BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Peran sistem informasi terhadap kemajuan organisasi sudah tidak diragukan lagi. Dengan dukungan sistem informasi yang baik maka sebuah perusahaan akan memiliki berbagai keunggulan kompetitif. Kualitas barang dan pelayanan yang maksimal bukanlah suatu jaminan untuk sebuah perusahaan menjadi maju namun tidak terlewatkan juga cara penyampain informasi yang cepat dan mudah diakses oleh semua orang dimanapun berada sangat mempengaruhi majunya sebuah perusahaan. Internet dan website merupakan media yang sangat tepat digunakan untuk penyebaran informasi.

Salah satunya adalah Ai Read yang ingin memajukan usahanya dalam menyewakan buku bacaan. Ai Read adalah sebuah taman bacaan yang menyewakan komik, buku dan majalah yang ingin memperkenalkan koleksi buku yang ada kepada khalayak banyak, sehingga dapat menarik minat pelanggan untuk menjadi member. Saat ini koleksi buku yang dimiliki oleh Ai Read sekitar 1000 buku dan jumlah member sekitar 150 member. Dalam usahanya untuk memperkenalkan katalog buku Ai Read masih kurang maksimal, dikarenakan media yang digunakan hanya menggunakan pamflet dan selebaran sehingga penyampaian informasi masih lambat. Ai Read juga mempunyai permasalahan dalam proses peminjaman dan pengembalian buku dikarenakan dalam pencatatan

peminjaman dan pengembalian buku Ai Read masih menggunakan Microsoft Acces yang tabel dan data-datanya belum saling terintegrasi.

Melihat permasalahan yang terjadi pada Ai Read sehingga penulis mempunyai ide untuk membuat media penyebaran informasi mengenai katalog buku, komik serta majalah melalui website Ai Read yang dapat diakses oleh khalayak banyak serta membuat aplikasi untuk proses peminjaman dan pengembalian buku.

Peranan pembangunan *website* Toko dan Penyewaan Buku Ai Read ini diwujudkan dalam bentuk pelayanan digital dengan mengedepankan pelayanan publik demi kepuasan semua pihak. Untuk itu penulis mengambil judul :

"PEMBANGUNAN WEBSITE SEBAGAI MEDIA PROMOSI KATALOG ONLINE PADA TOKO BUKU DAN PENYEWAAN KOMIK AI READ"

1.2 Identifikasi dan Rumusan Masalah

Dalam bab ini akan di jabarkan mengenai identifikasi masalah dan rumusan masalah berdasarkan latar belakang masalah.

1.2.1 Identifikasi Masalah

Dengan melihat salah satu proses bisnis yang terjadi di Toko Buku dan Penyewaan Komik Ai Read penulis mengambil sekelompok kecil masalah yang diambil yang berhubungan dengan judul penelitian yaitu :

a. Pada proses peminjaman dan pengembalian buku, pencarian status buku sering tidak ditemukan serta sering terjadi ketidakcocokan status buku, sehingga mengakibatkan sering terjadinya kehilangan buku dikarenakan

aplikasi yang ada hanya mengguanakan Microsoft acces yang tabel-tabelnya belum terintegrasi satu sama lain.

b. Sulitnya memberikan informasi kepada pelanggan mengenai buku-buku baru yang ada. Dikarenakan media untuk promosi hanya menggunakan pamflet saja.

1.2.2 Rumusan Masalah

Setelah melihat identifikasi masalah yang ada, penulis mencoba merumuskan masalah menurut pandangan penulis disesuaikan dengan teori-teori :

a. Bagaimana membangun aplikasi untuk menangani proses peminjaman dan pengembalian buku.

b. Bagaimanakah membangun *website* di Toko Buku dan Penyewaan Komik Ai Read sebagai sarana tambahan untuk penyampaian informasi serta promosi katalog buku online.

1.3 Maksud dan Tujuan Penelitian

Dalam bab ini penulis akan menjabarkan tentang maksud dan tujuan penelitian

1.3.1 Maksud Penelitian

Maksud dari penelitian tugas akhir ini adalah membangun sebuah website sebagai media promosi dan penyebaran informasi mengenai katalog buku dan komik yang dapat meningkatkan jumlah pelanggan.

1.3.2 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Membuat suatu aplikasi peminjaman dan pengembalian buku menggunakan bahasa pemrograman php untuk mempermudah dan mempercepat proses peminjaman dan pengembalian buku serta mencegah hilangnya buku karena pencarian status buku yang tidak benar.
2. Membuat sistem informasi berbasis *website* di Toko Buku dan Penyewaan Komik Ai Read yang baik sehingga dapat mempermudah masyarakat dalam mendapatkan informasi khususnya tentang buku serta untuk memudahkan penyampaian sebuah informasi yang disampaikan Toko Buku dan Penyewaan Komik Ai Read kepada pubik sehingga ada interaksi antar keduanya.

1.4 Kegunaan Penelitian

Pada bab ini penulis menjabarkan kegunaan penelitian yang dilakukan pada Toko penyewaan buku dan komik Ai Read

1.4.1 Kegunaan Teoritis

Dari penelitian yang telah dilakukan diharapkan dapat memberikan kegunaan-kegunaan antara lain :

Kegunaan secara teoritis penelitian ini adalah menemukan permasalahan-permasalahan yang ada di Toko Buku dan Penyewaan Komik Ai Read sesuai dengan teori yang ada sehingga nantinya untuk pengembangan teori-teori yang ada sebagai perkembangan ilmu pengetahuan.

1.4.2 Kegunaan Praktis

Adapun kegunaan praktis antara lain :

a. Bagi Penulis

Diharapkan penelitian ini bisa menambah pengetahuan dan wawasan bagi penulis dan sekaligus bisa memahami lebih dalam, khususnya dalam hal pembangunan website Ai Read sehingga kelak dapat digunakan baik dalam teori maupun praktiknya.

b. Bagi Ai Read, hasil penelitian ini mudah-mudahan bisa mempercepat penyebaran informasi, serta mempermudah pencarian status buku sehingga dapat menghindari kehilangan buku dikarenakan status buku yang tidak cocok dengan realita.

1.5 Lokasi dan Jadwal Penelitian

Penulis melakukan penelitian Pembangunan Website sebagai media promosi catalog online pada Toko Buku dan Penyewaan Komik Ai Read, yang berlokasi di Jalan Jalaprang No. 16 Bandung 40124

Tabel 1.1

Perencanaan Jadwal Penyelesaian

No.	Nama Kegiatan	Oktober				November				December			
		1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
1	Pembuatan Proposal												
2	Insepsi												
3	Elaborasi												
4	Konstruksi												
5	Transisi												

.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Pengertian sistem dapat dikelompokan menjadi dua kelompok dalam mendifinisikan sistem yaitu penekanan pada prosedurnya dan menekankan pada komponen atau elemennya. Definisi sisitem itu sendiri yang menekankan pada komponen atau elemen memiliki pengertian yang lebih luas dari pada penekanan pada prosedur karena pengertian tersebut lebih diterima dikarenakan suatu sistem terdiri dari beberapa subsistem-subsistem.

Menurut **[Jog5]** sistem adalah kumpulan dari elemen - elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

Berdasarkan difinisi diatas dapat dikemukakan bahwa sistem dibentuk dari bagian-bagian yang mempunyai tujuan yang sama dengan hubungan yang saling membutuhkan untuk mencapai tujuan tertentu. Pendefinisian sistem itu sendiri bisa diartikan berdasarkan penggunaan pendekatan komponen yang dilakukan oleh penggunanya.

2.1.1 Bentuk Dasar Sistem

Bentuk umum dari suatu sistem terdiri atas masukan (*input*), proses dan keluaran (*output*). Dalam bentuk umum sistem ini terdapat satu atau lebih masukan yang akan diproses dan akan menghasilkan suatu keluaran.



(Sumber: Jogiyanto, 2005 : 4)

Gambar 2.1 Bentuk Dasar Sistem

2.1.2 Karakteristik Sistem

[Jog5] Bahwa suatu sistem mempunyai karakteristik atau sifat - sifat tertentu, yaitu memiliki komponen – komponen (*components)*, batas sistem (*boundary*), lingkungan sistem (*Environment*), penghubung (*Interface*), masukan (*Input*), keluaran (*Output*), pengolahan (*Proses*), dan sasaran (*Objective*), dan tujuan (*Goal*).

1. Komponen sistem

   Suatu sistem terdiri dari jumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem atau elemen – elemen sistem dapat berupa suatu subsistem atau bagian – bagian dari sistem. Setiap subsistem mempunyai sifat – sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi suatu sistem secara keseluruhan.

2. Batas sistem (*boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem di pandang sebagai satu kesatuan. Batas suatu sistem menunjukan ruang lingkup dari sistem tersebut.

3. Lingkungan sistem (*environment*)

   Lingkungan luar dari sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan (harus dijaga dan merupakan energi dari sistem) dan dapat bersifat merugikan (harus ditahan dan dikendalikan).

4. Penghubung sistem (*interface*)

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber – sumber daya mengalir dari subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya melalui penghubung, dengan penghubung satu subsistem dapat berinteraksi yang lainnya membentuk satu kesatuan.

5. Masukan sistem (*input*)

   Masukan (*input*) adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). Maintenance input adalah energi yang masukan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran sistem (*output*)

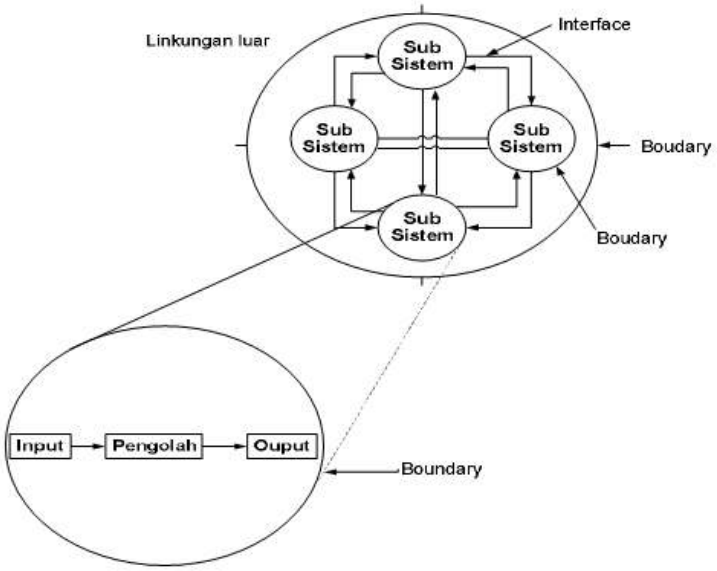
   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan subsistem yang lain atau kepada supra sistem.

7. Pengolah sistem

   Suatu sistem dapat mempunyai bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem

   Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan.



(Sumber: Jogiyanto, 2005 : 6)

Gambar 2.2 Karakteristik Sistem

2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut ini **[Jog5].**

1. Sistem diklasifikasikan berdasarkan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak, misalnya sistem teologi. Sistem fisik adalah sistem yang ada secara fisik misalnya sistem komputer.

2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia.

   Sistem alamiah adalah sistem yang terjadi melalui proses alam dan tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia yang melibatkan interaksi manusia dengan mesin yang disebut dengan *human-machine system* atau *man- machine system.*

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*).

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantaranya bagian-bagiannya dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Misalnya sistem pada komputer. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat dipredisikan karena mangandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*).

   Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara

otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataanya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara ralatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secare relatif tertutup karena sistem tertutup akan secara otomatis dan terbuka hanya untuk pengaruh yang baik.

2.1.4 Pengertian Informasi

Informasi merupakan hasil dari data yang di proses yang berasal dari inputan suatu kejadian yang nyata yang berguna bagi pemakainya. Pengertian informasi diatas merupakan pengembangan teori yang di kemukakan oleh **[Jog5]** yang menyatakan bahwa informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.1.5 Konsep Dasar Informasi

Informasi merupakan hasil dari pengolahan yang disajikan secara tepat dan akurat. Sumber dari informasi adalah data, data merupakan bentuk jamak dari bentuk tunggal atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata kejadian-kejadian (*event*). Kejadian adalah suatu yang terjadi pada saat tertentu yang menyangkut perubahan nilai yang disebut transaksi. Sedangkan kesatuan

nyata adalah berupa suatu objek yang nyata dan terjadi pada saat kejadian berlangsung. Data juga dapat diartikan suatu yang perlu diolah terlebih dahulu untuk mendapatkan suatu informasi. Menurut **[Jog5]** kualitas informasi yang diharapkan targantung 4 (empat) hal pokok yaitu:

1. Akurat

   Akurat mempunyai arti informasi yang dihasilkan harus bebas dari kesalahan-kesalahan, yang tidak biasa, tidak menyesatkan dan mencerminkan maksudnya.

2. Tepat waktu

   Tepat waktu berarti informasi yang disampaikan ke penerima tidak terlambat, kerena informasi adalah landasan untuk mengambil suatu keputusan. Untuk itu diperlukan suatu teknologi untuk dan mengirim dengan cepat dan tepat.

3. Relevan

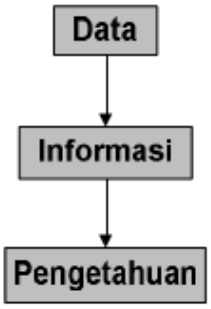
   Berarti informasi mempunyai manfaat dan berguna bagi pemakainya. Karena batas relevansi seseorang berbeda, maka informasi bisa dikatakan berguna jika benar-benar berguna dan dibutuhkan pemakainya.

4. Aman

   Aman berarti informasi harus terbebas dari penyadapan oleh pihak orang yang tidak berwenang dalam pengguna informasi tersebut.

2.1.6 Hirarki Informasi

Ingformasi memiliki urutan atau tingkatan yang disebut sebagai hirarki makna informasi, yaitu tingkatan informasi dilihat dari kegunaan atau asas manfaatnya.



(Sumber: Jogiyanto, 2005 : 9)

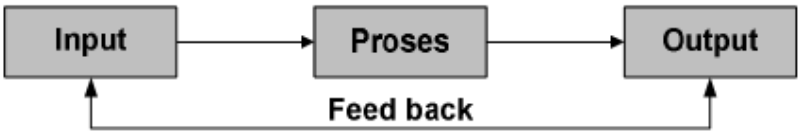
Gambar 2.3 Hirarki Informasi

Dari gambar diatas, menurut **[Jog5]** terjadi 3 (tiga) bagian utama, yaitu :

1. Data, yaitu gambaran dari suatu kejadian nyata.
2. Informasi, yaitu suatu hasil dari pengolahan data yang memiliki nilai guna bagi penerima yang membutuhkan.
3. Pengetahuan, yaitu nilai lebih yang didapat dari sebuah informasi.

2.1.7 Siklus Hidup Informasi

Dari penjelasan sebelumnya telah dikemukakan bahwa informasi adalah merupakan hasil dari pengolahan data. Supaya informasi yang didapat sesuai dengan yang diinginkan dan dapat melangsungkan hidupnya, maka diperlukan *feedback* atau umpan balik dari sistem yang berjalan.



(Sumber: Jogiyanto, 2005 : 9)

Gambar 2.4 Siklus Hidup Informasi

2.1.8 Sistem Informasi

Sistem yang ada berfungsi sebagai penghasil suatu keluaran, baik berupa informasi maupun berupa objek / benda. Untuk keluaran berupa informasi, maka sistem tersebut dikatakan sebagai sistem informasi.

2.1.9 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen-komponen sistem yang berapa didalam suatu ruang lingkup organisasi, saling berinteraksi untuk menghasilkan sebuah informasi yang bertujuan untuk pihak manajemen tertentu dan untuk mencapai tujuan tertentu.

Menurut **[Jog5]** factor – factor yang menentukan kehandalan dari suatu sistem informasi atau informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

1. Keunggulan (*usefulness*)

   Yaitu suatu sistem yang harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.

2. Ekonomis

   Kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal, sebesar biayanya.

3. Kehandalan (*Reliability*)

   Keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.

4. Pelayanan (*Customer Service*)

   Yakni suatu sistem memberikan palayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.

5. Kapasitas (*Capacity*)

   Setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

6. Sederhana dalam kemudahan (*Simplicity*)

   Sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti dan prosedure mudah diikutin.

7. Fleksibel (*Fleksibility*)

   Sistem informasi ini harus dapat digunakan dalam kondisi yang bagaimana yang diinginkan oleh organisasi tersebut atau pengguna tertentu.

2.1.10 Komponen Sistem Informasi

Istilah dalam komponen sistem informasi adalah blok bangunan (*building block*) yang dapat di bagi menjadi enam blok **[Jog5]**, yaitu :

1. Blok masukan (*input block*)

   Blok input merupakan data-data yang masuk ke dalam sistem informasi, yang dapat berupa document-document dasar yang dapat diolah menjadi suatu informasi tertentu.

2. Blok model (*model block*)

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan mengolah data input untuk menghasilkan suatu informasi yang dibutuhkan.

3. Blok keluaran (*output block*)

   Merupakan informasi yang menghasilakn sekumpulan data yang nantinya akan disimpan berupa data ceta laporan.

4. Blok teknologi (*technologi block*)

   Blok teknologi merupakan penunjang utama dalam berlangsungnya sistem informasi. Yang memiliki beberapa komponen yaitu diantaranya alat masukan data (*input device*), alat untuk menyimpan dan mengakses data (*stronge device*), alat untuk menghasilkan dan mengirimkan keluaran (*output device*), dan alat untuk membentuk pengendalian sistem secara keseluruhan (*control device*). Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *brainare*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok basis data (*database block*)

   Basis data merupakam kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan

digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dan perlu di organisasi sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

6. Blok kendali (*control block*)

   Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat di cegah bila terlanjur terjadi kesalahan-kesalahan kerusakan dalam penggunaan sistem.

2.2 Metoda Analisa dan Pengembangan Sistem

Metode yang dipakai dalam kegiatan analisa dan pengembangan sistem ini adalah metode analisa dan perancangan yang terstuktur menggunakan *Data Flow Diagram* dan *Entity Relation Diagram.* Data Flow Diagram sering digunakan untuk menggambarkan suatu sistem yang ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data itu mengalir. Data Flow Diagram merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Sedangkan objek utama dari pembuatan *Entity Relationship Diagram* adalah untuk menunjukan objek-objek (himpunan Entitas) apa saja yang ingin digunakan dalam sebuah basis data dan bagaimana hubungan yang terjadi diantara objek-objek tersebut. Langkah-langkah dasar dalam pembuatan sistem adalah sebagai berikut :

1. Identify, yaitu mengidentifikasi masalah
2. Understand, memahami cara kerja dari sitem yang telah ada
3. Analize, menganalisis sistem

4. Report, membuat laporan hasil analis

2.2.1 Analisis Sistem

Tahap Analisis Sistem dilakukan setelah tahap perancangan sistem dan sebelum tahap perancangan sistem. Tahap analis sistem merupakan tahap yang kritis dan sangat penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan di tahap berikutnya.

2.2.2 Perancangan Sistem

Perancangan sistem dilakukan setelah tahap analis sistem dilakukan, analis sistem telah menggambarkan dengan jelas apa yang harus dikerjakan dan bagi analis untuk memikirkan bagaimana membentuk suatu sistem tertentu.

Tujuan dan perancangan sistem adalah sebagai berikut :

1. Untuk memenuhi kebutuhan pada pemakai sistem
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemogram komputer.

2.2.3 Pengembangan Sistem

Secara umum tujuan pengembangan sistem informasi adalah untuk memberikan kemudahan dalam menyampaikan informasi, mengurangi biaya dan menghemat waktu, meningkatkan pengendalian, mendorong pertumbuhan, meningkatkan produktivitas serta profitabilitas organisasi.

Pengembangan sistem dapat berarti penyusunan suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau

memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki atau diganti disebabkan karena beberapa hal, yaitu sebagai berikut :

1. Adanya permasalahan-permasalahan yang timbul pada sistem yang lama. Permasalahan yang timbul dapat berupa :

   a. *Ketidakberesan*

      Ketidakberesan pada sistem yang lama menyebabkan sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan. Ketidakberesan ini dapat berupa:

      1. Kekacauan-kekacauan disengaja yang menyebabkan tidak amannya keuangan perusahaan dan kebenaran dari data menjadi kurang terjamin.
      2. Kesalahan-kesalahan yang tidak disengaja yang juga dapat menyebabkan kebenaran dari data kurang terjamin.
      3. Tidak efisiennya operasi.
      4. Tidak ditaatinya kebijaksanaan manajemen yang sudah ditetapkan.

   b. *Pertumbuhan Organisasi*

      Pertumbuhan organisasi yang menyebabkan harus disusunnya sistem yang baru. Pertumbuhan organisasi diantaranya adalah kebutuhan informasi yang semakin luas. Volume pengolahan data semakin meningkat, perubahan prinsip akuntasi yang baru. Karena adanya perubahan ini, maka menyebabkan sistem yang lama tidak efektif lagi sehingga sistem yang lama sudah tidak dapat memenuhi lagi semua kebutuhan informasi yang dibutuhkan oleh manajemen.

2. Untuk meraih kesempatan-kesempatan

   Teknologi informasi telah berkembang dengan cepatnya. Perangkat keras komputer, perangkat lunak dan teknologi komunikasi telah begitu cepat berkembang. Organisasi mulai merasakan bahwa teknologi informasi ini perlu digunakan untuk meningkatkan penyediaan informasi sehingga dapat mendukung dalam proses pengambilan keputusan yang akan dilakukan oleh manajemen. Dalam keadaan pasar bersaing, kecepatan informasi atau efisiensi sangat menentukan berhasil atau tidaknya strategi dan rencana-rencana yang telah disusun untuk meraih kesempatan-kesempatan yang asa. Bila pesaing dapat memanfaatkan teknologi ini, maka kesempatan-kesempatan akan jatuh ke tangan pesaing. Kesempatan-kesempatan ini dapat berupa peluang-peluang pasar, pelayanan yang meningkat kepada pelanggan dan lain sebagainya

3. Adanya instruksi-instruksi

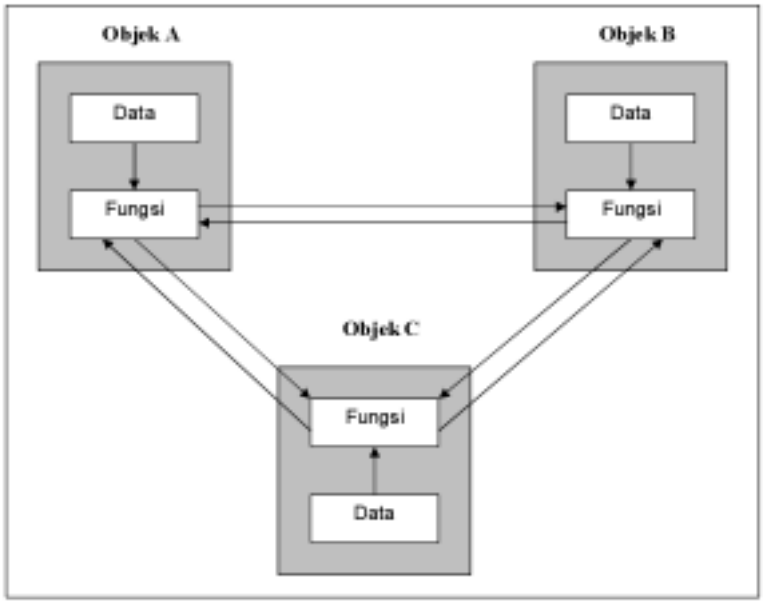
   Penyusunan sistem yang baru dapat juga terjadi karena adanya instruksi-instrukai dari atas pimpinan atau dari luar organisasi, seperti peraturan pemerintah dan sebagainya

2.3 Metode Pendekatan Sistem Berorientasi Objek

Menurut **[Adi05]** ada beberapa karakterisitik yang menjadi ciri-ciri dari pendekatan berorientasi objek adalah :

1. Pendekatan lebih pada data dan bukannya pada prosedur/fungsi.
2. Program besar dibagi pada apa yang dinamakan objek-objek.
3. Struktur data dirancang dan menjadi karakteristik dari objek-objek.

4. Fungsi-fungsi yang mengoperasikan data tergabung dalam suatu objek yang sama.
5. Data tersembunyi dan terlidung dari prosedur/fungsi yang ada di luar.
6. Objek-objek dapat saling berkomunikasi dengan saling mengirim *message* (pesan) satu sama lain.
7. Pendekatan adalah dari bawah ke atas (*bottom up approach*).



Gambar 2.5 (Sumber: Adi Nugroho,2005:17)

Pendekatan beorientasi objek membuat data terbungkus pada setiap fungsi/prosedur dan melindunginya terhadap perubahan tidak dikehendaki dari fungsi yang berada di luar.

[Lar04] mendefinisikan analisis adalah proses mendekomposisi atau menginvestigasi *domain problem* atau *requirements* menjadi bagian-bagian yang bisa dimengerti. Proses analisis lebih menekankan kepada pertanyaan seperti

apakah sistem yang akan dikerjakan, bagaimana sistem tersebut digunakan sehari-hari oleh user.

Perancangan adalah melakukan sintesa model software dari problem yang di analisis, menterjemahkan domain model kepada software model sehingga bisa dimengerti dan dikomunikasikan kepada anggota tim lain dan juga para professional software lainnya. Bisa disebut juga analisis adalah menentukan apa yang seharusnya dikerjakan, sedangkan perancangan adalah menentukan cara terbaik untuk mengerjakan sesuatu dengan benar setelah diketahui apa yang harus dikerjakan. Di sini terlihat bahwa analisis dilakukan sebelum perancangan. Analisis berorientasi objek menekankan kepada menemukan dan menjelaskan objek-objek dan interelasinya yang menjadi bagian atau pemain dalam sebuah sistem. Perancangan berorientasi objek memodelkan objek-objek yang diidentifikasi pada proses analisis dan interaksi antar objek ke dalam model software. Proses selanjutnya bisa mengidentifikasi atribut dan operasi objek-objek lainnya.

2.4 Metode Pengembangan Sistem

2.4.1 *Agile*

Metode ***agile*** sedang populer saat ini dalam metode pengembangan software. Dalam salah satu kamus bahasa Inggris, ***agile*** diterjemahkan sebagai "pergerakan yang cepat dan ringan" yang bersinonim dengan kata "*nimble*". Dengan demikian metode yang "***agile***" bisa didefinisikan sebagai metode yang berkarakter "cepat dan ringan".

[Lar04] mengemukakan selain itu metode yang ***agile*** juga mengutamakan kesederhanaan, komunikasi yang intens antar anggota tim, organisasi tim yang fleksible dan menyesuaikan dengan kebutuhan. Metode ***agile*** biasanya menerapkan iterasi yang dibatasi oleh waktu dan menghasilkan produk yang dihasilkan secara inkremental. Ciri lain dari metode ***agile*** adalah adaptasi dan respons yang cepat terhadap adanya perubahan.

Secara spesifik, praktek dari metode ***agile*** ini bermacam-macam di antaranya adalah ***eXtreme Programming*** (**XP**) yang dikonsep oleh *Kent Beck* yang cukup populer dan fenomenal di awal kemunculannya. Selain **XP**, metode lain yang berkarakter ***agile*** adalah *Scrum*, *Lean Development*, *DSDM*, *Feature-Driven Development*, dan tentu saja ***Unified Process*** (**UP**).

Manifesto ***Agile***:

1. Mengutamakan orang-orang dan interaksi (daripada proses dan tools)
2. Mengutamakan software yang berjalan (daripada dokumentasi yang lengkap)
3. Mengutamakan kerjasama pelanggan (daripada negosiasi kontrak)
4. Mengutamakan respons terhadap perubahan (daripada mengikuti rencana)

Prinsip ***Agile***:

1. Prioritas utama adalah memuaskan pelanggan dengan menghasilkan produk dengan cepat dan berkala.
2. Menerima dengan baik requirement yang berubah, walaupun pengerjaan software sudah masuk ke fase pengembangan. Prinsip

agile mengutamakan kepentingan kompetitif pelanggan yang kadang kala memerlukan perubahan requirement tersebut.

3. Menghasilkan produk berkala dengan cepat, dalam hitungan beberapa minggu sampai beberapa bulan, di mana berlaku prinsip lebih cepat lebih baik.
4. Bisnis analis dan pengembang software harus bekerja sama setiap hari selama proyek berlangsung.
5. Menjalankan proyek dengan orang-orang yang bermotivasi tinggi. Manajemen percaya, mendukung mereka, dan menjaga motivasi tinggi mereka dengan memberikan fasilitas yang mereka perlukan.
6. Mengutamakan mekanisme komunikasi secara langsung antar anggota tim melalui pertemuan-pertemuan yang diadakan berkala dalam sebuah ruangan dengan alat-alat yang mendukung seperti papan *whiteboard* yang lebar.
7. Ukuran utama keberhasilan adalah aplikasi yang bisa berjalan.
8. Proses ***agile*** mendukung proses pengembangan software yang terprediksi, dan memiliki kans keberhasilan yang tinggi (*sustainable*).
9. Manajemen, sponsor, pengembang, dan pelanggan harus bisa memelihara semangat yang konstan selama proyek berlangsung.
10. Perhatian secara terus menerus terhadap kemampuan teknis dan perancangan yang baik akan menambah karakter ***agility***.

11. Kesederhanaan - didefinisikan sebagai seni untuk memperkecil jumlah pekerjaan - adalah sangat penting.
12. Arsitektur terbaik, analisis, dan perancangan berasal dari tim yang terorganisasi dengan sendirinya.
13. Dalam interval tertentu, tim selalu berorientasi untuk menjadi lebih baik, dan mengubah langkah-langkahnya untuk penyesuaian.

2.4.2 *Unified Process (UP)*

Unified Procces (UP) juga biasa disebut RUP. Menurut **[Fow05]** RUP merupakan corak keseluruhan dan terminologi yang terdapat pada produk resmi Rational Software yang berdasarkan UP. UP pada dasarnya merupakan proses interatif dan incremental.

Pendapat dari [Lar04] UP mendefinisikan keseluruhan proses, tahapan, dan produk yang dihasilkan dalam setiap tahapan pembuatan software. Tahapan-tahapan ini biasanya terdiri atas: requirement, analisis, perancangan, implementasi, pengujian, dan penerapan. **UP** memiliki ciri-ciri utama sebagai berikut:

1. **UP** tergolong ke dalam metode yang berulang (iteratif). Metode yang berulang berarti membagi keseluruhan tahap dalam sebuah pembuatan software ke dalam beberapa tahap yang identik dan relatif singkat, di mana proses dan hasil dari tahap berikutnya bergantung kepada hasil *review* dari tahap sebelumnya dan *feedback* dari pihak-pihak yang berkepentingan.

2. **UP** mendefinisikan struktur untuk menjelaskan bagaimana tahapan-tahapan dalam keseluruhan proses dilakukan.
3. **UP** tergolong metode yang fleksibel yang cocok diterapkan terhadap pendekatan *agile* (ringan dan cepat). Pendekatan *agile* saat ini sedang populer karena kecepatan dan fleksibilitasnya.

Pendekatan iteratif dari metode UP ini kontras dengan pendekatan metode *waterfall* yang mencoba untuk mendefinisikan tahapan-tahapan pembuatan *software* ke dalam satu iterasi dimana tahapan *requirement,* analisis dan perancangan disesuaikan seluruhnya diawal sebelum tahapan implementasi dan pengujian dilakukan. Sebaliknya UP melihat keseluruhan proses sebagai iterasi dari sub proses yang terdiri atas tahapan yang lengkap di mana analisis, perancangan, dan implementasi dilakukan dalam setiap sub proses sesudahnya bersifat *evolutionary*. Artinya sub proses lanjutan dikembangkan berdasarkan hasil dari sub proses sebelumnya berdasarkan *feedback* dan *review* dari pihak-pihak yang berkepentingan. Karena itu metode pengembangan sistem model *waterfall* tidak sesuai dengan filosofi UP **[Fow05]**

Untuk menerapkan metode **UP dapat menggunakan** prinsip-prinsip *agile*. *Agile* adalah sebuah pendekatan yang mengutamakan kecepatan dan keringanan. Ringan dalam arti tidak banyak hal yang dilakukan dan tidak banyak dokumentasi yang dihasilkan. **UP** memang sekilas tidak tergolong dalam metode yang cepat dan ringan. Aktivitas yang didefinisikan dalam metode **UP** ini tergolong cukup banyak dengan dokumentasi yang juga tidak sedikit. Tetapi masih tetap bisa menerapkan pendekatan *agile* ini dalam metode

UP. Pertama jangan anggap semua aktifitas dan dokumentasi dalam **UP** sebagai hal yang wajib dilakukan. Ambil sebagian saja yang memang perlu karena memiliki nilai tambah apabila dilakukan. **UP** juga tergolong metode yang iteratif dan inkremental. Analisis dan perancangan tidak sepenuhnya dikerjakan di awal sebelum pemrograman dimulai. Pendekatan *agile* juga memiliki ciri demikian.

Pemodelan UML dengan pendekatan *agile* yang lebih mementingkan pemodelan untuk kepentingan komunikasi daripada untuk dokumentasi. Pendekatan *agile* juga lebih banyak menggunakan UML sebagai sketsa dibandingkan sebagai *blueprint* maupun bahasa pemrograman.

Terdapat empat fase besar yang mencirikan metode **UP**. Keempatnya dijelaskan secara singkat sebagai berikut:

1. Insepsi (*Inception*)

   Fase *inception* lebih menitikberatkan kepada pembahasan visi bisnis, penentuan-batasan-batasan system, dan estimasi-estimasi secara kasar. Pada fase ini kegiatan yang dilakukan lebih bersifat menentukan arah dan batasan-batasan tentang apa yang akan dilakukan, termasuk juga memperkirakan secara kasar besaran-besaran *project* seperti sumber daya dan waktu yang diperlukan.

2. Elaborasi (*Elaboration*)

   Pada fase *elaboration* mulai dibentuk arsitektur utama yang membentuk system secara iteratif. Identifikasi resiko tinggi juga dilakukan pada fase ini. Identifikasi dari requirement inti dan memiliki resiko tinggi sudah

dilakukan dalam fase ini. Implementasi (yang menghasilkan program yang bisa dijalankan) arsitektur dari aplikasi juga sudah mulai dilakukan. Perkiraan-perkiraan besaran yang didefinisikan dalam fase Inception dibuat lebih realistis.

3. Konstruksi (*Construction*)

   Construction adalah fase pengembangan yang melengkapi system secara keseluruhan. Pada fase ini dilakukan implementasi secara iteratif terhadap semua *requirement* yang ada.

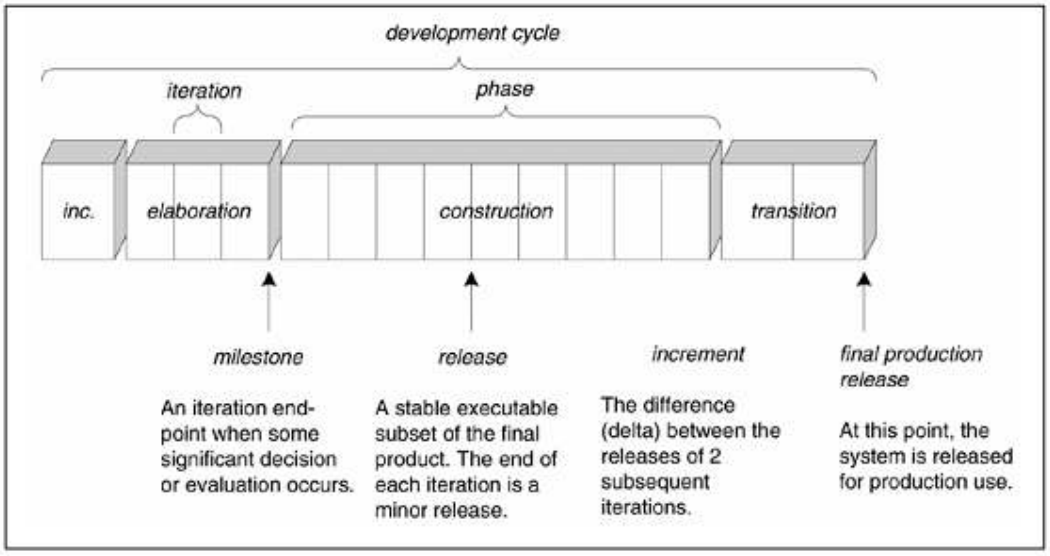
4. Transisi (*Transition*)

   Pada fase *transition*, pengujian sudah dilakukan sampai diserahkannya produk jadi ke pelanggan. *Beta test* dan persiapan penerapan aplikasi dilakukan dalam fase ini.

Walaupun urutan-urutan fase-fase UP mirip *waterfall*, secara karakteristik keduanya tentu saja berbeda. Karakter utama *waterfall* terdapat dalam iterasi tunggal dimana *requirement,* analisis, perancangan, konstruksi dan pengujian software dilakukan dengan sekuensial. Sebaliknya, UP tergolong metode yang iteratif.

Fase *inception* UP tidak sama dengan tahap *requirement waterfall. Inception* lebih menekankan kepada kegiatan-kegiatan yang lebih mendukung *feasibilitas* dari sebuah projek. Fase *elaboration* juga tidak sama dengan fase analisis dan perancangan pada *waterfall.* Dalam fase ini, arsitektur inti dari aplikasi didefinisikan, dianalisis, dirancang, dibuat dan diuji. Aktifitas-aktifitas tersebut dilakukan secara iteratif dan inkremental dengan batasan waktu yang

jelas dalam satu iterasinya. Demikian juga dengan fase *construction* dan *transition.*



(Sumber : Craig Larman, 2004: chapter 210)

Gambar 2.6 Proses Iterasi Tiap-Tiap Fase UP

Pendapat dari **[Lar04]** masing-masing fase dalam UP dapat dipecah ke dalam beberapa *iteration* / pengulangan. Suatu iterasi adalah suatu pengembangan mengulang yang akan menghasilkan rilis dari produk yang dapat di eksekusi, suatu **subset** dari produk akhir di bawah pengembangan, yang bertumbuh secara inkremental dari iterasi ke iterasi untuk menghasilkan sistem akhir.

Menurut **[Lar04]** beberapa manfaat pendekatan iteratif dalam rekayasa perangkat lunak dibandingkan dengan proses *waterfall* / air terjun yang tradisional, proses iteratif mempunyai keuntungan-keuntungan sebagai berikut :

1. Resiko-resiko sebelumnya yang dapat dikurangi
2. Perubahan lebih dapat dikendalikan
3. Tingkat penggunaan kembali yang lebih tinggi dan menguntungkan

4. Tim dalam proyek dapat belajar dalam perjalanannya
5. Secara menyeluruh mutu lebih baik

2.5 Konsep Dasar Pengolahan Data

Sistem pengolahan data merupakan suatu sistem di dalam perusahaan yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasional, bersifat manajerial dan kegiatan strategi dari suatu organisasi

2.5.1 Pengertian Basis Data

Setelah mengetahui dasar pengertian dari sistem informasi, maka dalam merancang ataupun membangun sistem informasi maka diperlukan sekumpulan data yang disebut sebagai basis data. Basis data adalah suatu kumpulan data yang saling berhubungan antara satu dengan yang lainnya, yang tersimpan dalam perangkat keras komputer (*Hardware*) dan digunakan perangkat lunak (*Software*) untuk memanipulasinya. Penerapan *database* didalam sistem informasi disebut database sistem. Sistem dasar data adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuat tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi. **[Fat99]** mendefinisikan "*basis data merupakan data tertentu yang secara bersama-sama dapat digunakan untuk aplikasi-aplikasi yang berbeda."*

Menutut karangan **[Fat99],** basis data dilihat dari strukrut katanya, yaitu **basis** yang diartikan sebagai gudang atau tempat berkumpul dan **data** sebagai representasi fakta yang mewakili satu objek tertentu. Basis data

sendiri dapat didefinisikan dalam sejumlah sudut pandang sebagai berikut **[Fat99]** :

1. Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file* / *table* / arsip yang saling berhubungan yang disimpan didalam media penyimpanan elektronik.

[Fat99] Sebelum membentuk suatu database, data mempunyai jenjang mulai dari karakter-karakter, data *item* (*field*), *records* dan kemudian database.

1. Karakter

   Karakter merupakan bagian data yang terkecil, karakter dapat berupa angka, huruf, abjad atau simbol-simbol khusus yang membentuk suatu data item.

2. *Field*

   Sebuah *field* menggambarkan satu atribut dari *record* yang menunjukan suatu item dari data tertentu.

3. *Record*

   Record merupakan kumpulan dari *field* dan *Record* menggambarkan suatu unit dan individu yang tertentu.

4. *File*

   File terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis.

5. Database

   Terbentuk dari kumpulan *file-file.*

Menurut **[Fat99]** penggunaan basis data untuk mencapai sejumlah tujuan, seperti :

1. Kecepatan dan Kemudahan (*Speed*)

   Pemenfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan atau manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah.

2. Efisiensi Ruang Penyimpanan (*Space*)

   Karena keterkaitannya yang erat antar kelompok data dalam sebuah basis data, maka redudansi (perulangan) pasti selalu ada. Banyaknya redudansi ini pasti akan memperbesar ruang penyimpanan yang harus disediakan. Tetapi dalam basis data hal ini dapat diatasi, yaitu dengan memberikan sejumlah pengkodean atau dengan membuat relasi-relasi.

3. Keakuratan (*Accuracy*)

   Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan atau batasan (*constraint*) tipe data, keunikan data, dan sebagainya, yang secara ketat dapat diterpakan yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidak akuratan pemesukan atau penyimpanan.

4. Ketersediaan (*Availability*)

   Pertumbuhan data (baik dari jumlah atau jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu kita butuhkan atau gunakan. Karena itu kita dapat memilah adanya data utama/ master/ referensi, data transaksi,data histori hingga data kadaluarsa.

5. Kelengkapan (*Completeness*)

   Lengkap atau tidaknya data yang kita kelola dalam sebuah basis data bersifat relative (baik terhadap kebutuhan pemakai maupun terhadap waktu). Bila seseorang pemakai sudah menganggap bahwa data yang dipelihara sudah lengkap, maka pemakai lain belum tentu sependapat. Atau, yang sekarang dianggap lengkap, belum tentu dimasa akan dating akan demikian. Oleh karena itu untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya menambah recor-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam emlakukan penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel baru.

6. Keamanan (*Security*)

   Dalam sistem yang besar maka diperlukan keamanan yang ekstra ketat, hal ini ditujuakan untuk melindungi data-data yang dianggap penting, Dalam hal ini basis data dapat memberikan fasilitas keamanan, diantaranya dalam suatu sistem dapat ditetapkan siapa-siapa pemakai yang boleh

menggunakan basis data beserta objek-ojbjek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan.

7. Kebersamaan Pemakaian (*Sharability*)

   Dalam penggunaan datanya basis data dapat digunakan oleh lebih dari satu orang, oleh karena itu fungsi dari basis data adalah menjaga atau menghindari terhadap munculnya suatu persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak orang atau pemakai pada saat yang sama).

2.5.2 Pengertian Sistem Basis Data

Menurut **[Jog99]** Sistem Basis Data adalah

"*Suatu sistem informasi yang mengintegarasikan kumpulan dari data yang berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi*."

Komponen-komponen utama sebuah sistem basis data :

1. Perangkat Keras (*Hardware*)
2. Sistem Operasi (*Operating System*)
3. Basis Data (*Data Base*)
4. Sistem Pengolahan Basis Data
5. Pemakai (*User*)
6. Aplikasi (perangkat lunak) lain (bersifat opsional)

2.6 Pengertian Internet

Internet merupakan suatu alat komunikasi yang sangat penting pada saat ini karena tanpa *Inernet* segala informasi dan perkembangan tegnologi di dalam negeri maupun luar negeri akan lebih sulit didapat. *Internet* berasal dari kata *Interconection Network* yang berarti hubungan antar satu komputer dengan komputer lain di seluruh dunia yang membenuk suatu *jaringan* dengan melalui berbagai jalur komunikasi.

Sedangkan secara umum, *Internet* adalah sebuah kelompok besar dari berbagai sumber-sumber informasi yang meliputi seluruh dunia. Jadi *Internet* dapat diartikan sebagai *jaringan* komunikasi yang mencakup seluruh dunia yang menyadiakan seluruh sumber informasi yang dapat diakses seiao saat oleh semua pamakainya di seluruh dunia

Ada juga yang mengartikan *Internet* sebagai *jaringan* terbesar yang menghubungkan jutaan komputer yang tersebar di seluruh enjuru dunia da tidak terikat pada satu organisasipun

Fasilitas yang ada di internet dapat dijabarkan sebagai berikut :

1. *E-Mail* atau surat elektronik merupakan fasilitas yang paling sederhana dan paling banyak digunakan di internet.
2. *Mailing List* atau kelompok diskusi merupakan fasiltas yang dibangun dengan teknik yang sama dengan proses penyebaran surat elektronik. Dengan fasilitas ini, berita atau file di distribusikan kebanyak pengguna sekaligus, sehingga pengguna dapat melakukan diskusi di internet.

3. *File Transfer*, dipergunakan untuk menyalin file dari satu komputer di intenet ke komputer lainnya.
4. *WWW* atau *World Wide Web*, dipergunakan untuk memudahkan dalam melakukan navigasi informasi di internet.
5. *Remote Login*, dipergunakan untuk melakukan login ke komputer manapun asalkan mempunyai account, koneksinya dengan menggunakan telnet, rlogin atau program aplikasi lainnya.
6. *Gophe*r, merupakan aplikasi yang dapat mencari informasi yang ada di internet, tetapi hanya text base saja atau berbasis teks.
7. *Chat*, merupakan program aplikasi intenet yang memungkinkan kita berbicara secara langsung (*on-line*) dengan lawan bicara kita.

2.7 World Wide Web (WWW)

Sistem pengaksesan informasi dalam *internet* yang paling terkenal adalah *World Wide Web (WWW)* atau bias dikenal dengan istilah *Web*. Pertama kali diciptakan pada tahun 1991 di CERN, laboratorium Fisika Partikel Eropa, Jenewa, Swiss. Tujuan awalnya adalah untuk menciptakan media yang mudah untuk berbagi informasi diantara pada fisikawan dan ilmuwan.

Web menggunakan *protocol* yang disebut *HTTP*(*Hyper Text Transfer Protocol*) yang berjalan pada *TCP/IP*. Adapun dokumen *Web* ditulis dalam format *HTML* (*Hyper Text Markup Language*). Dokumen ini diletakkan dalam *Web Server* (*Server* yang melayani permintaan halaman *Web*) dan

diakses oleh klien (pengakses informasi) melaluiperangkat lunak yang disebut *Web Browser* atau sering disebut *Browser* saja.

Konsep *hypertext* yang dipergunakan pada *Web* sebenarnya telah dikembangkan jauh hari sebelum kehadiran *Web*. Dengan mengunakan *hypertext* , pemakai dapat melompat dari satu dokumen ke dokumen lain dengan mudah, dengan cukup meng-klik teks-teks khusus yang pada awalnya ditandai dengan garis bawah.

Penggunaan *hypertext* pada *Web* juga telah dikembangkan lebih jauh menuju ke *hypermedia* . Dengan mengunakan pendekatan *hypermedia* tidak hanya teks yang dapat dikaitkan, melainkan juga gambar, suara, dan bahkan video.

Informasi yang terdapat pada *Web* disebut hal *Web*(*Web page*). Untuk mengakses sebuah halaman *Web* dari *Browser,* pemakai perlu meyebutkan *URL* (*Uniform Resource Locator*). *URL* tersusun atas tiga bagian:

1. Format Transfer atau *Protokol*
2. Nama *Host*
3. *Path* berkas dokumen

2.8 Hypertext Markup Language (HTML)

HTML adalah kependekan dari (*Hypertext Markup Language*), merupakan sebuah bahasa scripting yang berguna untuk menuliskan halaman web. Pada halaman Web, HTML dijadikan sebagai Bahasa script dasar yang berjalan bersama berbagai bahasa Scripting Pemrograman lainnya.

Semua Tag-tag HTML bersifat dinamis, artinya kode HTML tidak hanyalah dijadikan sebagai file Executable program. Hal ini disebabkan HTML hanyalah sebuah Bahasa scripting yang dapat berjalan apabila dijadikan di dalam browser (pengakses web). Semua bahasa scripting yang berjalan di bawah Web dapat didukung oleh HTML, biasanya bahasa-bahasa tersebut melakukan *Embedded Script* pada tag-tag HTML.

2.9 HTTP dan URL

HTTP (*Hypertext Transfer Protocol*) adalah protokol level aplikasi untuk sistem informasi *hypermedia* tersebar. Protokol ini generik dan *stateleese* yang dapat dimanfaatkan banyak tugas selain utuk *hyperteks*. HTTP adalah bahasa komunikasi antara *web browser* dengan *web server.*

2.10 PHP

PHP (akronim dari PHP : *Hypertext prepocessor*) adalah merupakan bahasa *scripting* yang menyatu dengan HTML dan berada pada *server* (*server-sideembeded scripting*). program dari *server* akan dijalankan dari jarak jauh (*remote*) melalui *web browser* meskipun *platform* (SO, tipe mesin, struktur *jaringan*, dsb) *browser* dan *server*nya berbeda.

PHP berbeda dengan HTML karena dengan PHP kita tidak langsung menuliskan yang ingin dimuculkan di *browser,* tetapi kita meuliskan kode yang memerintahkan *server* untuk meproses dan menghasilkan halaman *web* yang diinginkan. *Server* memberikan memberikan haslnya berupa halaman *web,* maka pengunung akan melihat halaman tersebut seperti halaman HTML biasa.

Aplikasi PHP memanfaatkan *protocol* HTTP untuk komunikasi antar *server* dengan *client* HTTP menjelaskan tata cara suatu *client* meminta informasi dari *server* dan mengembalikan informasi yang diminta, termasuk mengatur apabila informasi tersebut didapat atau tidak didapat. Informasi pada *server* biasanya mengacu pada *file*. Jasdi meminta informasi dapat dikatakan meminta *server* untuk membuka (*meload*) dari suatu *file* dan kemudian menyerahkan isi *file* tersebut kepada *client*, *client* (*browser*) sering disebut (*agen*).

2.11 MySQL

MySQl bukan lingkungan pengembang aplikasi basis data, tetapi satu DBMS kecil yang kompak. MySQL cocok untuk aplikasi berbasis web keperluan minimal dan menengah. MySQL adlah DBMS kecil, kompak dan mudah digunakan. MySQl ideal untuk aplikasi berukuran kecildan menengah, namun telah menjanjikan untuk penggunaan besar. Implementasi MySQL berupa *client/server* terdiri dari *daemon* mysqld (terdapat pada sistem UNIX) dan beberapa program *client.*

MySQL sebagai database server memiliki konsep database modern, MySQL mempunyai banyak keistimewaan, diantaranya adalah sebagai berikut

1. Portability

   MySQL dapat berjalan dengan stabil pada berbagai sistem operasi yang ada pada saat ini seperti Microsoft Windows, Linux, FreeBSD, Mac OS X dan sistem operasi lainnya.

2. Open Source

   MySQL merupakan database *server* yang didistribusikan secara *open source*, sehingga dapat digunakan secara bebas dan gratis.

3. Multi-User

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini terjadi karena sebuah database *server* MySQL dapat diakses *client* secara bersamaan.

4. Performance Tuning

   MySQL mempunyai kecepatan yang cukup tinggi dalam menangani *query*, sehingga dapat memproses lebih banyak SQL persatuan waktu.

5. Column Types

   MySQL mempunyai tipe kolom yang sangat kompleks, seperti signed, unsigned integer, float, double, char, varchar, text, blob, date, time, time stamp, year, set dan enum.

6. Command dan Function

   MySQl mempunyai operator dan fungsi secara penuh yang mendukung perintah *select* dan *where data query*.

7. Security

   MySQL mempunyai lapisan-lapisan sekuritas seperti subnetmask, nama host dan izin akses user dengan system perizinan yang mendetail serta password terenkripsi.

8. Scalability dan Limits

   MySQL mempunyai kemampuan untuk menangani databse dengan skalabilitas besar dengan jumlah record lebih dari 50 juta dan 60 juta table serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tablenya.

9. Connectivity

   MySQL mempunyai kemampuan untuk melakukan koneksi dengan client menggunakan berbagai protokol, seperti TCP/IP, Unix Socet (UNIX) atau Named Pipes (NT).

10. Localisation

    MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari 20 bahasa.

11. Interface

    MySQL mempunyai interface terhadap berbagai aplikasi dan program degan menggunakan fungsi API (*Application Programming Interface*).

12. Client dan Tools

    MySQL dilengkapi dengan berbagai tool-tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk onlinenya.

13. Structur Table

    MySQL mempunyai struktur table yang lebih fleksibel dalam menganai ALTER TABLE, dibandingkan database lainnya seperti PostgreSQL ataupun Oracle.

2.12 Dreamweaver 8

Dreameaver adalah suatu web authoring tool yang berfungsi untuk mendesain halaman web. Fasilitas yang diberikan oleh dreameaver MX yaitu menggunakan berbagai macam bahasa pemrograman web seperti JSP, PHP, ASP, Cold Fusion dan juga bahasa program berbasis dotnet

2.13 Apache

Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini mengunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah.

Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.

2.14 Enterprise Architect

Enterprise Architect merupakan salah satu software yang paling banyak digunakan untuk melakukan design software melalui pendekatan Unified Proses. Enterprise Architect merupakan software yang menyediakan

banyak fungsi - fungsi seperti design proses, generate code, reverse engineering, serta banyak fungsi-fungsi yang lain.

BAB III

OBJEK PENELITIAN DAN METODE

3.1 Objek Penelitian

Objek Penelitian dilakukan di Toko Ai Read penyewaan komik, buku dan majalah.

3.1.1 Sejarah Singkat Ai Read Komik

Toko Ai Read didirikan pada tanggal 27 maret 2008 oleh Ari dan Rama. Toko ini terletak di Jl. Jalaprang No. 16 Sukaluyu Bandung berawal dari kesamaan hoby membaca kemudian mereka merencanakan untuk membuat toko bacaan. Untuk pertama kali mereka membuka toko bacaan, komik dan majalah berasal dari buku-buku koleksi mereki dikumpulkan, namun lama kelamaan dari hasil peminjaman buku, komik dan majalah yang disewakan, maka toko Ai Read sudah dapat membeli buku, komik, majalah sendiri

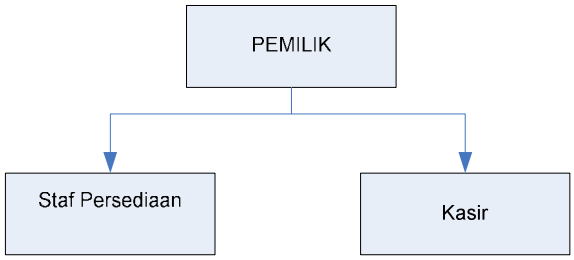
3.1.2 Visi dan Misi

Visi dan misi yang dimiliki oleh Toko penyewaaan buku, komik dan majalah Ai Read dalam menjalankan usahanya ialah :

1. Visi
    a. Menjadi taman bacaan yang lengkap bagi masyarakat
2. Misi
    a. Mendorong serta memberi motivasi, semangat dan meningkatkan minat baca masyarakat
    b. Membudayakan sikap masyarakat untuk gemar membaca

3.1.3 Struktur Organisasi

Struktur organisasi Toko Ai Read adalah sebagai berikut :



Gambrar 3.1 Struktur Organisasi Toko Ai Read

3.1.4 Deskripsi Jabatan (*Job Description*)

Job description atau deskripsi kerja merupakan serangkaian tugas yang dilakasanakan pada tiap-tiap bagian sesuai dengan wewenang yang diberikan. Deskripsi kerja di Toko Ai Read adalah sebagai berikut.

1. Pemilik orang yang memiliki atau penyandang dana serta pemegang saham serta juga mempunyai tanggung jawab untuk penerimaan pegawai, mengatur tata letak barang.
2. Persediaan (gudang) mempunyai wewenang mencatat dan melaporkan persedian buku kepada pmilik dan pembelian barang.
3. Kasir orang yang mempunyai tanggung jawab dalam mengelola laporan keuangan, transaksi peminjaman dan pengembalian buku, dan hal-hal administrasi lainnya.

3.2 Alat Bantu Pemodelan

Pemodelan adalah sebuah aktivitas yang telah berlangsung bertahun-tahun dalam pengembangan perangkat lunak. Sebuah model masih diperlukan untuk menulis sebuah aplikasi baik menggunakan bahasa pemograman yang paling

sederhana sampai bahasa yang kompleks. Pemodelan dapat berupa menggambar diagram alir (*flowchart*) yang menunjukkan langkat-langkah yang dikerjakan sebuah aplikasi. Pemodelan diperlukan untuk mendefinisikan sebuah model membuat mudah untuk memecah aplikasi yang komplek atau sebuah sistem yang besar menjadi sederhana, menjadi bagian-bagian diskrit yang mudah untuk dipelajari. Menurut **[SH02]** pemodelan harus memenuhi syarat berikut :

1. *Readability*

   Readability membawa kejelasan – kemudahan dalam memahami. Memahami sistem adalah langkah pertama baik dalam membangun atau mengembangkan sebuah sistem yang telah ada. Hal ini melibatkan pengetahuan tentang bagaimana sistem dibuat, bagaimana sistem berperilaku dan sebagainya. Memodelkan sebuah sistem membuat sistem mudah dibaca dan yang paling penting, mudah dalam pendokumentasian. Penggambaran sebuah sistem agar mudah dibaca melibatkan pemahaman struktur sistem dan tingkah laku sistem.

2. *Reusability*

   Reusability adalah efek dari membuat sebuah sistem mudah dibaca. Bila sebuah sistem telah dimodelkan sedemikian rupa sehingga mudah dipahami, karena para pengembang cenderung untuk mencari kesamaan atau pengulangan, baik dalam hal fungsionalitas, fitur ataupun struktur.

3.2.1 *Unified Modeling Language (UML)*

Ada beberapa difinisi dari *Unified Modeling Language* (UML) yang pada dasarnya memiliki maksud yang sama, antara lain :

[DR03] mendefinisikan:

"*Unified Modeling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan system piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system."

[SH02] mendefinisikan:

"*Unified Modeling Language* (UML) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifacts* dari sistem *software,* untuk memodelkan bisnis, dan sistem non *software* lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks."

Kemudian **[Bah99]** juga mengemukakan :

"The unified modeling language is a language for specifying, constructing, visualizing, and documenting the software system and its components. A UML is a graphical language with sets of rules and semantics. The rules and semantics are expressed in English, in a form known as object constraint language (OCL)."

UML mengintegrasikan konsep dari *booch,* OMT, OOSE dan juga *Class-Relation* dengan menggabungkan mereka menjadi suatu keasatuan bahasa modeling yang bisa berguna bagi siapa saja. UML bertujuan untuk menjadi standar bahasa modeling yang mampu untuk memodelkan system yang konkuren dan juga terdistribusi. UML sudah menjadi standar industri yang dibuat dibawah pengawasan *Object Management Group* (OMG). OMG ditugaskan untuk memberi informasi mengenai *object-oriented methodology* yang mampu untuk membuat bahasa untuk *modeling software*. Dengan menggunakan UML para

pengembang dapat membuat 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 dia lebih cocok untuk penulisan piranti lunak dalam bahasa pemrograman berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk *modeling* aplikasi procedural dalam VisualBasic atau C.

UML bukan merupakan metode dan juga tidak berusaha untuk menjadi suatu metode, namun demikian ia merupakan kebutuhan untuk *Rational Unified Process* yang telah dibuat oleh *Rational Software Corporation*. Metode *development* lainnya seperti *Dynamic Systems Development Method* juga menggunakan teknik dari UML.

UML menyediakan kemampuan untuk menangkap karakteristik sebuah sistem dengan menggunakan notasi-notasi tertentu. UML menyediakan sederetan notasi sederhana yang mudah dipahami untuk mendokumentasikan sistem berdasarkan prinsip-prinsip perancangan berorientasi objek. UML sudah menjadi *standardized modelling language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas seperti:

1. Spesifikasi
2. Visualisasi
3. Desain Arsitektur
4. Konstruksi

5. Simulasi dan testing
6. Dokumentasi
7. Sketsa

UML dikembangkan sebagai ide dasar untuk mempromosikan hubungan dan produktifitas antara para pengembang dari *object-oriented system*. Menurut **[Lar04]** ada tiga cara dalam memakai **UML** dalam melakukan pemodelan sistem:

1. **UML** sebagai sketsa

   UML digambarkan dalam sketsa coretan-coretan dalam kertas atau *whitboard* secara tidak formal. Biasanya digunakan dalam sesi diskusi tim untuk membahas aspek tertentu dalam tahap analisis dan perancangan

2. **UML** sebagai *blueprint* system

   Seperti diagram kelistrikan adalah blueprint dari komponen atau produk yang akan dihasilkan, **UML** juga bisa menggambarkan blueprint yang identik untuk sebuah sistem software.

3. **UML** sebagai bahasa pemrograman

   UML berfungsi sebagai bahasa pemrograman mencoba melakukan semuanya dengan **UML** sampai kepada produk jadinya. Analisis dan perancangan dilakukan dengan diagram-diagram yang ada dalam **UML**, sementara sebuah tool atau generator bisa menghasilkan produk akhir dari diagram-diagram ini.

Saat ini **UML** paling banyak digunakan dengan cara pertama dan kedua. Khusus dalam metode *agile* (cepat dan ringan), **UML** digunakan dengan cara pertama.

3.2.1.1 Konsep Dasar UML

Teknik dasar dalam membuat model dari sebuah ide atau dunia nyata adalah dengan menggunakan *abstraksi*. Setiap diagram UML yang digambar memiliki keterkaitan dengan dunia nyata. *Abstraksi* dikembangkan sebagai ketentuan untuk dipelajari dan sering digunakan. Analogi yang lebih mendekati adalah merupakan kumpulan dari sketsa atau *blueprint* yang menampilkan detail yang cukup dari suatu bangunan untuk memastikan tentang apa sebenarnya bangunan tersebut. *Abstraksi* model dan diagram juga berguna karena akan menjelaskan lebih rinci detail-detail yang dibutuhkan.

3.3 Model-Model Diagram UML

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. Pada saat ini UML telah mencapai versi 2.1 memiliki banyak perubahan dari versi sebelumnya seperti diagram kolobarasi diganti dengan diagram komunikasi dan terdapat tiga tambahan diagram. Diagram-diagram yang resmi tersedia ada delapan diagram yang diklasifikasikan berdasarkan tiga jenis diagram sebagai berikut.

Tabel 3.1
Klasifikasi Jenis Diagram UML

Structural Diagram	1. Class diagram 2. Component diagram 3. Deployment diagram
Behavioral Diagram	1. Use case diagram 2. Activity diagram 3. Statechart diagram
Interaction Diagram	1. Sequence diagram 2. Collaboration diagram

(Sumber: *Martin Fowler. UML Distilled*)

3.4.1 Diagram *Use Case*

Diagram *Use Case* adalah deskripsi fungsi dari sebuah sistem dari perspektif/sudut pandang para pengguna sistem. *Use Case* mendifinisikan "apa" yang dilakukan oleh sistem dan elemen-elemennya, bukan "bagaimana" sistem dan elemen-elemennya saling berinteraksi. *Use Case* bekerja dengan menggunakan *"scenario"*, yaitu deskripsi urutan-urutan langkah yang menerangkan apa yang dilakukan penggunaan terhadap sistem maupun sebaliknya. Diagram u*se case* mengidentifikasikan fungsionalitas yang dipunyai oleh sistem (*use case*), user yang berinteraksi dengan sistem (*actor*) dan asosiasi/keterhubungan antara user dengan fungsionalitas sistem. Komponen notasi dasar yang dipunyai oleh diagram *use case* adalah *actor*, *use-case*, dan *association*. Berikut adalah notasi yang terdapat pada diagram *use case*. Diagram *use case* memiliki sebuah model khusus yang terbatas untuk kondisi tertentu yang disebut *stereotype.* Untuk menunjukan *stereotype* digunakan symbol "<<" diawalnya dan ditutup dengan ">>" diakhirnya. Terdapat dua *stereotype* paling sering digunakan dalam *use case diagram* yaitu <<extend>> dan <<include>>. <<extend>> digunakan untuk menunjukan bahwa satu *use case* merupakan

tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu dipenuhi. Sedangkan <<include>> digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.

3.4.2 Diagram *Activity*

Diagram *activity* digunakan untuk mendokumentasikan alur kerja pada sebuah sistem, yang dimulai dari pandangan *business level* hingga ke *operational level.* Pada dasarnya, diagram *activity* merupakan variasi dari diagram *state machine*. Diagram *activity* mempunyai peran seperti halnua *flowchart,* akan tetapi perbedaannya dengan *flowchar* adalah diagram *activity* bisa mendukung perilaku *parallel* sedangkan *flowchar* tidak bisa.

3.4.3 Diagram Sequence

Diagram ini menunjukan sejumlah objek dan *message* (pesan) yang diletakan diantara objek-objek didalam *use case.* Perlu diingat bahwa di dalam diagram ini, kelas-kelas dan actor-aktor diletakan dibagian atas diagram dengan urutan dari kiri ke kanan dengan garis *lifeline* yang diletakan secara vertical terhadap kelas dan actor.

3.4.4 Diagram *Class*

Diagram *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/property) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan *object* beserta hubungan satu sama lain seperti *containment,* pewarisan, asosiasi, dan lain-lain. Sebuah *Class* memiliki tiga area pokok yaitu :

1. Nama, merupakan nama dari sebuah kelas
2. Atribut, merupakan peroperti dari sebuah kelas. Atribut melambangkan batas nilai yang mungkin ada pada obyek dari class
3. Operasi, adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau yang dapat dilakukan oleh *class* lain terhadap sebuah *class*.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.
4. *Package*, hanya dapat dipanggil oleh instance sebuah class pada paket yang sama.

Class juga dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *runtime*.

3.4.4.1 Relasi Pada Diagram *Class*

Relasi atau *relationship* merupakan keterhubungan antar kelas yang muncul pada saat sebuah kelas berinteraksi dengan kelas-kelas lainnya. Didalam *class diagram*, setiap kelas pasti akan berinteraksi dengan baik satu maupun lebih

kelas. Relasi yang muncul pada setiap keterhubungan antar kelas juga akan memiliki atribut-atribut yang akan lebih menjelaskan definisi dari sebuah relasi yang terjadi. Relasi yang terdapat pada diagram *Class* :

1. Asosiasi *(Association)*

   Para pemodel menggunakan pemahaman asosiasi adalah pada saat beberapa kelas saling terhubung satu sama lain secara konseptual. Asosiasi juga dapat menjadi lebih kompleks pada saat beberapa class terhubung ke satu class.

2. *Constraint* Pada Asosiasi

   Kadangkala sebuah asosiasi diantara dua *class* harus mengikuti sebuah aturan dan aturan ini bisa diletakkan dalam sebuah *constraint* pada garis asosiasi dan diletakkan dalam kurung kurawal. Bentuk lain dari tipe *constraint* adalah relasi OR yang ditulis dengan {or} dalam garis putusputus yang menghubungkan dua garis asosiasi. Kondisi OR ini menghadapkan kepada keadaan bahwa sebuah kelas terhubung dengan dua kelas tetapi pada saat mendefinisikan relasinya kelas harus memilih salah satu dari kedua kelas tersebut, sedangkan kondisi untuk memilih keduanya adalah *invalid*.

3. Kelas Asosiasi (*Associations Class*)

   Sebuah asosiasi dapat memiliki atribut dan operasi seperti halnya sebuah *class*. Sebuah *association class* sebenarnya diperlukan apabila salah satu dari kelas yang terhubung mempunyai sebuah atau beberapa atribut yang tidak layak dimiliki oleh kelas tersebut, karena secara logis atribut tersebut

lebih layak dimiliki oleh asosiasi yang menghubungkan kedua kelas tersebut.

4. Multiplisitas (*Multiplicity*)

   Multiplisitas atau kardinalitas adalah jumlah banyaknya obyek sebuah *class* yang berelasi dengan sebuah obyek lain pada *class* lain yang berasosiasi dengan *class* tersebut. Untuk menyatakan multiplisitas anda dapat meletakkannya diatas garis asosiasi berdekatan dengan *class* yang sesuai. Ada banyak multiplisitas yang mungkin untuk dipakai. Tabel berikut menjabarkan multiplisitas yang dapat digunakan.

Tabel 3.2
Nilai Multiplisitas Potensial

Indicator	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
1..*	One to more
N	Only n (where n > 1)
0..n	Zero to n (where n > 1)
1..n	One to n (where n > 1)

(Sumber : Modul Workshop UML Bab 2)

3.4.4.2 *Generalization & Inherintance*

Sebuah *class* (*child class* atau *subclass*) dapat mewarisi atribut-atribut dan operasioperasi dari *class* lainnya (*parent class* atau *super class*) dimana *parent class* bersifat lebih umum daripada *child class*.

Generalisasi pada konsep *Object Oriented* digunakan untuk menjelaskan hubungan kesamaan diantara *class*. Dengan menggunakan generalisasi bisa dibangun struktur logis yang bisa menampilkan derajat kesamaan atau perbedaan

diantara *class-class*. Manfaat lain dari struktur hirarkis juga memungkinkan untuk penambahan *subclass* (*child class*) baru tanpa harus merubah struktur yang sudah ada.

Aturan inheritance dapat secara umum bisa diklasifikasikan sebagai berikut :

1. *Subclass* selalu mewarisi semua sifat dari *superclass*-nya.
2. Definisi *subclass* selalu mencakup paling tidak satu detil yang tidak diturunkan dari *superclass*-nya.

Inheritance sangat dekat dengan asosiasinya dengan generalisasi. Generalisasi menjelaskan hubungan logis antar elemen-elemen yang mempunyai karakteristik yang sama. Sedangkan *inheritance* menerangkan mekanisme agar bagi pakai (*sharing*) bisa terjadi.

3.4.4.3 Kelas Ketergantungan (*Dependant Class)*

Pada penggunaan relasi kadangkala satu *class* menggunakan *class* yang lain, hal ini disebut *dependency*. Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. Notasi untuk *dependency* pada UML dapat menggunakan garis putus-putus dan tanda panah pada ujungnya.

3.4.4.4 Agregasi

Didalam UML, ada relasi dengan perlakukan khusus yang disebut dengan "bagian dari (*part of*)" yang menangani antar obyek-obyek dimana salah satunya adalah bagian dari yang lain. Dengan kata lain sebuah obyek terdiri atas obyek-obyek yang lain. Hal seperti ini di dalam UML disebut agregasi. Sebuah agregasi

adalah kasus khusus dari asosiasi. Agregasi disimbolkan dengan jajaran genjang yang diletakkan pada *class* yang mengandung objek. Kadangkala relasi OR harus digunakan pada agregasi. Untuk memodelkan hal tersebut, sebuah constraint dengan kalimat atau dalam tanda kurung kurawal pada garis putus-putus harus dibuat untuk menghubungkan dua objek.

Komposit (*composite*) adalah sebuah tipe agregasi yang kuat dimana bagian dari obyek tergantung penuh/secara keseluruhan terhadap objeknya sehingga bila sebuah obyek komposit dibuang maka bagian yang tergantung pada komponen tersebut akan terbuang juga pada saat yang bersamaan. Notasi komposit sama seperti agregasi hanya saja jajaran genjangnya terisi (solid).

3.4.4.5 *Interface* dan Realisi

Interface adalah satu set *operation* yang memberikan spesifikasi beberapa aspek dari perilaku dan operasi disuatu *class* ke *class* yang lain. Contohnya, *keyboard* pada komputer sebenarnya merupakan *interface* yang bisa dipakai ulang karena tombol-tombol *keyboard* sebenarnya berasal dari mesin ketik, hanya saja mungkin ada beberapa operasionalisasi tombol-tombol yang berbeda yang sudah ditransfer ke sistem yang lain. Pemodelan *interface* sama dengan pemodelan *class* hanya saja pada *interface* tidak mempunya atribut dan pada penamaannya perlu ditambahkan *stereotype* <<interface>> atau ditambahkan huruf "I" di depan nama interface. Relasi antara *class* dan *interface* disebut *Realization*. Realisasi dituliskan dengan garis putus-putus dengan segitiga yang mengarah ke *interface*.

3.4.4.6 Diagram *Component*

Komponen perangkat lunak adalah bagian fisik dari sebuah sistem yang menetap di komputer. komponen merupakan implementasi software dari sebuah class. Komponen bisa berupa tabel, *file* data, *file* exe, file DLL, dokumen dan lain-lain.

Component diagram mengandung komponen, *interface* dan *relationship*. Komponen diagram ini digunakan pada saat pengembang ingin memecah sistem menjadi komponen-komponen dan ingin menampilkan hubungan-hubungan mereka dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih rendah. Secara umum dapat disimpulkan bahwa component diagram yang digunakan untuk menjelaskan kebergantungan antar beragam komponen-komponen *software* seperti misalnya kebergantungan antara *file-file executable* dengan *file-file* sumbernya (*source file*) dll.

3.4.4.7 Diagram *Deployment*

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment diagram* dapat digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem.

3.5 Perangkat Lunak Pendukung Pemodelan UML

Hasil-hasil pemodelan yang dikomunikasikan dengan pelanggan agar lebih baik dan lebih professional jika diformat dengan baik menggunakan software yang bisa menyajikan gambar-gambar hasil pemodelan dengan rapi.

Kegunaannya bernilai tambah lebih tinggi lagi, terutama untuk tim pengembang, apabila software tersebut memiliki fungsi *auto-generate* dari model ke source code, dan fungsi *reverse engineering* dari *source code* menjadi sebuah model. Sehingga hasil kerja programmer yang dituangkan dalam source code bahasa pemrograman tertentu bisa divisualisasikan menggunakan model **UML** hasil *reverse engineer* dari software tersebut.

Nilai tambah lebih tinggi juga bisa diberikan oleh sebuah **CASE tool** apabila menyediakan integrasi dengan *Integrated Development Environment* (IDE) dari sebuah bahasa pemrograman seperti Microsoft Visual Studio (.NET) dan Eclipe (Java). IDE tertentu biasanya mempunyai kelebihan yang telah dioptimisasi untuk bahasa pemrograman dan platform tertentu, sehingga lebih menjadi pilihan *programmer* daripada membuat source code-nya langsung di sebuah **CASE tool**. Dengan adanya integrasi antara **CASE tool** dengan sebuah IDE, proses *auto generate* dan *reverse engineering* tentu bisa dilakukan dengan lebih mudah dan lebih cepat.

Salah satu **CASE tool** yang cukup lengkap untuk melakukan pemodelan dengan **UML** dan memiliki dua fungsi di atas adalah ***Enterprise Architect*** dari *Sparx System*. Gambar berikut menunjukkan tampilan *software* tersebut.

Hampir seluruh jenis diagram pemodelan **UML 2.0** terdapat dalam ***Enterprise Architect***. Layout *software* juga mirip dengan layout dari sebuah IDE, dimana *toolbox* ditempatkan di sebelah kiri, *project explorer* di sebelah kanan, dan content ditempatkan di tengah. Kemiripan layout ini tentunya memudahkan pengguna terutama programmer untuk cepat familiar dengan *software* ini.

Kelebihan-kelebihan software ini di antaranya adalah: sangat lengkap, fleksibilitas penggunaan, transformasi dari satu model ke model lain, korelasi satu model dengan model lain, *auto generate* model ke dalam dua platform yang saat ini sedang terkenal: .NET dan Java, *reverse engineering* dari source code ke model, dan fitur-fitur pendukung lain seperti auto generate RTF dan HTML report, serta transformasi dari sebuah model menjadi image: JPG, GIF, atau BMP.

Sedangkan kekurangannya adalah perintah-perintah atau pilihan-pilihan yang terlalu kompleks dan tidak mudah untuk segera dipahami. Untuk bisa memahami seluruh notasi dan istilah-istilah yang dipakai, terlebih dahulu harus mengetahui konsep dasar **UML**. Kekurangan kedua adalah tidak mudah untuk menemukan perintah atau shortcut untuk melakukan sesuatu yang mudah, misalnya memformat seluruh garis penghubung agar memiliki penampakan label yang sesuai keinginan.

Sesuai dengan prinsip *agile*, penulis tidak perlu menggunakan seluruh model yang ada dalam **UML** dalam melakukan pemodelan dengan software ini. Fitur-fitur yang cukup banyak dan menarik dipakai dari pandangan seorang *software architect* adalah *auto generate* dari sebuah *class diagram* menjadi *language specific diagram* dan source code. Bagi bisnis analis, apalagi yang telah

terbiasa menggunakan Microsoft Visio, menggunakan software ini akan terasa sulit di awal. Butuh waktu untuk merasa familiar dengan software ini.

3.6 Batasan dan Asumsi

Pembatasan masalah ini dilakukan agar dalam pengolahan sistem dapat lebih terarah dari tujuan yang telah ditetapkan. Berdasarkan permasalahan yang timbul, penyusun membatasi beberapa permasalahan sebagai berikut :

1. Prosedur pencatatan transaksi peminjaman dan pengembalian buku yang terjadi Ai Read
2. Program aplikasi yang dibangun menggunakan php dengan tujuan :
    a. Sebagai media promosi katalog buku, komik dan majalah.
    b. Peminjaman dan pengembalian buku serta pendaftaran member tidak dapat dilakukan secara online.
3. Aplikasi yang dibuat hanya menguraikan
        a. Pengolahan data sebagai admin yang terdiri dari :
            1) penginputan data buku, komik dan majalah
            2) penginputan data member
            3) penginputan data berita
            4) pencatatan peminjaman buku dan pengembalian buku
        b. Sebagai user hanya dapat menggunakan fasilitas informasi yang disediakan dalam web, seperti fasilitas informasi katalog buku dan komik, kecuali untuk fasilitas history member hanya dapat diakses oleh member.

3.7 Metode Pengumpulan Data

Metode yang digunakan dalam pelaksanaan pencarian dan pengumpulan data untuk membantu serta memudahkan penulis melakukan penelitian tugas akhir penulis menggunakan data primer dan data sekunder.

3.7.1 Sumber Data Primer

Data primer merupakan data yang diambil secara langsung, data ini diperoleh dari kegiatan observasi yaitu pengamatan langsung pada objek penelitian dan mengadakan wawancara dengan pihak yang terlibat.

3.7.1.1 Observasi

Yaitu penulis terjun langsung ke lapangan atau tempat penelitian guna melihat dan meneliti secara langsung sistem yang sedang berjalan di Ai Read Komik.

3.7.1.2 Wawancara

Yaitu suatu teknik mengumpulkan data dengan melakukan tanya jawab atas pertanyaan yang berkaitan dengan bahan yang dibutuhkan dalam penelitian. Interview dilakukan secara langsung pada pihak yang bersangkutan.

3.7.1.3 Studi Kepustakaan

Yaitu membaca dan mempelajari buku-buku serta artikel-artikel yang berhubungan dengan tema skripsi.

3.7.2 Sumber Data Sekunder (Dokumentasi)

Data sekunder merupakan data yang diperoleh secara tidak langsung yaitu berupa referensi dari buku, makalah atau dari beberapa sumber di internet yang berhubungan dengan tema penelitian yang sedang dilakukan oleh penulis.

BAB IV

ANALISIS SISTEM

4.1 Analisi Sistem Yang Berjalan

Pada analisis sistem yang berjalan ini menjelaskan sistem yang sedang berjalan pada tempat penelitian.

4.1.1 Gambaran Umum Ai Read Komik

Toko Ai Read dalam usaha mempromosikan produk-produknya kepada masyarakat masih menggunakan media brosur dan pamflet sehingga dalam mempromosikan produk dan penyebaran informasi yang berkaitan dengan Ai Read masih kurang cepat diterima oleh masyarakat luas, dan juga dalam proses peminjaman dan pengembalian buku masih menggunakan microrosoft acces sehingga apabila jumlah data semakin banyak maka aplikasi menjadi kurang cepat atau sering terjadi error, hal ini sangat mempengaruhi kenyamanan pelanggan.

Berdasarkan urain diatas maka penulis dapat menyimpulkan bahwa dalam usaha mempromosikan produk masih kurang maksimal dan juga dalam proses peminjaman dan pengembalian buku masih menggunakan aplikasi yang sederhana, sehingga hal ini dapat menimnulkan beberapa masalah yang berdampak kurang baik kepada Ai Read diantaranya :

a. Perkembangan usaha bisnis Ai Read menjadi kurang maksimal dikarenakan penyebaran informasi menggunakan media yang kurang cepat diterima oleh masyarakat
b. Banyaknya data-data peminjaman dan pengemabalian yang hilang dikarenakan aplikasi masih sederhana dan sering terjadi error serta

untuk penyimpanan data masih dimasukan dalam ruang penyimpanan yang kurang tepat

4.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan oleh sistem adalah perangkat lunak yang dapat digunakan oleh pengguna dimanapun dan kapanpun dengan asumsi bahwa pengguna merupakan pihak luar dari sistem yang membutuhkan kecepatan dalam informasi yaitu informasi pengolahan data katalog buku. Solusi yang ditawarkan adalah perangkat lunak berbasis *online* atau *internet*, dimana perangkat lunak yang berbasis *online* atau *internet* menawarkan kelebihan yaitu pengguna dapat mengakses dengan cepat dimana saja, kapan saja asalkan sebuah komputer harus memiliki fasilitas *browser* dan terkoneksi dengan jaringan internet. Secara umum perangkat lunak yang akan dikembangkan memiliki kemampuan :

1. Menginformasikan katalog buku dan komik secara online
2. Menangani proses peminjaman dan pengembalian buku namun tidak secara online.
3. Memfasilitasi untuk dapat memodifikasi katalog buku-buku dan komik.
4. Meng*upload* informasi / berita yang akan di informasikan kepada user / pengguna.
5. Menghapus / memodifikasi informasi yang di informasikan.
6. Memberikan fasilitas kepada user untuk melihat informasi yang berkaitan.
7. Memberikan fasilitas bagi member untuk melihat history member.

4.1.3 Kebutuhan Perangkat Lunak

Pada bagian ini akan dibahas kebutuhan perangkat lunak secara rinci yaitu meliputi lingkungan pengembangan dan operasi, kebutuhan pengguna, dan kebutuhan fungsional perangkat lunak.

4.1.4 Kebutuhan Pengguna

Pengguna perangkat lunak dibagi menjadi dua kategori, yaitu member dan bagian khusus Administrator. Berikut ini adalah asumsi terhadap pengguna perangkat lunak yang akan dikembangkan :

1. Data pengguna admin belum ada pada sistem.
2. Semua pengguna harus memiliki kemampuan dasar menggunakan komputer dan aplikasi berbasis web, khusus user atau member memahami alur sistem yang ada pada aplikasi ini.

a. Pengguna Member

Pengguna member memiliki kemampuan sebagai berikut :

Tabel 4.1
Kemampuan Pendaftar Member

No	Kemampuan	Keterangan
1	Login	Masuk kedalam sistem
2	Melihat History Member	
3	Melihat informasi member	
4	Melihat informasi katalog buku dan komik	
5	Logout	Keluar dari sistem

b. Administrator

Administrator memiliki kemampuan sebagai berikut

Tabel 4.2
Kemampuan Administrator

No	Kemampuan	Keterangan
1	Login	Masuk Kedalam Sistem
2	Memasukan informasi katalog buku dan komik	
3	Melihat data informasi katalog buku dan komik	
4	Mengubah dan menghapus data informasi katalog buku dan komik	
5	Melihat data member	
6	Merubah dan menghapus data member	
7	Menginputkan data peminjaman buku	
8	Mengecek data pengembalian buku	
9	Memasukan berita	
10	Melihat data berita	
11	Mengubah dan menghapus berita	
12	Logout	Keluar Dari Sistem

4.1.5 Kebutuhan Fungsional

Kebutuhan fungsional perangkat lunak merupakan penjabaran dari spesifikasi kebutuhan perangkat lunak menjadi fungsi - fungsi perangkat lunak. Perincian kebutuhan fungsional perangkat lunak Ai Read Komik Online yaitu :

a. Menangani Info Katalog Buku dan Komik

1. Menampilkan form pencarian katalog buku dan komik.
2. Melakukan proses pencarian katalog buku dan komik.

b. Menyimpan atau Merubah Info Katalog Buku dan Komik

1. Menampilkan form login.
2. Menampilkan form untuk menginputkan data info katalog buku dan komik.
3. Menampilkan form untuk merubah dan menghapus data info katalog buku dan komik, menginput info katalog buku dan komik.

4. Melakukan proses perubahan data info katalog buku dan komik.

c. Memberikan Fasilitas Melihat dan Merubah Data Pengguna

1. Menampilkan data Member
2. Menampilkan form untuk merubah data Member
3. Melakukan proses penyimpanan atau perubahan data

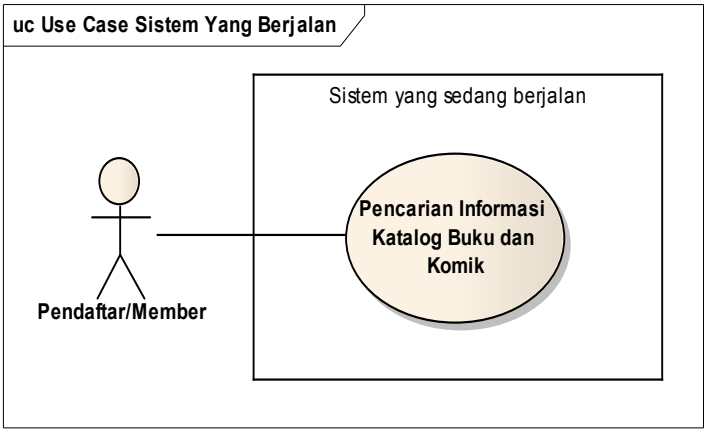
d. Menangani Proses Peminjaman dan Pengembalian Buku

1. Mencatat data peminjam berdasar id member
2. Menyimpan kedalam database
3. Mencatat data pengembalian berdasar id member dan id buku

4.1.6 Model *Use Case Diagram*

Permodelan ini dimaksudkan untuk menggambarkan kegiatan - kegiatan dan hubungan yang terjadi antara para aktor dan use case didalam sistem yang sedang berjalan. Kegiatan yang dilakukan dan hubungan antara para aktor dalam sistem yang sedang berjalan digambarkan menggunakan *use case* diagram sebagai berikut.

1. *Use Case* Sistem yang sedang berjalan



Gambar 4.1 *Use Case* Sistem yang berjalan

2. Skenario *Use Case*

Use case sekenario adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem berikut ini adalah sekenario *use case* yang ada.

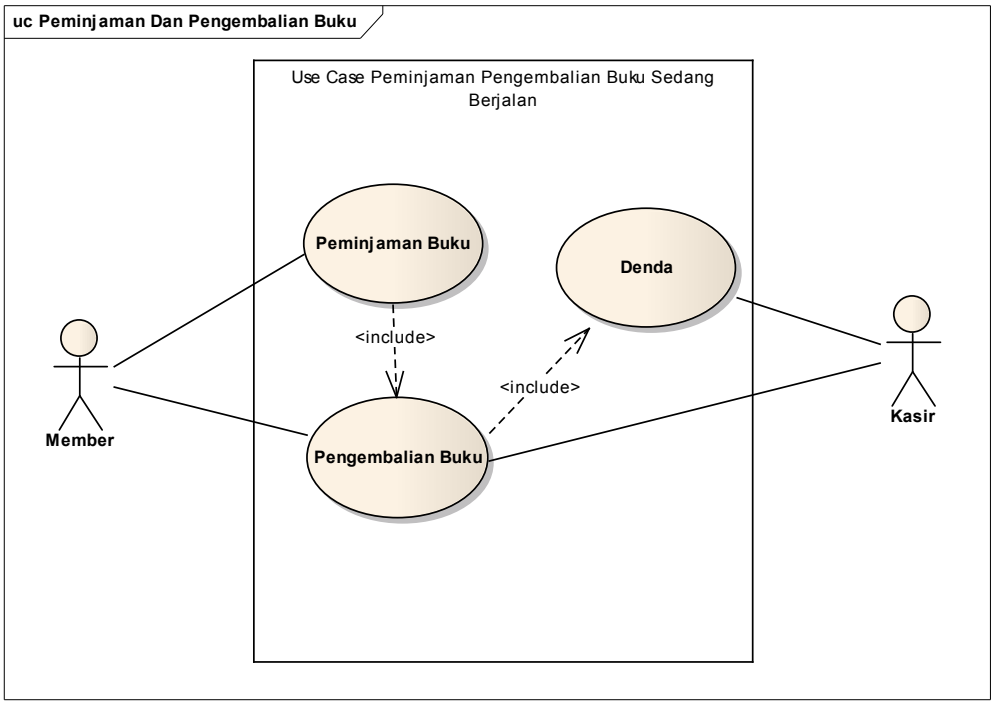
a. Skenario Informasi Katalog Buku dan Komik

Tabel 4.3
Skenario Informasi Katalog Buku dan Komik yang Sedang Berjalan

Use Case Sekenario	
Nama *use case*	Informasi Katalog Buku dan Komik
Aktor utama	User/Member
Tujuan	Mengetahui informasi seputar buku dan komilk terbaru
Type	Primary
Deskripsi	Datang langsung kebagian informasi sekaligus pendaftaran Ai Read dan menanyakan langsung kepada petugas yang ada

Actor	Sistem
1.Menanyakan informasi buku dan komik 3.Menerima informasi buku dan komik	2.Memberikan informasi tentang buku dan komik

3. Use Case Peminjaman dan Pengembalian Buku Berjalan



Gambar 4.2 *Use Case* Peminjaman dan Pengembalian Buku Sedang Berjalan

4.1.7 Evaluasi Sistem

Sistem yang sedang berjalan menunjukan beberapa kelemahan dimana kelemahan pada proses yang terjadi yang telah digambarkan adalah sebagai berikut.

1. Tidak adanya proses pemesanan peminjaman buku.
2. Belum adanya media untuk memperkenalkan katalog buku,komik dan majalah kepada masyarakat.
3. Belum ada fasilitas untuk berinteraksi antara user dengan Ai Read

4.2. Perancangan Sistem

Tujuan dari desain sistem secara umum adalah untuk memberikan gambaran secara umum kepada user tentang sistem yang baru.

4.2.1 Rancangan Skenario

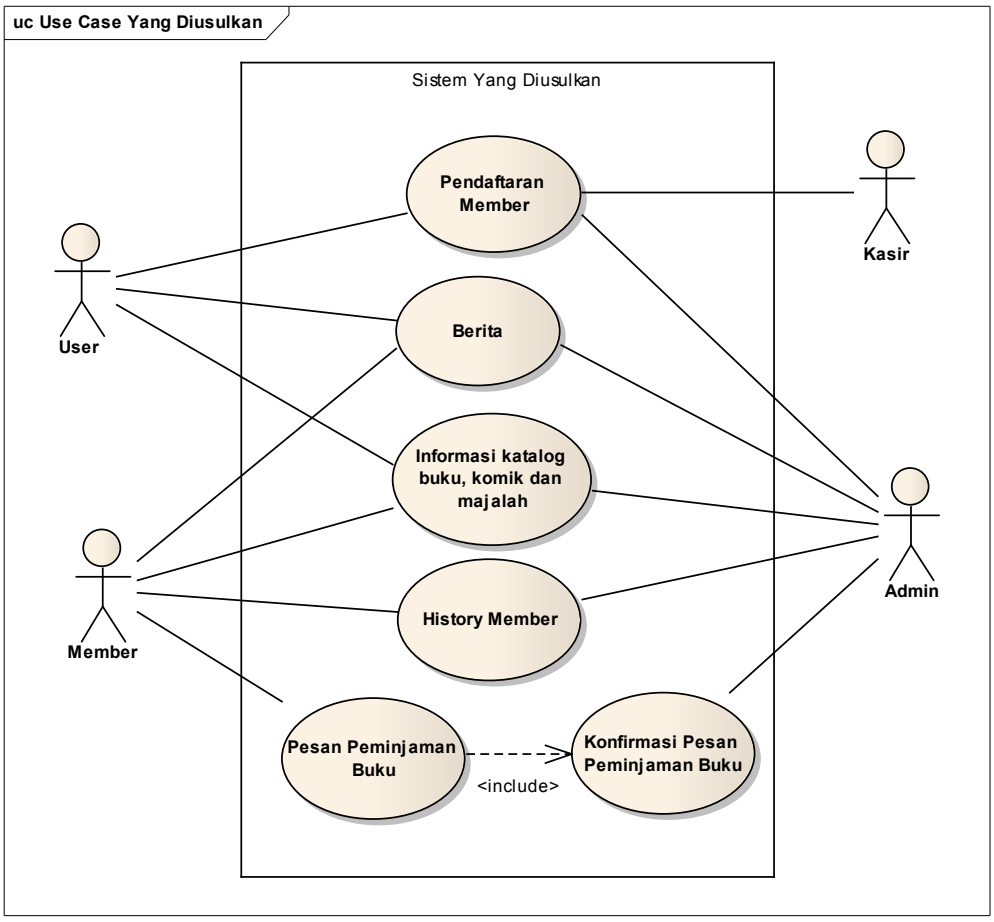
Skenario *use case* digunakan untuk memudahkan dalam menganalisis skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap scenario tersebut.

4.2.1.1 Skenario Yang Diusulkan

Tabel 4.4
Skenario Yang Diusulkan

Use Case Sekenario	
Nama *use case*	Sistem Yang Diusulkan
Aktor utama	Member
Tujuan	-
Type	Primary
Deskripsi	User dapat melakukan pendaftaran anggota, sebelum user menjadi member maka user hanya bisa mengakses fasilitas-fasilitas tertentu saja yaitu fasilitas berita dan informasi katalog buku saja. Sedangkan member mendapatkan fasilitas seperti berita, katalog buku, history, pesan peminjaman buku

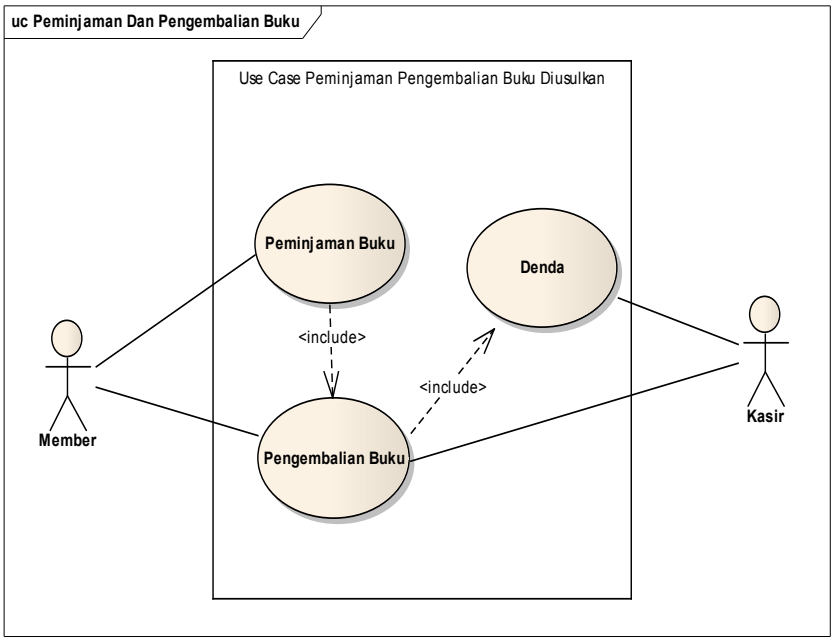
4.2.1.2 Diagram Use Case Yang Diusulkan



Gambar 4.3 (Diagram Use case Yang Diusulkan)

4.2.1.3 Diagram Use Case Peminjaman dan Pengembalian Buku Diusulkan

Pada gambar 4.4 berikut diperlihatkan *use case* peminjaman dan pengembalian buku yang diusulkan.



Gambar 4.4 (Diagram Use case Peminjaman dan Pengembalian Buku Diusulkan)

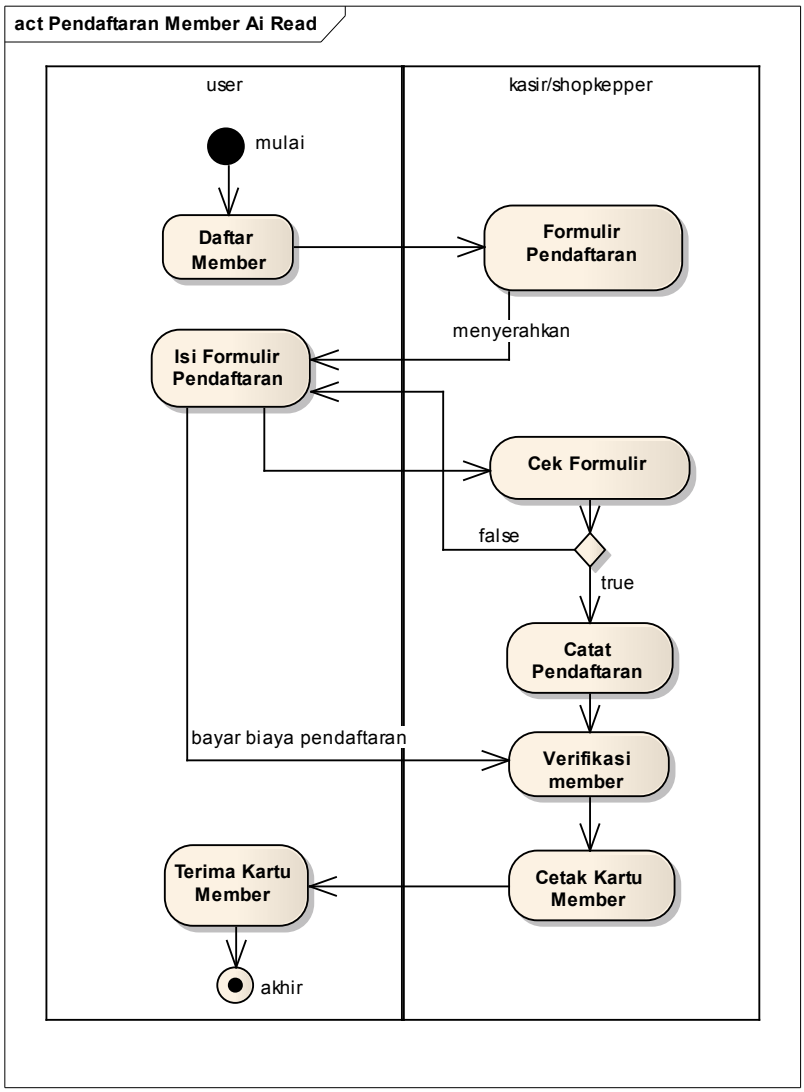
4.3 Activity Diagram

Jika Use case diagram merupakan gambaran mengenai proses apa saja yang dilakukan pada sistem. Maka activity diagram menggambarkan bagaimaana proses-proses tersebut dilakukan.

Seperti yang telah dijelaskan pada saat identifikasi use case, maka proses proses-proses yang terlibat dalam sistem adalah sebagai berikut:

4.3.1 Activity Diagram Pendaftaran Member Ai Read

Pada gambar 4.5 berikut akan dijelaskan activity diagram pendaftaran member yang dilakukan oleh actor user :



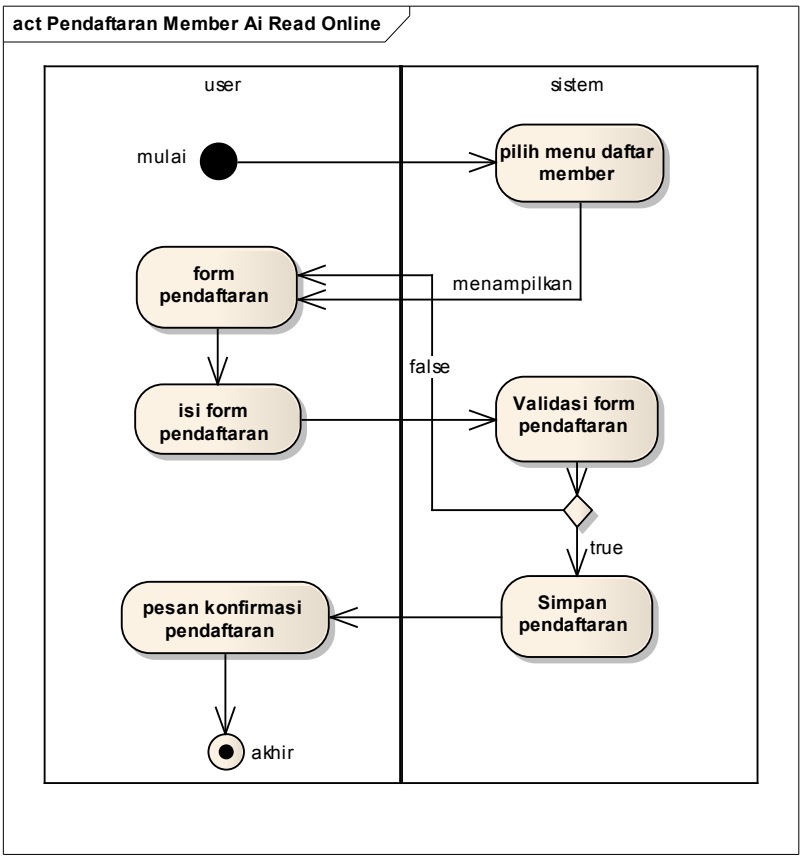
Gambar 4.5 (Activity Diagram Pendaftaran Member Ai Read)

Tabel 4.5 deskripsi Activity Diagram Pendaftaran Member Ai Read

DESKRIPSI ACTIVITY PENDAFTARAN MEMBER AI READ	
Activity	Pendaftaran Member
Deskripsi	Fungsi Pendaftaran Member
Aktor	User
Deskripsi proses	1. User datang ke Ai Read untuk mendaftar member 2. Kasir memberikan formulir pendaftaran 3. User mengisi formulir pendaftaran 4. Kasir cek pengisian formulir pendaftaran dan persyaratan pendaftaran member, jika proses salah maka formulir diserahkan kembali kepada user untuk membernarkan pengisian formulir pendaftaran, jika pengisian formulir pendaftaran benar dan persyaratan telah lengkap maka kasir mencatat pendaftaran member dan meminta biaya administrasi pendaftaran 5. User membeyar biaya administrasi kepada kasir 6. Kasir mencetak kartu anggota/ member dan menyerahkan kepada user 7. User menerima kartu keanggotaan 8. Selesai
Trigger/Event	-
Catatan	User datang langsung ke toko Ai Read

4.3.2 Acrivity Diagram Pendaftaran Member Ai Read Online

Pada gambar 4.6 berikut akan dijelaskan activity diagram pendaftaran member online yang dilakukan oleh actor user :



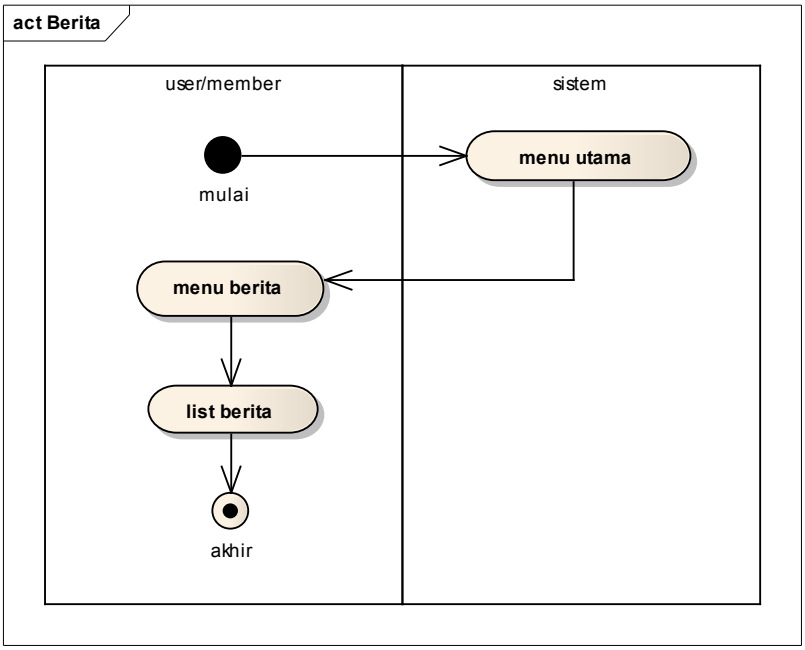
Gambar 4.6 (Activity Diagram Pendaftaran Member Online)

Tabel 4.6 deskripsi Activity Diagram Pendaftaran Member Online

DESKRIPSI ACTIVITY PENDAFTARAN MEMBER ONLINE	
Activity	Pendaftaran Member Online
Deskripsi	Fungsi Pendaftaran Member Online
Aktor	User
Deskripsi proses	1. User memilih menu *daftar member* yang disediakan pada website Ai Read 2. Sistem memunculkan form isian pendaftaran member 3. User mengisi form, klik tombol simpan 4. Sistem cek pengisian formulir, jika pengisian salah maka user kembali lagi ke form isian pendaftaran, jika pengisian benar maka proses penyimpanan berhasil 5. Sistim mengirimkan pesan 6. User menerima pesan konfirmasi pendaftaran 7. Selesai
Trigger/Event	-
Catatan	User memilih menu *daftar member*

4.3.3 Activity Diagram Berita

Pada gambar 4.7 berikut akan dijelaskan activity diagram berita yang dilakukan oleh actor user/member :



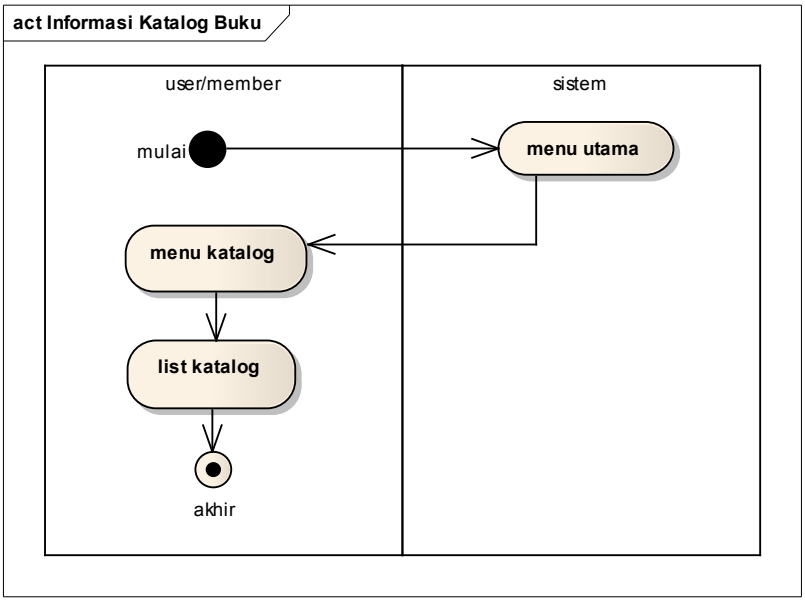
Gambar 4.7 (Activity Diagram Berita)

Tabel 4.7 deskripsi Activity Diagram Berita

DESKRIPSI ACTIVITY BERITA	
Activity	Melihat Berita
Deskripsi	Fungsi Menu Berita
Aktor	User/Member
Deskripsi proses	1. User memilih menu *berita* yang disediakan pada website Ai Read 2. Sistem menampilkan form berita 3. List berita 4. User membaca berita 5. Selesai
Trigger/Event	-
Catatan	User memilih menu *berita*

4.3.4 Activity Diagram Katalog

Pada gambar 4.8 berikut akan dijelaskan activity diagram katalog yang dilakukan oleh actor user/member :



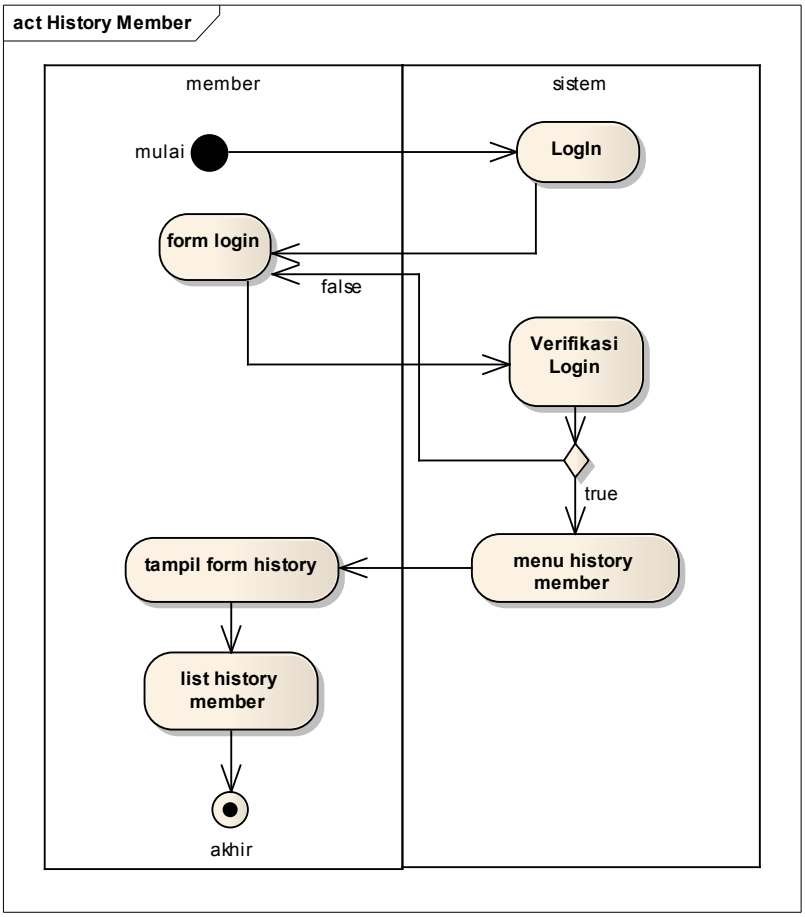
Gambar 4.8 (Activity Diagram Katalog)

Tabel 4.8 deskripsi Activity Diagram Katalog

DESKRIPSI ACTIVITY KATALOG	
Activity	Melihat Katalog Buku
Deskripsi	Fungsi Menu Katalog
Aktor	User/Member
Deskripsi proses	1. User memilih menu *katalog* yang disediakan pada website Ai Read 2. Sistem menampilkan form katalog 3. List katalog 4. User melihat katalog 5. Selesai
Trigger/Event	-
Catatan	User memilih menu *katalog*

4.3.5 Activity Diagram History Member

Pada gambar 4.9 berikut akan dijelaskan activity diagram history member yang dilakukan oleh actor member :



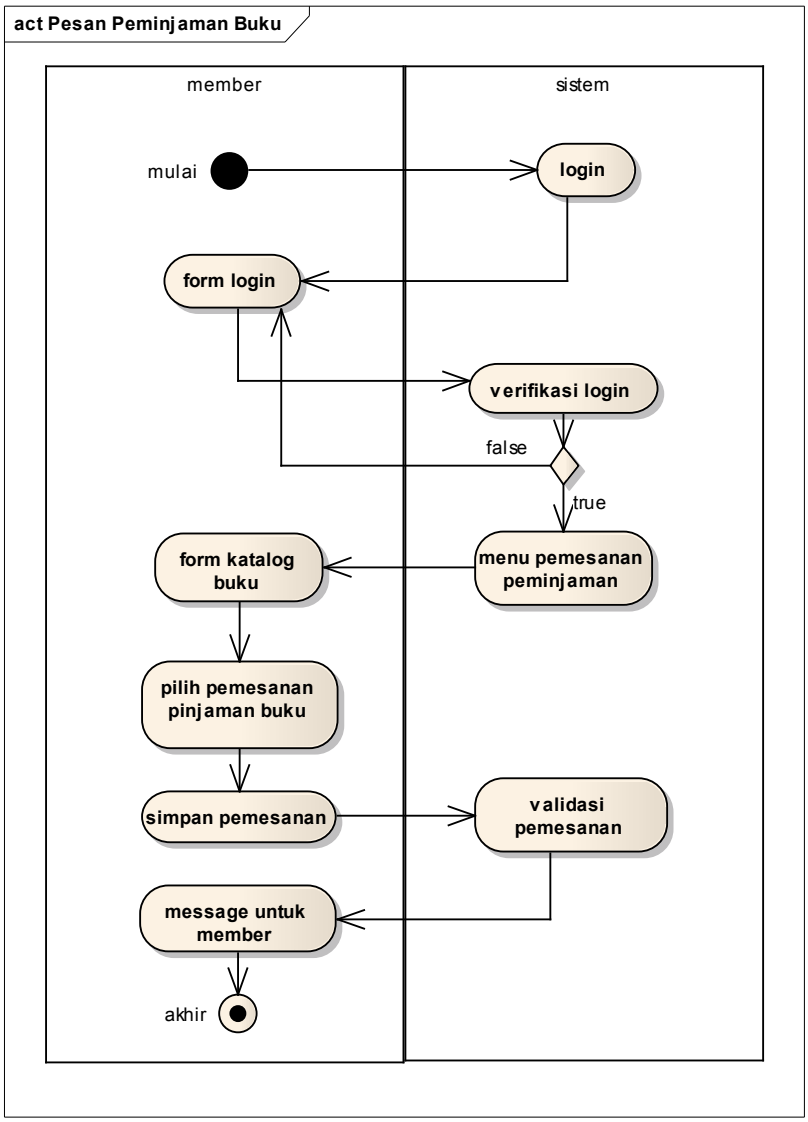
Gambar 4.9 (Activity Diagram History Member)

Tabel 4.9 deskripsi Activity Diagram History Member

DESKRIPSI ACTIVITY HISTORY MEMBER	
Activity	Melihat History Member
Deskripsi	Fungsi Menu History Member
Aktor	Member
Deskripsi proses	1. User login 2. Sistem cek login, jika salah maka tampilan form login kembali, jika benar member berhasil masuk ke fasilitas history member 3. Member mengakses menu history member 4. Form list history member tampil 5. Selesai
Trigger/Event	-
Catatan	User memilih menu *history member*

4.3.6 Activity Diagram Pemesanan Pinjam Buku

Pada gambar 4.10 berikut akan dijelaskan activity diagram pemesanan pinjam buku yang dilakukan oleh actor member :



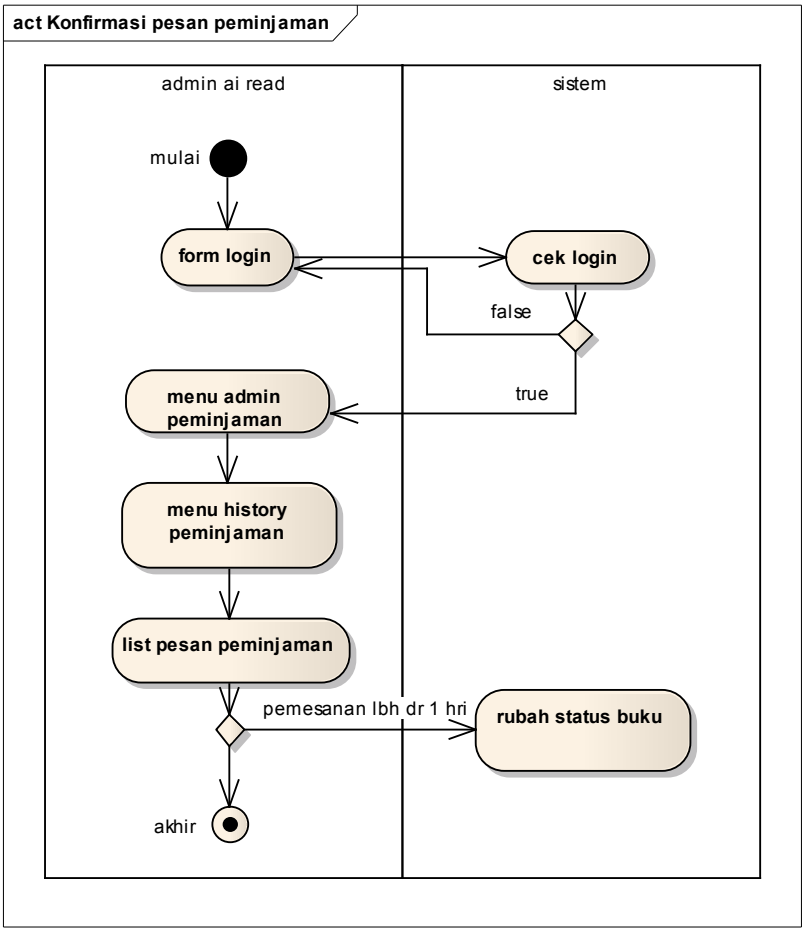
Gambar 4.10 (Activity Diagram Pesan Peminjaman Buku)

Tabel 4.10 deskripsi Activity Diagram Pesan Pinjam Buku

DESKRIPSI ACTIVITY PESAN PINJAM BUKU	
Activity	Pesan Pinjam Buku
Deskripsi	Pemesanan Peminjaman Buku
Aktor	Member
Deskripsi proses	1. Member Login 2. Sistem cek login 3. Jika salah login member dikembalikan ke form login 4. Jika benar member sukse masuk, dan dapat mengakses menu pemesanan peminjaman 5. Form katalog muncul 6. Member memilih buku yang akan dipesan untuk dipinjam 7. Member klik tombol proses 8. Sistem menyimpan proses pemesanan peminjaman buku 9. Sistem menguirimkan pesan kepada member untuk proses peminjaman 10. Selesai
Trigger/Event	-
Catatan	User memilih menu *pesan peminjaman*

4.3.7 Activity Diagram Konfirmasi Pesan Pinjam Buku

Pada gambar 4.11 berikut akan dijelaskan activity diagram konfirmasi pesan pinjam buku yang dilakukan oleh actor admin :



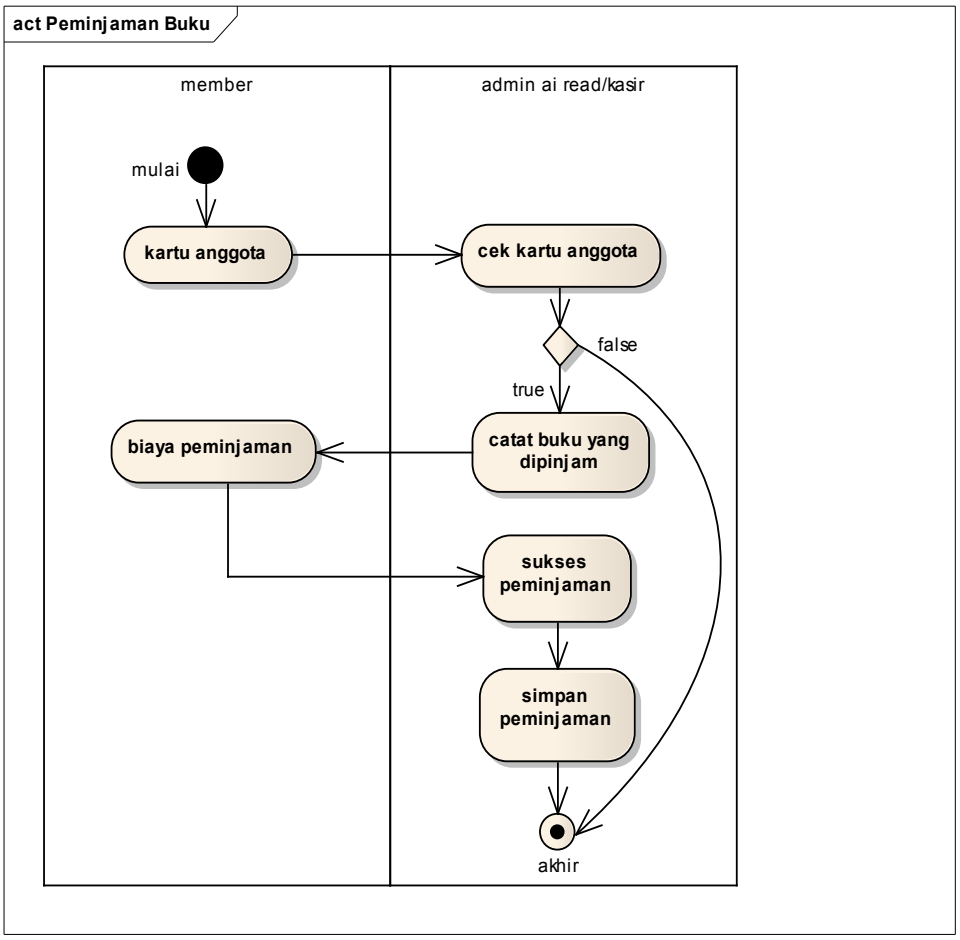
Gambar 4.11 (Activity Diagram Konfirmasi Pesan Peminjaman Buku)

Tabel 4.11 deskripsi Activity Diagram Konfirmasi Pesan Peminjaman Buku

DESKRIPSI ACTIVITY KONFIRMASI PESAN PINJAM BUKU	
Activity	Konfirmasi Pesan Pinjam Buku
Deskripsi	Konfirmasi Pemesanan Peminjaman Buku
Aktor	Admin
Deskripsi proses	1. Admin Login 2. Sistem cek login 3. Jika ada kesalahan login sistem mengembalikan ke halaman form login 4. Jika benar maka admin sukses masuk ke halaman admin 5. Admin klik menu peminjaman, klik menu history peminjaman 6. Sistem memunculkan halaman list peminjaman 7. Admin cek pesan peminjaman 8. Jika lebih dari 1 hari member tidak melanjutkan proses peminjaman, yaitu dengan datang ke ai read, maka admin merubah status buku yang tadinya sedang proses pemesanan, dirubah menjadi ada 9. Selesai
Trigger/Event	-
Catatan	Admin berhasil login

4.3.8 Activity Diagram Peminjaman Buku

Pada gambar 4.12 berikut akan dijelaskan activity diagram peminjaman buku yang dilakukan oleh actor member :



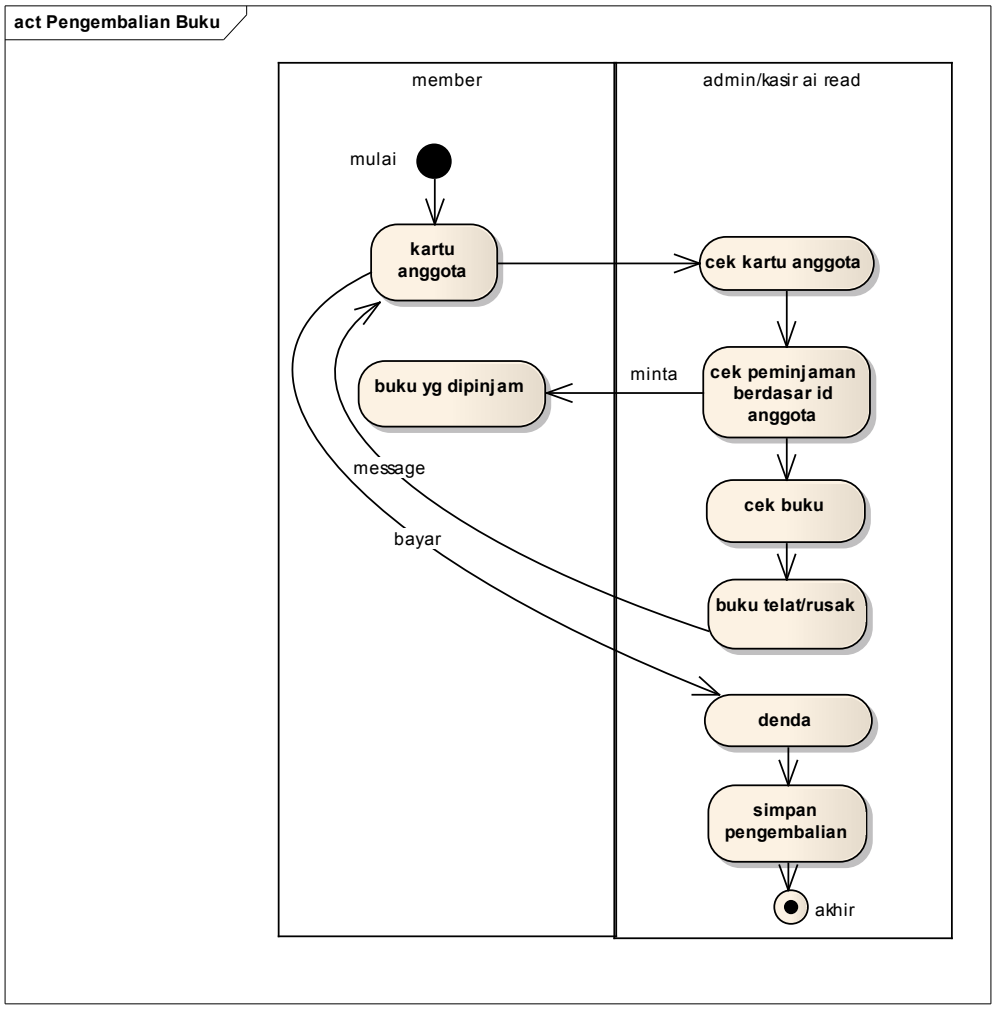
Gambar 4.12 (Activity Diagram Peminjaman Buku)

Tabel 4.12 deskripsi Activity Diagram Peminjaman Buku

DESKRIPSI ACTIVITY PEMINJAMAN BUKU	
Activity	Peminjaman Buku
Deskripsi	Peminjaman Buku
Aktor	Member
Deskripsi proses	1. Member datang ke ai read dan menyerahkan kartu anggota 2. Admin/kasir ai read cek kartu anggota apakah benar sudah terdaftar 3. Jika salah maka proses berakhir 4. Jika benar maka admin/kasir ai read mencatat buku yang akan dipinjam 5. Member membayar biaya peminjaman berdasarkan harga sewa dan jumlah buku 6. Admin/ kasir ai read menyimpan proses peminjaman 7. Selesai
Trigger/Event	-
Catatan	Member datang langsung ke toko Ai Read

4.3.9 Activity Diagram Pengembalian dan Denda Buku

Pada gambar 4.13 berikut akan dijelaskan activity diagram pengembalian dan denda buku yang dilakukan oleh actor member :



Gambar 4.13 (Activity Diagram Pengembalian dan Denda Buku)

Tabel 4.13 deskripsi Activity Diagram Pengembalian dan Denda Buku

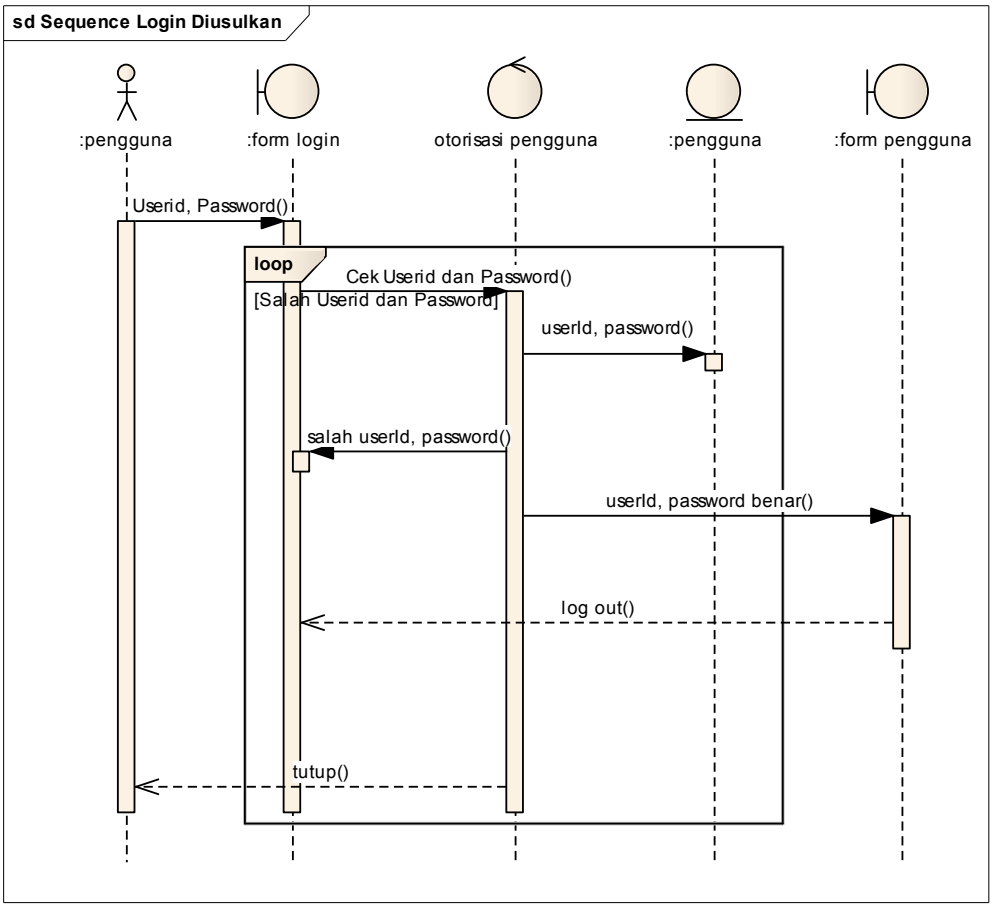
DESKRIPSI ACTIVITY PENGEMBALIAN DAN DENDA BUKU	
Activity	Pengembalian dan Denda Buku
Deskripsi	Pengembalian dan Denda Buku
Aktor	Member
Deskripsi proses	1. Member datang ke toko Ai Read 2. Member menyerahkan karu anggota dan buku yang dipinjam 3. Admin/kasir Ai Read cek peminjaman buku berdasarkan id anggota 4. Jika buku yang dipinjam telat dan ada kerusakan maka admin/kasir ai read meminta biaya denda berdasarkan berapa hari telat dan kerusakan buku 5. Jika buku yang dipinjam tidak bermasalah maka admin/kasir ai read menyimpan proses peminjaman 6. Selesai
Trigger/Event	-
Catatan	Member datang langsung ke toko Ai Read

4.4 Model Diagram Sequence yang Diusulkan

Sequnce Diagram menggambarkan interakasi antara sejumlah object di dalam dan disekitar system berupa message yang digambarkan terhadap waktu, sequence diagram terdiri antar dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait)

4.4.1 Sequence Login

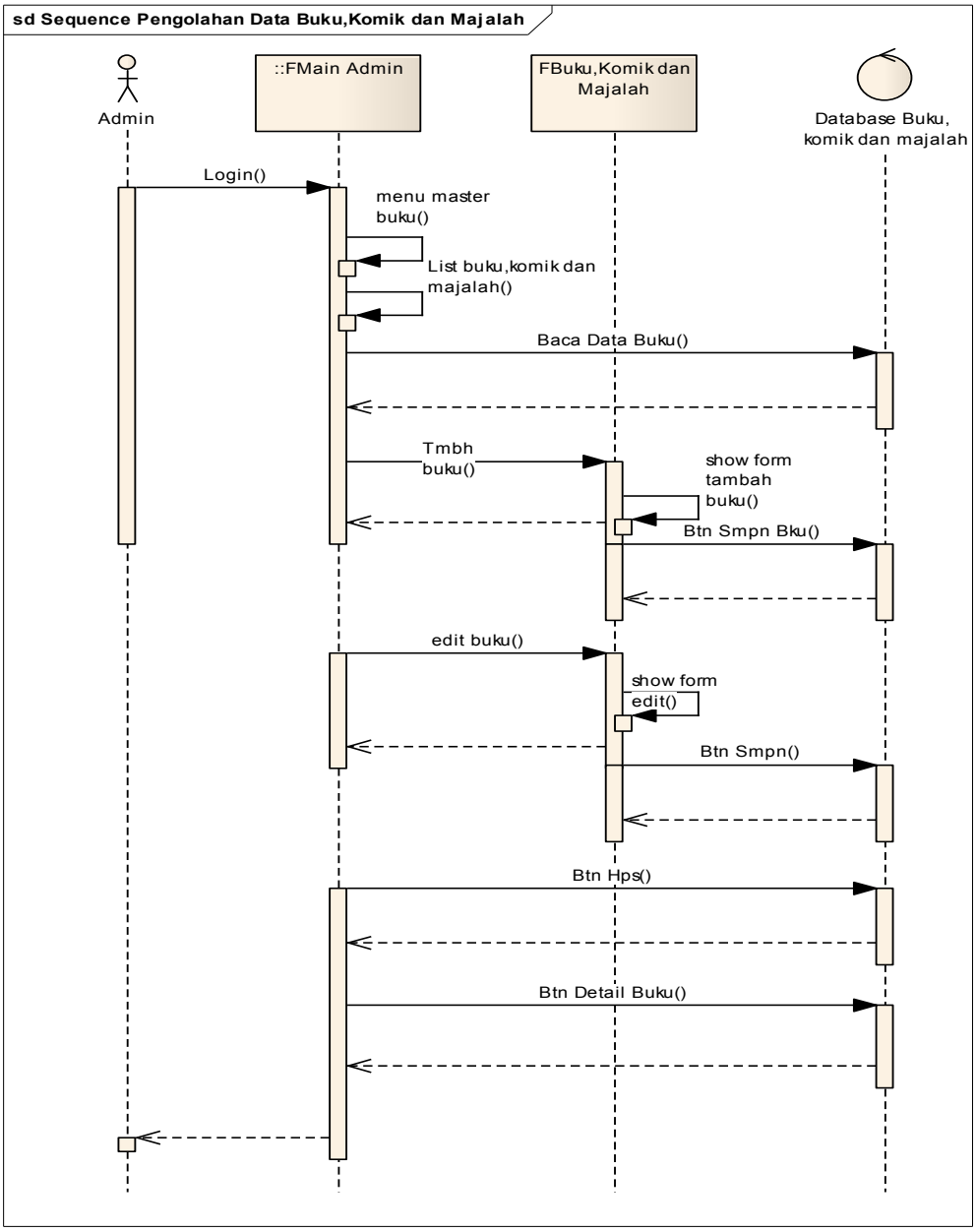
Pada gambar 4.14 berikut diperlihatkan *Sequence diagram* Login :



Gambar 4.14 (Sequence Diagram Login)

4.4.2 Sequence Pengolahan Data Buku

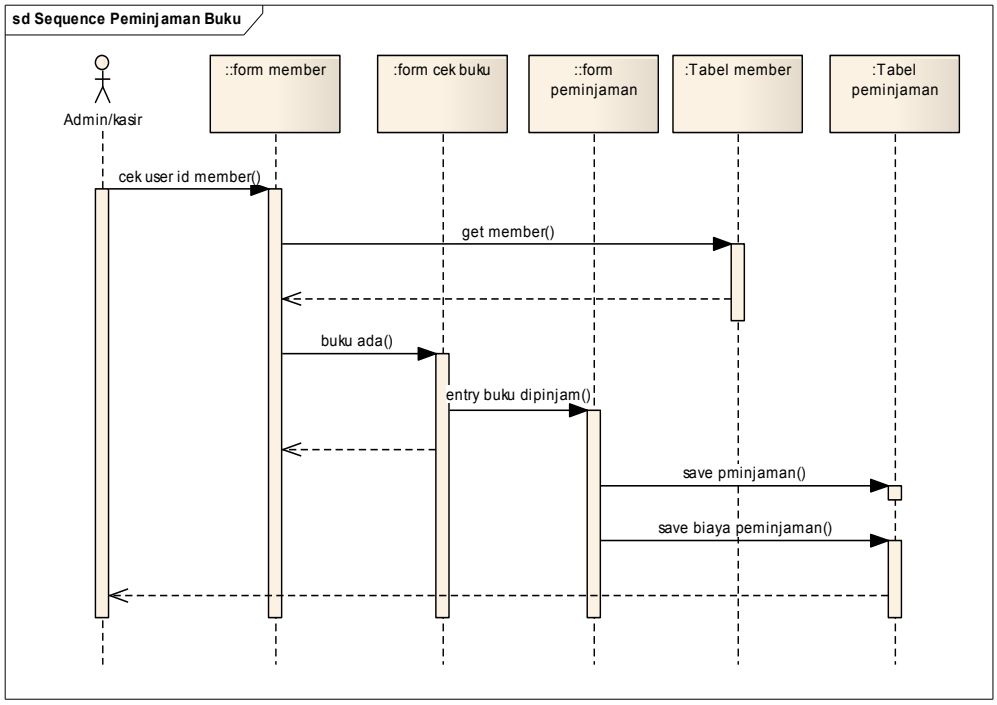
Pada gambar 4.15 berikut diperlihatkan *Sequence diagram* Pengolahan Data buku :



Gambar 4.15 (Sequence Diagram Pengolahan Data Buku, Komik dan Majalah)

4.4.3 Sequence Peminjaman Buku

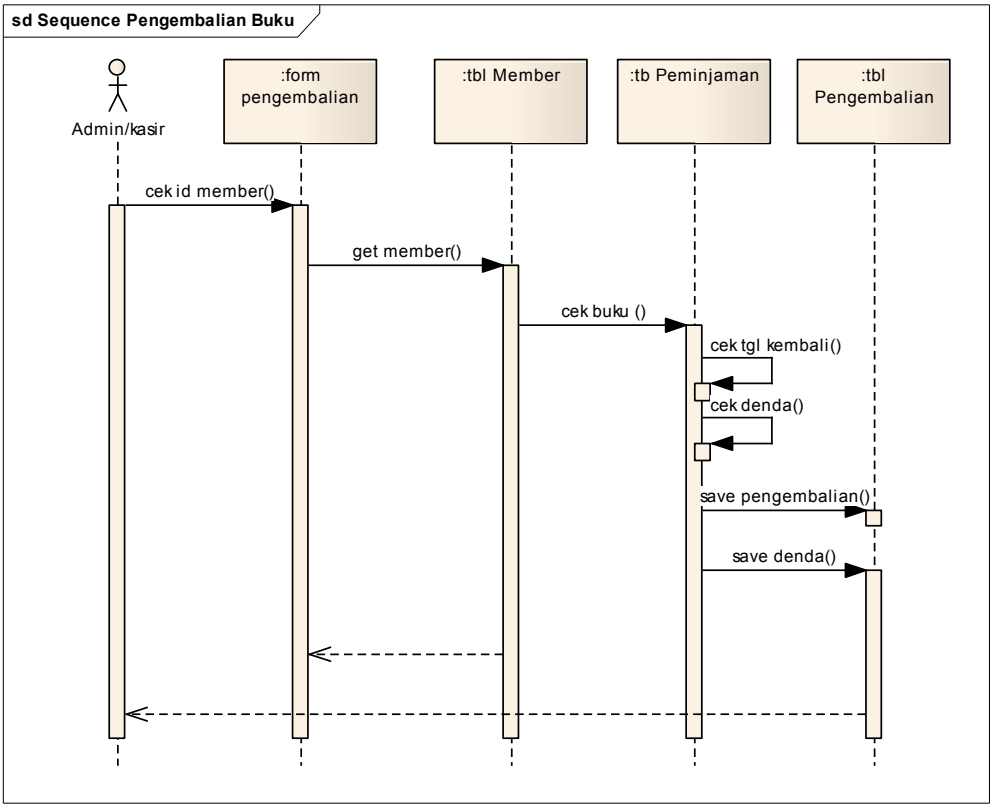
Pada gambar 4.16 berikut diperlihatkan *Sequence diagram* Peminjaman Buku, Komik dan Majalah :



Gambar 4.16 (Sequence Diagram Peminjaman Buku, Komik dan Majalah)

4.4.4 Sequence Pengembalian Buku

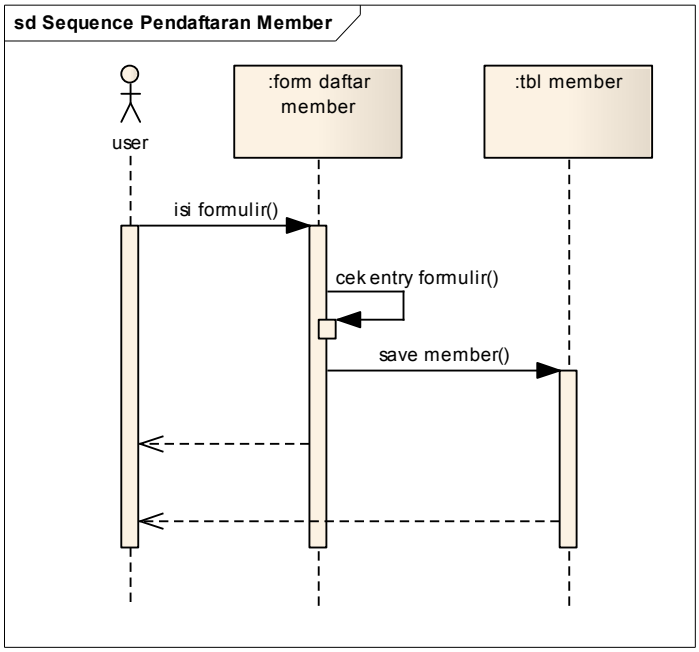
Pada gambar 4.17 berikut diperlihatkan *Sequence diagram* Pengembalian Buku, Komik dan Majalah :



Gambar 4.17 (Sequence Diagram Pengembalian Buku, Komik dan Majalah)

4.4.5 Sequence Pendaftaran Member

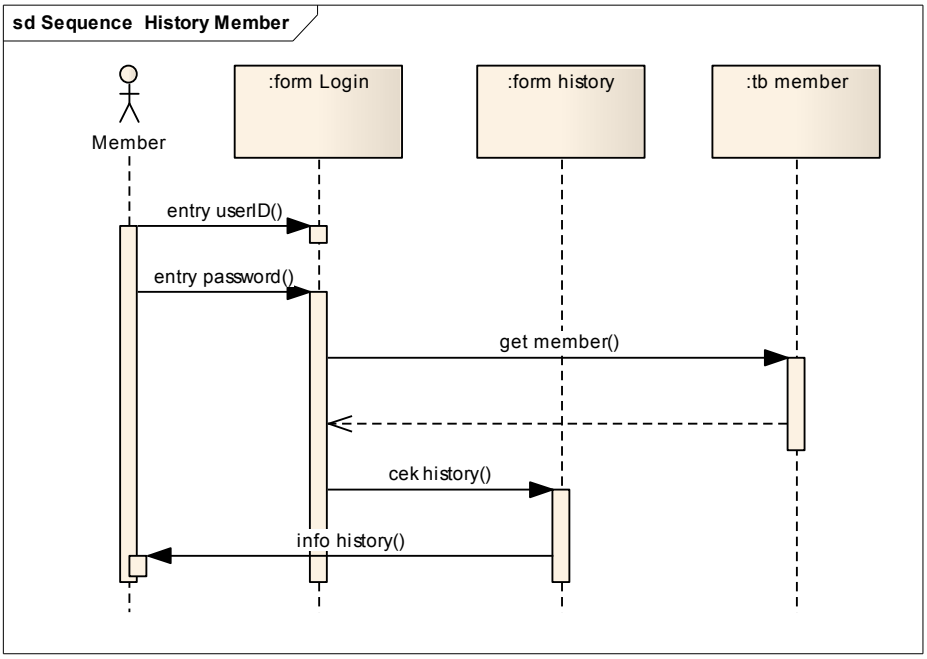
Pada gambar 4.18 berikut diperlihatkan *Sequence diagram* pendaftaran member :



Gambar 4.18 (Sequence Diagram Pedaftaran Member)

4.4.6 Sequence History Member

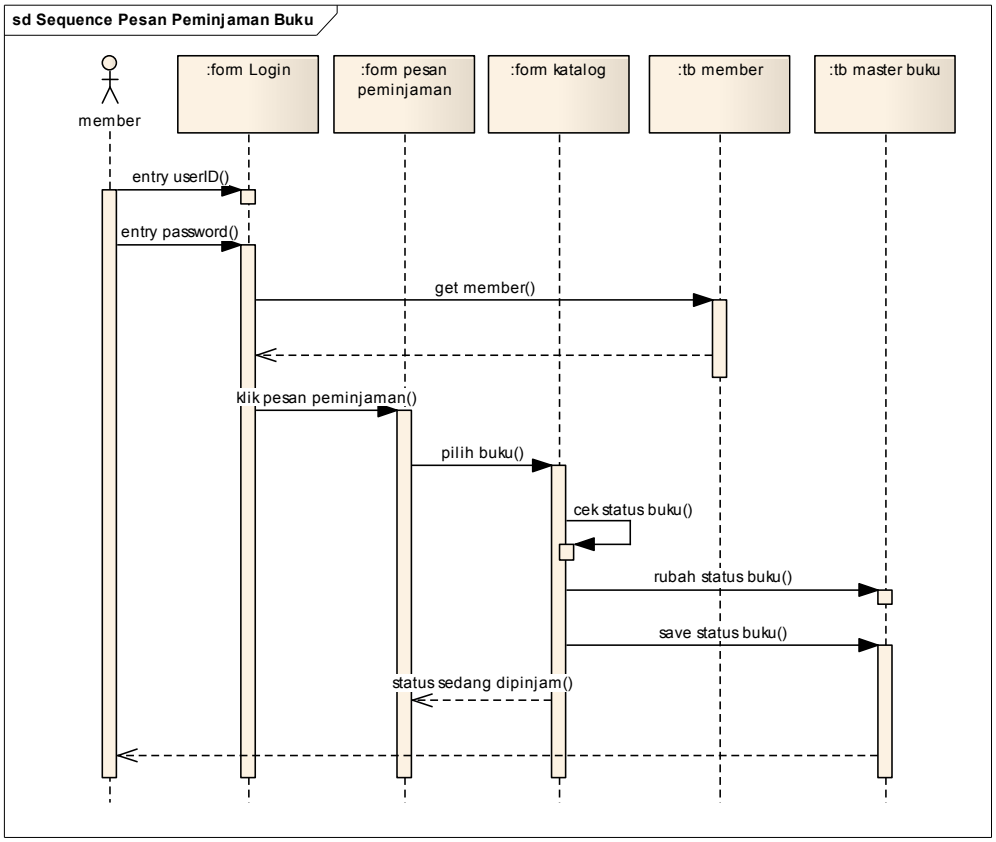
Pada gambar 4.19 berikut diperlihatkan *Sequence diagram* history member :



Gambar 4.19 (Sequence Diagram History Member)

4.4.7 Sequence Pesan Peminjaman Buku

Pada gambar 4.20 berikut diperlihatkan *Sequence diagram* pesan peminjaman buku :

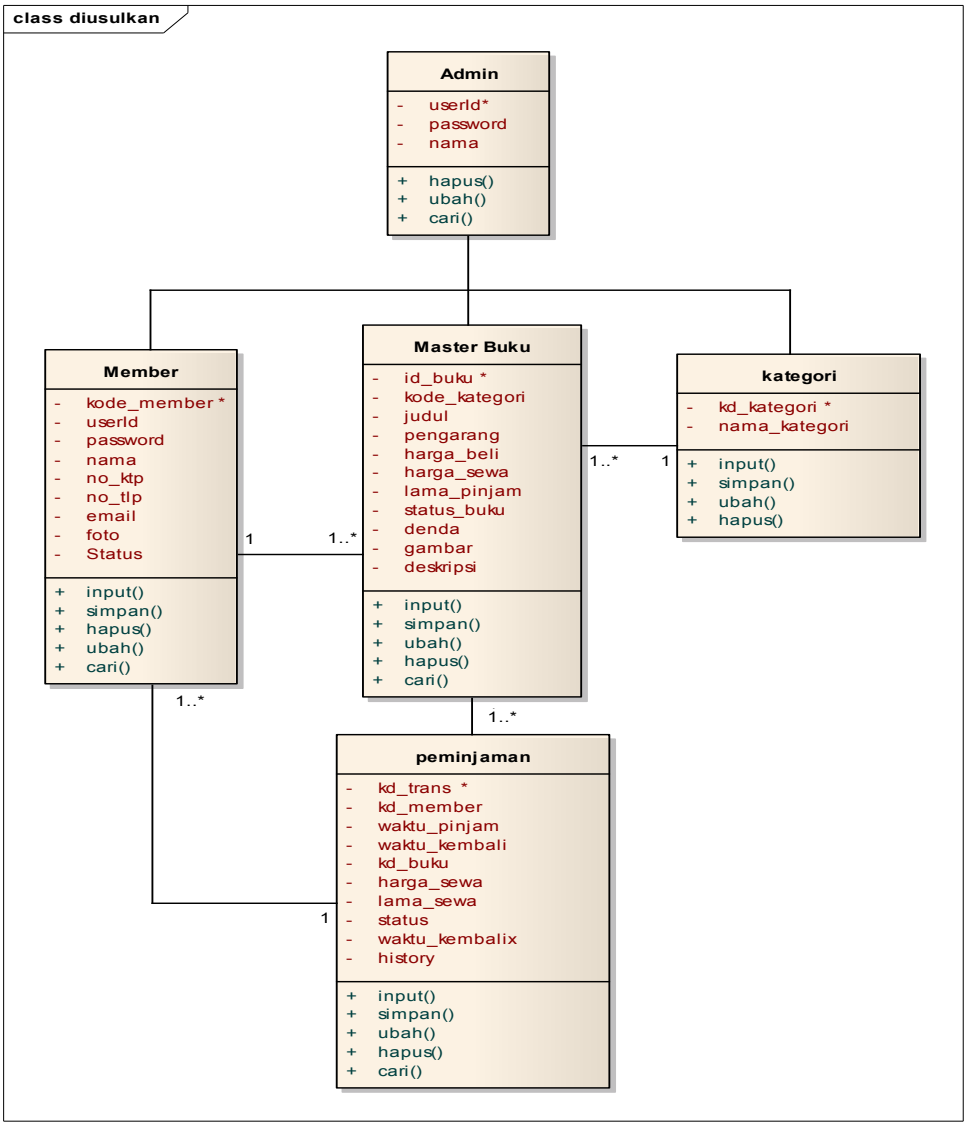


Gambar 4.20 (Sequence Diagram Pesan Peminjaman Buku)

4.5 Model Class Diagram yang Diusulkan

Class adalah sebuah spesifikasi yang jika di instansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

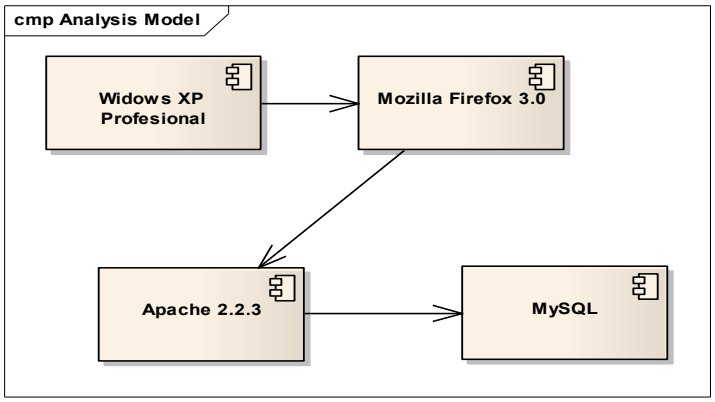
4.5.1 Relasi Class



Gambar 4.21 (Relasi Class)

4.6 Model Componen Diagram yang Diusulkan

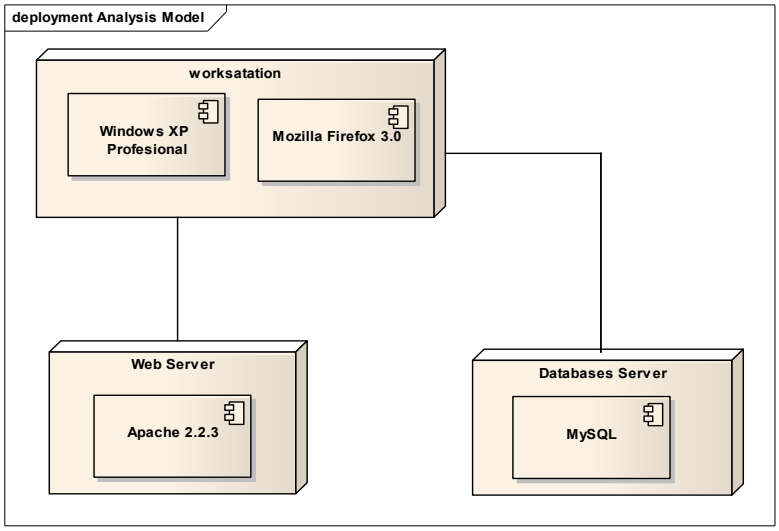
Komponen perangkat lunak adalah bagian fisik dari sebuah sistem yang menetap di komputer. komponen merupakan implementasi software dari sebuah class. Komponen bisa berupa tabel, *file* data, *file* exe, file DLL, dokumen dan lain-lain. Berikut gambar component diagram yang diusulkan :



Gambar 4.22 (Component Diagram)

4.7 Model Deployment Diagram Yang Diusulkan

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment diagram* dapat digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem.



Gambar 4.23 (Deployment Diagram)

4.8 Perancangan Tampilan

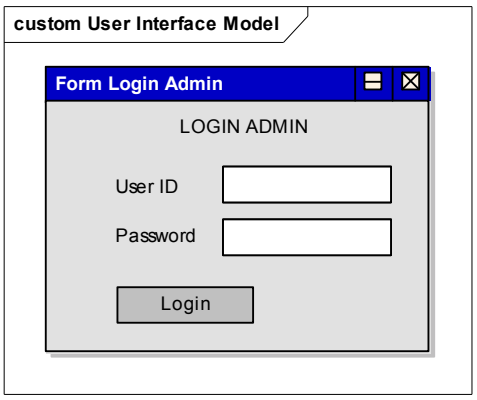
Perancangan tampilan / *interface* dalam melakukan proses-proses yang ada didalam system berdsarkan gambar diagram-diagram UML sebelumnya. Perancangan ini melibatkan perancangan *input-output.*

4.8.1 Perancangan Input

Perancangan Input merupakan awal dimulainya sesuatu proses sistem informasi. Bahan mentah dari informasi ini adalah data yang terjadi dari traksaksi-transaksi ataupun data dari luar sistem.

Sebagai database awal yang sedang dibangun, Pengolahan data Pengiriman membutuhkan data input awalan, sebagai hak untuk menggunakan amplikasi pengolahan data pengiriman, bentuk awal dari aplikasi tersebut adalah sebagai berikut ini :

1. Form login Admin

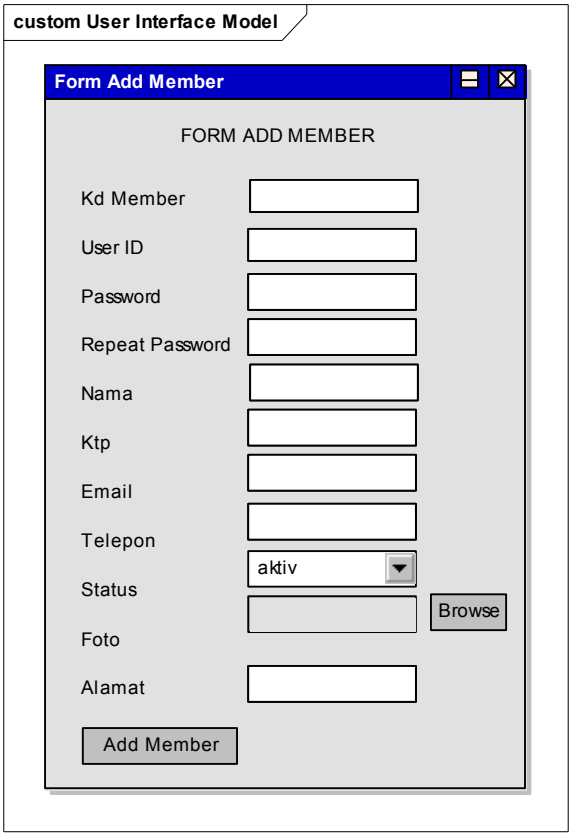


Gambar 4.24 (Tampilan Form Login Admin)

Keterangan :

1. User ID : untuk menginputkan userID
2. Password : untuk menginputkan password
3. Button Login : digunakan untuk proses login

2. Form Add Member



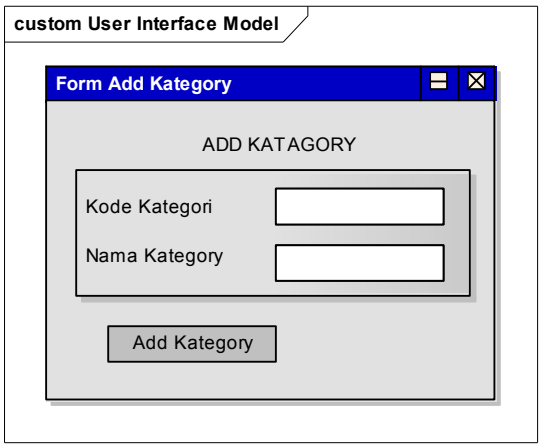
Gambar 4.25 (Tampilan Form Add Member)

Keterangan :

1. Kd Member : untuk memasukan inputan kode membe
2. User ID : untuk penginputan user id
3. Password : untuk menginputkan password
4. Repeat Password : untuk pengulangan password
5. Nama : untuk menginputkan nama
6. Ktp : untuk menginputkan ktp
7. Email : untuk menginputkan email
8. Telepon : untuk menginputkan telepon

9. Status : untuk menginputkan status member oleh admin
10. Foto : untuk menginputkan gambar foto
11. Botton Browse : untuk mencari lokasi foto yang akan diinputkan
12. Alamat : untuk menginputkan alamat
13. Button Add Member : tombol untuk penyimpanan

3. Form Add Katagory Buku

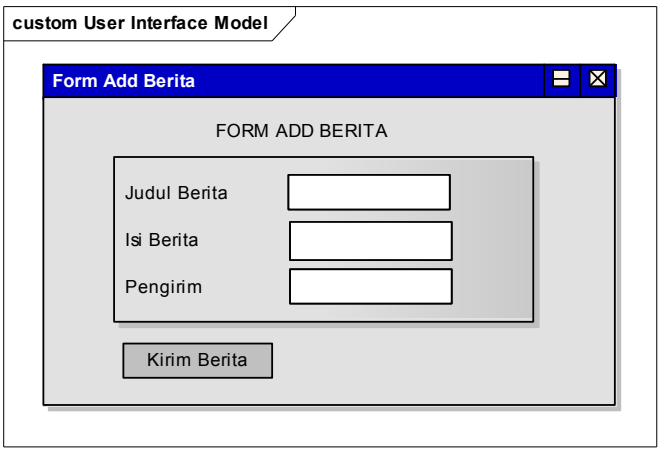


Gambar 4.26 (Tampilan Form Add Kategory)

Keterangan :

1. Kode kategori : untuk menginputkan kategori
2. Nama kategori : untuk menginputkan nama kategory
3. Button Add Kategory : untuk tombol penyimpanan

4. Form Add Berita

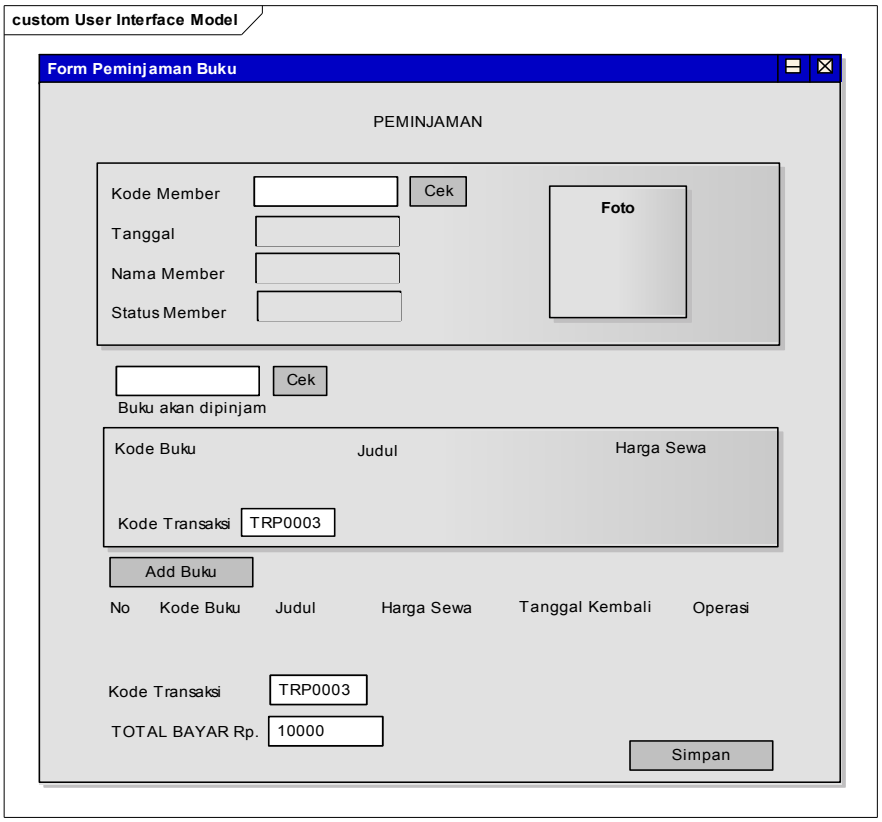


Gambar 4.27 (Tampilan Form Add Berita)

Keterangan :

1. Judul berita : untuk penginputan judul berita
2. Isi berita : untuk penginputan isi berita
3. Pengirim : untuk menginputkan pengirim berita
4. Button Kirim Berita : untuk tombol penyimpanan berita

5. Form Peminjaman Buku

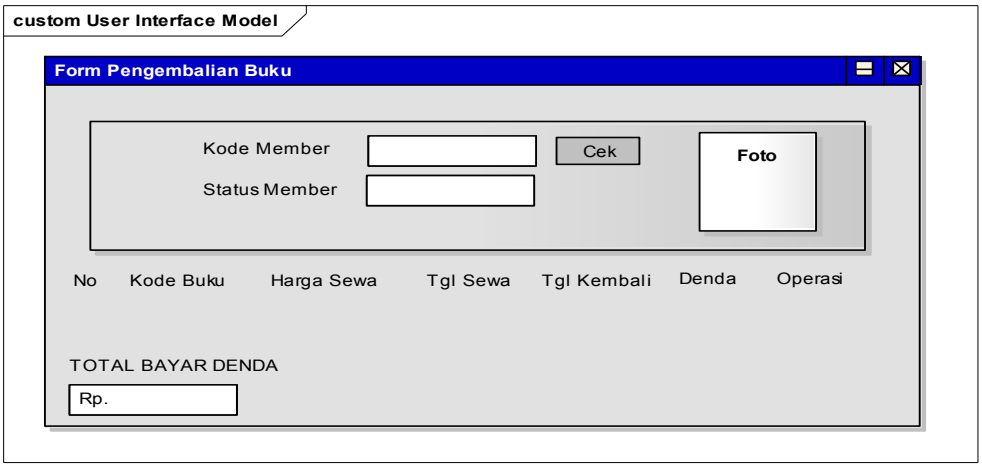


Gambar 4.28 (Tampilan Form Peminjaman Buku)

Keterangan :

1. Kode Member : untuk menginputkan kode member
2. Button cek : untuk mengecek apakah member ada atau tidak
3. Tanggal : tanggal pendaftaran memebr
4. Nama member : menampilkan nama member
5. Status member : menampilkan status member
6. Button add buku : untuk konfirmasi buku yang akan dipinjam
7. Button simpan : untuk menyimpan peminjaman

6. Form Pengembalian Buku



Gambar 4.29 (Tampilan Form Pengembalian Buku)

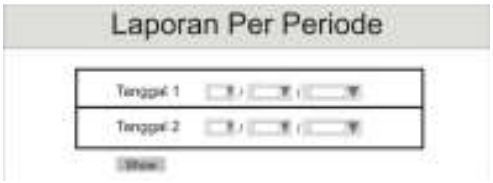
Keterangan :

1. Kode member : untuk menginputkan kode member
2. Status Member : untuk mengetahui status memeber
3. Button cek : tombol untuk mengecek member

3.8.1 Perancangan Output

Bentuk dari laporan yang dihasilkan sebagai output dari sisetm informasi yang berjalan dan umum dugunakan adalah dalam bisa berbentuk tabel, grafik dan bagan

1. Tampilan Laporan



Gambar 4.30 (Laporan Awal)

Keterangan :

1. Tanggal 1 : untuk set tanggal bulan tahun awal
2. Tanggal 2 : untuk set tanggal bulan tahun terakhir yang akan dilihat
3. Button show : tombol untuk melihat report

2. Laporan master buku



Gambar 4.31 (Laporan Master Buku)

Keterangan :

1. Merupakan tampilan laporan dimana terdapat beberapa field id buku, judul, pengarang, harga buku, dan harga sewa.

3. Laporan member



Gambar 4.32 (Laporan Member)

Keterangan :

1. Merupakan tampilan laporan dimana terdapat beberapa field id buku, judul, pengarang, harga buku, dan harga sewa.

4. Laporan peminjaman



Gambar 4.33 (Laporan Peminjaman Buku)

Keterangan :

1. Merupakan tampilan laporan dimana terdapat beberapa field id buku, judul, pengarang, harga buku, harga sewa dan grand total.

5. Laporan pengembalian

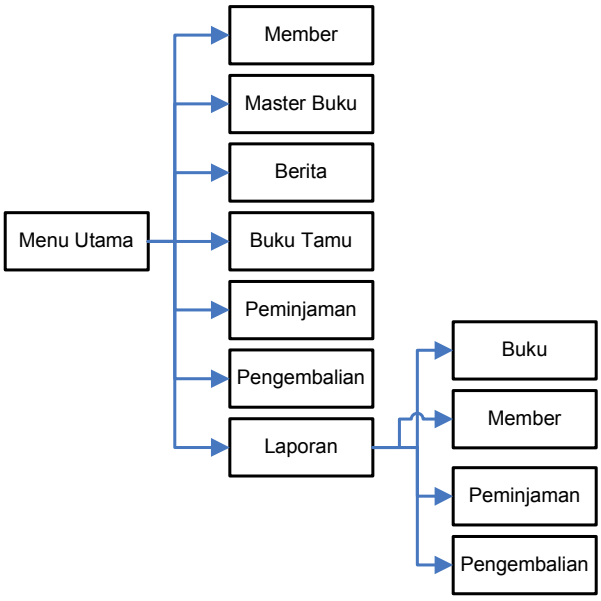


Gambar 4.34 (Laporan Pengembalian Buku)

Keterangan :

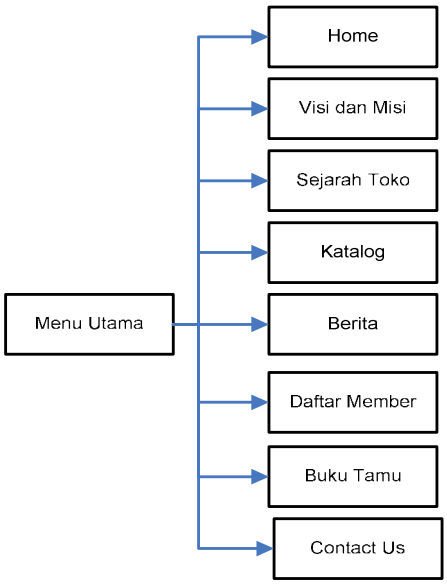
1. Merupakan tampilan laporan dimana terdapat beberapa field id buku, judul, pengarang, harga buku, harga sewa dan total denda.

3.8.2 Perancangan Menu Halaman Main Admin



Gambar 4.35 (Struktur Menu Main Admin)

3.8.3 Perancangan Menu Halaman Website Ai Read

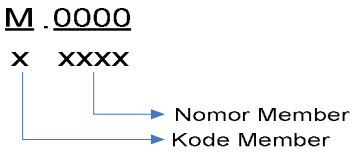


Gambar 4.36 (Struktur Menu Website Ai Read)

4.8.5 Pengkodean

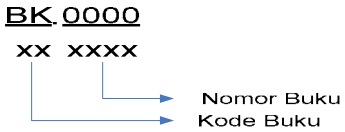
Pengkodean ini adapat berfungsi untuk mendefiniskan suatu objek secara singkat, mengklasifikasikan data dan memasukan data kedalam, selain itu kode dapat dibentuk dari kumpulan huruf, angka, dan karakter khusus pada program system informasi ini terdapat nomor urut pengkodean berguna untuk mempermudah dalam pencarian data. Adapun contoh pengkodean yaitu sebagai berikut :

a. Member



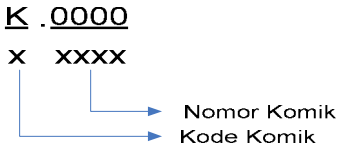
Contoh : M.001

b. Buku



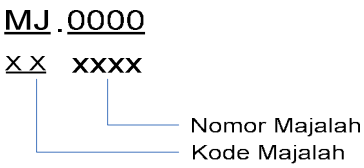
Contoh : B.001

c. Komik



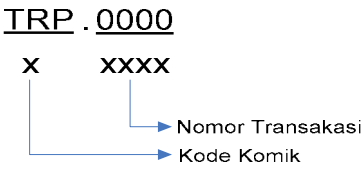
Contoh : K.001

d. Majalah



Contoh : MJ.001

e. Transaksi Peminjaman



Contoh : TRP.001

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1 Implementasi

Kegiatan Implementasi atau penerapan dilakukan dengan dasar yang telah direncanakan dalam rencana implementasi. Pada penerapan sistem yang diusulkan tentu saja harus disediakan beberapa aspek yang akan mendukung kelancaran operasi sistem dan hal yang akan dijadikan faktor pendukung adalah penentuan kebutuhan sumber daya.

Implementasi adalah proses penerapan rancangan program yang telah dibuat pada bab sebelumnya atau aplikasi dalam melaksanakan sisten informasi pemograman yang dibuat.

5.1.1 Batasan Implementasi

Pembuatan aplikasi Sistem Informasi Pembangunan website sebagai media promosi katalog online pada toko buku dan penyewaan komik Ai Read ini dibatasi pada Perangkat Lunak (*software*) dan Perangkat Keras (*hardware*) yang digunakan.

5.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk proses pengkodean adalah bahasa pemograman, adapun bahasa pemograman yang digunakan adalah PHP. Proses pengkodean adalah suatu kegiatan penulisan program yang akan dieksekusi oleh komputer dan juga suatu tahap menterjemahkan logika ke dalam aplikasi bahasa pemograman PHP. Kode program diisi oleh pemogram

berdasarkan dokumentasi hasil dari perancangan sistem secara rinci. Untuk menulis kode program yang akan dibuat membutuhkan bahasa pemograman yang sesuai dengan kebutuhan sistem berbasis komputer yang akan dibuat. Adapun hasil dari pengkodean itu adalah berupa listing program.

Sebelum program diterapkan, maka program harus bebas dari kesalahan. Oleh karena itu perlu dilakukan pengetesan program baik permodul atau maupun pengetesan untuk semua modul yang sudah dirangkaikan kesalahan-kesalahan yang mungkin terjadi adalah:

1. Kesalahan Bahasa

   Kesalahan penulisan program yang tidak sesuai dengan yang disyaratkan. Kesalahan ini relatif rendah diketahui karena *Compiler* akan memberitahukan letak dan sebab kesalahannya sewaktu program dikompilasi.

2. Kesalahan Waktu Proses

   Kesalahan ini terjadi sewaktu program dijalankan. Kesalahan ini menyebabkan program berhenti sebelum selesai. Kesalahan ini juga relatif mudah ditemukan ditunjukan letak dan sebab kesalahan.

3. Kesalahan Logika

   Kesalahan logika program yang dibuat. Kesalahan ini sulit ditemukan karena tidak ada pemberitahuan kesalahanya, tetapi dapat diketahui dari hasil proses program yang salah. Cara mencari kesalahan ini dapat dicari dengan test data yaitu dengan menjalankan program dengan data tertentu dan membandingkan hasil pengolahan dengan hasil yang sudah diketahui.

5.1.3 Implementasi Perangkat Keras

Perangkat Keras yang digunakan dalam Pembuatan aplikasi Pembangunan website sebagai media promosi katalog online pada toko buku dan penyewaan komik Ai Read sangat menentukan kecepatan dan ketepatan dalam proses penyimpanan data. Adapun Perangkat Keras yang digunakan dalam Pembuatan aplikasi Pembangunan website sebagai media promosi katalog online pada toko buku dan penyewaan komik Ai Read adalah sebagai berikut :

1. Processor Pentium II
2. VGA Card 32 Mb
3. Memory 128 Mb
4. Harddisk 4Gb
5. Monitor
6. Alat-alat pendukung seperti *mouse,keyboard,cd rom,* dll.

5.1.4 Implementasi Antar Muka

Dalam pembuatan aplikasi apapun, antar muka juga memegang peranan yang penting. Antar muka dapat memudahkan user dalam mengoperasikan aplikasi yang telah dibuat. Oleh karena itu, antar muka pada aplikasi Pembangunan website sebagai media promosi katalog online pada toko buku dan penyewaan komik Ai Read dibuat menarik dan sederhana sehingga petugas dan juga member dapat mengoperasikan aplikasi ini dengan mudah. Untuk lebih jelasnya akan digambarkan sebagai berikut :

5.1.4.1 Tampilan User

1. Tampilan Utama

Ini adalah tampilan awal pada saat seorang pengunjung masuk dalam alamat website Ai Read



Gambar 5.1 *Tampilan Utama User*

2. Tampilan Login Member

Tampilan dimana apabila seorang member ingin mendapatkan dan dapat mangakses fasilitas history member



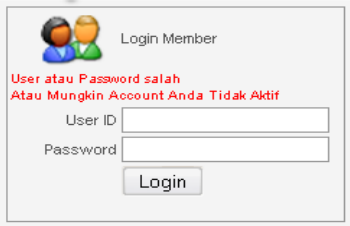
Gambar 5.2 *Tampilan Login Member*

Keterangan :

Pada saat form login yang harus diisikan adalah userID dan password, setelah userID dan password diisi maka tekan tombol login untuk masuk ke halaman.

3. Tampilan Pesan Kesalahan Saat Login Member

Tampilan pesan kesalahan pada saat login member mungkin dikarenakan kesalahan dalam penginputan userId ataupun Password



Gambar 5.3 *Tampilan Pesan Kesalahan Login Member*

4. Tampilan Pada Saat Login Member Berhasil

Tampilan saat berhasil login member sehingga member bisa mendapat fasilitas atau dapat mengakses history member



Gambar 5.4 *Tampilan Halaman Saat Berhasil Login*

Keterangan :

Pada saat login member berhasil maka anda akan mendapatkan tampilan akses member seperti gambar diatas dimana member dapat melihat :

1. *Profile* : member dapat melihat data diri mereka
2. *Pesan online* : member dapat melakukan pemesanan peminjaman buku secara online
3. *History member* : digunakan untuk melihat history member
4. *Logout* : keluar dari sistem

5. Tampilan Edit Profile Member

Ini adalah tampilan pada saat member meng klik profile



Gambar 5.5 *Tampilan Halaman Edit Profile Member*

Keterangan :

Pada form profil terdapat :

1. *Kode member* : merupakan identitas member
2. *Password* : isian password ini nantinya dapat digunakan untuk login member
3. *Nama* : nama member

4. *Ktp* : no ktp member
5. *Email* : email member
6. *Telpon* : nomor telpon member
7. *Alamat* : alamat member
8. *Button update member* : digunakan untuk menyimpan perubahan pada saat profil member dirubah

6. Tampilan Pesan Online

Ini adalah tampilan dimana member memilih fasilitas *pesan online*



Gambar 5.6 *Tampilan Halaman Pesan Online*

Apabila member ingin melanjutkan pemesanan online maka klik link pinjam next, dimana tampilan halaman pinjeam next seperti gambar dibawah ini :



Gambar 5.7 *Tampilan Halaman Pinjam Next*

Setelah member klik pinjam next maka akan diarahkan ke halaman katalog buku, dimana member dapat pesan buku yang akan dipinjam, klik pada

BK0003 sebagai buku yang akan di pesan pinjam, maka member diarahkan ke halaman lanjut pinjam seperti gambar dibawah ini :



Gambar 5.8 *Tampilan Halaman Pilih Buku*

Di halaman ini ada keterangan tentang deskripsi buku, dan saat member ingin melanjutkan pesan peminjaman maka member dapat mengklik link pinjam, sehingga akan tampil halaman seperti gambar dibawah ini :



Gambar 5.9 *Tampilan Halaman Konfirmasi Pesan Peminjaman*

Gambar diatas adalah tampilan halaman konfirmasi untuk pesan peminjaman, apakah member telah cocok dengan buku yang akan di pesan pinjam, atau member belum cocok, apabila member sudah cocok dengan buku yang akan di pesan pinjam member mengklik link *next*



Gambar 5.10 *Tampilan Halaman Selesai Pesan Peminjaman*

Gambar diatas adalah tampilan dimana proses akhir pesan peminjaman buku member mengklik tombol finish.

7. Tampilan History Member

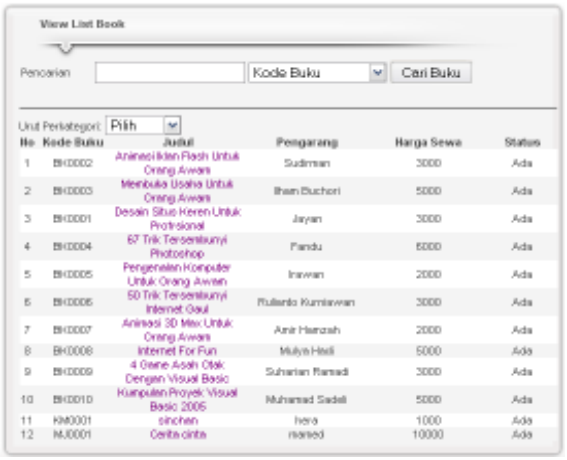
Tampilan dimana member berhasil Login dan menggunakan fasilitas untuk melihat history member



Gambar 5.11 *Tampilan Halaman History Member*

8. Tampilan Katalog Buku, Komik dan Majalah

Halaman katalog buku, komik dan majalah yang dapat diakses baik oleh user biasa maupun member



Gambar 5.12 *Tampilan Halaman Katalog Buku*

Keterangan :

Gambar diatas adalah gambar tampilan halaman katalog dimana banyak fitur yang ada diantaranya :

1. *Pencarian* : kita dapat melakukan pencarian berdasarkan kode buku, judul buku, pengarang, harga sewa buku.
2. *Urut per kategori* : dimana kita dapat mengurutkan katalog berdasar kategori buku, majalah dan komik.

9. Tampilan Halaman Detail Katalog

Tampilan halaman detail katalog buku, kita dapat melihat kode buku, pengarang, harga sewa dan deskripsi



Gambar 5.13 *Tampilan Halaman Detail Katalog*

10. Tampilan Halaman Berita

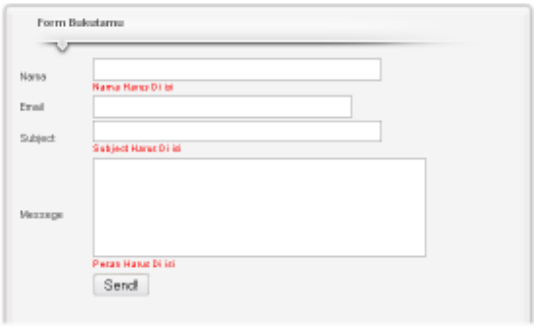
Tam,pilan halaman berita yang dapat diakses oleh user biasa maupun member



Gambar 5.14 *Tampilan Halaman Berita*

11. Tampilan Halaman Kesalahan Pengisian Buku Tamu

Tampilan kesalahan apabila salah satu field terjadi kesalahan penginputan atau juga bisa terjadi karena data tidak diisi



Gambar 5.15 *Tampilan Halaman Kesalahan Penginputan Buku Tamu*

5.1.4.2 Tampilan Halaman Administrator

Suatu form dimana jika seorang Admin ingin melakukan pengeditan dan pemesukan data dalam data base, maka administrator diwajibkan mengisi login Admin

1. **Tampilan Login Administrator**

   Tampilan halamana login administrator dimana ada dua field yaitu userID dan password, setelah mengisikan userID dan password klik tombol login untuk masuk ke halaman administrator



Gambar 5.16 *Tampilan Halaman Login Admin*

2. **Tampilan Gagal Login Administrator**

   Tampilan pada saat entry form login administeror salah dalam pengisian mungkin baik userID ataupun password



Gambar 5.17 *Tampilan Halaman Gagal Login Admin*

3. **Tampilan Sukses Login Administrator**

   Berikut adalah tampilan sukses login administrator dimana disediakan menu-menu seperti :

   1. *Log out :* digunakan untuk keluar dari sistem
   2. *Add new member* : digunakan untuk menambah member baru
   3. *Member list :* digunakan untuk melihat list member
   4. *Add kategori buku :* digunakan untuk menambah kategori buku

5. *Master buku :* master buku digunakan untuk melihat data buku
6. *Berita :* digunakan untuk entry berita yang akan dipublikasikan
7. *Buku tamu :* digunakan untuk melihat siapa yang sudah berkunjung ke website Ai Read.
8. *Peminjaman :* digunakan untuk pencatatan transaksi peminjaman
9. *Pengembalian :* digunakan untuk mencek peminjaman dan mecatat pengembalian buku
10. *Laporan :* digunakan untuk melihat laporan per hari, minggu ataupun bulan



Gambar 5.18 *Tampilan Halaman Sukses Login Admin*

4. Tampilan Input Member

Tampilan *add new member*, dalam pengisia member baru maka userID dan password harus diisi, sebab nantinya akan digunakan untuk login member agar mendapat fasilitas yang lebih



Gambar 5.19 *Tampilan Halaman Input Member*

5. Tampilan Edit Member

Tampilan edit member



Gambar 5.20 *Tampilan Halaman Edit Member*

6. Tampilan Detail Member

Berikut adalah tampilan detail member, dimana kita dapat melihat foto member dan deskripsi member lainnya.

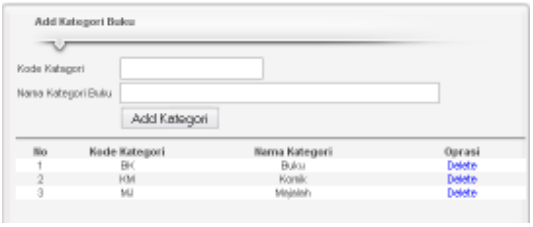


Gambar 5.21 *Tampilan Halaman Detail Member*

7. Tampilan Input Kategori Buku

Tampilan inputan kategori buku, pada halaman ini terdapat field seperti berikut :

1. *Kode kategori* : kita dapat mengisikan kode kategori yang akan kita tambahkan
2. *Nama kategori* : kita dapat mengisikan nama kategori yang berkaitan dengan kode kategori
3. *Button add kategori* : setelah pengisian kode kategori dan nama katergori diisikan mana tekan button add kategori untuk proses penyimpanan

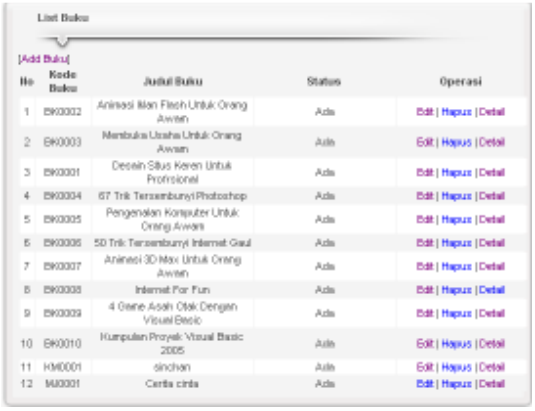


Gambar 5.22 *Tampilan Halaman Input Kategori Buku*

8. Tampilan List Master Buku

Ini adalah tampilan halaman list buku dimana administrator dapt menggunakan operasi seperti berikut :

1. *Edit* : untuk mengedit buku
2. *Hapus* : untuk proses menghapus buku
3. *Detail* : untuk melihat detail buku



Gambar 5.23 *Tampilan Halaman List Master Buku*

9. Tampilan Input Master Buku

Merupakan tampilan input buku :

1. *Kategori* : untuk memilih kategoro yang akan di masukan
2. *Kode buku* : secara otomatis akan ada sendiri
3. *Judul* : diisi judul buku
4. *pengarang* : diisi pengarang buku
5. *harga beli* : diisi harga beli buku
6. *harga sewa* : diisi harga sewa buku
7. *lama pinjam* : diisi lama peminjaman buku
8. *status buku* : menunjukan buku itu ada atau tidak

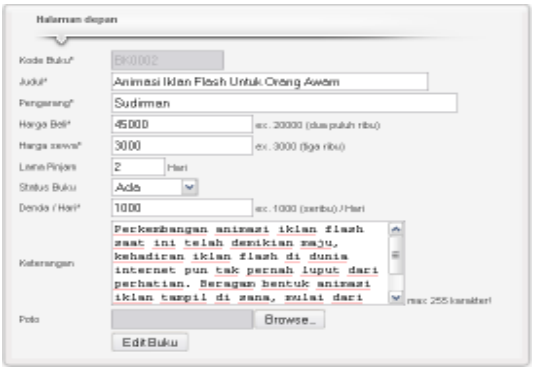
9. *denda* : diisi besarnya denda
10. *deskripsi* : keterangan detail buku
11. *gambar* : field untuk menginputkan gambar buku
12. *button browse* : untuk mencari lokasi gambar buku yang akan dimasukan
13. *button add buku* : proses penyimpanan



Gambar 5.24 *Tampilan Halaman Input Master Buku*

10. Tampilan Edit Master Buku

Berikut adalah gambari tampilan halaman edit master buku



Gambar 5.25 *Tampilan Halaman Edit Master Buku*

11. Tampilan Detail Master Buku

Tampilan detail buku, dimana kita dapat melihat gambar buku, deskripsi buku, pengarang, harga sewa dll



Gambar 5.26 *Tampilan Halaman Detail Master Buku*

12. Tampilan Form Peminjaman Buku

Tampilan halaman form peminjaman buku, dimana ada beberapa fasilitas seperti berikut :

1. *Lihat data peminjaman online* : untuk melihat data buku yang dipinjam secara online
2. *Cek history peminjaman* : untuk meliahat history pesan peminjaman buku online
3. *Cek* : digunakan untuk mengecek member apakah aktif atau tidak
4. *Button check* : untuk mengecek buku yang akan dipinjam apakah ada atau tidak
5. *Button add* : digunakan untuk menyimpan sementara buku yang akan dipinjam
6. *Button simpan* : digunakan untuk menyimpan buku yang akan dipinjam



Gambar 5.27 *Tampilan Halaman Form Peminjaman Buku*

13. Tampilan Pemilahan Buku Yang Akan Dipinjam

Berikut adalah gambar form pemilahan buku yang akan dipinjam, kita dapat mengetahui buku yang akan dipinjam



Gambar 5.28 *Tampilan Pemilihan Buku Yang Akan Dipinjam*

14. Tampilan Buku Yang Dipinjam Sudah Pasti

Gambar form halaman buku yang sudah pasti akan dipinjam



Gambar 5.29 *Tampilan Buku Yang Akan Dipinjam Sudah Pasti*

15. Tampilan Validasi Peminjaman Buku

Tampilan diman buku yang kita masukan statusnya sudah dipinja, maka proses tidak bisa berlanjut ke proses berikutnya



Gambar 5.30 *Tampilan Halaman Validasi Peminjaman Buku*

16. Tampilan Pengembalian Buku

Ini adalah tampilan pengembalian buku, kita dapat mengecek berdasarkan idmember, jika member tersebut pernah meminjam maka buku-buku yang dipinjam akan tampil, dan jika ingin dikembalikan maka klik operasi kembalikan



Gambar 5.31 *Tampilan Halaman Pengembalian Buku*

17. Tampilan Main Laporan

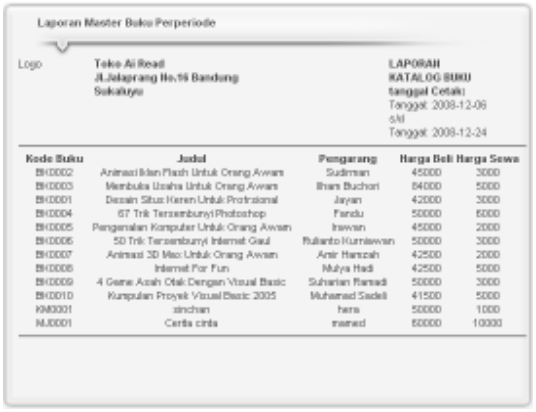
Tampilan halaman main laporan, kita dapat melihat laporan per hari, minggu dan bulan. Button show digunakan untuk melihat list laporan



Gambar 5.32 *Tampilan Halaman Main Laporan*

18. Tampilan Laporan Buku

Berikut adalah tampilan halaman laporan buku


Laporan Master Buku Perperiode

Logo

Toko Ai Read
Jl.Jelaprang No.16 Bandung
Sukabuyu

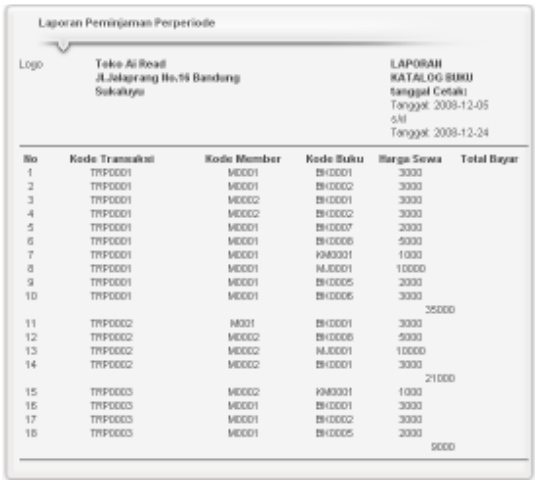
LAPORAN
KATALOG BUKU
tanggal Cetak:
Tanggal: 2008-12-06
s/d
Tanggal: 2008-12-24

Kode Buku	Judul	Pengarang	Harga Beli	Harga Sewa
BKD002	Animasi Iklan Flash Untuk Orang Awam	Sudirman	45000	3000
BKD003	Membuka Usaha Untuk Orang Awam	Ilham Buchori	84000	5000
BKD001	Desain Situs Keren Untuk Profesional	Jayan	42000	3000
BKD004	67 Trik Tersembunyi Photoshop	Fandu	50000	6000
BKD005	Pengenalan Komputer Untuk Orang Awam	Irawan	45000	2000
BKD006	50 Trik Tersembunyi Internet Gaul	Rulianto Kurniawan	50000	3000
BKD007	Animasi 3D Max Untuk Orang Awam	Amir Hamzah	42500	2000
BKD008	Internet For Fun	Mulya Hadi	42500	5000
BKD009	4 Game Asah Otak Dengan Visual Basic	Suharian Ramadi	50000	3000
BKD010	Kumpulan Proyek Visual Basic 2005	Muhamad Sadeli	41500	5000
KM0001	sinchan	hera	50000	1000
MJD001	Cerita cinta	mamad	60000	10000

Gambar 5.33 *Tampilan Laporan Buku*

19. Tampilan Laporan Peminjaman Buku

Berikut adalah tampilan halaman laporan peminjaman buku


Laporan Peminjaman Perperiode

Logo

Toko Ai Read
Jl.Jelaprang No.16 Bandung
Sukabuyu

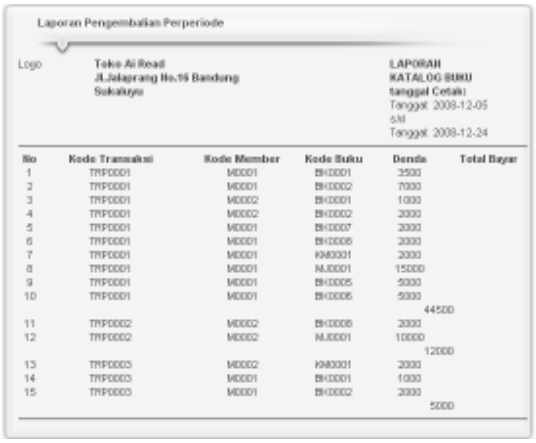
LAPORAN
KATALOG BUKU
tanggal Cetak:
Tanggal: 2008-12-06
s/d
Tanggal: 2008-12-24

No	Kode Transaksi	Kode Member	Kode Buku	Harga Sewa	Total Bayar
1	TRP0001	MD001	BKD001	3000	
2	TRP0001	MD001	BKD002	3000	
3	TRP0001	MD002	BKD001	3000	
4	TRP0001	MD002	BKD002	3000	
5	TRP0001	MD001	BKD007	2000	
6	TRP0001	MD001	BKD008	5000	
7	TRP0001	MD001	KM0001	1000	
8	TRP0001	MD001	MJD001	10000	
9	TRP0001	MD001	BKD005	2000	
10	TRP0001	MD001	BKD006	3000	
					35000
11	TRP0002	M001	BKD001	3000	
12	TRP0002	MD002	BKD008	5000	
13	TRP0002	MD002	MJD001	10000	
14	TRP0002	MD002	BKD001	3000	
					21000
15	TRP0003	MD002	KM0001	1000	
16	TRP0003	MD001	BKD001	3000	
17	TRP0003	MD001	BKD002	3000	
18	TRP0003	MD001	BKD005	2000	
					9000

Gambar 5.34 *Tampilan Laporan Peminjaman Buku*

20. Tampilan Laporan Pengembalian Buku

Berikut adalah tampilan halaman pengembalian buku



Laporan Pengembalian Perperiode

Logo

Toko Al Read
Jl.Jelaprang No.16 Bandung
Sukakuyu

LAPORAH
KATALOG BUKU
tanggal Cetak:
Tanggal: 2008-12-05
s/d
Tanggal: 2008-12-24

No	Kode Transaksi	Kode Member	Kode Buku	Denda	Total Bayar
1	TRPD001	MD001	BHD001	3500	
2	TRPD001	MD001	BHD002	7000	
3	TRPD001	MD002	BHD001	1000	
4	TRPD001	MD002	BHD002	2000	
5	TRPD001	MD001	BHD007	2000	
6	TRPD001	MD001	BHD008	2000	
7	TRPD001	MD001	KM0001	2000	
8	TRPD001	MD001	MJD001	15000	
9	TRPD001	MD001	BHD005	5000	
10	TRPD001	MD001	BHD006	5000	
					44500
11	TRPD002	MD002	BHD006	2000	
12	TRPD002	MD002	MJD001	10000	
					12000
13	TRPD003	MD002	KM0001	2000	
14	TRPD003	MD001	BHD001	1000	
15	TRPD003	MD001	BHD002	2000	
					5000

Gambar 5.35 *Tampilan Laporan Pegngembalian Buku*

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari pelaksanaan Tugas Akhir ini adalah dengan adanya Perangkat Aplikasi sebagai media promosi dan penyebaran informasi yaitu dengan media website serta aplikasi peminjaman dan pengembalian buku dapat membantu toko Ai Read untuk mengembangkan usahanya serta mendapatkan kemudahan khususnya masyarakat dalam mendapatkan kemudahan untuk :

a. Bagi pihak Ai Read, dengan dibuatnya aplikasi peminjaman dan pengembalian buku mengguanakan php, proses pencatatan peminjaman dan pengembalian buku menjadi semakin mudah serta data saling terintegrasi dengan data yang di onlinekan.
b. Mendapatkan informasi dengan cepat mengenai buku-buku terbaru serta berita-berita terbaru berkaitan dengan Ai Read
c. Adanya media untuk saling berinteraksi dan bertukar pikiran antara toko bacaan dengan masyarakat luas.

6.2 Saran

Hasil perancangan sistem informasi untuk mempromosikan buku-buku terbaru, berita-berita terbaru melalui website ini masih dimungkinkan untuk dilakukan pengembangan, sehingga hasil yang diperoleh dapat lebih baik dan bisa lebih optimal.

Untuk mendapatkan hasil yang lebih baik, pada program aplikasi ini masih dapat dikembangkan dengan cara menambahkan pendaftaran serta peminjaman dan pengembalian buku yang dapat dilakukan secara online.