

PERANCANGAN
SISTEM INFORMASI PERPUSTAKAAN
BERBASIS *WEB APPLICATION*

Tesis

Untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-2
Program Studi Magister Sistem Informasi



Oleh :

Yudie Irawan

NIM : J4F008036

PROGRAM PASCASARJANA
UNIVERSITAS DIPONEGORO
SEMARANG

2011

Abstraks

Digital library system dan *library management system* menjawab perkembangan kebutuhan masyarakat dalam hal pelayanan pada lembaga perpustakaan. *Digital library system* memberikan kontribusi pada pengembangan *digital resource* sehingga *digital resource* dapat diakses melalui internet tanpa mengenal waktu dan tempat. *Library management system* memberikan kontribusi pada perkembangan otomatisasi pengolahan data keanggotaan, sirkulasi dan kataloging. Peneliti mengembangkan *digital library system* dan *library management system* dengan mengintegrasikan arsitektur dua sistem tersebut. Integrasi arsitektur ditempuh dengan memasukkan komponen *library management system* ke dalam arsitektur *digital library system*. Penerapan teknologi *web application* diperlukan agar komponen tersebut dapat terintegrasi dengan komponen *digital library system*.

Sistem yang dihasilkan dari pengembangan ini memiliki keunggulan pemanfaatan aplikasi peminjaman, keanggotaan dan *kataloging* yang *sharable* melalui internet, sehingga aplikasi itu dapat digunakan bersama – sama. Informasi katalog dapat disampaikan antar perpustakaan, tanpa meninggalkan fungsi *digital library* dalam pemanfaatan bersama *digital resource* yang bersumber dari *upload* masing – masing pustakawan.

Kata Kunci : *Digital Library System, Library Management System, Integrasi, Web Application*

Abstract

Digital library systems and library management system has been able to meet the development needs of society in terms of library service in institutions. Digital library system contributes the development of digital resource digital resource that can be accessed via the Internet. Library management system contributed to the development of automation membership data processing, circulation and cataloging. In this thesis develop digital library systems and library management system by integrating these two systems architecture. In this thesis is to develop a new concept of digital library systems and library management system by integrating these two systems architecture. Integration architecture implemented by inserting component library management system into the digital library system architecture. Web application technology required for these components in order to be integrated with the digital library system components.

The new system has the advantage of this development application utilization of borrowing, membership and kataloging to a sharable over the internet, so applications that can be used together. Information can be delivered between the library catalog, without leaving the digital library function in the utilization of shared digital resources derived from uploading by each librarian.

Key words : digital library system, library management system, integration, web application

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan digital library tidak lepas dari perkembangan teknologi informasi. *Digital library* dibangun diatas teknologi *web*, yang memungkinkan pengaksesan koleksi oleh anggota, kapan dan dimanapun posisi pengguna berada melalui internet. Di awal pengembangan *digital library*, para peneliti lebih memberikan perhatian pada layanan sistem dan peningkatan jumlah *digital resource* yang dihasilkan, seperti halnya yang disampaikan oleh Cleveland (1998), bahwa *digital library* – *digital library* yang ada akhirnya menjadi kumpulan sumber daya yang berbeda dan juga sistem yang berbeda, hal ini disebabkan karena sistem – sistem itu dikembangkan untuk melayani komunitas tertentu dan kelompok - kelompok *user* yang ada.

Pada perkembangan selanjutnya, isu yang muncul adalah interoperabilitas. Seperti yang disebutkan oleh Vullo (2010), interoperabilitas (*interoperability*) adalah sebuah aktifitas yang mengacu pada kemampuan sistem yang beragam dan organisasi untuk bekerja sama. Aktifitas ini sebenarnya telah diramalkan oleh Cleveland, yaitu bahwa pasca perkembangannya(*digital library*), interoperabilitas antara digital library (dalam hal arsitektur, *metadata* dan format dokumen) juga mungkin terjadi dalam pengembangan

sistem yang dibangun relatif terbatas untuk tujuan dan komunitas yang spesifik.

Isu interoperabilitas ini kemudian melahirkan *digital library system* yang *sharable* penggunaannya, salah satunya dibangun oleh Information Society Technology, yaitu DELOS Digital Library Management System (DLMS). DelosDLMS merupakan aplikasi *open source digital library system* yang dapat didownload dan bebas digunakan sebagai sistem *digital library system* oleh pihak manapun. Selain DelosDLMS ada juga proyek yang disebut Koha Integrated Library System (ILS) yaitu sistem *open source* yang juga bebas *download* dan digunakan oleh institusi manapun untuk pengelolaan *library information system*.

Dari penjelasan tersebut terdapat dua istilah dalam pengembangan sistem perpustakaan, yaitu *digital library system* dan *integrated library system* atau *library management system*. Persamaan dari kedua sistem tersebut adalah *open source*, diperbolehkan untuk digunakan dan dikembangkan pada perpustakaan mana saja. Perbedaannya yaitu *digital library system* digunakan untuk mengelola *digital resource* dan cara pengaksesannya, sedangkan *library management system* digunakan untuk mengelola sirkulasi, keanggotaan, *kataloging* dan *online public access catalog*.

Pada sisi lain kebutuhan mencari referensi menggunakan otomatisasi perpustakaan meningkat pada semua tingkatan penyelenggara pendidikan, baik dasar, menengah dan perguruan

tinggi, karena disadari bahwa pemanfaatan *non-digital resource* tetap tidak dapat digantikan oleh *digital resource*. Oleh karena itu kebutuhan pencarian *non-digital resource* tetap akan berlangsung dan berkembang semakin meluas. Keunggulan *digital library system* adalah memberikan kontribusi akses *download digital resource* dari manapun melalui internet. Namun *digital library system* tidak menyediakan informasi tentang *non-digital resource*. Sebaliknya, pada *library management system* memberikan kontribusi pada pengelolaan operasional perpustakaan yang secara otomatis menyediakan informasi *non-digital resource*. Namun pada *library management system* tidak menyediakan *digital resource* dan pemanfaatannya.

1.2 Perumusan Masalah

Pemanfaatan *digital resource* tidak menggantikan pentingnya *non-digital resource*. Kebutuhan pengguna untuk mencari *non-digital resource* meningkat keluar dari lingkup lokal. Keberadaan *digital resource* dan *non-digital resource* saling mendukung dalam memenuhi kebutuhan pengguna. Teknologi web yang diterapkan dalam *digital library* mampu memperluas cakupan pencarian referensi, namun tidak menyediakan informasi *non-digital resource*. Sebaliknya *library management system* mampu menyediakan informasi *non-digital resource* secara baik pada lingkup lokal, namun tidak menyediakan *digital resource*.

Dari ulasan tersebut, sistem perpustakaan perlu dikembangkan agar mampu memenuhi kebutuhan pemanfaatan *digital resource* dan *non-digital resource* pada skala yang lebih luas. Pengembangan sistem informasi perpustakaan yang dibahas

dalam tesis ini adalah dengan mengintegrasikan arsitektur antara *digital library system* dan *library management system* agar dapat mengadopsi masing – masing keunggulan yang dimiliki oleh kedua sistem tersebut.

1.3 Batasan Masalah

Pengembangan sistem baru dilakukan dengan cara menggabungkan dua arsitektur dari *digital library system* dan *library management system*. Untuk memenuhi hal tersebut maka seluruh aplikasi dibangun menggunakan *web application*. Dalam tesis ini, pembangunan *web application* menggunakan tahapan *Waterfall* model. Pada tahap perancangan menggunakan *unified modelling language* dan pada tahap penerapan sistem menggunakan *open source object oriented programming Hypertext PreProcessor* (PHP). Dan penyimpanan data menggunakan *MySQL database management system*.

Dalam tesis ini tidak membahas tingkat keamanan sistem dan jumlah maksimal user yang dapat dilayani oleh sistem.

1.4 Keaslian Penelitian

Dalam penyusunan tesis ini telah dilakukan analisa dari beberapa jurnal internasional mengenai pengembangan sistem perpustakaan digital. Jurnal tersebut antara lain:

1. Artikel yang disampaikan oleh Candella, dkk (2006) menyebutkan bahwa DelosDLMS adalah prototipe dari *Digital Library (DL) Management System* generasi berikutnya. DL ini diwujudkan dengan menggabungkan berbagai fungsi DL yang khusus disediakan oleh mitra jaringan DELOS. Saat ini DelosDLMS

menggabungkan pencarian teks dan audio visual, menawarkan visualisasi terbaru untuk informasi, dan fasilitas umpan balik yang relevan, menyediakan Novel *interfaces*, memungkinkan informasi yang diakses tercatat dan terproses, mengintegrasikan dan memproses sensor data stream, dan terakhir, dari sudut pandang rekayasa sistem, mudah dikonfigurasi dan disesuaikan dengan kehandalan dan pengembangannya yang mudah. Prototipe ini didasarkan pada platform OSIRIS/ISIS, sebuah middleware environment yang dikembangkan oleh ETH Zurich dan diterapkan di Universitas Basel.

Arsitektur Delos mengadopsi pendekatan berlapis, dan mengatur komponen konstituen menjadi tiga tingkatan:

- a. *Application Frame*, yaitu set perpustakaan dan subsistem yang mendukung pengoperasian komponen sistem *Digital Library* (DL) lainnya;
- b. *Enabling Component*, yang menyediakan fungsionalitas untuk mendukung kerjasama antar komponen mengimplementasikan aplikasi DL,
- c. *DL Application Component*, yang menyediakan fungsionalitas DL khusus untuk pengguna akhir.

2. Menurut Rosasco dan Bakke (2005), sistem Koha dikembangkan berdasarkan pada kebutuhan komunitas perpustakaan yang berkolaborasi yang sedang berkembang untuk mencapai tujuannya secara teknologi. Koha terus mengembangkan fasilitasnya untuk memperluas dan memenuhi kebutuhan dasar penggunanya. Perpustakaan umum pertama di Amerika Utara mendanai dukungan penggunaan MARC dan mulai menggunakan Koha pada tahun 2003. Koha sekarang digunakan di seluruh dunia, terutama di perpustakaan kecil. Pada arsitektur Koha terdiri dari beberapa modul atau *function* yang dikirim atau diakses ke web antara lain Circulation, Acquisition/Budget, Patrons, Non MARC,

MARC, Fines, Admin, Export, Import. Modul – modul tersebut mengakses database dengan DB Wrapper. Daemon adalah sistem *client server* yang digunakan untuk mendukung fungsi – fungsi agar dapat menjalankan aplikasi.

Tesis ini mengembangkan sistem perpustakaan dengan cara menggabungkan arsitektur *digital library system* dan *library management system*. Penggabungan kedua arsitektur ini ditempuh dengan memasukkan beberapa komponen *library management system* (*Circulation, Fines, Admin, Import dan Export*) ke dalam arsitektur *digital library system*. Dengan metode ini maka komponen tersebut dapat digunakan bersama melalui *web browser* (*sharable*) secara *online*. Pada lapis *Enabling Component* terdapat *DBWrapper* aplikasi DBMS yang mengelola alur kerja sama dengan *Process Engine*. Sedangkan dalam *Application Framework* terdapat komponen *Web Server*, tempat dimana *web application* diletakkan.

Dengan arsitektur ini koleksi yang dapat diakses tidak hanya *e-book*, namun juga informasi ketersediaan buku (*non digital*), status peminjaman, dan jumlah eksemplar. Aplikasi ini berjalan dalam satu *platform* sehingga tidak memerlukan penyesuaian teknologi maupun migrasi data diantara para pengguna sistem. Proses migrasi ditangani oleh SysAdmin apabila diperlukan.

1.5 Manfaat Penelitian

Manfaat yang dihasilkan dari perancangan sistem ini adalah memberikan kontribusi terhadap ilmu pengetahuan pada bidang sistem informasi khususnya dalam

pengembangan sistem informasi perpustakaan dan perkembangan teknologi yang mampu diadopsi untuk meningkatkan pelayanan kepada pengguna akhir.

1.6 Tujuan Penelitian Tesis

- a. Membuat perancangan terintegrasi *library management system* dengan *digital library system* menggunakan rekayasa sistem *Waterfall Model* dan perangkat pemodelan *Unified Modelling Language (UML)*.
- b. Mengimplementasikan rancangan sistem ke dalam sistem aplikasi menggunakan bahasa pemrograman PHP dan *database MySQL*.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Artikel yang disampaikan oleh Candella, dkk (2006) menyebutkan bahwa DelosDLMS adalah prototipe dari *Digital Library (DL) Management System* generasi berikutnya. DL ini diwujudkan dengan menggabungkan berbagai fungsi DL yang khusus disediakan oleh mitra jaringan DELOS. Saat ini DelosDLMS menggabungkan pencarian teks dan audio visual, menawarkan visualisasi terbaru untuk informasi, dan fasilitas umpan balik yang relevan, menyediakan *Novel interfaces*, memungkinkan informasi yang diakses tercatat dan terproses, mengintegrasikan dan memproses sensor *data stream*, dan terakhir, dari sudut pandang rekayasa sistem, mudah dikonfigurasi dan disesuaikan dengan kehandalan dan pengembangannya yang mudah. Prototipe ini



leware
rapkan
ambar

Gambar 1 : *The Digital Library System Reference Architecture*

(Delos)

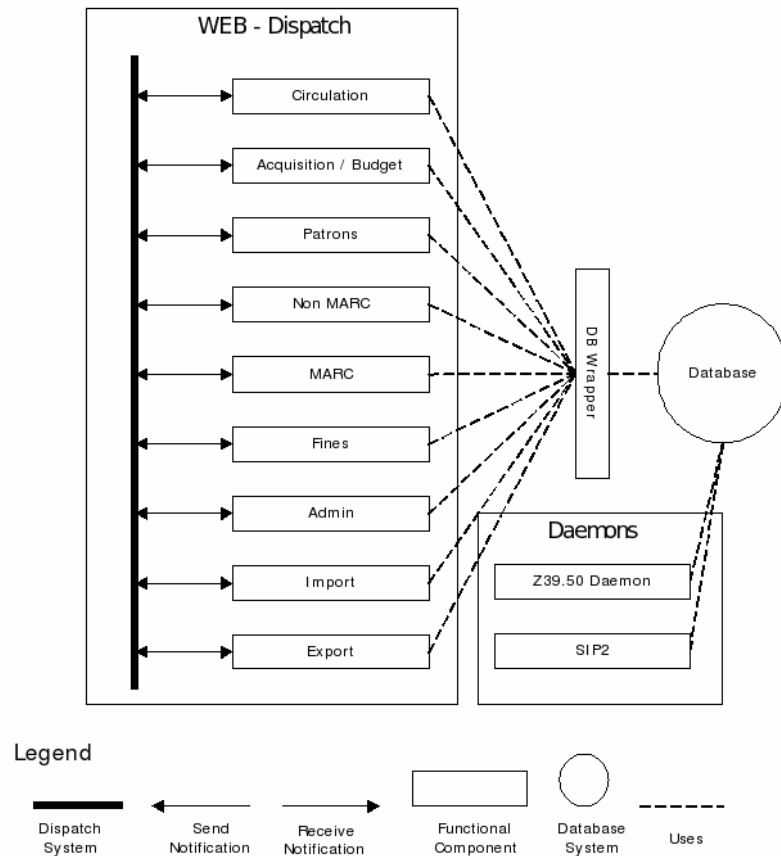
Sumber: Candella, 2006

Arsitektur Delos mengadopsi pendekatan berlapis, dan mengatur komponen konstituen menjadi tiga tingkatan:

- d. *Application Frame*, yaitu set perpustakaan dan subsistem yang mendukung pengoperasian komponen sistem *Digital Library* (DL) lainnya;
- e. *Enabling Component*, yang menyediakan fungsionalitas yang dibutuhkan untuk mendukung kerjasama antar komponen mengimplementasikan aplikasi DL,
- f. *DL Application Component*, yang menyediakan fungsionalitas DL khusus untuk pengguna akhir.

Menurut Rosasco dan Bakke (2005), sistem Koha dikembangkan berdasarkan pada kebutuhan komunitas perpustakaan yang berkolaborasi yang sedang berkembang untuk mencapai tujuannya secara teknologi. Koha terus mengembangkan fasilitasnya untuk memperluas dan memenuhi kebutuhan dasar penggunanya. Perpustakaan umum pertama di Amerika Utara mendanai dukungan penggunaan MARC dan mulai menggunakan Koha pada tahun 2003. Koha sekarang digunakan di seluruh dunia, terutama di perpustakaan kecil. Pada arsitektur Koha terdiri dari beberapa modul atau *function* yang dikirim atau diakses ke web antara lain Circulation, Acquisition/Budget, Patrons, Non MARC, MARC, Fines, Admin, Export, Import. Modul – modul tersebut mengakses database dengan DB Wrapper. Daemon adalah sistem

client server yang digunakan untuk mendukung fungsi – fungsi agar dapat menjalankan aplikasi. Arsitektur Koha dapat dilihat pada Gambar 2.



Gambar 2 : General Koha Usage/Architecture Layout

Rosasco dan Bakke (2006)

Menurut Chen, dkk.(2007), dalam paper The Online Bioinformatics Resources Collection at the University of Pittsburgh Health Sciences Library Sistem--a one-stop gateway to online bioinformatics databases and software tools, adalah aplikasi pencarian yang berbasis *web application* dengan fungsi menjembatani gap antara berkembangnya informasi yang

dibutuhkan oleh para ahli biologi dan peneliti medis dan juga meningkatnya pertumbuhan sumber – sumber *online bioinformatics* yang cepat. Aplikasi bernama Online Bioinformatica Resource Collection (OBRC) adalah sebuah aplikasi yang menggunakan teknologi pencarian data dan berisi 1542 database utama *bioinformatics* yang *online*, Aplikasi ini dioperasikan diatas *Zope Web Application server*, menggunakan implementasi *vivisimo clustering engine*, untuk menyempurnakan hasil pencarian dimana secara otomatis dapat mengatur hasil pencarian dalam kategori – kategori yang diciptakan secara dinamis dan berdasarkan pada informasi secara tekstual dari *record* yang dihasilkan. OBRC juga dibangun dengan sistem manajemen konten (Content Management System) dimana dapat dijumpai pada situs web University of Pittsburgh, Health Science Library System yaitu pada <http://www.hslls.pitt.edu/guides/genetics/obrc>

Abdullah dan Zainab (2007), dengan paper The digital library as an enterprise: the Zachman approach, menguji kebutuhan perpustakaan digital bagi para *stakeholder* dan bagaimana sebuah perpustakaan digital yang kolaboratif dirancang untuk memenuhi kebutuhan-kebutuhan yang mendesak. Perpustakaan digital kolaboratif sudah dipahami; dikandung untuk mendukung sekolah menengah dan informasi dari siswa diperlukan di dalam melaksanakan proyek-proyek berbasis sekolah. Pembahasan dalam paper ini mengacu pada *Zachman Frameworks* yaitu matrik yang

menyediakan bentuk baku dan sangat terstruktur tentang cara memandang dan mendefinisikan suatu perusahaan.

Gütl and Schmaranz (1999), menjelaskan prototype bernama Digilib2000 yang menggunakan *Web-based technology*. Inti dari sistem ini adalah sebuah sistem cerdas yang memiliki peran sebagai penyimpan, penganalisa, dan modul perantara informasi. Aplikasi ini didukung dengan kamus dan ensiklopedia yang dapat dimanfaatkan secara bersamaan pada saat aktifitas eksplorasi dilakukan. Sistem ini juga menyediakan konten yang sangat terstruktur sebagaimana halnya sebuah jurnal virtual. Pengguna juga dapat memiliki sub – perpustakaan dan memberikan nama pada sub perpustakaan tersebut. Aplikasi ini memiliki sistem pembayaran yang terintegrasi.

2.2. Landasan Teori

Landasan teori yang digunakan untuk menjadi dasar penelitian ini bersumber pada buku referensi, jurnal penelitian dan halaman web dari internet.

2.2.1. *Web Application*

Dewasa ini *web application* dikenal sebagai aplikasi yang diakses melalui *web browser* dan melalui jaringan seperti Internet atau intranet. Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstal perangkat lunak pada kemungkinan ribuan komputer klien merupakan keunggulan teknologi ini, selain juga untuk *cross-platform compatibility*. Termasuk

aplikasi *web common webmail*, penjualan ritel *online*, *online* pelelangan, wiki dan banyak fungsi lainnya ([en.wikipedia.org/wiki/ Web_application](http://en.wikipedia.org/wiki/Web_application), 2009). Pada jurnal yang ditulis oleh Xu, dkk, (2005), menitikberatkan pada efektifitas dan efisiensi sebuah testing terhadap aplikasi yang berbasis *web application* dengan membandingkan dua metode yaitu *Semantic Label* dan *XML description technique*. Lei Xu dan timnya mengembangkannya dengan melengkapi mekanisme *feedback control* pada pembangunan aplikasi agar lebih menyempurnakan kualitas sistem.

Edinburgh (2005) membahas sebuah pendekatan pengujian pada *web application*. Dalam metode pendekatannya analisa aliran data akan dianggap sebagai *Function Level Testing*, *Function Cluster Level Testing*, *Object Level Testing* dan *Web Application Level Testing*, dari level terendah hingga level tertinggi.

2.2.2. PHP

PHP (Hypertext Preprocessor), merupakan bahasa pemrograman pada sisi server yang memperbolehkan programmer menyisipkan perintah – perintah perangkat lunak web server (apache, IIS, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke browser yang me-*request*-nya, contohnya adalah bagaimana memungkinkannya memasukkan tanggal sekarang pada sebuah halaman web setiap kali tampilan tanggal dibutuhkan. Sesuai dengan fungsinya yang berjalan di sisi server maka PHP adalah bahasa pemrograman yang digunakan untuk membangun teknologi *web application*. (Kevin Yank, 2002)

PHP telah menjadi bahasa *scripting* untuk keperluan umum yang pada awalnya hanya digunakan untuk pembangunan web yang

menghasilkan halaman web dinamis. Untuk tujuan ini, kode PHP tertanam ke dalam dokumen sumber *HTML* dan diinterpretasikan oleh server web dengan modul PHP prosesor, yang menghasilkan dokumen halaman web. Sebagai bahasa pemrograman untuk tujuan umum, kode PHP diproses oleh aplikasi penerjemah dalam modul baris - baris perintah dan melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program di channel output standar. Hal ini juga dapat berfungsi sebagai aplikasi grafis. PHP tersedia sebagai prosesor untuk server web yang paling modern dan sebagai penerjemah mandiri pada sebagian besar sistem operasi dan komputer *platform*. (wikipedia.org, 2010)

2.2.3. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional ([RDBMS](#)) yang didistribusikan secara gratis dibawah [lisensi GPL](#) (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; [SQL](#) (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata ([DBMS](#)) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh

pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis web ([wordpress](#)), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL memiliki beberapa keistimewaan, antara lain :

1. **Portabilitas.** MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. **Perangkat lunak sumber terbuka.** MySQL didistribusikan sebagai [perangkat lunak sumber terbuka](#), dibawah lisensi [GPL](#) sehingga dapat digunakan secara gratis.
3. **Multi-user.** MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. **'Performance tuning'**, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. **Ragam tipe data.** MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. **Perintah dan Fungsi.** MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. **Keamanan.** MySQL memiliki beberapa lapisan keamanan seperti level [*subnetmask*](#), nama [*host*](#), dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. **Skalabilitas dan Pembatasan.** MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman ([*records*](#)) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. **Konektivitas.** MySQL dapat melakukan koneksi dengan klien menggunakan protokol [TCP/IP](#), [Unix socket \(UNIX\)](#), atau [Named Pipes \(NT\)](#).
10. **Lokalisasi.** MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. **Antar Muka.** MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi [API](#) (*Application Programming Interface*).
12. **Klien dan Peralatan.** MySQL dilengkapi dengan berbagai peralatan yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. **Struktur tabel.** MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam [PostgreSQL](https://www.postgresql.org/) ataupun [Oracle](https://www.oracle.com/). (id.wikipedia.org/wiki/MySQL)

2.2.4. UML(*Unified Modeling Language*)

UML telah menjadi sebuah standar bahasa dalam pembuatan *software blueprints*. Pemodelan dengan *UML* mampu memberikan visualisasi, spesifikasi, konstruksi dan dokumentasi dari sebuah sistem perangkat lunak. Penggunaan *UML* tepat untuk pemodelan sistem perusahaan sampai *web-based application* dan bahkan untuk *real time embedded system*. (Booch,1999)

UML merupakan sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan *UML* dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena *UML* juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, *UML* tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. (Yanti, 2003)

Model *UML* terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model. *UML* mendefinisikan diagram-diagram sebagai berikut:

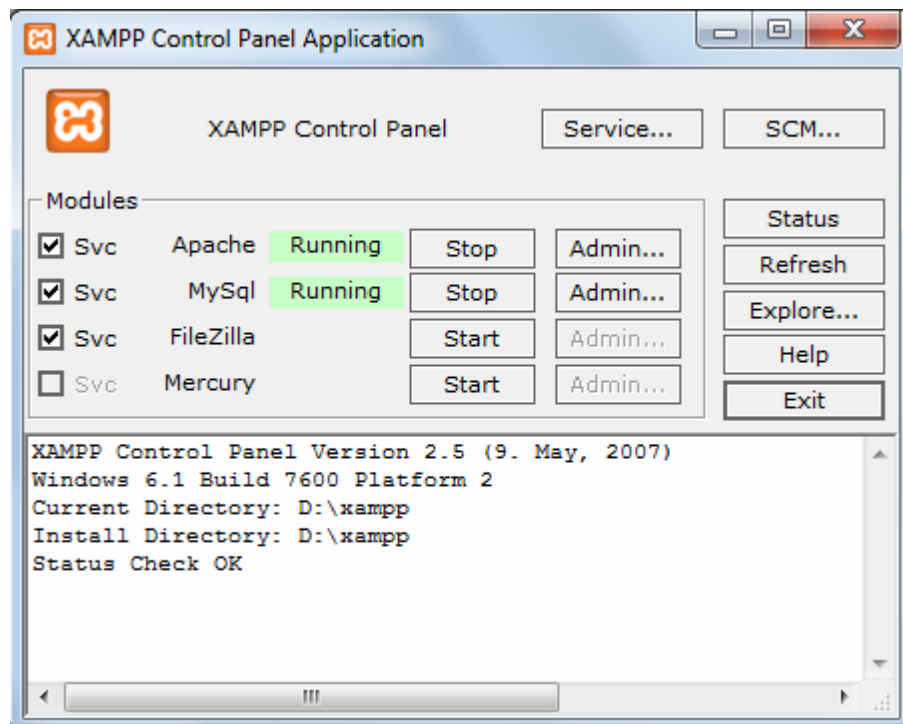
- a. *class diagram*
- b. *object diagram*
- c. *component diagram*
- d. *deployment diagram*
- e. *use case diagram*
- f. *sequence diagram*
- g. *collaboration diagram*
- h. *statechart diagram*
- i. *activity diagram*

Pendekatan *object oriented* menunjukkan kinerja yang signifikan dibandingkan dengan pemrograman prosedural klasik, karena penggunaan memory lebih sedikit sesuai dengan jumlah instruksi yang digunakan.

2.2.5. XAMPP

XAMPP adalah sebuah software *web server* apache yang didalamnya sudah tersedia database *server* MySQL dan dapat mendukung pemrograman PHP. XAMPP merupakan *software* yang

mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support* (PHP 4 dan PHP 5) dan beberapa *module* lainnya.



Gambar 3. Jendela dan panel pada XAMPP

Perbedaan versi untuk *Windows operating system* sudah dalam bentuk instalasi grafis dan yang *Linux* dalam bentuk file terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk *Windows* adalah memiliki fitur untuk mengaktifkan sebuah *server* secara grafis, sedangkan *Linux* masih berupa perintah-perintah di dalam *console*. Oleh karena itu yang versi untuk *Linux* sulit untuk dioperasikan. Dulu XAMPP untuk *Linux* dinamakan LAMPP, sekarang diganti namanya menjadi XAMPP FOR LINUX. Tampilan awal XAMPP dapat dilihat pada Gambar 3.

2.2.6. CSS (*Cascading Style Sheets*)

Pada versi HTML yang terdahulu, *web browser* mengontrol tampilan (*rendering*) dari setiap halaman web. Jika menggunakan elemen H1 pada (*large heading*) pada web dokumen, *browser* akan merender elemen tersebut. Dengan adanya CSS, programmer dapat mengontrol bagaimana browser me-render halaman web. Mengaplikasikan CSS pada halaman web dapat memberikan tampilan yang lebih menarik dan spesifik sesuai dengan tema pada sebuah *web site*. Teknologi CSS memberikan fasilitas untuk menentukan *style* (misal; *spacing, margins*) dari elemen halaman web terpisah dari struktur dokumen web (*section headers, body text, , links*). Pemisahan tersebut memberikan peningkatan yang lebih besar dalam pengaturan *web pages*, dan membuat perubahan – perubahan *style* dalam dokumen dapat dilakukan lebih cepat dan lebih mudah. (en.wikipedia, 2010)

2.2.7. Sistem Informasi Perpustakaan

Pengertian perpustakaan digital berkembang menjadi sebuah organisasi yang menyediakan sumber daya, termasuk didalamnya staff khusus, bertugas memilih, menyusun, dan menawarkan akses intelektual, menerjemahkan, mendistribusikan, memelihara integritas, menjamin keutuhan dari waktu ke waktu hasil koleksi digital sehingga karya – karya tersebut dapat dibaca dan secara

ekonomis tersedia untuk dimanfaatkan oleh komunitas tertentu maupun sekumpulan komunitas. (Waters,1998)

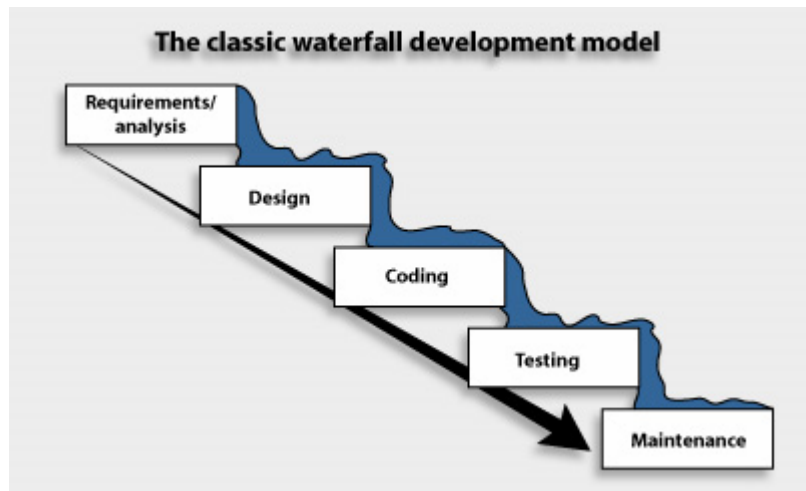
Disampaikan oleh Cleveland (1998), bersumber pada beberapa jurnal dan hasil diskusi sebelumnya maka definisi karakteristik perpustakaan digital antara lain :

- a. perpustakaan digital merupakan perpustakaan yang mewakili perpustakaan tradisional yang menyediakan baik koleksi digital dan koleksi tradisional, termasuk koleksi media. Sehingga perpustakaan tersebut memangkas biaya koleksi elektronik dan biaya kertas.
- b. Perpustakaan digital juga termasuk didalamnya adalah materi digital yang sebenarnya berada diluar perpustakaan secara fisik namun memiliki *link* dari perpustakaan digital lainnya.
- c. Perpustakaan digital juga akan berisi segala proses dan pelayanan yang menjadi tulang belakang dan jaringan syaraf dalam perpustakaan digital. Walau bagaimanapun, beberapa tradisional proses yang akan membangun pola kerja perpustakaan digital, yang akan disempurnakan dan ditingkatkan untuk mengakomodasi perbedaan antara media digital yang baru dan media tradisional .

Sushan Dhakal (2007), memfokuskan pada pemanfaatan *Open Digital Library* pada rencana pendidikan masa kini, dan deskripsi teknik dari arsitekturnya. Paper tersebut berdasarkan pada penelitian dan pembangunan *Digital Library Research Lab* menggunakan OAI_PMH dan PHP *based harvester*.

2.2.8. *Waterfall Model*

Dalam pembangunan perangkat lunak menggunakan *Linear Sequential / Waterfall Model*. Metode ini merupakan model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Seiring dengan kebutuhan maka *Waterfall Model* mengalami modifikasi yaitu ; *Shasimi* dan *Roice's Final model*. *Classic Waterfall Model* dapat dilihat pada gambar 4.



Gambar 4. Classic Waterfall Model

Pada *Third Computer Journal Lecturer* terdapat ulasan dengan judul *Discussion on The Ideal of Program Correctness by Tony Hoare* , artikel tersebut membandingkan efektifitas beberapa metode pengembangan sistem termasuk *Waterfall Model*. Pada perkembangannya terdapat *Waterfall* model telah dimodifikasi, salah satunya adalah *Roice's Final Model*. (en.wikipedia, 2010).

Fase-fase dalam *Waterfall Model* menurut Pressman, (2005), yaitu:

1. *Analysis*

Mengumpulkan kebutuhan data perbandingan dan perkembangan teknologi *web service* dan *web application* secara lengkap kemudian dianalisa kelayakannya untuk dijadikan metode dalam pengembangan sistem informasi perpustakaan beserta kebutuhan database yang harus dipenuhi oleh program yang akan dibuat. Fase ini dikerjakan agar menghasilkan desain sistem yang lengkap.

2. *Design*

Desain software memiliki berbagai tahapan yang berfokus pada atribut program yang jelas yaitu : *data structure*, *software architecture*, *interface representations*, dan detail prosedur(*algorithm*). Proses desain menterjemahkan kebutuhan pengguna dalam sebuah dokumen aplikasi yang dapat diperkirakan kualitasnya sebelum proses *coding* dimulai. Pada tahap ini menggunakan model *Unified Modelling Language* sebagai perangkat pembuatan desain *software*.

3. *Code*

Tahap *Coding* adalah tahap dimana hasil desain *software* diterjemahkan ke dalam bahasa yang dapat dimengerti oleh komputer. Dalam penelitian ini menggunakan bahasa pemrograman PHP karena bahasa pemrograman ini mendukung aplikasi berteknologi web. Database yang dihasilkan disimpan dalam aplikasi database MySQL.

4. *Test*

Pengujian sistem menggunakan *Black box testing*, yang menganggap aplikasi sebagai sebuah kotak hitam dimana user mengabaikan sistem bisnis yang diadopsinya. *Blackbox testing* menitikberatkan pada kesesuaian suatu komponen terhadap spesifikasi.

BAB III

CARA PENELITIAN

3.1 Bahan Penelitian

Bahan penelitian diambil dari sumber data primer dan sekunder.

3.1.1 Data primer diteliti secara langsung dari perpustakaan dengan metode wawancara dan observasi di lapangan, antara lain ;

- a. data buku, dari data – data buku diperoleh atribut untuk disimpan dalam *record* yang berkaitan dengan buku yaitu : judul buku, pengarang, penerbit, edisi buku, jenis buku, jumlah halaman, jumlah eksemplar dan tahun terbit
- b. data anggota, meliputi atribut anggota yang akan digunakan untuk melengkapi field pada *storage* anggota. Field tersebut antara lain ; nomor anggota, nama anggota, jenis kelamin, alamat, kota dan status pendidikan.
- c. data sirkulasi peminjaman, meliputi atribut yang berkaitan dengan sirkulasi peminjaman, antara lain ; tanggal peminjaman, id peminjam, tanggal kembali, denda, judul buku yang dipinjam, dan lama pinjam.
- d. data petugas, dari data petugas dapat dicatat beberapa atribut yang berkaitan dengan penyimpanan data petugas, yaitu ; nomor induk petugas, nama petugas, jenis kelamin, alamat, kota, dan jabatan petugas.

3.1.2 Data sekunder meliputi data – data yang diperoleh dari buku literatur, studi pustaka mengenai arsitektur *digital library sistem*, arsitektur *library management sistem*,, *web application*, *MySQL database*, *waterfall model* dan teknologi pemrograman *PHP*.

3.2 Piranti dan Alat Penelitian

3.2.1 Perangkat Keras (*Hardware*)

Perangkat keras dibutuhkan dalam melakukan desain sistem dan implementasi program. Perangkat keras yang dibutuhkan dalam penelitian ini adalah :

- a. Komputer server, komputer *server* digunakan untuk menempatkan program dan database sistem perpustakaan online. Server yang digunakan memiliki spesifikasi teknis sebagai berikut :

Processors	(1) Intel® Xeon® Processor E5504 (2.00 GHz, 4MB L3 Cache, 80 Watts, DDR3-800) Intel® 5520 Chipset
Memory	4 GB (2 x 2 GB) PC3-10600R (DDR3-1333)
Storage Controller	Smart Array P410i/Zero Memory Controller (RAID 0/1/1+0) Available upgrades: 256MB, 512MB with BBWC, Battery kit upgrade (for BBWC), and Smart Array Advanced Pack (SAAP)
Hard Drive	1 x HP 146GB 10K SAS 2.5 DP HDD
Optical	12.7mm SATA DVD-RW

Drive

Kit

Network
Controller

1 x NC382i Dual Port
Multifunction
Gigabit Server Adapter
(two ports total)
with TCP/IP Offload
Engine, including support
for Accelerated iSCSI

Expansion
Slots

Two PCI-Express Gen 2
expansion slots:
(1) full-length, full-height
slot; (1) low-profile slot
Optional PCI-X Riser
expansion slot

Ports

USB 2.0 Support
4 Total: (1) front, (1)
internal and (2) rear
accessible ports1 internal
Secure Digital (SD) slot

Power
Sources

Dual Common Slot Bays
for High Efficiency,

Right Sized, Hot plug and Redundant Power options , 460W CS HE Gold Power Supply

Fans

3 total (3 fans in 1 Processor Models, 1 additional fans with 2 Processor Model) redundancy standard

- b. Komputer klien, komputer klien digunakan untuk mengakses program yang berada pada komputer server melalui jaringan. Spesifikasi tekni komputer klien yang digunakan adalah sebagai berikut:

Processor

Intel® Core™2 Duo Processor E7400 (3M Cache, 2.80 GHz, 1066 MHz FSB)

Chipset

Intel® G31 Express Chipset

Memory

type

1GB DDR2-Synch DRAM PC2-6400 (800 MHz) Non-ECC

Maximum memory	4 GB DDR2-Synch DRAM
Memory slots	2 DIMM
Internal drives	160 GB SATA NCQ HDD SMART IV 7200 rpm
Storage controller	Serial ATA 3.0 Gb/s hard drive
Optical drives	SATA SuperMulti LightScribe DVD Writer Drive
Graphic card	Integrated Intel® Graphics Media Accelerator 3100
Internal audio	Realtek ALC662-GR HD Audio compatible codec with 5.1 channel audio 3D audio
External I/O ports	Rear: 4 USB 2.0, 1 serial port, 1 parallel port, 2 PS/2, 1 RJ-45, 1 VGA, audio in,

	audio out, Mic In;
	Front: 2 USB 2.0, audio ports, Card Reader
Network interface	Integrated Realtek 8111C-VB-GR GbE Ethernet Controller
Fax/modem	Agere 56K PCI Modem
Keyboard	USB Standard Keyboard
Pointing device	USB 2-Button Optical Scroll Mouse
Power Supply	250 Watt
Monitor	LCD 15"

- c. Printer , perangkat keras printer digunakan untuk mencetak laporan yang dihasilkan oleh sistem informasi perpustakaan online. Spesifikasi teknis printer yang digunakan adalah sebagai berikut :

Platform	Business Laser
----------	----------------

Printing

Method Laser

Max.

Media

Sizes A4

Max.

Resolution 600

Effective

Print

Resolution 600 x 600 dpi

Print Speed

Black 17 ppm

Monthly

Usage 5,000 pages per month

Volume (maximum)

1,500 pages per month

(average)

PC

Connectivity

y

Processor 266 MHz

Memory	8 MB embedded inside
Standard	SIP ASIC, 16 k bit NVRAM
Language	
Supports	Host-based printing
Network	
Supports	Not Available
Input Tray	
#1	150 sheets
Duplex	Manual two-sided
Printing	printing
Compatible	A4, A5, A6, B5, 16K;
Media	Latter, Legal,
Sizes	Execitive; postcards, envelopes (C5, DL, B5, No.10, Monarch)

3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam perancangan terdiri dari beberapa jenis aplikasi :

- a. aplikasi pengolah kata : digunakan untuk mencatat laporan dalam bentuk dokumen dan lembar kerja. Untuk kegiatan dokumentasi tersebut menggunakan Microsoft Office Word dan Excell.
- b. Aplikasi pengolah desain : digunakan untuk menggambar perancangan dalam bentuk model desain sistem dan database. Penulis menggunakan *Rational Rose* untuk membuat desain pemodelan sistem.
- c. Aplikasi pembangun sistem/bahasa pemrograman : bahasa pemrograman merupakan aplikasi yang digunakan untuk mengimplementasikan perancangan ke dalam program yang siap digunakan Bahasa pemrograman yang digunakan adalah PHP versi 5.2.8 .
- d. Aplikasi web server : yaitu aplikasi yang berfungsi sebagai server dan pengolah database. Aplikasi yang digunakan adalah XAMPP, aplikasi ini bersifat *open source*, mendukung *dynamic web*, dan telah mencakup didalamnya adalah HTTP Server dan MySQL database.

3.3 Metode Penelitian

3.3.1 Metode Pengembangan Sistem

Untuk memecahkan problem yang sebenarnya diperlukan model untuk menggabungkan strategi pengembangan yang terdiri dari proses, metode, dan perangkat perancangan. Strategi ini seringkali mengarah pada sebuah model proses atau model perancangan sistem. Oleh karena itu pengembangan sistem perpustakaan digital ini menggunakan metode pengembangan sistem *Linear Sequential / Waterfall Model*. Metode ini merupakan

model klasik yang bersifat sistematis dan mudah dipahami karena berurutan dalam tahapan membangun *software*.

Sesuai dengan tahapan yang dijadikan referensi sehingga deskripsi aktifitas perancangan sistem yang dilakukan penulis dapat disampaikan sebagai berikut :

3.3.1.1 Tahap Analisis

Pada tahap ini dilakukan identifikasi permasalahan dan pencarian solusi untuk mengatasi permasalahan. Permasalahan yang diangkat dalam tesis ini adalah bagaimana mengembangkan perpustakaan digital dengan menggabungkan arsitektur *digital library sistem* dan *library information sistem*. Berdasarkan arsitektur kedua sistem terdahulu, maka integrasi arsitektur dapat dilakukan dengan memasukkan komponen *library management sistem* ke dalam arsitektur *digital library sistem*. Untuk melakukan integrasi ini maka dibutuhkan *web application* sebagai teknologi untuk membangun komponen *library managements system* yang diintegrasikan.

Seperti telah dijelaskan sebelumnya bahwa integrasi kedua arsitektur ini ditempuh dengan memasukkan beberapa komponen *library management sistem* (*Circulation, Fines, Admin, Import dan Export*) ke dalam arsitektur *digital library sistem*. Dengan metode ini maka komponen tersebut dapat digunakan bersama melalui *web browser* (*sharable*) secara *online*. Arsitektur digital library sistem pendekatan berlapis. Pada lapis paling atas disebut *DL Application*

Components, yaitu lapisan yang menyediakan fungsionalitas DL khusus bagi *end user* atau disebut sebagai *client side*. Pada layer tersebut komponen *circulation, fines, import/download, export/upload, dan admin space management*. Dengan meletakkan komponen tersebut pada lapis atas maka memungkinkan user mengaksesnya, sesuai fungsinya, termasuk komponen *admin space management*. Pada lapis tengah yaitu *Enabling Component* terdapat komponen *DBWrapper* yang ditambahkan, yaitu aplikasi DataBase Management System yang mengelola alur kerja sama dengan *Process Engine*. Pada lapis terbawah yang disebut *Application Framework*, atau *server side* terdapat penambahan komponen *Web Server*, tempat dimana *web application* diinstall dan dijalankan. Pada lapis ini hanya SysAdmin yang memiliki hak akses mengelolanya.

3.3.1.2. Desain

Desain *software* memiliki berbagai tahapan yang berfokus pada atribut program yang jelas yaitu : *software architecture*, detail prosedur(*algorithm*), *data structure* dan *interface representations*.

Dalam *Unified Modelling Language* bentuk disain disajikan dalam berbagai diagram, antara lain:

- a. Software Architecture : *Use Case Diagram* Perpustakaan Online.
- b. Detail Procedure : *Sequence Diagram, Activity Diagram, State Machine Diagram*
- c. Data Structure : *Class, Class Diagram*

Mengacu pada fungsi – fungsi hasil *integrasi digital library system* dan *library management sistem* maka dapat dibuat beberapa *use case* antara lain :

1. Mencari Koleksi
2. Membaca Koleksi
3. Mendownload Koleksi
4. Merekomendasikan Koleksi
5. Mengubah Password
6. Melihat Status Koleksi
7. Menambah Koleksi
8. Membuat Laporan
9. Mengelola Pinjaman Koleksi
10. Mengupload Koleksi
11. Mengelola Data Anggota
12. Mengelola Data Pustakawan
13. Mengelola Data SysAdmin
14. Mengelola Migrasi Data

Tabel Fungsi Utama Perangkat Lunak

Tabel 1 : Fungsi Utama Perangkat Lunak Pengunjung

No.	Nama Fungsi	Deskripsi
1	Login	Merupakan fungsi untuk memverifikasi pengunjung dalam mengoperasikan sistem.
2	Mencari Koleksi	Mencari koleksi baik dalam bentuk <i>digital</i> maupun <i>non digital</i> dengan memasukkan satu kata kunci atau beberapa kata kunci.

No.	Nama Fungsi	Deskripsi
3	Mendownload Koleksi	Merupakan fungsi untuk mendownload digital resource.

Tabel 2 : Fungsi Utama Perangkat Lunak Anggota

No.	Nama Fungsi	Deskripsi
1	Login	Merupakan fungsi untuk memverifikasi anggota dalam mengoperasikan sistem.
2	Mencari Koleksi	Mencari koleksi baik dalam bentuk <i>digital</i> maupun <i>non digital</i> dengan memasukkan satu kata kunci atau beberapa kata kunci.
3	Mendownload Koleksi	Merupakan fungsi untuk mendownload digital resource.

Tabel 3 : Fungsi Utama Perangkat Lunak Pustakawan

No.	Nama Fungsi	Deskripsi
1	Login	Merupakan fungsi untuk memverifikasi pengguna dalam mengoperasikan sistem.
2	Mencari Koleksi	Mencari koleksi baik dalam bentuk <i>digital</i> maupun <i>non digital</i> dengan memasukkan satu kata kunci atau beberapa kata kunci.
3	Mengupload Koleksi	Merupakan fungsi untuk mengupload digital resource.
4	Mengelola Sirkulasi	Merupakan fungsi untuk membantu aktifitas pustakawan pada transaksi peminjaman dan pengembalian koleksi.
5	Mengambah Anggota	Merupakan fungsi untuk menambah, memperbaiki dan menghapus data anggota perpustakaan.

Tabel 4 : Fungsi Utama Perangkat Lunak SysAdmin

No.	Nama Fungsi	Deskripsi
1	Login	Merupakan fungsi untuk memverifikasi pengguna dalam mengoperasikan sistem.
2	Menambah Pustakawan	Merupakan fungsi untuk menambah, memperbaiki dan menghapus data pustakawan.
3	Mengelola Migrasi	Membuka hasil upload data dari pustakawan untuk kemudian dikonversi ke format yang sesuai dengan sistem.

Tabel 5 : Karakteristik Pengguna

No.	Pengguna	Karakteristik
1	SysAdmin	Menguasai sistem secara keseluruhan
2	Pengunjung	Pengguna selain anggota perpustakaan.
3	Anggota	Pengguna yang menjadi anggota perpustakaan.
4	Pustakawan	Petugas yang menangani pengelolaan administrasi pada perpustakaan

Antarmuka Pengguna

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*), dengan piranti input menggunakan *keyboard, mouse, monitor LCD* atau *CRT*. *Interface* perangkat keras yang akan digunakan adalah berupa server dan *client* dengan infrastruktur jaringan., yaitu *Ethernet card, switch, RJ45*.

Lingkungan Operasi

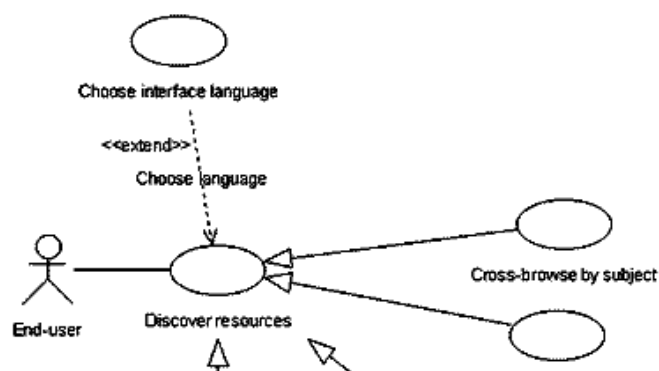
Server : Apache Server 2.2
 DBMS : MySQL
 OS Server : Windows Server 2003 atau Linux / BSD
 OS Client : Minimum Windows XP
 Development Tools : *PHP*

Use Case Diagram

Use case diagram sistem baru dibuat mengacu pada *use case diagram* masing – masing sistem, yaitu *use case diagram digital library* dan *usecase diagram library management system*. Dalam penamaannya sedikit berbeda namun memiliki arti yang sama.

Menurut Heery (2001) *Use case diagram digital library* terdiri dari 7 (tujuh) *use case*, dapat dilihat pada gambar 5, yaitu :

1. *discover resource*, fungsi menemukan *resource* atau *digital collection*, fungsi ini memiliki dua fungsi turunan yaitu *search(simple)* maupun *search(complex)*
2. *search(simple)*, fungsi untuk mencari digital collection dengan memasukkan filter sederhana.
3. *search(complex)*, fungsi untuk mencari digital collection dengan memasukkan filter yang lebih detail.
4. *choose interface language*, fungsi ini disediakan untuk mengganti bahasa pengantar sesuai bahasa user. Fungsi ini minimal menyediakan bahasa setempat dan bahasa internasional.
5. *cross browse by subject*, fungsi yang digunakan untuk mencari lintas web menggunakan subjek
6. *cross browse by metadata* fungsi yang digunakan untuk mencari lintas web menggunakan metadata.
7. *data information landscape* fungsi yang menyediakan peta informasi yang ada.



Gambar 5. Use Case Diagram *Digital*

End user dapat menggunakan pencarian sederhana dan pencarian kompleks dengan terlebih dahulu untuk membaca atau mendownload (*discovery resource*) *digital resource* yang dibutuhkan. Penyajian *digital resource* juga dikelompokkan menurut bidang ilmu (*data information landscape*) untuk mempercepat proses pencarian. Pada use case diagram ini hanya memiliki satu aktor yaitu *End-User*.

Pada *use case diagram library information system*, sesuai yang dibahas oleh Rosasco dan Bakke (2005), seperti yang terlihat pada Gambar 6, terdapat tiga *use case diagram* yang memiliki masing – masing use case , yaitu : *patron use case diagram*, *circulation use case diagram* dan *cataloging use case diagram*.

Pada *Patron use case diagram* atau *OPAC (Online Public Access Catalog)* terdapat 4 (empat) *use case* yaitu ;

1. *Find items*, yaitu fungsi untuk menemukan informasi items (katalo/buku)
2. *Advance search*, fungsi yang menyediakan pencarian lanjut/mahir, yaitu pencarian dengan banyak filter.

3. *Search*, fungsi yang menyediakan fasilitas pencarian item secara sederhana, yaitu dengan satu filter.

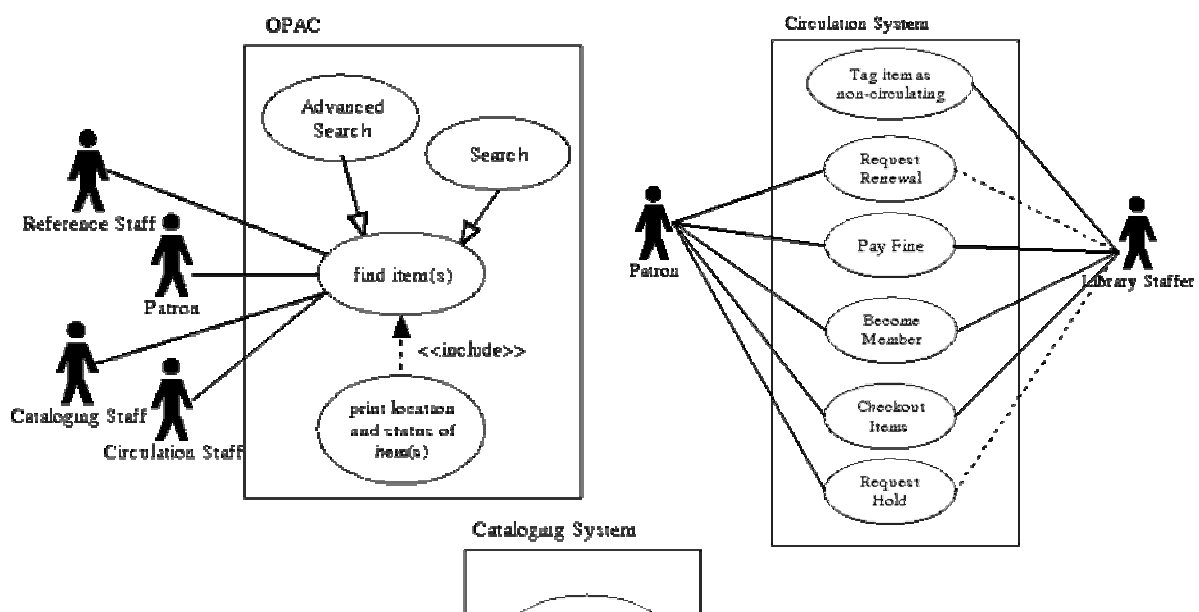
4. *Print location and status items*, fungsi yang menyediakan fasilitas cetak lokasi dan status buku.

Pada *Circulation use case diagram* terdapat 6 (enam) *use case* yaitu;

1. *Tag item as non-circulating* : fungsi yang memberikan fasilitas untuk buku referensi saja atau buku yang boleh dipinjam.
2. *Request Renewal* ; fungsi yang menyediakan fasilitas mengusulkan buku baru.
3. *Pay fine* ; fungsi yang menyediakan fasilitas pembayaran denda
4. *Become member*; fungsi untuk mencata pendaftaran anggota baru
5. *Checkout item* ; fungsi untuk memeriksa status buku
6. *Request hold*; fungsi untuk mencatat pemesanan buku yang akan dipinjam

Pada *Cataloging system use case diagram* terdapat 4(use case), yaitu ;

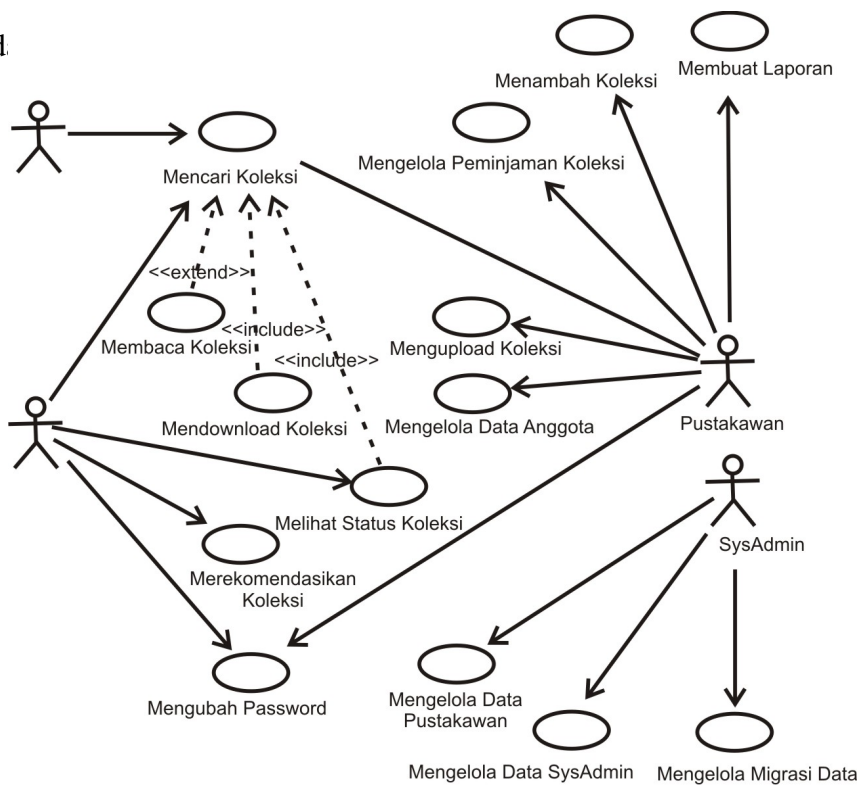
1. *Want List item*; fungsi yang menyampaikan daftar permintaan pinjam buku
2. *Order item*; fungsi yang memberikan fasilitas mengorder buku ke *supplier*
3. *Accession item*; fungsi untuk memberikan hak akses pada buku
4. *Deaccession item*; fungsi untuk menolak hak akses pada buku.



Gambar 6 . Use Case Diagram Library Management

Pada *use case diagram* tersebut diatas digambarkan terdapat banyak aktor yang berinteraksi dengan sistem. Dari aktor - aktor tersebut dapat dikelompokkan menjadi aktor user (*patron* dan *book supplier*) dan pustakawan (*reference, cataloging, circulation staff* dan *library staffer*).

Berdasarkan *use case* dari dua sistem yang diintegrasikan, maka *use case diagram* integrasi *digital library system* dan *library management system* disajikan pad.



Gambar 7. Use Case Diagram integrasi Digital

Pada diagram ini tidak semua actor dan use case ditampilkan mengingat keterbatasan waktu penyelesaian laporan. Namun fungsi – fungsi utama yang mencirikan kedua sistem tetap dipertahankan dalam diagram integrasi.

Berikut adalah detail spesifikasi dari use case yang ditampilkan pada gambar 18.

Deskripsi Use Case (Mencari Koleksi)

<p>Aktor Utama : Anggota</p> <p>Aktor Tambahan: Pengunjung, Pustakawan</p> <p>Nama Use Case: Mencari Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada anggota perpustakaan untuk melakukan pencarian koleksi baik digital maupun non digital.</p> <p>Kondisi awal: Halaman utama sistem sudah terbuka.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. User membuka alamat web site sistem perpustakaan online. 2. User memasukkan kata kunci untuk mencari koleksi yang dikehendaki. 3. User mengklik tombol pencarian untuk menjalankan proses pencarian. 4. Hasil pencarian ditampilkan. <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 1. 4.a Hasil pencarian tidak ditemukan – Tampilkan pesan hasil pencarian tidak ada.
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Textbox pencarian siap digunakan.</p> <p>Kondisi akhir:</p> <p>Tampil hasil pencarian sesuai dengan kata kunci yang dimasukkan.</p> <p>Trigger: User memasukkan kata kunci dan mengklik tombol pencarian.</p> <p>Frekwensi penggunaan:</p> <p>beberapakali sehari</p>	
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Deskripsi Use Case (Membaca Koleksi)

<p>Aktor Utama:</p> <p>Anggota</p> <p>Aktor Tambahan:,</p> <p>Pustakawan</p> <p>Nama Use Case:</p> <p>Membaca Koleksi</p> <p>Tujuan Use Case:</p> <p>use case ini menyediakan layanan kepada anggota perpustakaan untuk membaca koleksi digital.</p> <p>Kondisi awal:</p> <p>Tampil daftar koleksi digital yang berasal dari hasil pencarian</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar koleksi. 2. User mengklik tombol baca. 3. Sistem menampilkan isi koleksi digital untuk dibaca 4. User membaca koleksi digital . <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 2. a . Sumber koleksi digital tidak ditemukan maka tampil pesan penolakan.
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>maupun yang ditampilkan sistem secara otomatis.</p> <p>Kondisi akhir: Web browser menampilkan isi koleksi digital untuk dibaca</p> <p>Trigger: User mengklik tombol baca.</p> <p>Frekwensi penggunaan: beberapa kali sehari</p>	
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Deskripsi Use Case (Mendownload Koleksi)

<p>Aktor Utama: Anggota</p> <p>Aktor Tambahan: Pustakawan</p> <p>Nama Use Case: Mendownload Koleksi</p> <p>Tujuan Use Case: use case ini digunakan untuk mendownload koleksi digital.</p> <p>Kondisi awal: Tampil daftar koleksi digital yang berasal dari hasil pencarian maupun yang</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar koleksi. 2. User mengklik tombol download. 3. Sistem menampilkan dialog window penyimpanan file koleksi digital. 4. User menentukan lokasi penyimpanan file . <p>Pesimistic Flow:</p> <p>2.a . Sumber koleksi digital tidak ditemukan maka tampil</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>ditampilkan sistem .</p> <p>Kondisi akhir: Web browser menampilkan pesan penyimpanan file koleksi digital.</p> <p>Trigger: User mengklik tombol download.</p> <p>Frekwensi penggunaan: beberapa kali sehari</p>	<p>pesan penolakan.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------

Deskripsi Use Case (Melihat Status Koleksi)

<p>Aktor Utama: Anggota</p> <p>Aktor Tambahan: Pustakawan</p> <p>Nama Use Case: Melihat Status Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada anggota perpustakaan untuk melihat status koleksi non-digital.</p> <p>Kondisi awal: Sistem menampilkan menu Cek Status Buku</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu Cek Status Buku. 2. User mengklik tombol menu. 3. Sistem menampilkan form pengisian buku yang diinginkan . 4. User mengisi judul buku. 5. User mengklik tombol Cari 6. Sistem menampilkan hasil pencarian dan status koleksi non-digital yang terdiri dari ;dipinjam, dipesan, tersedia. <p>Pesimistic Flow:</p> <p>5.a . Judul koleksi non-digital tidak</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Kondisi akhir: Web browser menampilkan daftar status koleksi non-digital.</p> <p>Trigger: User mengklik tombol Cari.</p> <p>Frekwensi penggunaan: beberapa kali sehari</p>	<p>ditemukan maka sistem menampilkan pemberitahuan.</p> <p>5.b. Judul koleksi dibiarkan kosong maka tampil peringatan.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------

Deskripsi Use Case (Merekomendasikan Koleksi)

<p>Aktor Utama: Anggota</p> <p>Aktor Tambahan: Pustakawan</p> <p>Nama Use Case: Merekomendasikan Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada anggota perpustakaan untuk merekomendasikan koleksi.</p> <p>Kondisi awal: Sistem menampilkan menu Rekomendasi</p> <p>Kondisi akhir: Web browser menampilkan pesan proses Rekomendasi telah</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu Rekomendasi. 2. User mengklik tombol menu. 3. Sistem menampilkan form pengisian rekomendasi koleksi. 4. User mengisi form dan mengklik Process Change 5. Sistem menampilkan informasi penyimpanan berhasil dilakukan. <p>Pesimistic Flow:</p> <p>4.a . User tidak mengisi lengkap form rekomendasi , sistem menampilkan peringatan.</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>berhasil.</p> <p>Trigger: User mengklik tombol Process Change.</p> <p>Frekwensi penggunaan: beberapa kali sehari</p>	<p>b. User mengklik Cancel Change</p>
---------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------

Deskripsi Use Case (Mengubah Password)

<p>Aktor Utama: Anggota</p> <p>Aktor Tambahan: Pustakawan</p> <p>Nama Use Case: Mengubah Password Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada anggota dan pustakawann untuk ubah password</p> <p>.Kondisi awal: Sistem menampilkan menu Profil Anggota</p> <p>Kondisi akhir: Web browser menampilkan pesan proses penggantian password telah berhasil.</p> <p>Trigger: User mengklik tombol Process Change.</p> <p>Frekwensi</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu Profil Anggota. 2. User mengklik tombol menu. 3. Sistem menampilkan form pengisian Ubah Password. 4. User mengisi form dan mengklik Process Change 5. Sistem menampilkan informasi penggantian password berhasil dilakukan. <p>Pesimistic Flow:</p> <p>4.a User mengklik Cancel Change maka sistem kembali ke Menu Utama.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

penggunaan: beberapa kali sehari	
--------------------------------------------	--

Deskripsi Use Case (Mengelola Pinjaman Koleksi)

<p>Aktor Utama: Pustakawan</p> <p>Aktor Tambahan: Anggota</p> <p>Nama Use Case: Mengelola Pinjaman Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada pustakawan untuk mengelola pinjaman koleksi.</p> <p>Kondisi awal: Sistem menampilkan menu Pustakawan</p> <p>Kondisi akhir: Web browser menampilkan koleksi yang dipinjam pada daftar peminjaman koleksi.</p> <p>Trigger: Pustakawan mengklik tombol Daftar Peminjaman.</p> <p>Frekwensi penggunaan: beberapa kali sehari</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu Pustakawan. 2. Pustakawan mengklik tombol Daftar Peminjaman. 3. Sistem menampilkan daftar koleksi yang dipinjam. 4. Pustakawan mengklik icon penambahan peminjaman. 5. Sistem menampilkan form peminjaman 6. Pustakawan memilih data anggota yang akan meminjam 7. Pustakawan memilih data buku yang dipinjam 8. Sistem kembali ke daftar peminjaman buku. <p>Pesimistic Flow:</p> <p>4.a . Pustakawan mengklik menu Kembali pada daftar pinjaman maka sistem menghapus data pinjaman yang telah dikembalikan</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>8. a. Pustakawan tidak mengisi lengkap form peminjaman maka sistem menampilkan peringatan.</p> <p>b. Pustakawan mengklik Cancel Change maka sistem kembali ke daftar peminjaman.</p> <p>c. Pustakawan</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Deskripsi Use Case (Menambah Koleksi)

<p>Aktor Utama: Pustakawan</p> <p>Aktor Tambahan: Anggota</p> <p>Nama Use Case: Menambah Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada Pustakawan untuk menambah koleksi.</p> <p>Kondisi awal: Sistem menampilkan menu daftar koleksi</p> <p>Kondisi akhir:</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu daftar koleksi. 2. User mengklik tombol ikon penambahan koleksi. 3. Sistem menampilkan form pengisian tambah koleksi. 4. User mengisi form dan mengklik Process Change 5. Sistem menampilkan informasi penyimpanan berhasil dilakukan. <p>Pesimistic Flow:</p> <p>4.a . Pustakawan tidak mengisi</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Sistem menampilkan pesan proses penambahan telah berhasil.</p> <p>Trigger: User mengklik tombol Process Change.</p> <p>Frekwensi penggunaan: beberapa kali sehari</p>	<p>lengkap form penambahan koleksi maka sistem menampilkan peringatan.</p> <p>b. Pustakawan mengklik Cancel Change maka sistem kembali ke Menu Utama.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------

Deskripsi Use Case (Membuat Laporan)

<p>Aktor Utama: Pustakawan</p> <p>Aktor Tambahan: Anggota</p> <p>Nama Use Case: Membuat Laporan Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada pustakawan untuk membuat laporan koleksi.</p> <p>Kondisi awal: Sistem menampilkan menu Statistik.</p> <p>Kondisi akhir: Sistem menampilkan laporan.</p> <p>Trigger: User</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu Statistik. 2. User mengklik tombol menu. 3. Sistem menampilkan form pengisian pilihan laporan. 4. User mengisi form dan mengklik Statistik 5. Sistem menampilkan laporan. <p>Pesimistic Flow:</p> <p>4.a . Pustakawan tidak mengisi lengkap form laporan maka sistem menampilkan peringatan.</p> <p>b. Data laporan</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>mengklik tombol Statistik.</p> <p>Frekwensi penggunaan:</p> <p>beberapa kali sebulan</p>	<p>tidak ada maka laporan kosong.</p>
----------------------------------------------------------------------------------------------------	---------------------------------------

Deskripsi Use Case (Mengelola Data Anggota)

<p>Aktor Utama: Pustakawan</p> <p>Aktor Tambahan: Anggota</p> <p>Nama Use Case: Mengelola Anggota</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada pustakawan untuk mengelola data anggota.</p> <p>Kondisi awal: Sistem menampilkan menu Pustakawan</p> <p>Kondisi akhir: Sistem menampilkan pesan proses perubahan data pada daftar anggota.</p> <p>Trigger: Pustakawan mengklik tombol Data Anggota.</p> <p>Frekwensi</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu Pustakawan. 2. User mengklik tombol Data Anggota. 3. Sistem menampilkan daftar anggota. 4. Pustakawan mengisi ikon penambahan anggota. 5. Sistem menampilkan form penambahan anggota. 6. Pustawakan mengisi form penambahan anggota 7. Pustakawan mengklik Process Change. 8. Sistem menampilkan perubahan data pada daftar anggota. <p>Pesimistic Flow:</p> <p>4.a . Pustakawan mengklik menu Hapus pada daftar anggota maka data</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>penggunaan: beberapa kali sebulan</p>	<p>anggota tersebut dihapus dari daftar anggota.</p> <p>7.a. Pustakawan mengklik Cancel Change maka sistem kembali ke Menu Utama.</p>
-----------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------

Deskripsi Use Case (Mengupload Koleksi)

<p>Aktor Utama: Pustakawan</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Mengupload Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada pustakawan untuk mengupload digital koleksi</p> <p>Kondisi awal: Sistem menampilkan menu Pustakawan</p> <p>Kondisi akhir: Sistem menampilkan pesan proses upload berhasil.</p> <p>Trigger: User mengklik tombol Upload.</p> <p>Frekwensi penggunaan:</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu Pustakawan. 2. Pustakawan mengklik menu Upload. 3. Sistem menampilkan form pengisian upload koleksi. 4. Pustakawan mengisi form dan mengklik tombol Upload 5. Sistem menampilkan informasi progres upload. 6. Sistem menampilkan proses upload berhasil. <p>Pesimistic Flow:</p> <p>4.a . Pustakawan mengklik Cancel Change maka sistem kembali ke menu Pustakawan.</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

beberapa kali sehari	
----------------------	--

Deskripsi Use Case (Mengelola Data Pustakawan)

<p>Aktor Utama: SysAdmin</p> <p>Aktor Tambahan: Pustakawan</p> <p>Nama Use Case: Mengelola Data Pustakawan</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada SysAdmin untuk mengelola data pustakawan.</p> <p>Kondisi awal: Sistem menampilkan menu SysAdmin</p> <p>Kondisi akhir: Sistem menampilkan perubahan daftar Pustakawan.</p> <p>Trigger: User mengklik tombol ikon penambahan Pustakawan.</p> <p>Frekwensi penggunaan: beberapa kali sebulan.</p>	<p style="text-align: center;">Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu SysAdmin. 2. SysAdmin mengklik ikon penambahan. 3. Sistem menampilkan form pengisian penambahan pustakawan. 4. SysAdmin mengisi form dan mengklik Process Change 5. Sistem menampilkan informasi penyimpanan berhasil dilakukan. <p>Pesimistic Flow:</p> <p>2.a . SysAdmin mengklik tombol Hapus maka data pustakawan yang bersangkutan dihapus.</p> <p>4.a. SysAdmin tidak lengkap mengisi form penambahan pustakawan maka sistem menampilkan</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>peringatan.</p> <p>b. User mengklik Cancel Change.</p>
--	-----------------------------------------------------------

Deskripsi Use Case (Mengelola Data SysAdmin)

<p>Aktor Utama: SysAdmin</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Mengelola Data SysAdmin</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada sysadmin untuk mengelola koleksi.</p> <p>Kondisi awal: Sistem menampilkan daftar SysAdmin</p> <p>Kondisi akhir: Web browser menampilkan daftar perubahan data SysAdmin.</p> <p>Trigger: User mengklik tombol ikon tambah SysAdmin.</p> <p>Frekwensi penggunaan: beberapa kali dalam beberapa bulan.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar SysAdmin. 2. SysAdmin mengklik tombol ikon penambahan SysAdmin. 3. Sistem menampilkan form pengisian penambahan SysAdmin. 4. SysAdmin mengisi form dan mengklik Process Change 5. Sistem menampilkan informasi penyimpanan berhasil dilakukan. <p>Pesimistic Flow:</p> <p>2.a. SysAdmin mengklik tombol hapus pada daftar SysAdmin maka data yang bersangkutan dihapus.</p> <p>4.a . SysAdmin tidak mengisi lengkap form rekomendasi</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>sistem menampilkan peringatan.</p> <p>b. SysAdmin mengklik Cancel Change maka sistem kembali ke Menu Utama.</p>
--	--------------------------------------------------------------------------------------------------------------------

Deskripsi Use Case (Mengelola Migrasi)

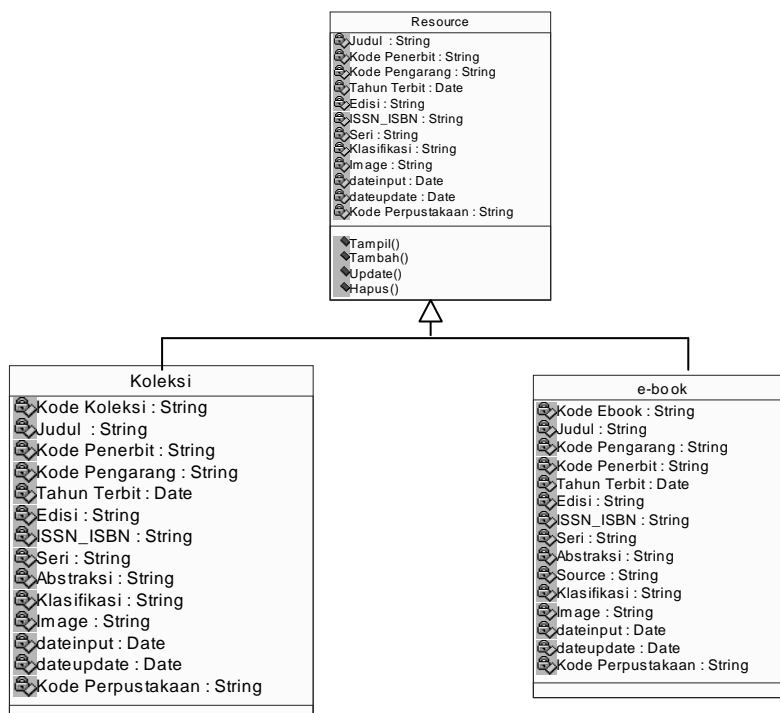
<p>Aktor Utama: SysAdmin</p> <p>Aktor Tambahan: Pustakawan</p> <p>Nama Use Case: Mengelola Migrasi Koleksi</p> <p>Tujuan Use Case: use case ini menyediakan layanan kepada sysadmin untuk mengelola migrasi.</p> <p>Kondisi awal: Sistem menampilkan menu SysAdmin</p> <p>Kondisi akhir: Web browser menampilkan pesan download file migrasi telah selesai</p> <p>Trigger: User mengklik tombol</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu SysAdmin. 2. SysAdmin mengklik tombol menu Migrasi. 3. Sistem menampilkan daftar upload file dari Pustakawan yang akan di migrasi. 4. SysAdmin memilih file pada daftar dan klik download. 5. Sistem menampilkan informasi download file migrasi telah selesai. <p>Pesimistic Flow:</p> <p>4.a . User mengklik tombol Cancel maka menu kembali ke SysAdmin</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>download.</p> <p>Frekwensi penggunaan:</p> <p>beberapa kali dalam beberapa bulan</p>	
------------------------------------------------------------------------------------------------	--

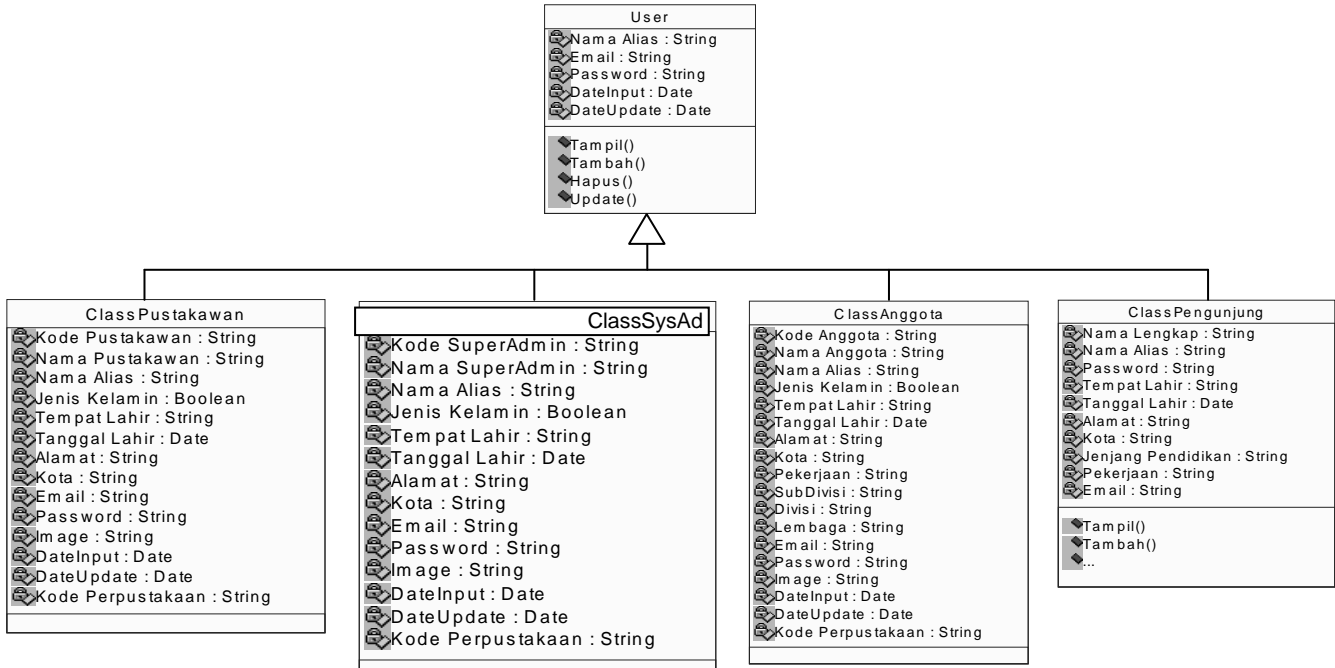
Class Diagram

Setelah memodelkan sistem utama dengan use case diagram, langkah selanjutnya penulis menganalisa *class – class* yang dapat disimpulkan dalam use case diagram. Pengertian *class* yaitu sekumpulan objek yang memiliki persamaan atribut dan operasionalnya. Sedangkan objek adalah benda, baik berbentuk maupun tidak berbentuk yang dapat dilihat, disentuh maupun dirasakan efeknya.

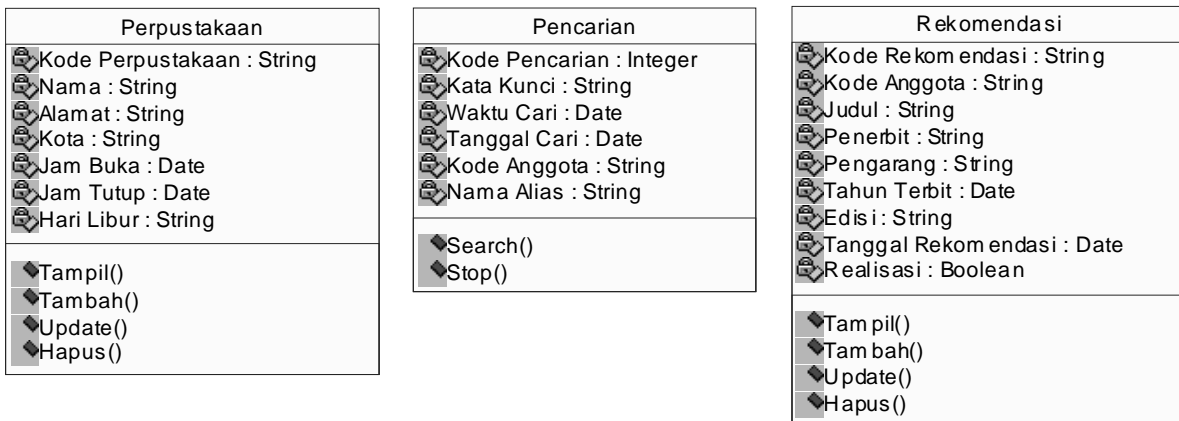
Class yang diperoleh dari hasil analisa disajikan dalam Gambar 8 s.d Gambar 12:



Gambar 8. Class Resource, Class Koleksi dan Class e-Book

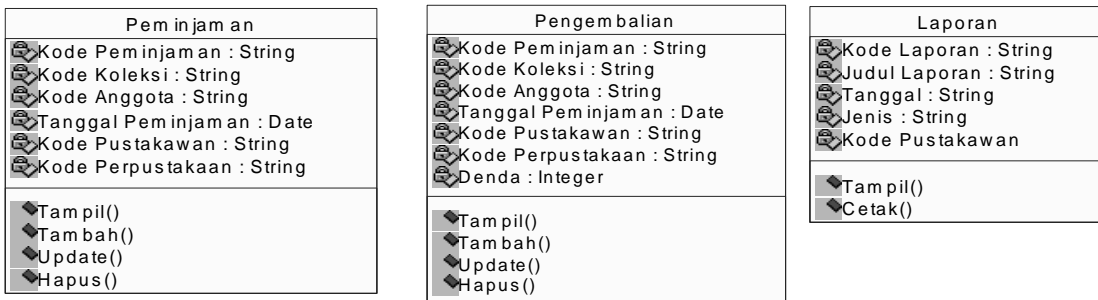


Gambar 9. Class User, Class Pustakawan, Class Anggota dan Class Pengunjung.

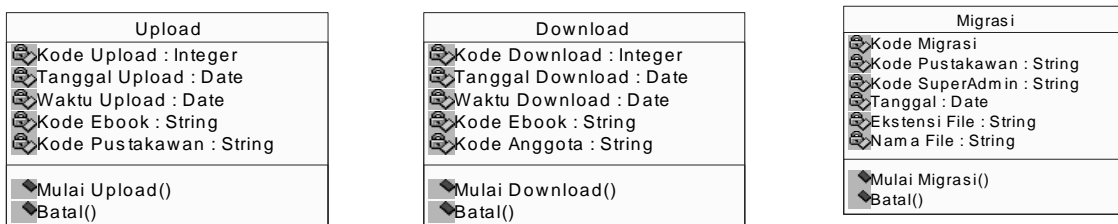


Gambar 10. Class Perpustakaan, Class Pencarian, Class

Rekomendasi



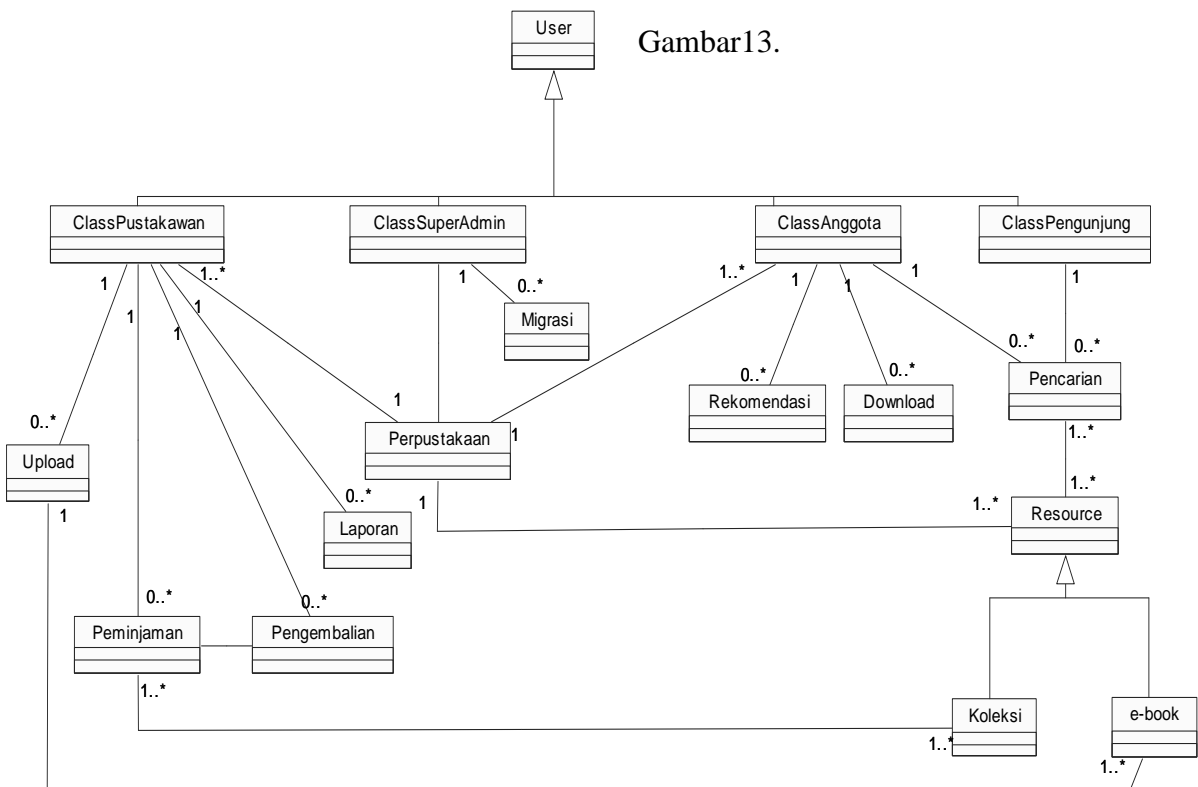
Gambar 11. Class Peminjaman, Class Pengembalian, Class Laporan



Gambar 12. Class Upload, Class Download, Class Migrasi

Hasil analisa asosiasi class disajikan dalam class diagram

Gambar13.

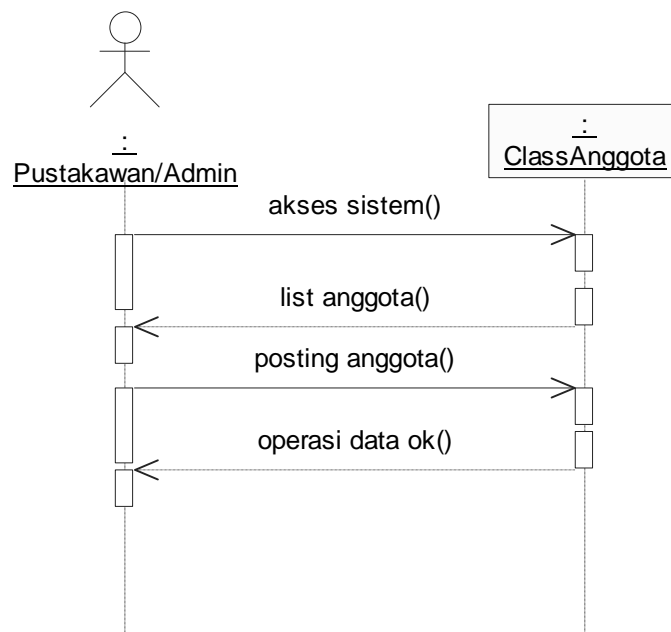


S Gambar 13 . Class Diagram Sistem Perpustakaan Online

Sequence Diagram

Sequence diagram digunakan untuk menjabarkan aktifitas yang ada pada use case kepada level yang lebih detail.

Sequence Kelola Data Anggota

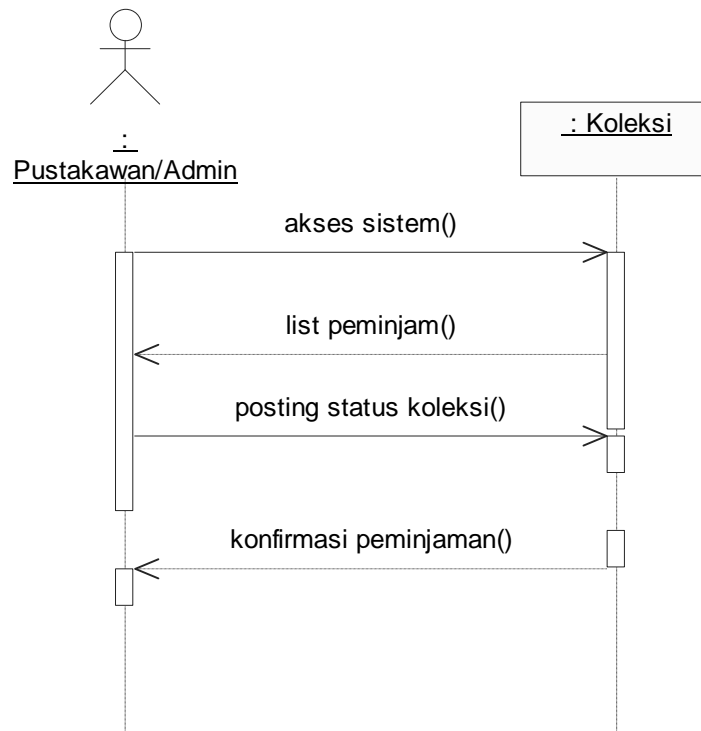


Gambar 14 . Sequence Diagram Kelola Data Anggota

Pada sequence diagram kelola data anggota menerangkan bahwa aktor Pustakawan selaku admin lokal membuka sistem. *Class* Anggota menampilkan data anggota, pustakawan melakukan penambahan data anggota. Setelah melakukan verifikasi *class* Anggota kemudian melakukan update data anggota dan memberikan

informasi kepada aktor Pustakawan bahwa penambahan anggota telah berhasil.

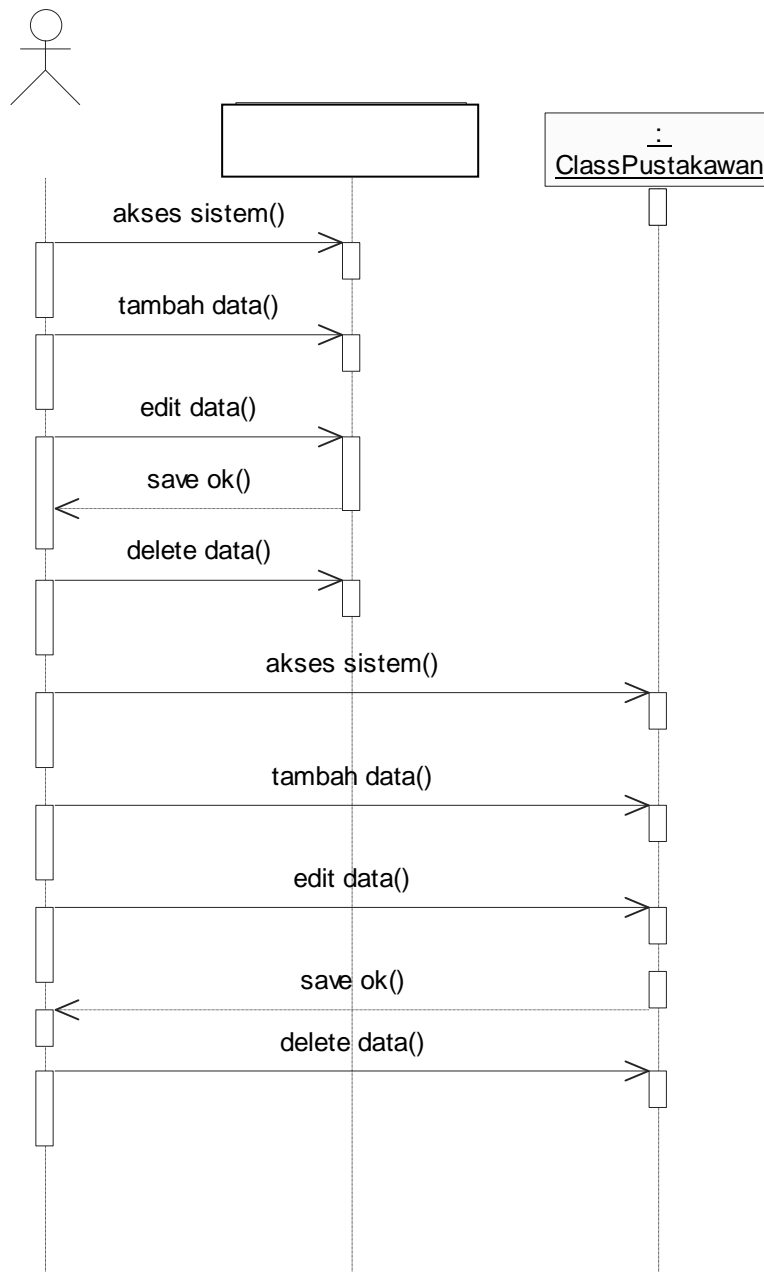
Sequence Kelola Peminjaman



Gambar 15 . Sequence Diagram Kelola Peminjaman

Pada gambar Sequence Diagram Kelola Peminjam menjelaskan bahwa aktor Pustakawan dapat melakukan penambahan peminjaman pada koleksi. Aktor pustakawan membuka sistem, kemudian koleksi memberikan data peminjam. Aktor pustakawan melakukan perubahan status koleksi kemudian koleksi memberikan konfirmasi bahwa perubahan status peminjam telah berhasil.

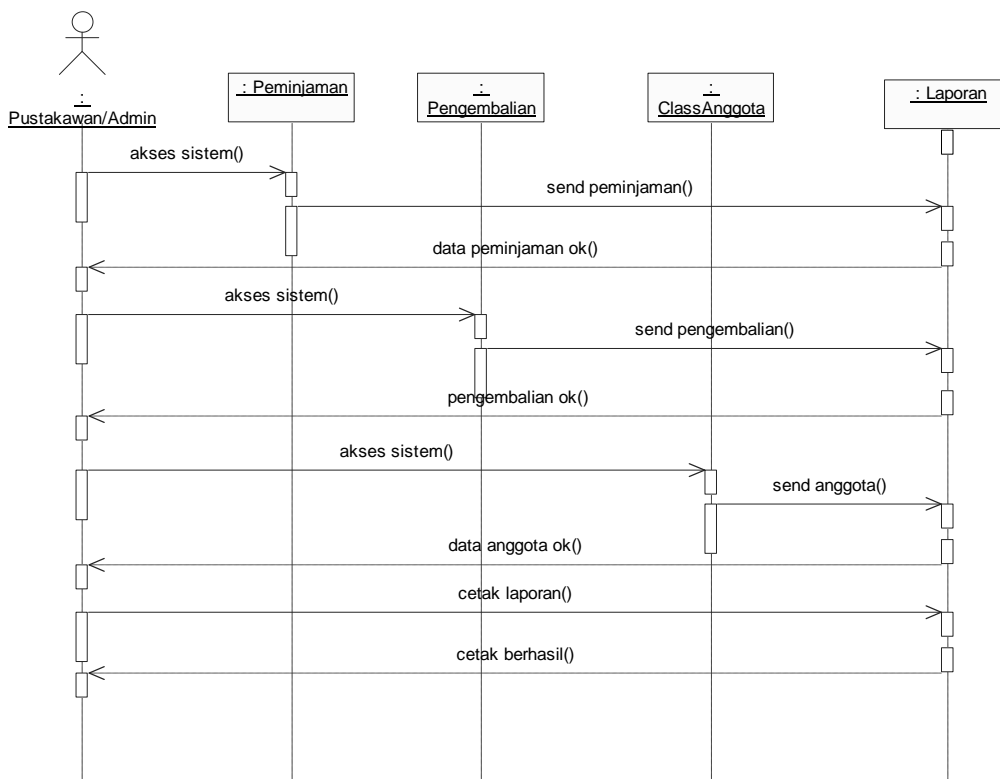
Sequence Kelola User (SysAdmin)



Gambar 16 . Sequence Diagram Kelola User (SysAdmin)

Pada sequence diagram Kelola User (SysAdmin), SysAdmin membuka sistem pada *Class SysAdmin*, SysAdmin menambahkan data pada *Class SysAdmin* kemudian melakukan perubahan data pada *Class SysAdmin* kemudian *Class SysAdmin* menerima perubahan data dan data telah tersimpan. Kemudian SuperAdmin menghapus data pada *Class SysAdmin*. SysAdmin membuka sistem pada *Class Pustakawan* kemudian melakukan penambahan data dan perubahan data pada *Class SysAdmin* dan data telah tersimpan. Kemudian SysAdmin menghapus data pada *Class Pustakawan*.

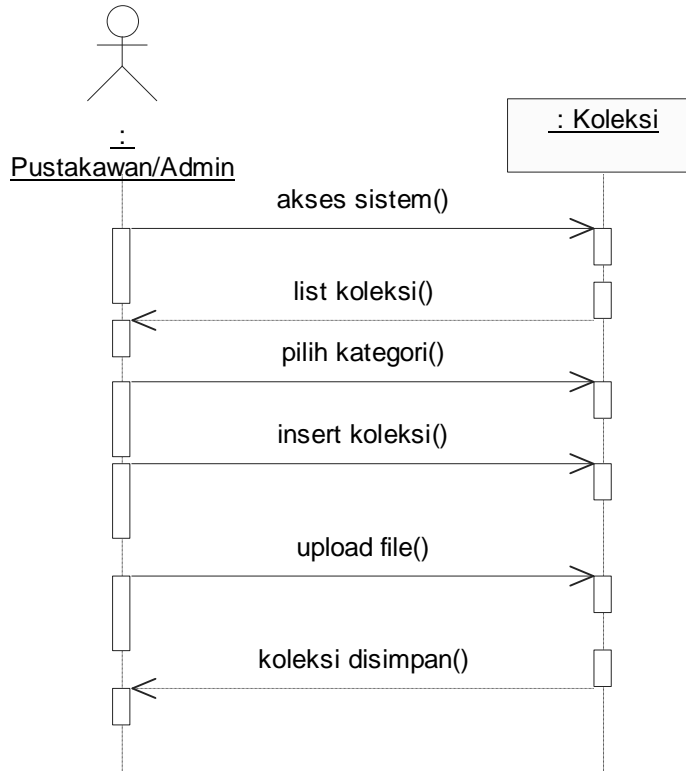
Sequence Laporan



Gambar 17 . Sequence Diagram Laporan

Pada sequence diagram Laporan Pustakawan membuka sistem pada peminjaman kemudian peminjaman mengirim data peminjam pada laporan, Laporan mengkonfirmasi bahwa data peminjam telah berhasil disimpan. Pustakawan membuka sistem pada pengembalian kemudian pengembalian mengirim data pengembalian pada laporan , Laporan mengkonfirmasi bahwa pengembalian telah berhasil disimpan. Pustakawan membuka sistem pada class anggota kemudian class anggota mengirim data anggota pada laporan. Laporan mengkonfirmasi bahwa data anggota telah berhasil disimpan. Pustakawan mencetak laporan, Laporan mengkonfirmasi bahwa cetak laporan telah berhasil.

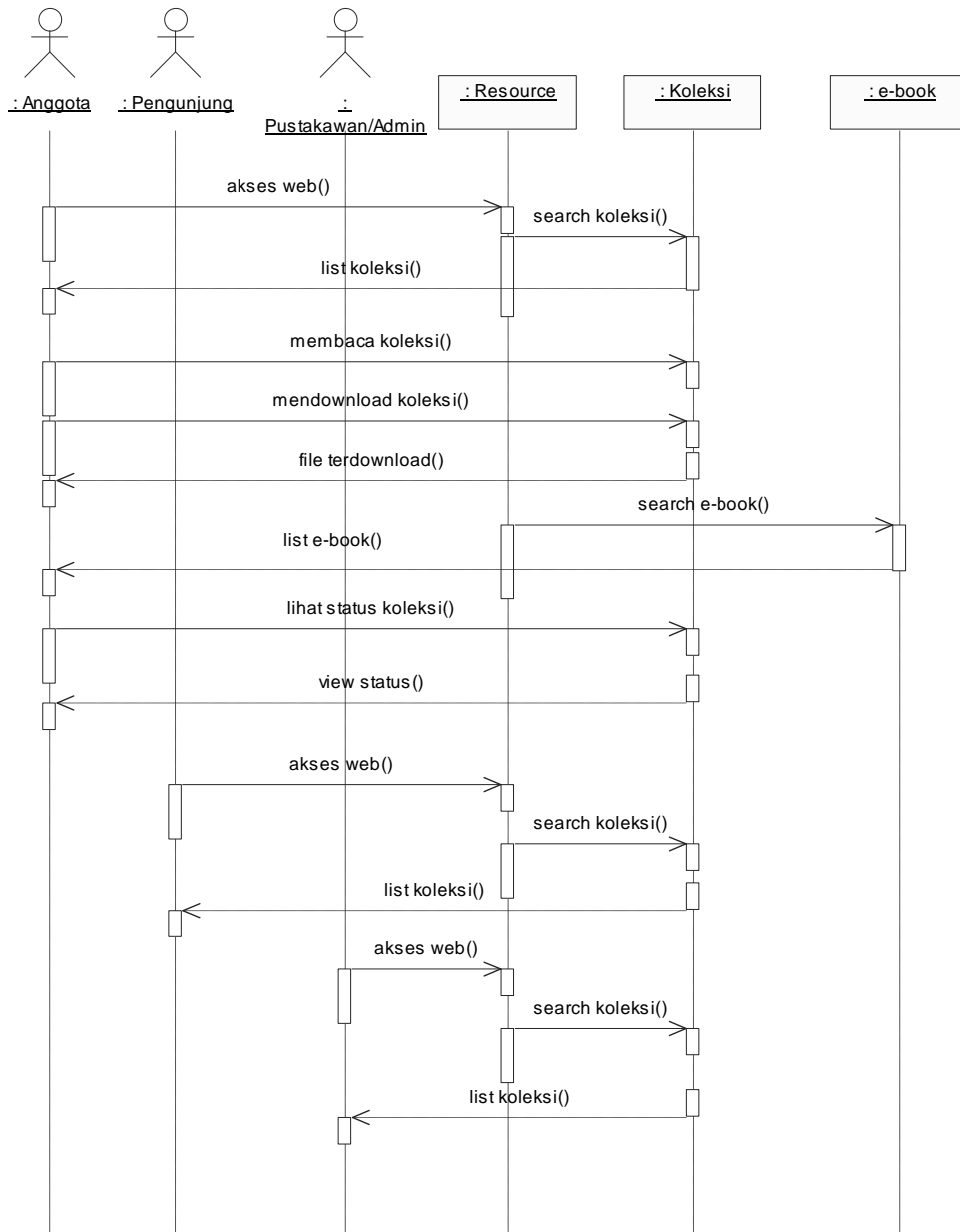
Sequence Menambah koleksi



Gambar 18 . Sequence Diagram Menambah Koleksi

ustakawan membuka system pada Koleksi, Koleksi memberikan daftar koleksi. Pustakawan memilih kategori pada Koleksi lalu pilih, kemudian Pustakawan Upload file ke Koleksi. Koleksi mengkonfirmasi bahwa koleksi telah tersimpan.

Sequence Mencari Koleksi

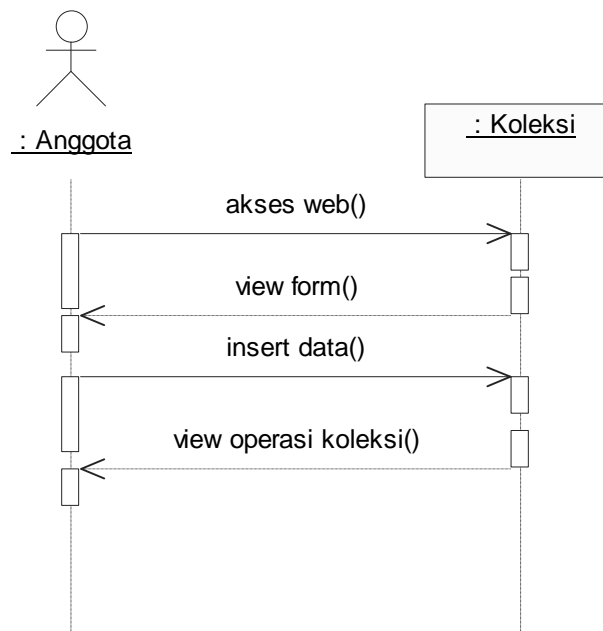


Gambar 19. Sequence Diagram Mencari Koleksi

Pada Sequence Mencari Koleksi Anggota membuka web pada *Resource* kemudian *Resource* melakukan pencarian koleksi pada Koleksi, Koleksi memberikan daftar koleksi pada Anggota. Anggota membaca koleksi, Anggota mendownload koleksi, file terdownload. *Resource* mencari *e-book* di *e-book*, kemudian *e-book* memberikan daftar *e-book* ke Anggota, Anggota lihat status koleksi di Koleksi.

Koleksi membuka status anggota, kemudian Koleksi membaca status anggota. Pengunjung membuka web pada *Resource*, *Resource* mencari koleksi pada Koleksi. Koleksi memberi daftar koleksi ke Pengunjung. Pustakawan membuka web pada *Resource*, *Resource* mencari koleksi pada Koleksi. Koleksi memberi daftar koleksi ke Pustakawan.

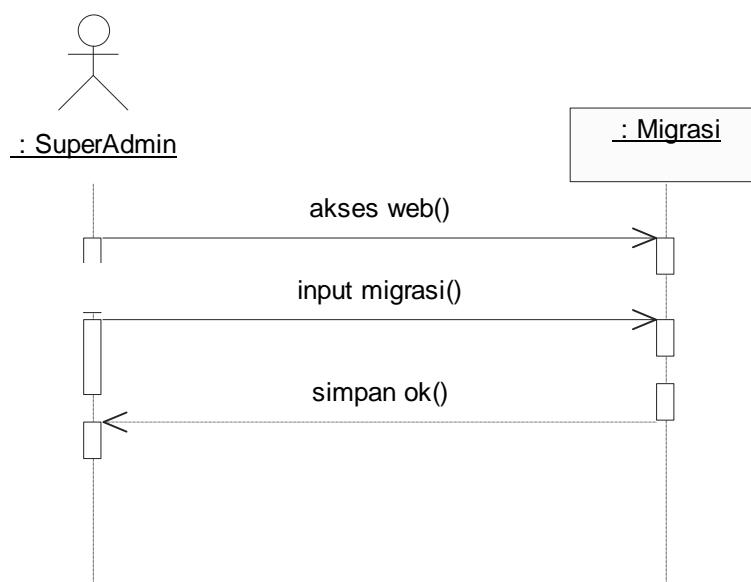
Sequence Merekomendasikan Koleksi



Gambar 20. Sequence Diagram Merekomendasikan Koleksi

Pada Sequence merekomendasikan Koleksi, Anggota membuka web pada Koleksi kemudian Koleksi memberikan formulir. Anggota memberikan data , selanjutnya Koleksi memberikan informasi bahwa koleksi diterima.

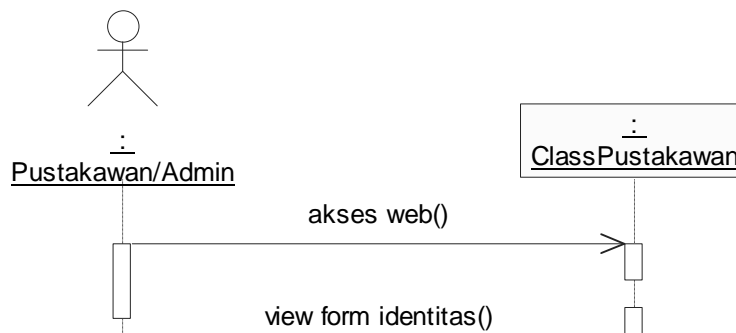
Sequence Migrasi Data



Gambar 21. Sequence Diagram Migrasi Data

Pada Sequence *Migrasi* Data SysAdmin membuka web *Migrasi* kemudian SysAdmin melakukan input *migrasi* . *Migrasi* memberikan informasi bahwa data *migrasi* telah tersimpan.

Sequence Ubah Password Pustakawan

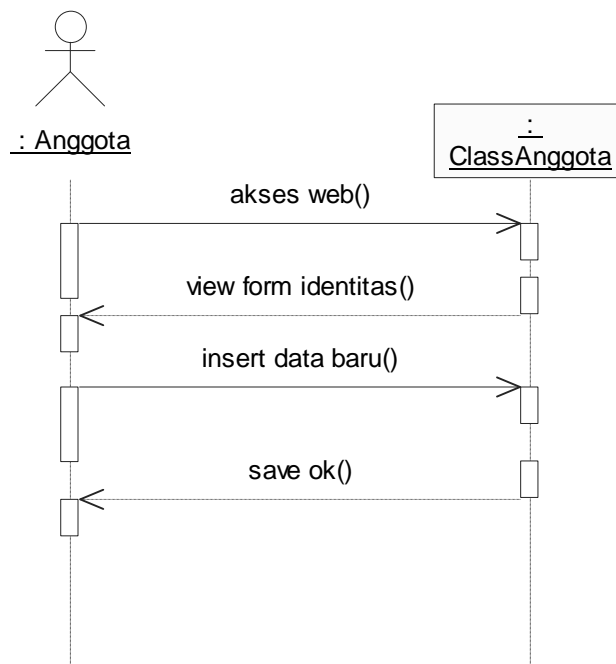


Gambar 22. Sequence Diagram Ubah Password Pustakawan

Pada Sequence Ubah Password Pustakawan ini Pustakawan membuka web, kemudian *class* pustakawan memberikan formulir identitas. Pustakawan mengganti data yang baru kemudian *class*

pustakawan memberikan informasi bahwa data yang baru telah tersimpan.

Sequence Ubah Password Anggota



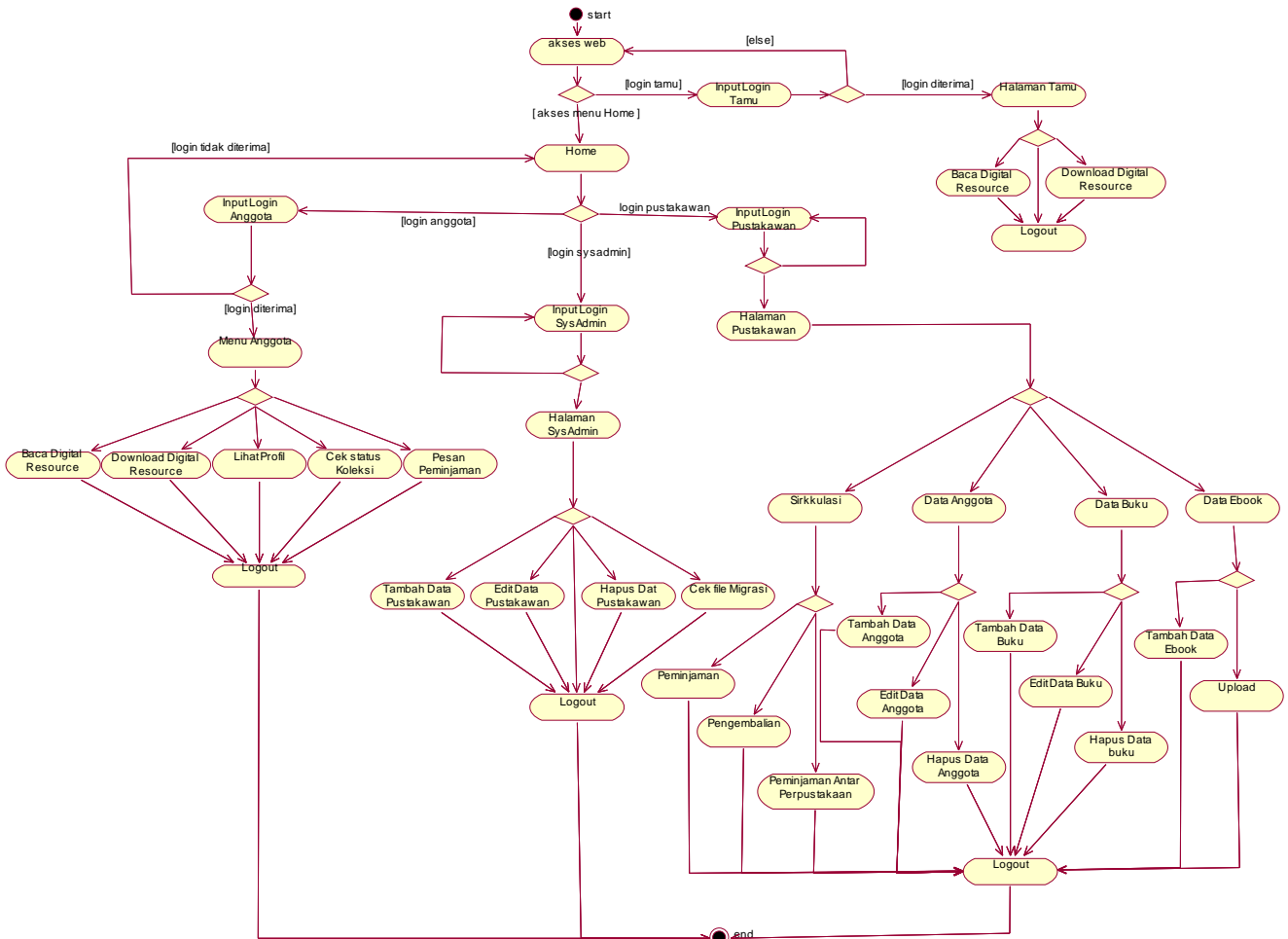
Gambar 23. Sequence Diagram Ubah Password Anggota

Pada Sequence Ubah Password Anggota ini anggota membuka web pada *class* anggota kemudian *class* anggota memberikan formulir

identitas. Anggota mengganti data yang baru, kemudian *class* anggota memberikan informasi bahwa data yang baru telah tersimpan.

Activity Diagram

Sebelum menyajikan activity diagram pada tingkat modul, terlebih dahulu disajikan activity diagram pada level sistem menyeluruh.

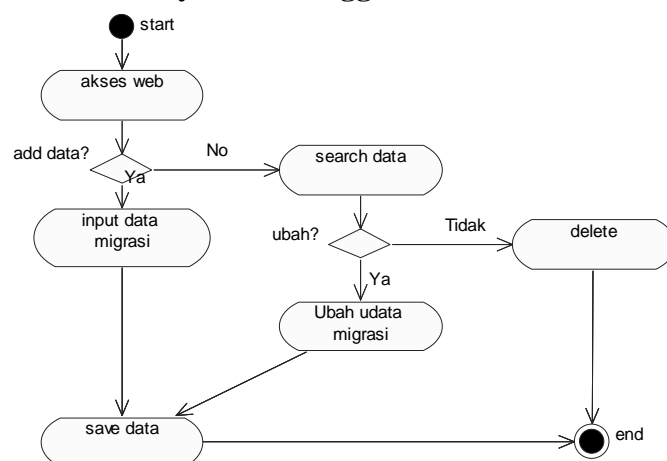


Gambar 24. Activity Diagram Sistem Menyeluruh

Aktifitas pengguna dimulai dengan mengakses web, pada halaman ini tampil daftar koleksi digital tanpa fasilitas download, menu utama dan menu alternatif. Pada divisi area sebelah kanan terdapat 2 (dua) form masing – masing untuk login anggota dan login tamu atau pengunjung umum.

Bagi pengunjung umum yang telah berhasil login dapat mengakses digital *resource* untuk dibaca dan didownload. Bagi anggota setelah berhasil login dapat mengakses profil anggota, cek status peminjaman, cek status koleksi dan pesan peminjaman. Selain itu terdapat akses untuk membaca dan mendownload *digital resource*. Bagi pustakawan yang telah berhasil login maka dapat mengakses peminjaman dan pengembalian, data buku, data e-book, data keanggotaan dan upload digital *resource*. Dalam menu data buku, ebook dan keanggotaan terdapat fasilitas penambahan, pengeditan dan penghapusan data. Terdapat pula fasilitas peminjaman antar perpustakaan apabila ada peminjaman buku oleh anggota perpustakaan lain. Pada sysadmin yang telah berhasil melakukan login maka dapat mengakses pengelolaan data pustakawan, juga melihat file migrasi yang diupload oleh pustakawan.

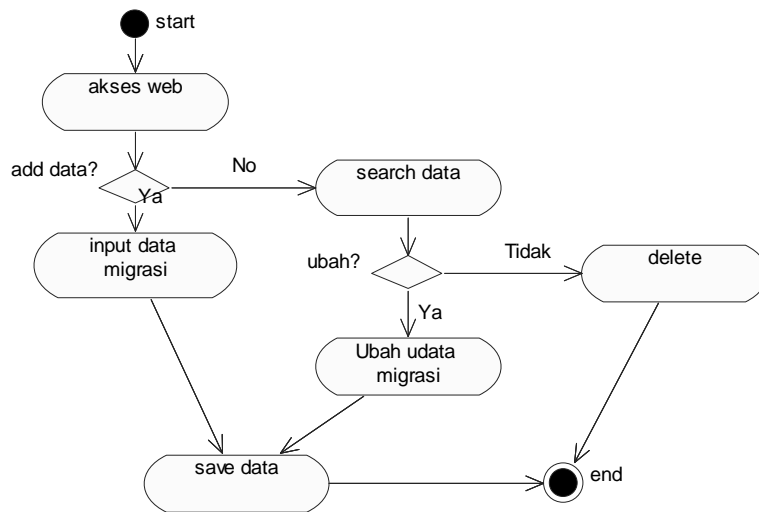
Activity Kelola Anggota



Gambar 25. Activity Diagram Kelola Anggota

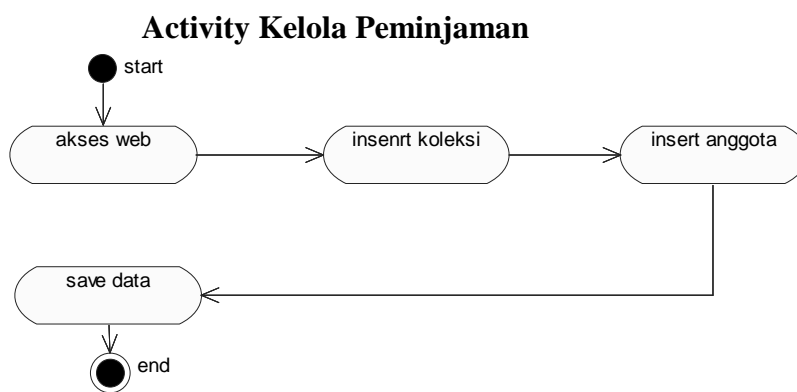
Pada Activity Kelola Anggota diawali dengan membuka web, jika ada penambahan data maka user memasukkan data kemudian data disimpan. Pilihan kedua selain menginput data adalah pencarian data, jika ada pencarian data ditujukan untuk aktifitas update maka update data dilakukan terlebih dahulu sebelum disimpan. Pilihan selain update adalah hapus. Jika data dihapus maka kemudian proses selesai.

Activity Kelola Migrasi



Gambar 26 Activity Diagram Kelola Migrasi

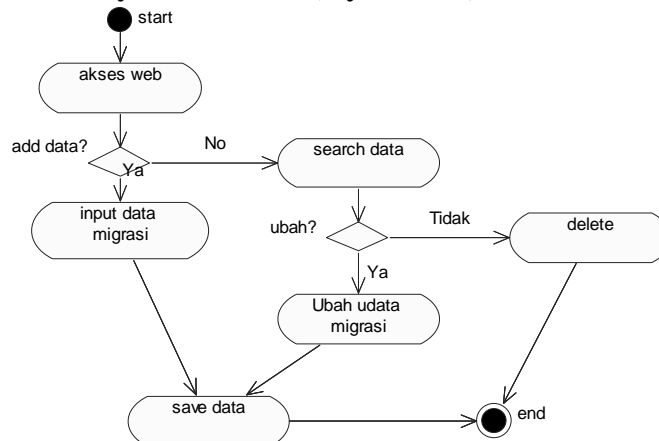
Pada Activity Kelola Migrasi dimulai dari membuka web kemudian oleh Pustakawan jika ada penambahan data maka masukkan data migrasi kemudian data disimpan, jika tidak ada penambahan data maka melakukan pencarian data jika ada perubahan data maka data migrasi di ubah lalu disimpan. Jika tidak maka data dihapus kemudian selesai.



Gambar 27. Activity Diagram Kelola Peminjaman

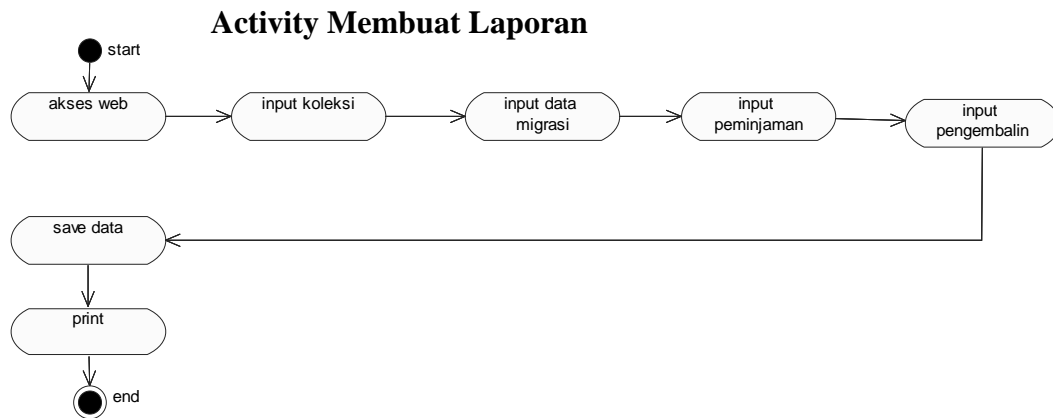
Pada Activity Kelola Peminjaman dimulai dari membuka web kemudian memasukkan koleksi yang akan dipinjam lalu memasukkan data anggota kemudian data di simpan dan proses selesai.

Activity Kelola User (SysAdmin)



Gambar 28. Activity Diagram Kelola User (SysAdmin)

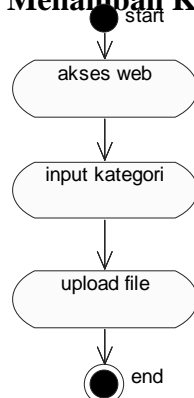
Pada Activity Kelola User di mulai dari membuka web, jika ada penambahan data maka masukkan data migrasi lalu data disimpan, jika tidak ada penambahan data maka melakukan pencarian data, jika ada perubahan data maka ubah data migrasi kemudian data disimpan, jika tidak ada maka data dihapus kemudian proses selesai.



Gambar 29. Activity Diagram Membuat Laporan

Pada Activity Membuat Laporan dimulai dari membuka web kemudian memasukkan koleksi, memasukkan data migrasi, memasukkan peminjaman, memasukkan pengembalian lalu data disimpan kemudian cetak proses selesai.

Activity Menambah Koleksi

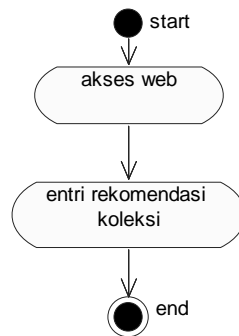


Gambar 30. Activity Diagram Menambah Koleksi

d

a Activity Menambah Koleksi dimulai dari membuka web, mengklik kategori untuk memasukkan data koleksi kemudian upload file proses selesai.

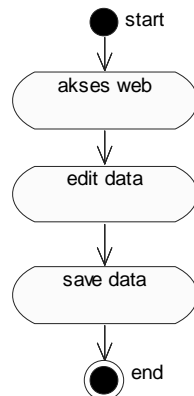
Activity Merekomendasikan Koleksi



Gambar 31. Activity Diagram Merekomendasikan Koleksi

Pada Activity Merekomendasikan Koleksi dimulai dari membuka web, kemudian inputkan rekomendasi koleksi proses selesai.

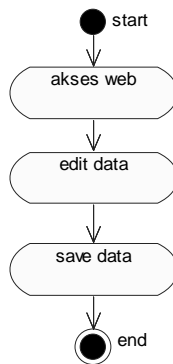
Activity Ubah Password Anggota



Gambar 32. Activity Diagram Ubah Password Anggota

Pada Activity Diagram Ubah Password Anggota dimulai dari membuka web lalu mengedit data kemudian data disimpan, proses selesai.

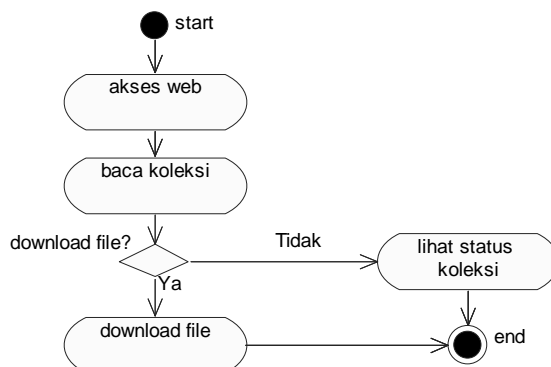
Activity Ubah Password Pustakawan



Gambar 33. Activity Diagram Ubah Password Pustakawan

Pada Activity Ubah Password Pustakawan dimulai dari membuka web kemudian mengedit password kemudian data disimpan, proses selesai.

Activity Mencari Koleksi



Gambar 34. Activity Diagram Mencari Koleksi

Pada Activity Mencari Koleksi dimulai dari membuka web, kemudian membaca koleksi pilihan setelah membaca adalah mendownload file jika tersedia, atau melihat status koleksi.

Desain *Layout*

Desain *Layout* berfungsi sebagai dasar pembuatan interface dalam setiap modul dan halaman web yang dibuat.

Berikut adalah desain layout untuk sistem yang akan di implementasikan.

Nama	
Daftar	

Gambar 35. Desain Lay out Halaman Utama

1. Nama layout : Home

Fungsi : Halaman utama *address* web

Keterangan : Layout halaman web terdapat menu utama, menu linkbar, login anggota dan daftar katalog. Menu utama dapat diakses oleh setiap tingkatan user. Menu linkbar untuk mengakses

ke halaman pustakawan dan sysadmin. Login anggota digunakan untuk ke halaman anggota.

Nama	
Daftar	

Gambar 36. Desain Lay out Halaman Anggota

2. Nama layout : Halaman Anggota

Fungsi : Halaman anggota menampilkan fasilitas untuk anggota

Keterangan : Layout halaman anggota tampil setelah anggota melakukan login. Halaman ini menyediakan fasilitas untuk kepentingan

Nama	
Profil	

Gambar 37. Desain Lay Out Profil Anggota

3. Nama layout : Halaman Profil

Fungsi : Menampilkan profil anggota dan ubah password

Keterangan : Layout mengubah password dengan input password lama

Na
Cek
Daftar Buku Yang Masih

Gambar 38. Desain Lay Out Cek Peminjaman

4. Nama layout : Halaman cek peminjaman
 Fungsi : Untuk menampilkan buku yang dipinjam anggota
 Keterangan : Layout menampilkan buku yang masih dipinjam

Nama
Cek
Cek Status

Gambar 39. Desain Lay Out Cek Koleksi

5. Nama layout : Halaman cek koleksi
 Fungsi : Untuk menampilkan status buku
 Keterangan : Layout menampilkan status buku yang diinginkan

Nama
Pemesa
Input

Gambar 40. Desain Lay Out Pemesanan Koleksi

6. Nama layout : Halaman pemesanan koleksi
Fungsi : Untuk melakukan pemesanan buku / koleksi
Keterangan : Layout menampilkan koleksi yang dipesan

Nama
Katalog
Input
Hasil

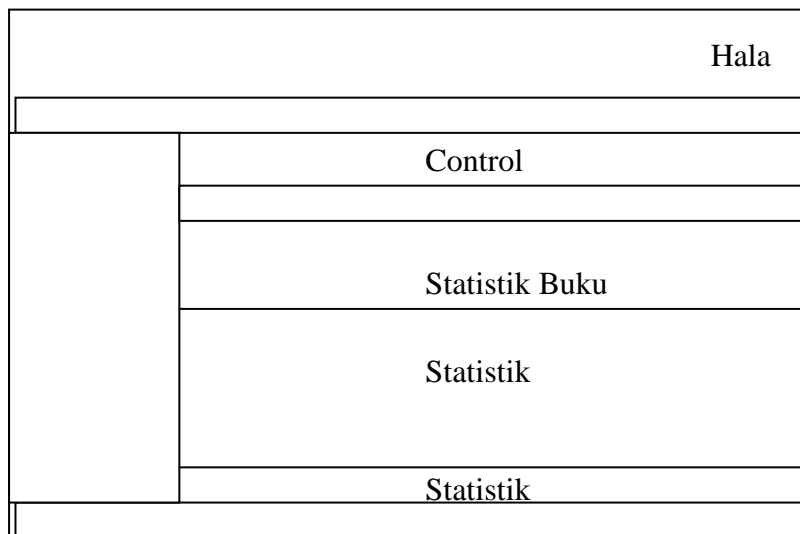
Gambar 41. Desain Lay Out Aplikasi Pencarian

7. Nama layout : Halaman Aplikasi Pencarian
Fungsi : Untuk melakukan pencarian buku / koleksi
Keterangan : Layout menampilkan hasil pencarian buku / koleksi

Nama
Usulan
Inputan

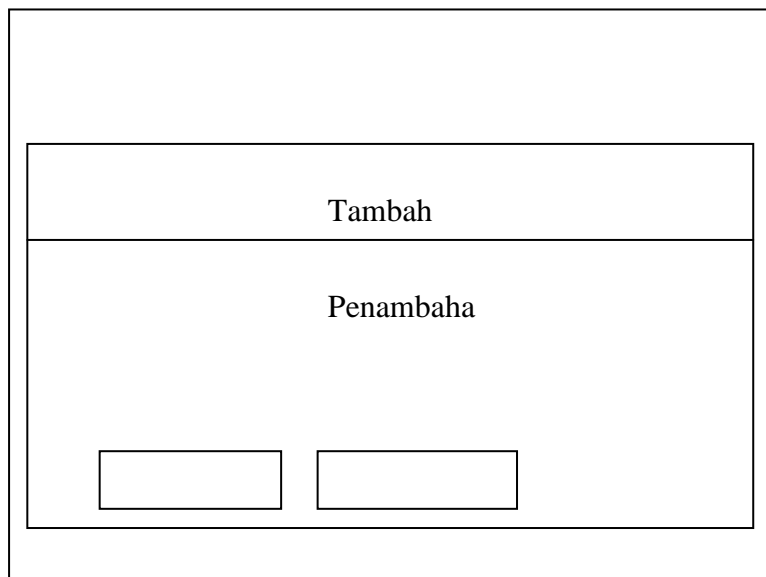
Gambar 42. Desain Lay Out Rekomendasi / Usulan

8. Nama layout : Halaman Rekomendasi / Usulan
Fungsi : Untuk melakukan Usulan Buku
Keterangan : Layout menampilkan hasil Masukan Rekomendasi



Gambar 43. Desain Lay Out Halaman Pustakawan

9. Nama layout : Halaman pustakawan
Fungsi : Untuk melakukan kegiatan pustakawan
Keterangan : Layout menampilkan laporan dan menu pustakawan



Gambar 54. Desain Lay Out Kelola Peminjaman Buku

Gambar 44. Desain Lay Out Kelola Peminjaman Buku

9. Nama layout : Halaman Peminjaman Buku
Fungsi : Untuk mendata layanan peminjaman buku
Keterangan : Layout menampilkan field yang harus di isi

Edit Data	
Edit Data	
<input type="text"/>	<input type="text"/>

Gambar 45. Desain Lay Out Add / Edit Anggota

9. Nama layout : Halaman edit tambah data anggota
Fungsi : Untuk menambah dan update data anggota
Keterangan : Layout menampilkan field yang harus di isi

	Tambah
	Field Buku

Gambar 46. Desain Lay Out Add / Edit Katalog

9. Nama layout : Halaman edit tambah data buku
Fungsi : Untuk menambah dan update data buku
Keterangan : Layout menampilkan field yang harus di isi

3. Code

Tahap Code adalah tahapan dimana desain yang dihasilkan diterjemahkan dengan bahasa pemrograman menjadi bentuk fisik form maupun tabel, fungsi dan prosedur. Tahap Code menggunakan bahasa pemrograman PHP dan database MySQL sebagai penyimpan data. Sistem dijalankan pada localhost sebelum diupload pada *web hosting*. Gambaran mengenai sistem yang dihasilkan dibahas pada Bab IV. Hasil Pembahasan.

4. Tes

Tahapan Test dilakukan dengan menggunakan metode BlackBox Testing. Blackbox testing memberikan hasil tes kesesuaian aplikasi dengan fungsi atau kemampuan sistem secara fungsional. Penjelasan Test ini dibahas pada Bab IV.