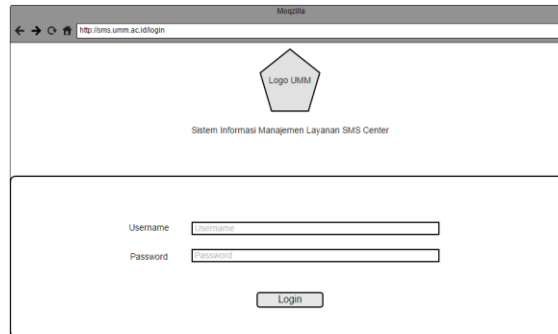


Gambar 5. Arsitektur Sistem dari SIM Layanan SMS Center

Pada Gambar 5 menjelaskan arsitektur sistem dari sistem informasi manajemen layanan SMS center. Sistem dapat diakses oleh aktor petugas INFOKOM (super admin) layaknya petugas fakultas, jurusan, dan unit kerja (admin). Saat diakses melalui web browser sistem akan meminta request web yang kemudian direspons oleh web server. Server berisi layanan web dan database, pada sub sistem data request terhubung ke database akademik, keuangan dan radius milik mahasiswa via API (Application Programming Interfaces) atau web service. Kemudian sistem juga terhubung pada modem sebagai gateway dari SMS yang mengirim dan menerima SMS dari dan ke orang tua mahasiswa, dosen, serta karyawan.

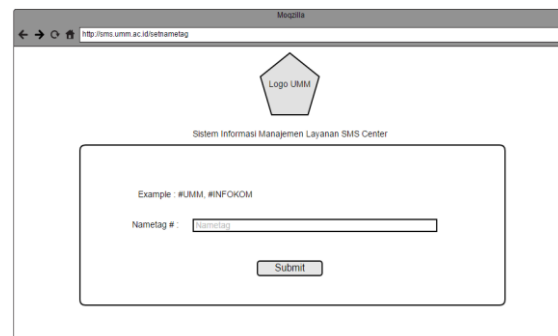
Adapun rancangan *interface* dari sistem informasi manajemen layanan *SMS center* ini menggunakan satu *interface* untuk aktor super *admin* dan *admin* akan dipisah berdasarkan level pada *source code (logic)* rancangan *interface* dapat dilihat melalui gambar 6, Gambar 7, dan Gambar 8.

1. Tampilan halaman *login*



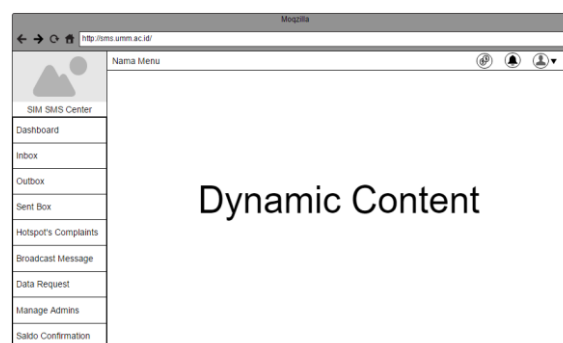
Gambar 6. Tampilan Halaman Login

2. Tampilan halaman *set name tag*



Gambar 7. Tampilan Halaman Set Name Tag

3. Tampilan halaman utama



Gambar 8. Tampilan Halaman Utama

3. Hasil Penelitian dan Pembahasan

Pada penelitian ini akan dilakukan 3 pengujian terhadap sistem, yakni pengujian fungsional, non fungsional, dan pengujian untuk membandingkan sistem informasi manajemen layanan *SMS center* dengan *legacy system*, yaitu *SMS gateway*. Hasil dari pengujian tersebut dapat menyimpulkan hasil penelitian apakah penelitian sukses atau tidak. Berikut hasil pengujian terhadap fungsional, non fungsional, dan hasil perbandingan sistem informasi manajemen layanan *SMS center* dengan *legacy system (SMS gateway)*.

3.1. Pengujian fungsional

Adapun hasil pengujian terhadap fungsionalitas dari sistem informasi manajemen layanan SMS center dapat disimpulkan melalui Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1. Pengujian Fungsional pada Aktor Super Admin

Case	Kesimpulan Pengujian
<i>Login</i>	Berhasil
<i>Broadcast message</i>	Berhasil
Melihat log data <i>request</i>	Berhasil
Melihat pulsa virtual	Berhasil
Mengonfirmasi pembelian pulsa	Berhasil
Mengelola <i>admin</i>	Berhasil
Mengelola keluhan <i>hotspot</i>	Berhasil
Melihat kotak masuk	Berhasil
Melihat kotak keluar	Berhasil
Melihat kotak terkirim	Berhasil
Mengelola pengaturan akun	Berhasil

Tabel 2. Pengujian Fungsional pada Aktor Admin

Case	Kesimpulan Pengujian
<i>Login</i>	Berhasil
<i>Broadcast message</i>	Berhasil
Melihat log data <i>request</i>	Berhasil
Melihat pulsa virtual	Berhasil
Mengonfirmasi pembelian pulsa	Berhasil
Mengelola <i>admin</i>	Berhasil
Mengelola keluhan <i>hotspot</i>	Berhasil
Melihat kotak masuk	Berhasil
Melihat kotak keluar	Berhasil
Melihat kotak terkirim	Berhasil
Mengelola pengaturan akun	Berhasil

Tabel 3. Pengujian Fungsional pada Aktor Mahasiswa dan Orang Tua

Case	Kesimpulan Pengujian
Melihat IPK	Berhasil
Melihat IP dan nilai	Berhasil
Melihat status pembayaran	Berhasil
Melihat kekurangan pembayaran	Berhasil

3.2. Pengujian non fungsional

Adapun hasil pengujian terhadap non fungsional dari sistem informasi manajemen layanan SMS center dapat disimpulkan melalui Tabel 4.

Tabel 4. Pengujian Non Fungsional

Pengujian	Kesimpulan Pengujian
Level hak akses	Berhasil
Tampilan responsif	Berhasil
<i>Long text SMS</i>	Berhasil
<i>Name tag</i>	Berhasil
<i>Format helper</i>	Berhasil
<i>Automatic broadcast</i>	Berhasil
<i>Automatic reply</i>	Berhasil
<i>Data request authentication</i>	Berhasil
Aksebilitas	Berhasil

3.3. Perbandingan SIM layanan SMS center dengan legacy system

Adapun hasil perbandingan dari sistem informasi manajemen layanan SMS center dengan legacy system (SMS gateway) dapat disimpulkan melalui Tabel 5 dan Tabel 6.

Tabel 5. Perbandingan Fungsionalitas antara SIM Layanan SMS Center dengan Legacy System

No	Fungsional	Legacy System (SMS Gateway)	SIM Layanan SMS Center
1	Single Message	√	—
2	Broadcast Message	√	√
3	Keluhan Hotspot	√	√
4	Inbox	√	√
5	Outbox	√	√
6	Sent Box	√	√
7	Data Request	—	√
8	Pulsa Virtual	—	√
9	Kelola Admin	—	√
10	Pengaturan Akun	√	√

Tabel 6. Perbandingan Non Fungsional antara SIM Layanan SMS Center dengan Legacy System

No	Non Fungsional	Legacy System (SMS Gateway)	SIM Layanan SMS Center
1	Level Hak Akses	—	√
2	Tampilan Responsif	—	√
3	Long Text SMS	—	√
4	Name tag	—	√
5	Format Helper	—	√
6	Automatic Broadcast	—	√
7	Automatic Reply	—	√
8	Data Request Authentication	—	√
9	Aksesibilitas	√	√

4. Kesimpulan

Dari hasil pengujian dan dilakukannya perbandingan antara sistem informasi manajemen layanan SMS center dengan legacy system (SMS gateway) maka dapat diambil kesimpulan, yaitu:

- Seluruh pengujian baik kebutuhan fungsional dan non fungsional bekerja dengan baik pada sistem informasi manajemen layanan SMS center sesuai dengan apa yang telah direncanakan.
- Hasil perbandingan antara sistem informasi manajemen layanan SMS center dengan legacy system (SMS gateway) membuktikan bahwa transformasi perangkat lunak berhasil dilakukan dengan mengikuti proses-proses dari *software re-engineering*.

Penelitian ini masih memiliki banyak kekurangan, dan berharap semoga kekurangan tersebut dapat diperbaiki dan disempurnakan. Berikut adalah beberapa saran penulis untuk menjadikan sistem informasi manajemen layanan SMS center ini jadi jauh lebih baik ke depannya:

- Implementasikan teknologi Nodejs, Ajax atau teknologi lainnya untuk setiap fungsionalitas agar sistem menjadi *real-time* dan tidak butuh untuk melakukan *load content* atau *page refresh*.
- Membuat format pesan data *request* dapat dikelola menggunakan *database* sehingga jika ada perubahan format pesan tidak perlu masuk ke dalam *script*.
- Dari sub sistem data *request* dapat dikembangkan menggunakan teknologi lainnya selain dengan teknologi SMS, yakni dapat membuat *Bot* pada aplikasi *chat messaging* seperti *Telegram*, *Hangout*, atau yang lainnya sehingga mahasiswa atau orang tua dapat menggunakan opsi selain mengirim SMS yang mengeluarkan biaya pulsa per SMS. *Bot* adalah program otomatis yang menjalankan tugas sesuai kehendak *programmer*, di mana *Bot* akan ditugaskan untuk menjawab pertanyaan dari pengguna secara otomatis terkait data *request*.
- Lakukan pengujian UAT (*User Acceptance Test*) atau berarti pengujian terhadap seluruh pengguna yang terkait dengan sistem informasi manajemen layanan SMS center untuk

mengukur tingkat penerimaan pengguna terhadap sistem, apakah memenuhi kebutuhan pengguna atau belum, sehingga sistem yang dibuat benar-benar dapat diterima dan sesuai dengan kebutuhan pengguna.

Referensi

- [1] Demeyer Serge, Stephane Ducasse dan Oscar Nierstrasz, "Object Oriented Reengineering Patterns", Switzerland: Square Bracket Associates, 2013.
- [2] R. S. Pressman, "Software Engineering : A Practitioner's Approach Fifth Edition", New York: McGraw-Hill Companies, Inc, 2001.
- [3] I. Sommerville, "Software Engineering Seventh Edition", United Kingdom: Addison-Wesley, 2004.
- [4] Abbas, Jeberson dan Klinsega, "The Need of Re-engineering in Software Engineering", United Kingdom: IJET Publications, 2012.
- [5] Eilam, Eldad & Chikofsky, Elliot J, "Reversing: secrets of reverse engineering", John Wiley & Sons, 2007.