APLIKASI PERPUSTAKAAN SMAN-LIB DENGAN JAVA (STUDI KASUS SMA NEGERI DARMARAJA)

LIBRARY APPLICATION SMAN-LIB WITH JAVA (CASE STUDY SMA NEGERI SITURAJA)

Muhammad Fikri Akbar[1]

[1] Prodi D3 Manajemen Informatika, Fakultas Ilmu Terapan, Universitas Telkom,
[1]fx.akbar119@gmail.com

Abstrak

Perpustakaan adalah tempat untuk membaca dan meminjam buku. Di setiap sekolah biasanya terdapat perpustakaan, seperti halnya di SMA Negeri Situraja. Pengelolaan perpustakaan sangatlah penting, untuk melihat berapa besarnya minat siswa untuk membaca dan meminjam buku, serta mengelola dan mencatat buku apa saja yang tersedia di perpustakaan. Pengelolaan buku di perpustakaan SMA Negeri Situraja masih menggunakan pencatatan manual yang menjadikannya kurang praktis untuk mengetahui ketersediaan buku, sulitnya mengetahui letak buku yang akan di pinjam, sulitnya dalam membuat laporan bulanan, sulitnya mengetahui buku yang belum dikembalikan, dan sulitnya memberitahu kepada anggota bahwa sudah lewat masa peminjaman. Pada aplikasi ini terdapat fitur-fitur yang di harapkan dapat membantu dalam proses pengelolaan perpustakaan seperti dapat menyimpan data-data dalam bentuk digital kedalam database, mencatat dan menampilkan data pengunjung, pencatatan buku baru, pencatatan peminjaman, pencatatan pengembalian, melihat laporan bulanan, dan pengiriman *sms* pemberitahuan kepada anggota yang telat mengembalikan buku. Dalam pengerjaan aplikasi ini menggunakan metode waterfall, bahasa pemrograman yang digunakan adalah Java, netbeans untuk *IDE* nya, dan Gammu sebagai *library* untuk proses pengiriman *sms* pemberitahuannya. Aplikasi perpustakaan ini dapat digunakan dengan baik dan mempermudah dalam proses pencatatan buku yang masuk, pencatatan peminjaman buku, pencatatan pengembalian buku, pencatatan pengunjung perpustakaan, pencarian buku, pembuatan laporan, dan pengiriman *sms* pemberitahuan kepada anggota yang telat mengembalikan buku.

Kata kunci: Aplikasi, Perpustakaan, Waterfall, Java, Gammu

Abstract

The library is a place to read and borrow books. In every school there is usually a library, as well as in SMA Situradja. Library management is very important, to see how much interest the students to read and borrow books, as well as manage and record books are available in libraries. Management books in the library SMA Situradja still use manual recording which makes it impractical to determine the availability of the book, the difficulty of knowing the location of the books to be borrowed, the difficulty in making a monthly report, the difficulty of knowing the book that has not been restored, and the difficulty of telling the members that have passed loan period. In this application, there are features that are expected to assist in the management of such libraries can store data in digital form into a database, record and display the data of visitors, recording a new book, recording borrowing, recording returns, see the monthly reports, and delivery sms notification to members of the late return a book. In this application work using waterfall method, the programming language used is Java, IDE netbeans for her, and Gammu as a library for sending sms notification process. This library application can be used with both and simplify the process of recording incoming books, borrowing books recording, recording the return of books, library patrons recording, book search, report generation, and sms delivery notification to members of the late return a book.

Keywords: Applications, Library, Waterfall, Java, Gammu

1 Pendahuluan

1.1 Latar Belakang

Perpustakaan adalah tempat untuk membaca dan meminjam buku. Disetiap sekolah biasanya terdapat perpustakaan, seperti halnya di SMA Negeri Situraja. Pengelolaan perpustakaan sangatlah penting, untuk melihat seberapa besarnya minat siswa untuk membaca dan meminjam buku, serta mengelola dan mencatat buku apa saja yang tersedia di perpustakaan.

Pengelolaan buku di perpustakaan SMA Negeri Situraja masih menggunakan pencatatan manual yang menjadikannya kurang praktis untuk mengetahui ketersediaan buku, sulitnya mengetahui letak buku yang akan dipinjam, sulitnya dalam membuat laporan bulanan, sulitnya mengetahui buku yang belum dikembalikan, dan sulitnya memberitahu kepada anggota bahwa sudah lewat masa peminjaman. Ini yang melatarbelakangi untuk pembuatan Proyek Akhir ini.

Berdasarkan permasalahan tersebut, diusulkan untuk membuat aplikasi perpustakaan yang dapat menyimpan data-data dalam bentuk dijital ke dalam database, mencatat dan menampilkan data pengunjung, pencatatan buku baru, pencatatan peminjaman, pencatatan pengembalian, melihat laporan bulanan, dan pengiriman *sms* pemberitahuan kepada anggota yang telat mengembalikan buku.

1.2 Rumusan Masalah

Adapun yang menjadi rumusan masalah Proyek Akhir ini adalah sebagai berikut:

1. Bagaimana mengetahui ketersediaan buku, dan letak buku yang akan di pinjam oleh anggota?
2. Bagaimana mempermudah pencatatan buku masuk, pengunjung, peminjaman, dan pengembalian buku sehingga dapat menghasilkan laporan bulanan?
3. Bagaimana memberitahukan keterlambatan peminjaman kepada anggota?

1.3 Tujuan

Adapun yang menjadi tujuan dari Proyek Akhir ini adalah sebagai berikut:

1. Membuat aplikasi yang dapat menampilkan ketersediaan buku, dan letak buku yang akan dipinjam.
2. Membuat aplikasi yang dapat mencatat peminjaman, dan pencatatan pengembalian sehingga dapat menghasilkan laporan bulanan.
3. Membuat aplikasi yang dapat mengirimkan *sms* pemberitahuan informasi pengembalian buku kepada anggota yang terlambat mengembalikan buku.

1.4 Batasan Masalah

Adapun yang menjadi batasan masalah dari proyek akhir ini adalah sebagai berikut :

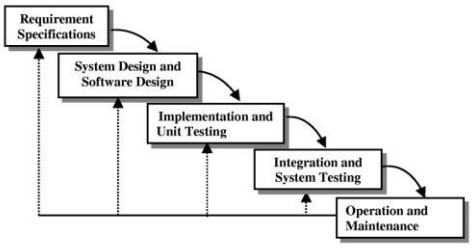
1. Aplikasi ini dipakai hanya pada SMA Negeri Situraja.
2. Aplikasi ini menggunakan bahasa Java sebagai bahasa pemrogrammanya, Apache dan MySQL dari aplikasi XAMPP untuk database dan servernya, Gammu sebagai *library* untuk *sms gateway*, ApachePOI sebagai *library* untuk pembuatan file excel, JfreeChart sebagai *library* untuk pembuatan grafik, dan JCalendar sebagai *library* untuk komponen Swing Jcalendar.
3. Aplikasi ini menggunakan jaringan LAN untuk menghubungkan aplikasi dengan databasenya.
4. Dalam proyek akhir ini tidak dilakukan integration, operation, dan maintenance.

1.5 Definisi Operasional

Aplikasi perpustakaan ini adalah aplikasi yang dibuat dengan menggunakan bahasa Java yang menyediakan layanan dan fitur untuk kegiatan mengelola data perpustakaan yang ada di SMA Negeri Situraja. Data-data yang dikelola meliputi pengelolaan data pengunjung, pengelolaan data petugas, data anggota perpustakaan, data buku, data peminjaman, data pengembalian, dan laporan bulanan. Aplikasi ini dapat mengirimkan sms pemberitahuan apabila anggota perpustakaan belum mengembalikan buku sampai batas waktu peminjaman. SMAN-LIB adalah nama yang diberikan untuk aplikasi perpustakaan ini.

1.6 Metode Pengerjaan

Dalam pembuatan perangkat lunak ini digunakan metode *Software Development Life Cycle* (SDLC) dengan metode *Waterfall*. Adapun proses dalam model *waterfall* ini antara lain :



Gambar 1.1
Model Waterfall [1]

a. Analisis Kebutuhan (*Requirement Specifications)*

Pada tahap ini dimulai dari identifikasi masalah serta mengumpulkan data-data tentang pengelolaan perpustakaan dengan cara:

1. Studi literatur

Teknik pengumpulan data dengan cara mengumpulkan literatur yang diperoleh dari jurnal, artikel internet, buku, serta *e-book* yang berkaitan dengan permasalahan yang sedang dibahas.

2. Wawancara

Wawancara dilakukan secara langsung kepada pihak sekolah sebanyak 2 kali, yaitu pada tanggal 4 Maret 2014 dan 8 Mei 2014. Wawancara dengan pihak sekolah ini bertujuan untuk mendapatkan keterangan, pendapat, serta ide untuk membantu menyelesaikan permasalahan ini. Hasil wawancara dapat dilihat pada lampiran 1.

Kemudian data tersebut akan digunakan sebagai acuan dalam membangun aplikasi perpustakaan ini.

b. Desain Sistem dan Desain Perangkat Lunak (*System Design and Software Design)*

Tahap selanjutnya dilakukan *system and software design* atau perancangan perangkat lunak dengan menggunakan Use Case Diagram, Class Diagram, Sequence Diagram, flowmap, ERD, dan relasi antar tabel.

c. Implementasi dan Pengujian Unit (*Implementation and Unit Testing)*

Tahap pembuatan program dengan melakukan transformasi dari desain yang telah dirancang ke bahasa yang dikenal oleh komputer (*coding*) dengan menggunakan bahasa pemrograman JAVA. Untuk pengujiannya akan menggunakan blackbox dan whitebox testing.

d. Integrasi dan Pengujian Sistem (*Integration and System Testing)*

Setelah perangkat lunak selesai dibuat maka tahap selanjutnya menguji perangkat lunak tersebut. Pengujian bertujuan untuk memastikan semua fungsionalitas yang disusun bekerja dengan benar.

e. Pengoperasian dan Pemeliharaan (*Operation and Maintenance)*

Perangkat lunak yang telah lolos uji langsung diimplementasikan atau diterapkan. Pada proyek akhir ini tidak dilakukan implementasi.

1.7 Jadwal Pengerjaan

Berikut ini adalah jadwal perencanaan pengerjaan Proyek Akhir ini.

Tabel 1.1
Jadwal Pengerjaan Proyek Akhir

Kegiatan	Bulan Ke-1	Bulan Ke-2	Bulan Ke-3	Bulan Ke-4	Bulan Ke-5	Bulan Ke-6	Bulan Ke-7
Analisis Kebutuhan							
1. Studi Literatur							
2. Wawancara							
Desain Sistem							
Membuat Desain UML							
Membuat Desain ERD							
Membuat Desain Mock Up							
Implementasi Dan Pengujian Unit							
Membuat Aplikasi SMAN-LIB Anggota							
Membuat Aplikasi SMAN-LIB ADMIN							
Melakukan Pengujian White Box							
Melakukan Pengujian Black Box							
Membuat Dokumentasi							

2 Tinjauan Pustaka

2.1 Perpustakaan

Berdasarkan Undang-Undang Republik Indonesia Nomor 43 Tahun 2007 Tentang Perpustakaan. Perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak, dan/atau karya rekam secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka. Perpustakaan Sekolah/Madrasah adalah perpustakaan yang merupakan bagian integral dari kegiatan pembelajaran dan berfungsi sebagai pusat sumber belajar untuk mendukung tercapainya tujuan pendidikan yang berkedudukan di sekolah/madrasah.

2.2 Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output.* [2]

2.3 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.[3]

2.4 Gammu

Gammu adalah nama proyek serta nama utilitas baris perintah, yang dapat di gunakan untuk mengontrol telepon. Gammu ditulis dalam bahasa C dan dibangun di atas libGammu. [4]

2.5 Xampp

XAMPP adalah software web server apache yang di dalamnya tertanam server MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat website yang dinamis. XAMPP sendiri mendukung dua system operasi yaitu windows dan Linux. Untuk linux dalam proses penginstalannya menggunakan command line sedangkan untuk windows dalam proses penginstalannya menggunakan interface grafis sehingga lebih mudah dalam penggunaaan XAMPP di Windows di banding dengan Linux.

DIdalam XAMPP ada 3 komponen utama yang di tanam di dalamnya yaitu web server Apache, PHP, dan MySQL. [5]

2.6 Flowmap

Flowmap adalah diagram yang menggambarkan aliran dokumen pada suatu prosedur kerja di organisasi, dan merupakan diagram alir yang menunjukkan arus dari dokumen, aliran data fisis, entitas sistem informasi dan kegiatan operasi yang berhubungan dengan sistem informasi. [6]

2.7 Entity Relationship Diagram (ERD)

Diagram Entitas-Relasi atau Entity Relationship Diagram (ERD) adalah diagram yang menggambarkan keterhubungan antar data secara konseptual. Penggambaran keterhubungan antar data ini didasarkan pada anggapan bahwa dunia nyata terdiri dari kumpulan objek yang disebut entitas (entity), dan hubungan yang terjadi diantaranya yang disebut relasi (relationship). [7]

2.8 Unified Modeling Languange (UML)

Unified Modeling Languange(UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. [8]

2.9 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavio*r) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case*: [8]

2.10 Sequence Diagram

Sequence diagram adalah diagram interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu [8].

2.11 Class Diagram

Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem dan merupakan tipe diagram yang paling ditemui dalam pemodelan sistem berbasis *object-oriented*. *Class Diagram* memperlihatkan sekumpulan *class, interface*, dan relasi yang ada di dalamnya [8].

2.12 SMS

SMS (Short Message Service) secara umum dapat diartikan sebagai sebuah service yang memungkinkan ditransmisikannya pesan text pendek dari dan ke mobile phone, fax, mesin, atau IP address. Disebut pesan text pendek karena pesan yang dikirimkan hanya berupa karakter text dan tidak lebih dari 160 karakter. Pentransmisian SMS menggunakan kanal signalling, bukan kanal suara, sehingga kita dapat saja menerima SMS walaupun kita sedang melakukan komunikasi suara. [9]

2.13 SMS Gateway

SMS Gateway adalah aplikasi SMS dimana pesan yang diterima dan dikirimkan menggunakan bantuan *Gateway Device* yang terintegrasi dengan *database server* yang dapat mendistribusikan pesan SMS secara otomatis. [10]

2.14 Black Box Testing

Black box testing merupakan strategi *testing* dimana hanya memperhatikan atau memfokuskan kepada faktor fungsionalitas dan spesifikasi perangkat lunak. Berbeda dengan *white box*, *black box testing* tidak membutuhkan pengetahuan mengenai, alur internal (internal path), struktur atau implementasi dari *software under test* (SUT). Tidak seperti *white box testing* yang dilakukan pada awal proses pengujian, *black box testing* dilakukan dibeberapa tahapan berikutnya. Karena *black box testing* memang ditujukan untuk mengabaikan struktur kontrol tetapi lebih terfokus terhadap *information domain*. Pengujian dirancang untuk menjawab beberapa pertanyaan sebagai berikut:

1. Bagaimana validitas fungsionalitasnya diuji?
2. Bagaimana perilaku sistem dan performasi diuji?
3. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik?
4. Apakah sistem secara khusus sensitif terhadap nilai input tertentu?
5. Bagaimana batasan-batasan kelas data diisolasi?
6. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem? [11]

2.15 White Box Testing

White box testing adalah metode desain kasus uji yang menggunakan struktur kontrol desain prosedural untuk memperoleh kasus uji. Dengan menggunakan metode pengujian *White box*, perekayasa sistem dapat melakukan kasus uji yang:

1. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali
2. Menggunakan semua keputusan logis pada sisi true dan false
3. Mengeksekusi semua loop pada batasan mereka dan pada batas operasional mereka
4. Menggunakan struktur data internal utuk jaminan validitasnya. [12]

2.16 Flow Graph

Flow graph terdapat notasi standar yang terdiri dari 2 (dua) buah notasi, yaitu lingkaran dan panah. Lingkaran atau *flow graph node*, digunakan untuk menyatakan satu atau beberapa *statement* prosedural yang ada dalam *source code* atau algoritma. Panah atau disebut sebagai *edge* atau *link*, digunakan untuk menyatakan aliran kendali atau alur perjalanan logika. Tanda panah pada *flow graph* ini memiliki fungsi yang sama dengan tanda panah pada *flow chart*. Sebuah *node* dapat digunakan untuk menggambarkan beberapa baris program sekaligus selama baris-baris program itu berada pada satu kelompok. Yang dimaksud dengan satu kelompok adalah baris-baris program dapat dieksekusi secara berurutan (sekuensial) dan di dalamnya tidak terdapat perubahan struktur kendali program, misalnya terdapat sintaks kondisional atau pengulangan. *Edge* mulai digambarkan apabila terdapat perubahan struktur kendali program. [11]

2.17 Cyclomatic Complexity

Cyclomatic complexity merupakan suatu sistem pengukuran untuk perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logika suatu program. Pada metode *basis path testing*, hasil dari penghitungan *cyclomatic complexity* digunakan untuk menentukan banyaknya *independent paths* (alur bebas).

Independent path adalah jalur yang melintasi minimal satu kumpulan *statement* program atau

sebuah kondisi pada program dan menghubungkan *node* awal *(start)* dengan *node* akhir *(end)*. Sebuah *independent path* minimal melewati sebuah *edge* baru (belum pernah dilewati) dan alur tersebut belum pernah ditemukan (unik).

Cyclomatic complexity pada sebuah dipengaruhi oleh 3 (tiga) hal, yaitu:

1. Banyaknya *edge* (panah)
2. Banyaknya *node* (lingkaran)
3. Banyaknya *predicate node*

Terdapat 2 (dua) persamaan yang digunakan, yaitu :

V(G)=E-N+2 **atau**

V(G)=P+1

Keterangan :

V(G) : *Cylomatic complexity* untuk *flow graph* G

E : Jumlah *edge* (panah)

N : Jumlah *node* (lingkaran)

P : Jumlah *predicate node*

Independent path yang ditemukan dalam *flow graph* merupakan *basis path* yang mewakili seluruh alur logika. [13]

Basis path testing merupakan suatu metoda yang digunakan dalam teknik *white box testing*. Metoda *basis path* ini sangat bermanfaat bagi seorang penguji perangkat lunak dalam menentukan:

1. Ukuran kompleksitas logika dari suatu struktur program, *procedure* atau *function*.
2. Menggunakan nilai kompleksitas untuk menentukan *basis set* (himpunan dasar) alur logika yang akan dieksekusi.

Dengan metode ini, penguji perangkat lunak dapat menentukan kasus uji yang diterapkan pada *basis set* dan menjamin bahwa setiap baris program akan dieksekusi minimal satu kali.

Metoda *basis path testing* ini memerlukan masukan berupa *source code* atau algoritma dari suatu perangkat lunak. Setelah *source code* atau algoritma ini didapatkan, maka langkah-langkah yang dilakukan dalam menjalankan metoda ini adalah sebagai berikut :

1. Menggambarkan alur logika menggunakan notasi yang telah ditentukan. Gambar alur logika ini disebut sebagai *flow graph*.
2. Menentukan *cyclomatic complexity*.
3. Menentukan *basis set* dari alur-alur yang independen.

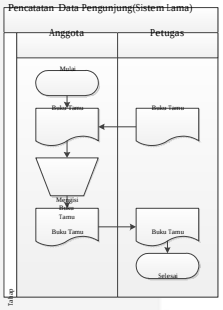
Membuat data uji untuk kemudian dieksekusi pada setiap alur. [13]

3 Analisis dan Perancangan

3.1 Analisis Sistem Lama

3.1.1 Flowmap Proses Pengisian Buku Tamu

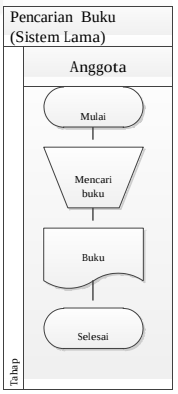
Berikut adalah flowmap dari proses pengisian buku tamu yang sedang berjalan. Anggota meminta buku tamu kepada petugas, petugas menyerahkan buku tamu, anggota mengisi buku tamu, lalu anggota menyerahkan buku tamu yang sudah diisi kepada petugas.



Gambar 3.1
Flowmap Proses Pengisian Buku Tamu

3.1.2 Flowmap Proses Pencarian Buku

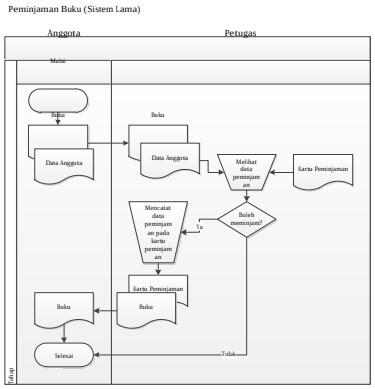
Berikut adalah flowmap dari proses pencarian buku yang sedang berjalan. Anggota langsung mencari buku ke tiap rak.



Gambar 3.2
Flowmap Proses Pencarian Buku

3.1.3 Flowmap Proses Peminjaman Buku

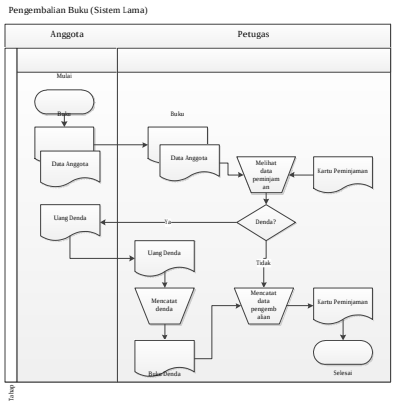
Berikut adalah flowmap dari proses peminjaman buku yang sedang berjalan. Anggota menyerahkan buku kepada petugas, petugas mengambil kartu peminjaman dan mengecek data peminjaman, jika anggota boleh meminjam, maka petugas akan mencatat data peminjaman.



Gambar 3.3
Flowmap Proses Peminjaman Buku

3.1.4 Flowmap Proses Pengembalian Buku

Berikut adalah flowmap dari proses pengembalian buku yang sedang berjalan. Anggota menyerahkan buku dan data anggota, petugas melihat data peminjaman dari kartu peminjaman, jika ada denda maka anggota harus membayar denda, dan petugas mencatat pada buku denda, lalu mencatat data pengembalian pada kartu peminjaman, jika tidak ada denda petugas langsung mengisi kartu peminjaman dengan data pengembalian.



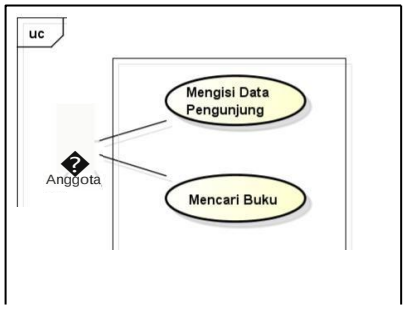
Gambar 3.4

Flowmap Proses Pengembalian Buku

3.1.5 Use Case Diagram

3.1.5.1 Use Case Diagram SMAN-LIB Anggota

Berikut adalah use case diagram dari aplikasi perpustakaan SMAN-LIB Anggota pada SMA Negeri Situraja.



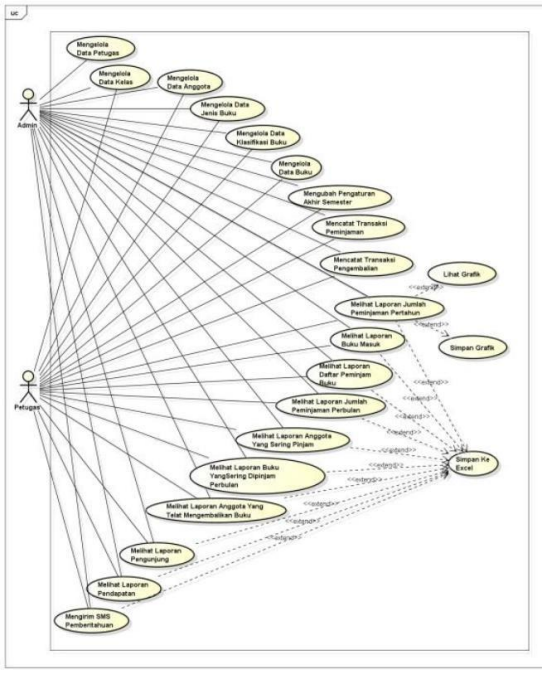
Gambar 3.5

Use Case SMAN-LIB Anggota

Keterangan : Pada aplikasi SMAN-LIB Anggota aktornya adalah hanya anggota. Disana anggota dapat melakukan pengisian data pengunjung dan pencarian data buku dengan memasukan kata pencarian dari buku yang akan dicari, nanti sistem akan menampilkan data buku secara lengkap termasuk apakah buku tersedia dan tempat dimana buku itu disimpan.

3.1.5.2 Use Case Diagram SMAN-LIB Admin

Berikut adalah use case diagram dari aplikasi perpustakaan SMAN-LIB Admin pada SMA Negeri Situraja.



Gambar 3.6
Use Case SMAN-LIB Admin

Keterangan: Fitur-fitur yang ada pada aplikasi SMAN-LIB Admin adalah mengelola data petugas, mengelola data anggota, mengelola data kelas, mengelola data buku, mengelola data jenis, mengelola data klasifikasi, mengatur akhir semester, mencatat data peminjaman, mencatat data pengembalian, melihat laporan jumlah peminjaman pertahun, melihat laporan buku masuk, melihat laporan daftar peminjam buku, melihat laporan jumlah peminjaman perbulan, melihat laporan

anggota yang sering pinjam, melihat laporan buku yang sering dipinjam, melihat laporan anggota yang telat mengembalikan buku, melihat laporan pengunjung, melihat laporan pendapatan, dan mengirimkan sms pemberitahuan kepada anggota yang telat mengembalikan buku. Kata mengelola disini adalah mencari, melihat, menambah, mengubah, dan menghapus. Pada laporan jumlah peminjaman pertahun laporan dapat dilihat dalam bentuk tabel maupun bentuk grafik, laporan yang berbentuk tabel dapat disimpan menjadi *file* excel dan laporan yang berbentuk grafik dapat disimpan menjadi *file* gambar. Selain laporan jumlah peminjaman pertahun semua laporan dapat disimpan menjadi *file* excel. Pada aplikasi SMAN-LIB Admin aktornya adalah admin dan petugas, dimana yang membedakan kedua aktor ini adalah admin berhak mengelola data petugas sedangkan petugas tidak.

4 Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan tujuan dan hasil pengujian aplikasi telah sesuai dengan apa yang direncanakan. Aplikasi dapat digunakan untuk mencatat data pengunjung, mencari data buku, mengelola data petugas, mengelola data anggota, mengelola data kelas, mengelola data buku, mengelola data jenis, mengelola data klasifikasi, mengatur akhir semester, mencatat data peminjaman, mencatat data pengembalian, melihat laporan jumlah peminjaman pertahun, melihat laporan buku masuk, melihat laporan daftar peminjam buku, melihat laporan jumlah peminjaman perbulan, melihat laporan anggota yang sering pinjam, melihat laporan buku yang sering dipinjam, melihat laporan anggota yang telat mengembalikan buku, melihat laporan pengunjung, melihat laporan pendapatan, dan mengirimkan sms pemberitahuan kepada anggota yang telat mengembalikan buku.

4.2 Saran

Berikut adalah saran untuk pengembangan aplikasi kedepan:

1. Pendaftaran anggota lewat sms.
2. Pencetakan kartu anggota.

5 Daftar Pustaka

[1] I. Sommerville, Software Engineering, Jakarta: Erlangga, 2003.

[2] H. Jogiyanto, Analisis dan Desain Sistem Informasi: Pendekatan terstruktur teori dan praktis aplikasi bisnis, Yogyakarta: Andi, 1999.

[3] W. Y. Anto, "JAVA," 3 September 2011. [Online]. Available: https://wh3lly.wordpress.com/2011/09/21/pengantar-java/#more-88.

[4] Wammu, "Gammu," 2014. [Online]. Available: http://wammu.eu/gammu/.

[5] Yogajiwanjaya, "Pengertian dan Kegunaan XAMPP," 29 Juni 2014. [Online]. Available: http://www.pusatdesainweb.com/2014/06/29/pengetian-dan-kegunaan-xampp/. [Accessed 22 Januari 2015].

[6] I. Gartina, "Pemodelan dengan Flowmap," Bandung, 2009.

[7] d. Eddy Prasetyo Nugroho, Rekayasa Perangkat Lunak, 1 ed., Bandung: Politeknik Telkom, 2009.

[8] R. A.S and M. Shalahuddin, Rekayasa Perangkat Lunak, Bandung: Informatika, 2013.

[9] Abusajid, "Layanan Pesan Singkat," 15 November 2006. [Online]. Available: http://www.mobileindonesia.net/mengenal-sms-short-message-service/.

[10] A. Budidoyo, "Sms Gateway," 30 Juni 2010. [Online]. Available: http://www.slideshare.net/ABYSlides/sms-gateway. [Accessed 30 Juni 2014].

[11] B. Beizer, Control Flow testing. Cognizant Technology Solutions: Software Testing, 2010.

[12] E. P. Nugroho, Rekayasa Perangkat Lunak, Bandung: Politeknik Telkom, 2009.

[13] R. R. Pressman, Software Enginnering, A Practitioner's Approach, McGraw-Hill, 2005.

[14] d. I. G. Siska Komala Sari, Analisis Desain dan Sistem Informasi, Bandung: Politeknik Telkom.