BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Objek yang dijadikan penelitian didalam penulisan skripsi ini adalah Perpustakaan SMA Karya Pembangunan 2 Bandung. Berikut adalah sejarah singkat SMA karya Pembangunan 2 serta Perpustakaannya , visi dan misi, struktur organisasi dan deskripsi tugas yang penulis peroleh dari tempat yang penulis teliti.

3.1.1. Sejarah Singkat Sekolah

SMA Karya Pembangunan 2 adalah sekolah menengah umum yang didirikan pada tahun 1977 dan berada dibawah naungan Yayasan Pembina Pendidikan Karya Pembangunan. Dalam perkembangannya, sekolah ini mengawali kegiatannya dengan menumpang di sekolah SD Impres Andir kidul, baru pada tahun 1979 SMA Karya Pembangunan 2 telah menempati gedung sendiri yang terdiri dari 4 lokasi dan sekarang telah mempunyai 21 lokasi untuk kelas serta 2 lokasi yang sedang dibangun, perpustakaan dan Lab IPA dan Lab. Komputer. Perubahan kearah perbaikan dan penyempurnaan terus diupayakan dan dilaksakan dari awal sampai sekarang untuk mewujudkan sebuah lembaga pendidikan yang betul-betul bisa memberikan bekal pendidikan yang paripurna bagi siswanya. Perubahan – perubahan yang sedang dan akan dilaksakan adalah penataan disiplin sekolah, penataan tata tertib, penyusunan program kerja, pengembangan sarana/prasarana dan pengembangan penghayatan dan pengamalan

ajaran agama. Hal hal tersebut dilaksanakan dengan berorientasi wawasan lingkungan, nilai-nilai agama dan berdasarkan Sistem Management Berbasis Sekolah.

3.1.2 Lokasi Sekolah

SMA Karya Pembangunan 2 Bandung terletak di Jl. AH. Nasution 25A Kelurahan Pasir Endah, Kecamatan Ujung Berung, Kota Bandung, Provinsi Jawa Barat dengan luas sekitar 2050 meter persegi.

3.1.3 Visi dan Misi Sekolah

Visi sekolah adalah berbunyi SMA Karya Pembangunan 2 merupakan wadah pembentukan Karakter manusia yang berkualitas tinggi. Visi tersebut dimaksudkan bahwa seluruh Masyarakat sekolah diharapkan “menempatkan” diri sebagai manusia-manusia yang bertaqwa, berwawasan ibadah kepada Allah swt, cerdas, kreatif, inovatif dan senang terhadap ilmu pengetahuan.

Selanjutanya Visi tersebut dituangkan kepada “misi” yang berbunyi SMA Karya Pembangunan 2 siap mendidik, mengajar, membimbing seluruh Civitas dengan suasana bergembira,berpikir positif dan kritis, belajar, bermunajat kepad Allah, berkreasi dan berprestasi. Pola-pola tersebutdiharapkan dapat mengarahkan proses pendidikan yang kondusif.

Visi : MAHABBAH (Cinta kepada Ilahi)

* Mandiri dalam berjuang mencapai cita-cita
* Aktif, inisiatif , kreatif dan positif dalam menjalankan program-program
* Hidup yang bermamfaat bagi agama, bangsa dan Negara
* Adaptif dengan masyarakat dan lingkungan sekolah

* Berahlaq mulia sesuai tuntutan agama

* Aman, nyaman, tertib, dan menyenangkan

* Hijau, segar, berbunga

Misi Sekolah :

* Mewujudkan sekolah sebagai tempat melahirkan manusia yang bertaqwa kepada Tuhan Yang Maha Esa

* Menciptakan manusia yang berkarakter kualitas tinggi

* Menciptakan manusia yang cager, bener, pinter, bageur, dan singer (sehat, taqwa, pinter, baik, dan kreatif)

* Menanamkan budaya disiplin diri yang tinggi

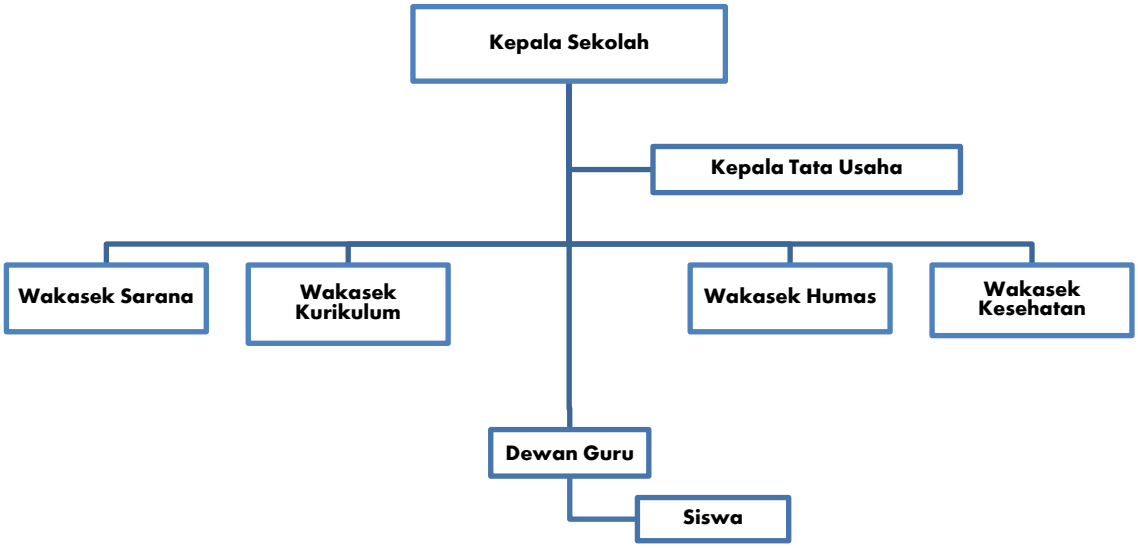
* Mewujudkan lingkungan yang kompetitif positif bagi warga sekolah

* Menanamkan kesadaran akan kewajiban tugasnya masing-masing bagi setiap warga sekolah

* Menciptakan lulusan yang siap bersaing untuk melnjutkan ke jenjang pendidikan yang lebih tinggi

3.1.4 Struktur Organisasi Sekolah

Berikut Ini adalah gambaran Umum Struktur Organisasi yang ada di sekolah SMA Karya Pembangunan 2 Bandung.



Gambar 3.1. Struktur Organisasi

(Sumber : SMA Karya Pembangunan 2 Bandung)

3.1.5 Deskripsi Tugas

Deskripsi tugas adalah rincian yang menunjukan fungsi dan tugas-tugas yang harus dilakukan. Deskripsi tugas yang ada di SMA Karya Pembangunan 2 yaitu;

1. Kepala Sekolah: bertugas memimpin dan mengkoordinasikan semua pelaksanaan rencana kerja harian, mingguan, bulanan catur wulan dan tahunan. Mengadakan hubungan dan kerjasama dengan pejabat-pejabat resmi setempat dalam usaha pembinaan sekolah.

2. Kepala Tata Usaha, bertugas mengkoordinasikan seluruh kegiatan yang berkaitan dengan administrasi sekolah, meliputi penyusunan program tahunan, kepegawaian, keuangan, pelaporan, inventaris dan kesiswaan.
3. Wakil Kepala Sekolah Bidang Sarana dan Prasarana Pendidikan, bertugas mengkoordinasikan segala kegiatan yang berkaitan dengan pengadaan, pemeliharaan dan penghapusan barang-barang inventaris/non inventaris baik fisik maupun non-fisik milik sekolah.
4. Wakil Kepala Sekolah Bidang Kurikulum, bertugas membuat perencanaan dan mengkoordinasikan pembagian tugas guru-guru per catur wulan, merekap daya serap dan target pencapaian kurikulum per catur wulan dan per tahun pelajaran, serta segala kegiatan yang berhubungan dengan urusan kurikulum dan pengajaran bidang intra-kurikuler.
5. Wakil Kepala Sekolah Budang Humas, bertugas untuk Hubungan dengan Pihak Luar ,menerima tamu luar yang datang kedalam lingkungan internal Sekolah, Mengatur dan menyelenggarakan hubungan sekolah dengan orang tua/wali siswa, Membina hubungan sekolah sengan POMG/Dewan sekolah, Menyusun laporan hubungan sekolah dengan masyarakat secara berkal, dll.
6. Wakil Kepala Sekolah Bidang Kesehatan tugasnya yaitu bertugas di UKS kesehatan yang berkaitan dengan pengobatan apabila ada yang sakit dan lain- lain.
7. Guru bertugas mengajar siswa
8. Siswa tugasnya belajar atau menuntut ilmu di sekolah

3.2 Sejarah Perpustakaan

Perpustakaan SMA Karya Pembangunan 2 berdirinya sekitar tahun 80an yaitu dua tahun setelah sekolah SMA Karya Pembangunan 2 didirikan pada tahun 1977 dan setelah SMA Karya Pembangunan 2 menempati gedung sendiri saat ini yang sebelumnya masih menumpang di SD Andir kidul.. Perpustakaan SMA Karya Pembangunan 2 telah banyak mengalami perombakan seiring dengan berkembangnya sekolah SMA Karya Pembangunan 2. Buku yang tersedia hingga saat ini yang ada di perpustakaan yaitu sekitar 1500 buku. Sayangnya jumlah buku yang makin banyak dan bertambah tidak diiringi dengan banyaknya jumlah pengunjung ke perpustakaan ini.maka dari itu diperlukan sesuatu untuk menarik minat pengunjung.

Peminjaman buku di perpustakaan SMA Karya Pembanguna 2 Bandung ini tidak dipungut biaya untuk setiap peminjaman buku, dan lamanya peminjaman buku selama tiga hari peminjaman, dan jika pengunjung terlambat dalam pengembalian buku maka perpustakaan SMA Karya Pembanguna 2 Bandung memberikan denda sebesar Rp.500,untuk satu hari keterlambatannya.

3.2.1. Visi dan Misi Perpustakaan

Adapun visi dan misi dari Perpustakaan SMA Karya Pembangunan 2, yaitu :

3.2.1.1 Visi

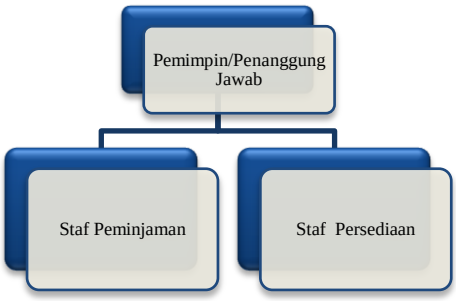
Visi Perpustakaan SMA Karya Pembangunan 2 Bandung adalah ” Perpustakaan sebagai pusat informasi ilmu pengetahuan, dengan mendukung kegiatan belajar

mengajar, serta menyediakan fasilitas informasi yang memungkinkan pengguna mengakses informasi secara nyaman diperpustakaan.”

3.2.1.2. Misi

1. Menyediakan informasi - informasi berkualitas untuk mendukung kegiatan pendidikan disekolah.
2. Menyediakan tempat yang nyaman bagi pengguna perpustakaan.
3. Menyediakan layanan khusus bagi pengguna perpustakaan.
4. Menyelenggarakan pendidikan pemakai bagi pengguna.

3.2.2. Struktur Organisasi Perpustakaan



Gambar 3.2 Struktur Organisasi

(Sumber : SMA Karya Pembangunan 2 Bandung)

3.2.3. Deskripsi Tugas

Deskripsi tugas adalah rincian yang menunjukan fungsi dan tugas-tugas yang harus dilakukan. Deskripsi tugas yang ada di perpustakaan SMA Karya Pembangunan 2 yaitu;

1. Pemimpin atau Penanggung Jawab adalah sebagai penanggungjawab perpustakaan dan sebagai orang yang menerima hasil laporan transaksi yang terjadi diperpustakaan SMA Karya Pembangunan 2 Bandung.

2. Staf Peminjaman Bagian peminjaman adalah orang yang mempunyai tanggung jawab dalam mengelola laporan keuangan, transaksi peminjaman buku, dan hal - hal administrasi lainnya.
3. Staf Persediaan barang (buku) mempunyai wewenang mencatat dan melaporkan persediaan buku kepada pemimpin.

3.3. Metode Pengumpulan Data

Pengumpulan data yang lengkap berhubungan dengan penelitian ini dilakukan dengan survei. Data yang diperlukan adalah data primer dan data sekunder. Kedua jenis data tersebut akan dikumpulkan melalui teknik sebagai berikut.

3.3.1. Sumber Data Primer

Sumber data / informasi dalam penelitian ini berdasarkan kepada jenis data yang diperlukan. Data primer yaitu data yang diperoleh dari responden secara langsung yang dikumpulkan melalui survei lapangan dengan menggunakan teknik pengumpulan data tertentu yang dibuat secara khusus untuk itu.

a. Wawancara

Dilakukan dengan pihak - pihak terkait yaitu bagian peminjaman, bagian persediaan buku dan pemimpin perpustakaan dengan harapan dapat melengkapi data data yang diperlukan dalam penelitian ini. Dalam penelitian ini, ada beberapa pertimbangan untuk menentukan seseorang sebagai sumber informasi. Dalam menentukan sumber informasi pertimbangannya yaitu.

1. Keakuratan dan validitas informasi yang diperoleh. Berdasarkan hal ini maka jumlah pemberi informasi sangat tergantung pada hasil yang

dikehendaki. Bila mereka yang menjadi sumber pemberi informasi adalah orangorang yang benar benar menguasi masalah yang diteliti, maka informasi tersebut dijadikan bahan analisis.

2. Jumlah sumber pemberi informasi sangat bergantung pada pencapaian tujuan penelitian, artinya bila masalah - masalah dalam penelitian yang diajukan sudah terjawab dari 5 sumber informasi, maka jumlah tersebut adalah jumlah yang tepat.
3. Peneliti diberi kewenangan dalam menentukan siapa saja yang menjadi sumber pemberi informasi, tidak terpengaruh jabatan seseorang. Bisa saja peneliti membuang pemberi sumber informasi yang dianggap tidak layak.

b. Observasi

Peneliti langsung mengunjungi lokasi penelitian ke bagian peminjaman dan bagian persediaan buku yang dijadikan objek penelitian, hal ini dilakukan untuk melihat dari dekat masalahmasalah yang berhubungan dengan pokok bahasan yang diperlukan dalam penelitian ini.

3.3.2 Sumber Data Sekunder *(Dokumentasi)*

Sumber data sekunder diperoleh dengan cara melakukan kajian kepustakaan, yaitu mempelajari buku buku literatur dan informasi melalui internet yang relevan dengan variabel yang diteliti, serta laporanlaporan triwulan, tahunan tentang perkembangan. Selain itu sumber data juga diperoleh dari instansi terkait yang ada hubungannya dengan variabel yang digunakan dalam penelitian ini.

Dokumentasi dilakukan dengan menelaah data sekunder yang digunakan untuk menunjang, melengkapi dan menyempurnakan data primer yang diperoleh dari dokumentasi perusahaan, kepustakaan, internet, serta hasil penelitian sebelumnya yang berhubungan dengan masalah dalam penelitian ini. Beberapa dokumen yang telah diberikan seperti:

1. Data Buku yang Tersedia
2. Data Peminjaman Buku
3. Kartu Perpustakaan

3.4 Metode Pendekatan / Pengembangan Sistem

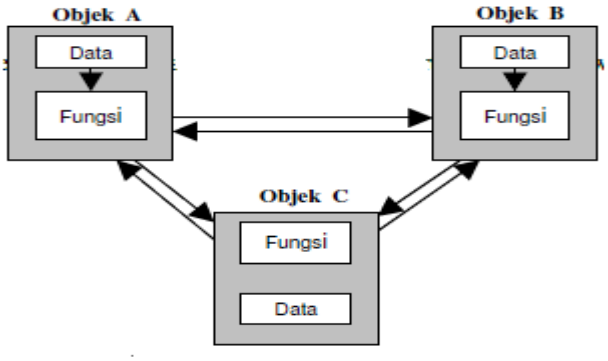
Metode pendekatan dan pengembangan suatu sistem harus diperhatikan dalam mengerjakan suatu proses pembuatan perangkat lunak, karena dampak daripendekatan dan pengembangan ini akan mempengaruhi terhadap kesuluruhan sistem.

3.4.1 Metode Pendekatan Sistem

Menurut ***Adi Nugroho*** 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. Sturktur data dirancang dan menjadi karakteristik dari objek objek.
4. Fungsi fungsi yang mengoperasikan data tergabung dalam suatu objek yang sama.
5. Data tersembunyi dan terlindung dari prosedur/fungsi yang ada di luar.

6. Objekobjek dapat saling berkomunikasi dengan saling mengirim *message* (pesan) satu sama lain.
7. Pendekatan adalah dari bawah ke atas, seperti gambar dibawah ini :



Gambar 3.3 Pengorganisasian Data Serta Fungsi Pendekatan Berorientasi Objek (Adi, Nugroho." *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*",[ADI05]

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

Analisis adalah proses mendekomposisi atau menginvestigasi *domain problem* atau *requirements* menjadi bagianbagian yang bisa dimengerti. Proses analisis lebih menekankan kepada pertanyaan seperti apakah sistem yang akan dikerjakan, bagaimana sistem tersebut digunakan seharihari oleh user.

Perancangan adalah melakukan sintesa model *software* dari masalah yang di analisis, menterjemahkan *domain* model kepada *software* model sehingga bisa dimengerti dan dikomunikasikan kepada anggota tim lain dan juga para profesional *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. Terlihat bahwa analisis dilakukan sebelum perancangan. Analisis berorientasi objek menekankan kepada menemukan dan menjelaskan objekobjek dan interelasinya yang menjadi bagian atau pemain dalam sebuah sistem. Perancangan berorientasi objek memodelkan objekobjek yang diidentifikasi pada proses analisis dan interaksi antar objek ke dalam model *software*. Proses selanjutnya bisa mengidentifikasi atribut dan operasi objekobjek lainnya.

3.4.2 Metode Pengembangan Sistem

Metode pengembangan perangkat lunak yang digunakan dalam penyusunan skripsi ini adalah prototipe, yang dimulai dengan pengumpulan kebutuhan, kemudian pengembang dan pemakai bertemu dan mendefinisikan objektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan area garis besar dimana definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat. Perancangan kilat membawa kepada konstruksi sebuah prototipe. Prototipe tersebut dievaluasi dan dipakai untuk menyaring kebutuhan pengembang perangkat lunak. Interaksi terjadi pada saat prototipe dirancang untuk memenuhi kebutuhan pelanggan dan pada saat yang sama memungkinkan pengembang untuk secara lebih baik memahami apa yang harus dilakukannya.

Secara ideal prototipe berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. Bila prototipe yang sedang bekerja

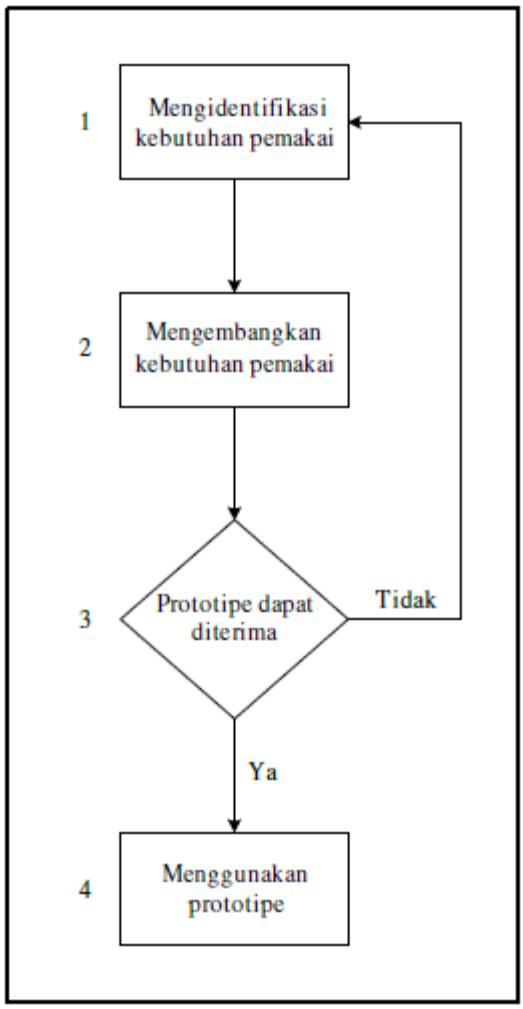
dibangun, pengembang harus mempergunakan fragmenfragmen program yang ada sehingga memungkinkan program yang bekerja untuk dimunculkan secara cepat.

Dalam pembangunan sistem, representasi pengetahuan ini terlebih dahulu dilakukan peninjauan keinginan persyaratan objektif para pemakai terhadap perangkat lunak yang akan dirancang. Setelah tahapan tersebut kemudian dilanjutkan dengan perancangan perangkat lunak yang memungkinkan untuk diperbaharui kembali melalui evaluasi program aplikasi yang dibangun dari perspektif pemakai dengan metode sampling.

Alasan mengapa penulis memakai metode prototipe ini adalah karena metode ini cukup efektif dalam mendapatkan kebutuhan secara cepat yang diperlukan untuk merancang suatu aplikasi khususnya aplikasi berbasis *web site*, serta dapat disetujui pelanggan ataupun pembuat perangkat lunak itu sendiri. Walaupun pada umumnya prototipe ini akan dihilangkan dan dibuat perangkat yang sebenarnya. Dengan prototipe ini juga, ketika diperlihatkan *working version*nya pelanggan bisa langsung merasakan seakanakan itu adalah sistem yang sebenarnya.

Prototyping adalah proses iteratif dalam pengembangan sistem dimana kebutuhan diubah ke dalam sistem yang bekerja *(working system)* yang secara terus menerus diperbaiki melalui kerjasama antara pengguna dan analis.

Adapun langkahlangkah pengembangan sistem metode prototipe dapat dilihat pada gambar 3.4 di bawah ini.



Gambar 3.4. Tahapan Pengembangan Prototype

(McLeod, Raymond, ***Management Information System***,)

Uraian tahapan pengembangan perangkat lunak menggunakan paradigm prototipe dapat dijelaskan sebagai berikut :

1) Mengidentifikasi kebutuhan pemakai Pada tahap ini, analisis sistem akan melakukan studi kelayakan dan studi terhadap kebutuhan pemakai baik yang meliputi model *interface,* teknik prosedural, maupun dalam teknologi yang akan digunakan. Pada tahap ini penulis melakukan penelitian terhadap sistem yang sedang berjalan dan menganalisis kebutuhan

pemakai, apa saja yang kirakira akan menjadi kebutuhan pemakai dalam menjalankan aplikasi berbasis *web* yang akan penulis rancang.

2) Mengembangkan kebutuhan pemakai

Pada tahap kedua ini, analis sistem bekerjasama dengan programmer mengembangkan prototipe sistem untuk memperlihatkan kepada pemesan pemodelan atau pihak yang membutuhkan rancangan aplikasi sistem yang akan dibangun. Seperti kegiatan mendefinisikan masalah yang mencakup pendefinisian *input*, proses dan *output* dari sistem yang akan dibangun serta memilih alternatif yang akan dibangun sesuai dengan kebutuhan pemakai. Mengenai implementasi alternatif yang telah dipilih pada tahap ini penulis mulai membuat desain sistem sesuai dengan kebutuhan pemakai yang telah dikembangkan.

3) Menentukan Prototype

Prototipe yang dibangun sudah mencapai tahap apakah akan diterima atau tidak oleh pihak pemakai aplikasi. Analisis sistem pada tahap ini akan mendeteksi dan mengidentifikasi sejauh mana pemodelan yang dibuat dapat diterima oleh pengguna. Perbaikanperbaikan apa yang diinginkan pengguna atau bahkan harus merubah secara keseluruhan.

4) Menggunakan Prototype

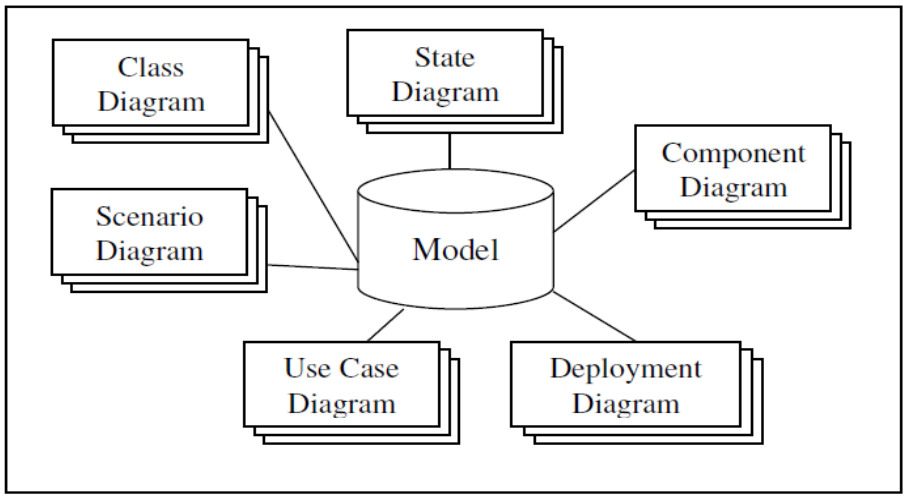
Pada tahap akhir ini, analis sistem akan menyerahkan kepada programmer untuk mengimplementasikan pemodelan yang sudah dibuat menjadi suatu sistem. Penulis akan melakukan beberapa test, seperti memasukan

sejumlah data ke dalam sistem dan dilihat hasilnya serta cara pemrosesan yang dilakukan oleh aplikasi sistem berbasis *web site* ini.

Dengan langkah langkah seperti yang disebutkan diatas diharapkan system perpustakaan *online* dan sistem otomasi perpustakaan yang dibangun dapat berjalan sesuai dengan yang dikehendaki. Program yang digunakan dapat berupa sebuah web site perpustakaan yang kemudian dikembangkan dan didesign sesuai dengan keinginan perpustakaan maupun program yang dikembangkan secara mandiri oleh tim teknologi informasi perpustakaan.

3.4.3 Alat Bantu Pemodelan

Pemodelan adalah sebuah aktivitas yang telah berlangsung bertahuntahun dalam pengembangan perangkat lunak. Sebuah model masih diperlukan untuk menulis sebuah aplikasi baik menggunakan bahasa pemograman yang paling sederhana sampai bahasa yang kompleks. Pemodelan sangatlah diperlukan karena membantu tim pengembang untuk visualisasi, spesifikasi, dan mendokumentasikan struktur dan perilaku dari arsitektur sistem. Dengan menggunakan pemodelan yang baku seperti UML, anggota tim pengembang tidak akan ada keraguan dalam mengkomunikasikan keputusan mereka satu dengan yang lainnya.



Gambar 3.5 Model Diagram

(Munawar, "Pemodelan Visual", [MUN05])

3.4.3.1 Konsep Dasar UML (*Unified Modelling Language*)

UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT), dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu :

1. Identifikasi kelaskelas dan obyekobyek

2. Identifikasi semantic dari hubungan obyek dan kelas tersebut

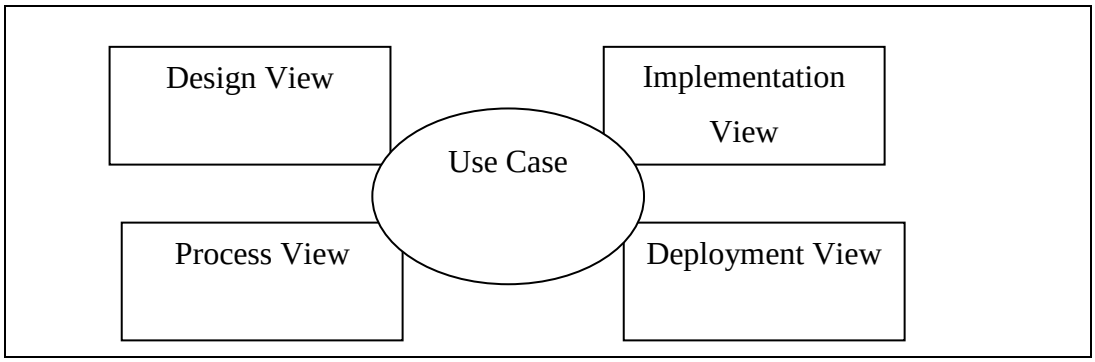
3. Perincian interface

4. Implementasi

Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan *entityrelationship*.

Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih member penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, design dan implementasi, dan model pengujian (*test model*). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

UML dibangun atas model 4+1 *view*. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam 5 *view* dimana salah satu diantaranya *use case view*. *Use case view* ini memegang peran khusus untuk mengintegrasikan *content* ke *view* yang lain.



Gambar 3.6 Model 4+1 *View*

(Munawar, "Pemodelan Visual", [MUN05])

Kelima view terserbut tidak berhubungan dengan diagram yang dideskripsikan di UML. Setiap view berhubungan dengan perspektif tertentu dimana sistem akan diuji. View yang berbeda akan menekankan pada aspek yang berbeda dari sistem yang mewakili ketertarikan sekelompok stakeholder tertentu. Penjelasan lengkap tentang sistem bisa dibentuk dengan menggabungkan informasiinformasi yang ada pada kelima view tersebut.

Use case view mendefinisikan perilaku eksternal sistem. Hal ini menjadi daya tarik bagi end user, analis dan tester. Pandangan ini mendefinisikan kebutuhan sistem karena mengandung semua view yang lain yang mendeskripsikan aspek - aspek tertentu dan rancangan sistem. Itulah sebabnya use case view menjadi pusat peran dan sering dikatakan yang mendrive proses pengembangan perangkat lunak.

Design view mendeskripsikan struktur logika yang mendukung fungsifungsi yang dibutuhkan di use case. Design view ini berisi definisi komponen program, classclass utama bersamasama dengan spesifikasi data, perilaku dan interaksiny. Informasi yang terkandung di view ini menjadi perhatian para programmer karena menjelaskan secara detil bagaimana fungsionalitas system akan diimplementasikan.

Implementation view menjelaskan komponenkomponen fisik dari system yang akan dibangun. Hal ini berbeda dengan komponen logic yang dideskripsikan pada design view. Termasuk disini diantaranya file exe, library dan database. Informasi yang ada di view ini relefan dengan aktifitasaktifitas seperti manajemen konfigurasi dan integrasi sistem.

Process view berhubungan dengan halhal yang berkaitan dengan *concurrency* didalam sistem. Sedangkan deployment view menjelaskan bagaimana komponenkomponen fisik didistribusikan ke lingkungan fisik seperti jaringan computer dimana sisetm akan dijalankan. Kedua view ini menunjukan kebutuhan non fungsional dari sistem seperti toleransi kesalahan dan halhal yang berhubungan dengan kinerja.

3.4.3.2 Model – Model Diagram UML

UML terdiri atas banyak elemenelemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemenelemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masingmasing 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 lainnya. Diagram di klasifikasikan berdasarkan tiga jenis diagram sebagai berikut.

Struktural Diagram	1. *Class Diagram* 2. *Composite Structure Diagram* 3. *Object Diagram* 4. *Component Diagram* 5. *Deployment Diagram* 6. *Pacakage Diagram*
Behavioral Diagram	1. *Use Case Diagram* 2. *Activity Diagram* 3. *State Machine Diagram*
Interaction Diagram	1. *Sequence Diagram* 2. *Communication Diagram* 3. *Interaction Overview Diagram* 4. *Timing Diagram*

Gambar 3.7 Klasifikasi Jenis Diagram UML
(Martin, Flower. "*UML Distilled, Edisi Tiga,"*[MAR05])

1 Class Diagram

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/properti) 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 lainlain. 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 objek 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 anakanak 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*.

A. Relasi Pada Diagram *Class*

Relasi atau *relationship* merupakan keterhubungan antar kelas yang muncul pada saat sebuah kelas berinteraksi dengan kelaskelas lainnya. Didalam diagram *class*, setiap kelas pasti akan berinteraksi dengan baik satu maupun lebih kelas. Relasi yang muncul pada setiap keterhubungan antar kelas juga akan memiliki atributatribut yang akan lebih menjelaskan definisi dari sebuah relasi yang terjadi.

Berikut 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 objek sebuah *class* yang berelasi dengan sebuah objek 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 3.1 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)

(Munawar, “Pemodelan Visual”, [MUN05])

B. *Generalization & Inheritance*

Sebuah *class* (*child class* atau *subclass*) dapat mewarisi atribut atribut Dan operasi operasi 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 *classclass*. Manfaat lain dari struktur hirarkis juga memungkinkan untuk penambahan *subclass* (*child class*) baru tanpa harus merubah struktur yang sudah ada.

Inheritance adalah sebuah mekanisme pengimplementasian generalisasi dan spesialisasi. 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 elemenelemen yang mempunyai karakteristik yang sama. Sedangkan *inheritance* menerangkan mekanisme agar bagi pakai (*sharing*) bisa terjadi.

C. 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 putusputus dan tanda panah pada ujungnya.

D. Agregasi

Didalam UML, ada relasi dengan perlakukan khusus yang disebut dengan "bagian dari (*part of*)" yang menangani antar objekobjek dimana salah satunya adalah bagian dari yang lain. Dengan kata lain sebuah objek terdiri atas objekobjek 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 putusputus harus dibuat untuk menghubungkan dua objek.

Komposit (*composite*) adalah sebuah tipe agregasi yang kuat dimana bagian dari objek tergantung penuh/secara keseluruhan terhadap objeknya sehingga bila sebuah objek 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).

E. *Interface* & Realisasi

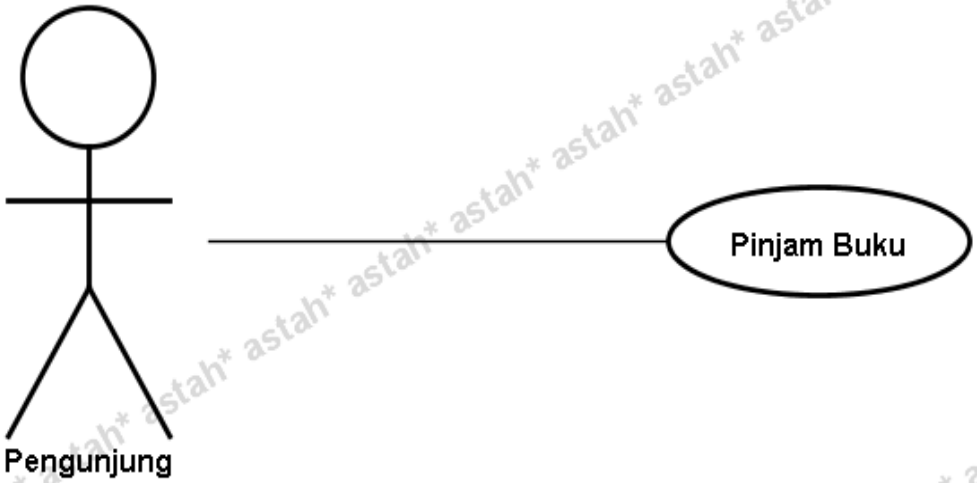
Interface adalah satu set *operation* yang memberikan spesifikasi beberapa aspek dari perilaku dan operasi disuatu *class* ke *class* yang lain. Contohnya, *keyboard* pada komputer sebenarnya merupakan *interface* yang bisa dipakai ulang karena tomboltombol *keyboard* sebenarnya berasal dari mesin ketik, 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 putusputus dengan segitiga yang mengarah ke *interface*.

2 Use Case Diagram

Diagram *use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif/sudut pandang para pengguna sistem. *Use case* mendefinisikan "apa" yang dilakukan oleh sistem dan elemenelemennya, bukan "bagaimana" system dan elemenelemennya saling berinteraksi. *Use case* bekerja dengan menggunakan "*scenario*", yaitu deskripsi urutan urutan langkah yang menerangkan apa yang dilakukan penggunan terhadap sistem maupun sebaliknya. Diagram *use case* mengidentifikasikan fungsionalitas yang dipunyai oleh system (*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, usecase,* 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 menunjukkan *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 menunjukkan 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.

Contoh salah satu *use case* model pada sebuah pemesanan buku diperpustakaan :



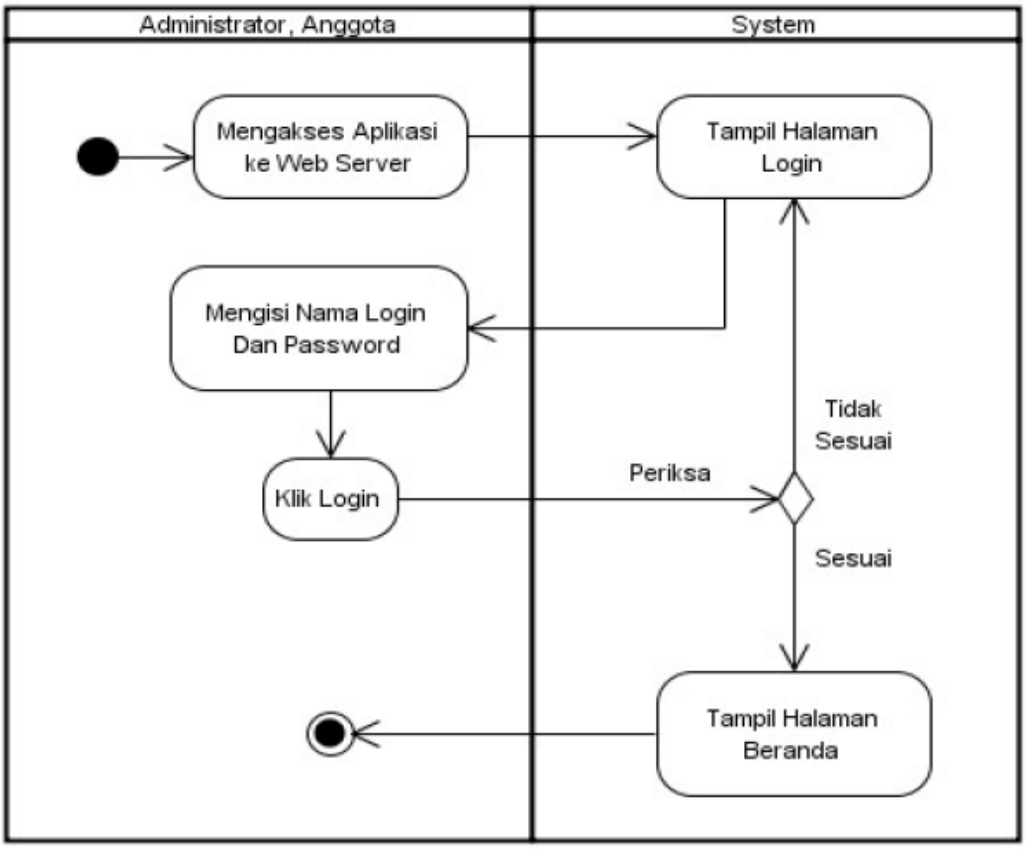
Gambar 3.8 Model *use case* peminjaman buku

Pada model diatas, digambarkan jika pengunjung perpustakaan dapat melakukan pemesanan buku didalam sebuah aplikasi perpustakaan *online* berbasis website.

3 Activity Diagram

Diagram *Sequence* mendokumentasikan komunikasi/interaksi antar kelas kelas. Diagram ini menunjukkan sejumlah objek dan *message* (pesan) – yang diletakkan diantara objekobjek didalam *use case*. Perlu diingat bahwa di dalam diagram ini, kelaskelas dan aktoraktor diletakkan dibagian atas diagram dengan urutan dari kiri ke kanan dengan garis *lifeline* yang diletakkan secara vertical terhadap kelas dan aktor.

Contoh salah satu Activity Diagram pada Activity diagram login :



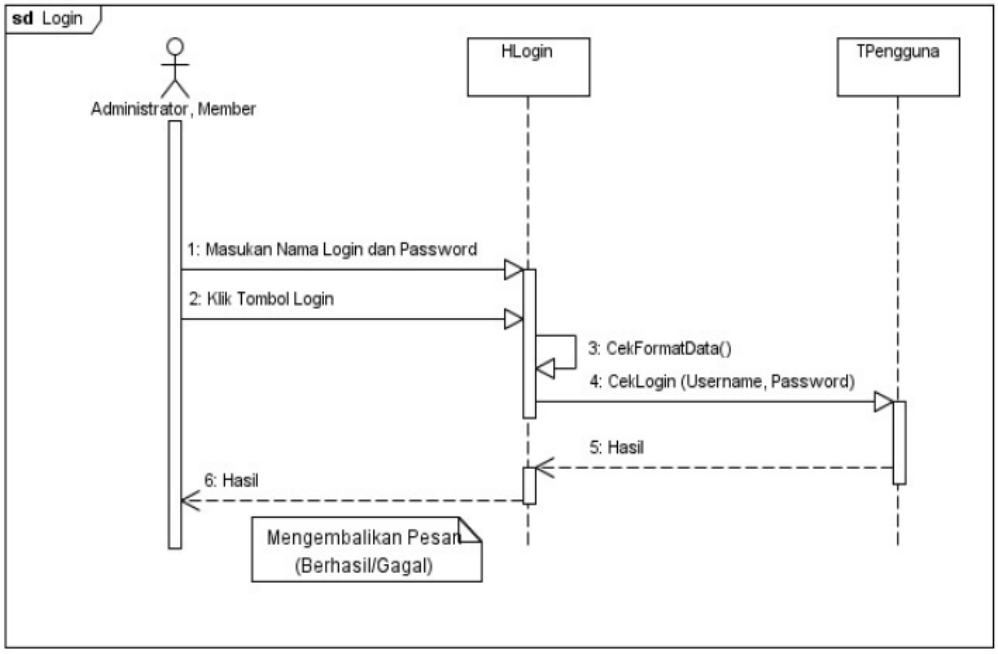
Gambar 3.9 Activity Diagram Login

4 Sequence Diagram

Diagram *Sequence* mendokumentasikan komunikasi/interaksi antar kelaskelas. Diagram ini menunjukkan sejumlah objek dan *message* (pesan) – yang diletakkan diantara objekobjek didalam *use case*. Perlu diingat bahwa di dalam diagram ini, kelaskelas dan aktoraktor diletakkan dibagian atas diagram dengan urutan dari kiri ke kanan dengan garis *lifeline* yang diletakkan secara vertical terhadap kelas dan aktor.

Diagram sequence merupakan diagram untuk menggambarkan perilaku aktor pada sistem tersebut berdasarkan garis waktu.

Dibawah ini adalah Gambar dari *Sequence Diagram* Login :



Gambar 3.10 *Sequence diagram* login

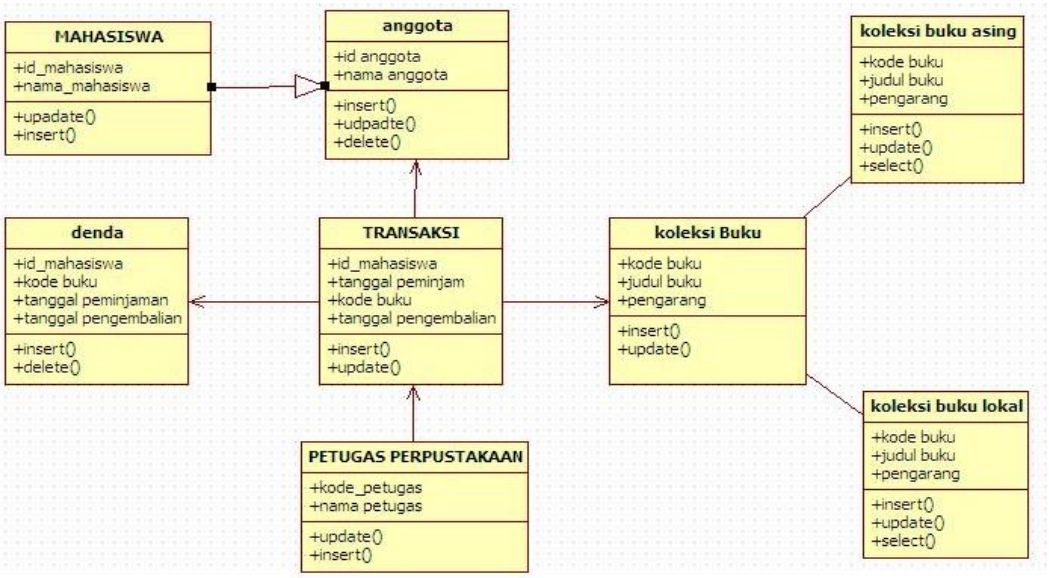
5 Diagram class

Class Diagram adalah suatu diagram yang memperlihatkan atau menampilkan struktur dari sebuah sistem,sistem tersebut akan menampilkan system kelas,atribut dan hubungan antara kelasketika suatu sistem telah selesai membuat diagram. Objek diagram adalah suatu diagram yang berfungsi untuk mengatur atribut, objek dan hubungan antara contoh dalam diagram, objek diagram juga dapat menampilkan struktur model system dalam waktu tertentu.

Diagram dibawah ini merupakan diagram class untuk peminjaman buku di perpustakaan. Didalam diagram tersebut kita memiliki beberapa object diantaranya Mahasiswa, anggota, denda, transaksi, koleksi buku, petugas perpustakaan, koleksi buku, koleksi buku local dan koleksi buku asing.

Object Anggota diatas merupakan turunan dari object mahasiswa bila ingin meminjam sebuah buku haruslah mahasiswa tersebut terdaftar menjadi anggota yang harus memiliki atribut pendukung yaitu, id_anggota dan nama anggota dan memiliki suatu aksi yaitu insert untuk menambahkan jumlah anggota untuk perpustakaan tersebut dan juga di perbaharui datanya, dan juga di delete bila ia sudah lulus dari universitas tersebut.

Petugas perpustakaan dapat melakukan transaksi pencatatan peminjaman dan pengembalian buku yang dilakukan oleh anggota perpustakaan tersebut. Lalu, di dalam sebuah transaksi kemungkinan ada sebuah denda yang dapat diberikan bila pengembalian buku melewati batas maksimum. Selain itu juga transaksi buku juga dapat dilakukan oleh anggota untuk memilih buku yang ada dalam perpustakaan tersebut, yang terdiri dari 2 buah koleksi yaitu koleksi buku asing dan koleksi buku local.

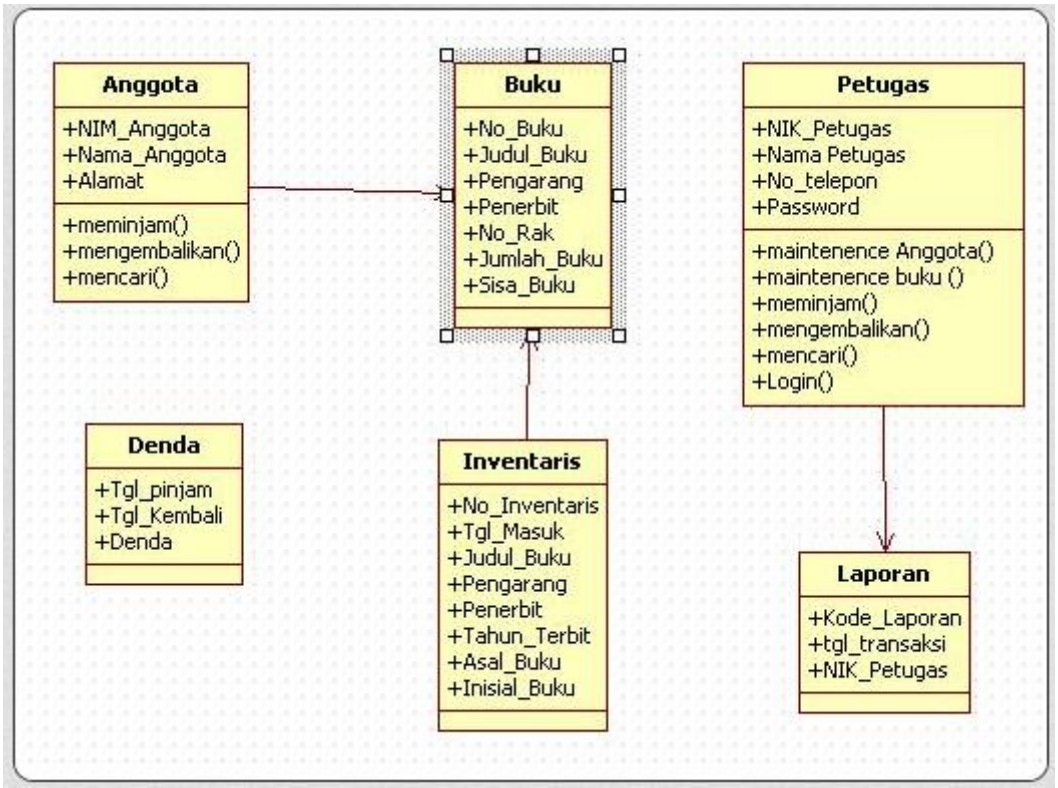


Gambar 3.11 Diagram class Sistem Peminjaman buku

6 Diagram Object

Object Diagram adalah Salah satu perancangan sistem yang digunakan untuk menjelaskan tentang nama obyek, atribut dan metode yang dipakai. Sebuah Object Diagram adalah gambaran dari objek-objek dalam sebuah system pada satu waktu. Diagram ini sering juga disebut sebagai Diagram Perintah, karena pada diagram ini perintah-perintah nya lebih ditonjolkan daripada kelasnya.

Dibawah ini contoh gambaran diagram object system perpustakaan :

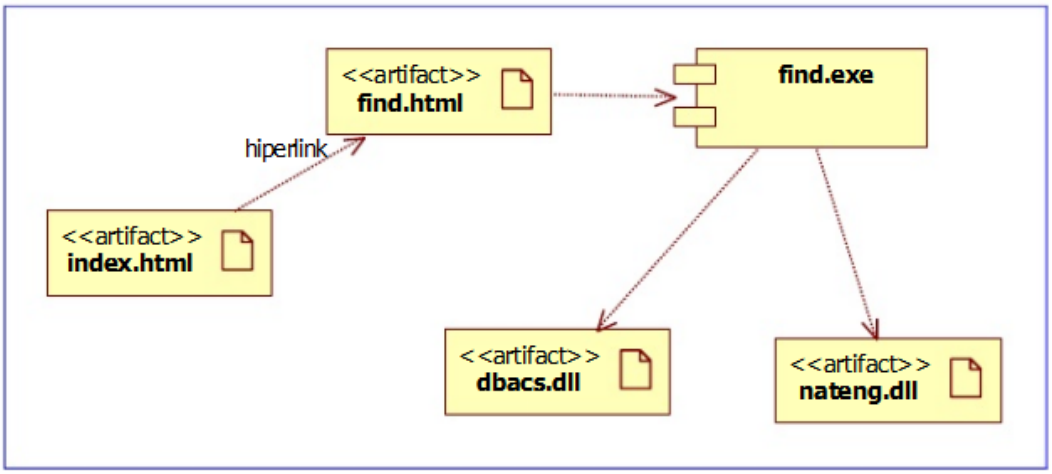


Gambar 3.12 Diagram Objek Sistem Perpustakaan

7 Diagram Component

Component diagram adalah diagram UML yang menampilkan komponen dalam system dan hubungan antara mereka. Pada component View, akan

difokuskan pada organisasi fisik system. Pertama, diputuskan bagaimana kelas-kelas akan diorganisasikan menjadi kode pustaka. Kemudia akan dilihat bagaimana perbedaan antara berkas eksekusi, berkas dynamic link library (DDL), dan berkas runtime lainnya dalam system.



Gambar 3.13 Contoh Diagram Komponen

Ket : - page = index.html, find.html

- executable file = find.exe

- library = dbacs.dll, nateng.dll

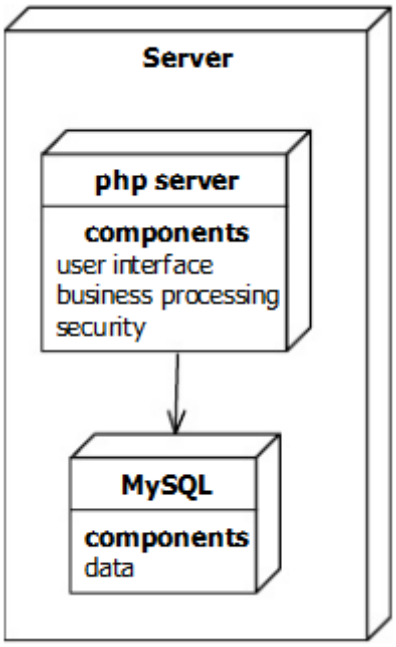
-komponen = semua hal diatas termasuk komponen

8 Diagram Deployment

Deployment View adalah pandangan yang terkait dengan penyebaran fisik aplikasi. Hal ini termasuk persoalan layout jaringan dan lokasi komponen-komponen dalam jaringan. Deployment View berisikan prosesor-prosesor, peralatan-peralatan, proses-proses dan hubungan antar prosesor dan antar perlatan. Semua informasi ini digambarkan dalam suatu deployment Diagram. Hanya ada satu deployment Diagram dalam setiap system , sehingga hanya satu deployment

dalam setiap model. Suatu deployment diagram menampilkan semua titik (node) dalam suatu jaringan, hubungan antar mereka, dan proses-proses yang dijalankan pada masing-masing node.

Berikut adalah diagram deployment dari system informasi manajemen perpustakaan:



Gambar 3.14 Diagram Deployment system manajemen perpustakaan.

Aplikasi di- deploy pada sebuah komputer server dimana di dalamnya sudahterdapat php server dan MySQL sebagai DBMS.

3.5. Pengujian Software

Pengujian Software (perangkat Lunak) adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.Roger Pressman (2002 : 59).

Metode pengujian yang digunakan oleh penulis adalah *black-box.Black Box Tasting* digunakan untuk menguji fungsi-fungsi dari perangkat lunak yang dirancang.

Pengujian *balck Box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujuan black box memungkinkan perekayasa perangkat lunak mendapat serangkain kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian black box merupakan pendekatan komlementer yang kemungkinan besar mampu mengungkap kelas kesalahan.

Pengujian black box berusaha menemukan kesalahan dalm kategori sebagai berikut :

1. Fungsi yang tidak benar atau hilang
2. Kesalahan antar muka
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi kesalahan terminasi