

BAB I

PENDAHULUAN

Pada bagian ini akan dijelaskan tentang pendahuluan dalam penyusunan Laporan Tugas Akhir, yang meliputi latar belakang masalah, identifikasi masalah, rumusan masalah, maksud dan tujuan penelitian, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1 Latar Belakang Masalah

Di permukaan bumi terdapat kira-kira 1,5 juta jenis makhluk hidup. Dari jumlah itu kira-kira 1 juta tergolong dalam dunia satwa. Indonesia termasuk dalam jajaran negara yang mempunyai keanekaragaman satwa. Keanekaragaman satwa di Indonesia sangat menonjol dikarenakan keadaan tanah, letak geografis serta keadaan iklim yang sesuai dengan tempat tinggal untuk satwa-satwa tersebut. Indonesia terletak di daerah yang cukup menguntungkan dan nyaman untuk tempat tinggal satwa-satwa tersebut, yakni di daerah tropis sehingga memiliki keanekaragaman hayati dibanding daerah subtropis (iklim sedang) dan kutub (iklim kutub). Tingginya keanekaragaman hayati di Indonesia ini terlihat dari berbagai macam ekosistem yang ada di Indonesia, seperti ekosistem pantai, ekosistem hutan bakau, ekosistem padang rumput, ekosistem hutan hujan tropis, ekosistem air tawar, ekosistem air laut, ekosistem savanna, dan lain-lain. Masing-masing ekosistem ini memiliki keanekaragaman hayati tersendiri.

Banyak diantara satwa-satwa Indonesia mempunyai keistimewaan yang tidak dapat ditemui di benua lain. Penyebaran dari jenis-jenis hewan liar/satwa yang ada di Indonesia mempunyai ciri-ciri tertentu yang tidak sama satu sama lain, beberapa diantaranya termasuk jenis yang langka dan hampir punah. Meskipun kaya, Indonesia dikenal juga sebagai negara yang memiliki daftar panjang tentang satwa liar yang terancam punah. Saat ini jumlah jenis satwa liar Indonesia terancam punah adalah 147 jenis mamalia, 114 jenis burung, 28 jenis reptil, 91 jenis ikan dan 28 jenis invertebrata [1]. Karena banyaknya keanekaragaman hayati di Indonesia ini maka sebenarnya informasi mengenai fauna di Indonesia penting untuk menambah pengetahuan masyarakat mengenai kekayaan alam di Indonesia sendiri. Apalagi masyarakat

Indonesia sendiri masih banyak yang belum menyadari akan kekayaan alam di negaranya sendiri terutama mengenai satwa-satwa langka di Indonesia yang unik dan menarik tapi langka dan hampir punah.

Yang menjadi dasar lokasi perancangan berada di Yayasan Margasatwa Tamansari Bandung (YMTB) atau yang dulu dikenal sebagai Kebun Binatang Bandung yakni karena YMTB sendiri adalah merupakan salah satu objek wisata yang populer di kota Bandung, terletak di jalan Tamansari no.6 Bandung. Jumlah satwa di YMTB terdiri 1.135 ekor dari 218 jenis satwa. Termasuk didalamnya satwa langka Indonesia maupun dunia terdiri dari mamalia, burung, reptil, ikan. YMTB merupakan sarana pendidikan yang secara tidak langsung berguna untuk mengenal berbagai macam satwa yang ada, supaya tertanam sejak dini perasaan mencintai seluruh alam dan isinya. Selain itu, YMTB merupakan taman satwa yang artinya tempat atau wadah yang mempunyai fungsi utama konservasi ex-situ yang melakukan usaha perawatan dan penangkaran berbagai jenis satwa dalam rangka membentuk dan mengembangkan habitat baru sebagai sarana perlindungan dan pelestarian alam yang dimanfaatkan untuk pengembangan IPTEK dan sarana rekreasi yang sehat.

Program pendidikan dan penelitian di YMTB melaksanakan pengembangan ilmu pengetahuan dan teknologi yang merupakan wahana keilmuan bagi masyarakat dan merupakan laboratorium hidup untuk lebih mencintai dan menghargai flora dan fauna sebagai kekayaan alam milik bersama.

Pengunjung YMTB yang datang dari berbagai daerah, kota, bahkan turis mancanegara. Kalau dilihat sebatas perkembangan YMTB sangat baik dan pantas untuk dibanggakan sebagai obyek wisata di Jawa Barat pada umumnya dan Bandung pada khususnya. Akan tetapi sayangnya pada YMTB ini kurang adanya fasilitas yang dapat menginformasikan segala sesuatu mengenai satwa serta informasi mengenai YMTB itu sendiri. Hal ini yang menjadi permasalahan karena visi dan misi dari YMTB merupakan pusat pendidikan, penelitian, rekreasi dan konservasi. Jadi sebenarnya di YMTB harus diberi fasilitas yang maksimal untuk tujuan pendidikan. Apalagi YMTB juga sering mengadakan kerjasama dengan pihak-pihak sekolah untuk mengunjungi YMTB dalam

rangka pendidikan dan rekreasi. Berangkat dari permasalahan diatas maka saya memutuskan untuk membuat Aplikasi Multimedia Informasi Yayasan Margasatwa Tamansari Bandung (YMTB).

1.2 Identifikasi Masalah

Berdasarkan pada latar belakang di atas, maka dapat diidentifikasi beberapa permasalahan pokok yang akan dijadikan bahan kajian, antara lain :

1. Papan Nama satwa di YMTB kurang memberikan informasi yang lengkap mengenai satwa.
2. Buku sebagai sarana atau alat bantu memberikan informasi mengenai satwa-satwa akan tetapi karena terlalu banyaknya satwa menjadikan pengunjung cenderung malas membaca.
3. Belum adanya fasilitas yang menyediakan informasi mengenai YMTB seperti informasi harga tiket masuk YMTB, map YMTB serta informasi satwa yang terdapat di YMTB.
4. Pemanfaatan teknologi komputer sebagai media yang bersifat "Edutainment" yaitu perpaduan antara *education* (pendidikan) dan *entertainment* (hiburan).

1.3 Rumusan Masalah

Dalam pembahasan pengembangan aplikasi terdapat rumusan masalah yang timbul dengan Identifikasi permasalahan diatas yaitu:

1. Mengapa Aplikasi multimedia informasi YMTB ini penting diterapkan sebagai sarana informasi kepada para pengunjung YMTB?
2. Bagaimana memanfaatkan teknologi komputer yang memadukan antara *education* (pendidikan) dan *entertainment* (hiburan)?

3. Apa saja fasilitas-fasilitas yang harus ada pada aplikasi ini sebagai media informasi yang dapat menampung segala informasi mengenai YMTB?

1.4 Maksud dan Tujuan

Maksud dari aplikasi ini adalah untuk menyediakan media sebagai fasilitas berupa sarana informasi kepada pengunjung YMTB dengan memberikan informasi yang terdapat di YMTB yang dapat digunakan sebagai sarana untuk menggali pengetahuan, terutama pengetahuan tentang satwa yang berada di Indonesia.

Sedangkan tujuan pembangunan aplikasi multimedia YMTB ini adalah :

1. Membuat media informasi yang dikhususkan kepada pengunjung YMTB yang dirancang dengan penampilan yang lebih menarik dan memberikan sebuah gambaran ilustrasi sederhana berupa animasi.
2. Memberikan kemudahan Pengunjung YMTB untuk memperoleh informasi yang terdapat di YMTB.
3. Memotivasi para pengunjung YMTB untuk lebih mengetahui mengenai keanekaragaman satwa yang terdapat di Indonesia.

1.5 Batasan masalah

Penulis memfokuskan penelitian ini dengan membuat ruang lingkup sebagai berikut :

1. Materi yang digunakan dalam media pembelajaran ini disesuaikan dengan acuan Metode terbaru menghitung Math Flash (Drajat Permadi, ST.2005) meliputi penjumlahan, pengurangan, perkalian dan pembagian.
2. Pembelajaran berhitung (aritmatika) ini ditujukan untuk anak kelas 2 sampai 3 Sekolah Dasar.

3. Tingkat pemahaman mata pelajaran berhitung adalah sudah melalui tingkat Pengenalan angka yaitu mengenal angka 1 sampai 10 , puluhan, ratusan serta ribuan.

1.6 Metodologi Penelitian

Adapun metodologi yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Studi pustaka, yaitu dengan mempelajari berbagai buku, referensi, skripsi, dan makalah yang dapat digunakan sebagai sumber dan pedoman dalam pembuatan laporan Tugas Akhir.
2. Studi sistem, yaitu mempelajari metode pembelajaran secara konvensional dan metode pembelajaran yang lainnya, mempelajari kelebihan dan kekurangannya, serta kendala yang dihadapi siswa/anak sekolah dasar dalam belajar berhitung aritmatika.
3. Pengembangan sistem, yaitu menggunakan rancangan interface program multimedia interaktif, yang terdiri dari desain, *storyboard*, komposisi tampilan, warna interface dan navigasi. Model pengembangan sistem yang penulis gunakan dalam penelitian ini adalah Perancangan dan pengembangan Sistem Multimedia Interaktif Dastbaz. Tahapan-tahapan pengembangan yang dilakukan dalam rekayasa sistem meliputi analisa sistem, desain/perancangan, mengimplementasikan hasil rancangan/pengkodean kedalam aplikasi menggunakan bahasa pemograman macromedia flash, serta pengujian.

1.7 Sistematika Penulisan

Bab I Pendahuluan, bab ini menjelaskan secara umum mengenai latar belakang masalah, identifikasi masalah, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi pembangunan sistem dan sistematika penulisan dalam pembuatan laporan ini.

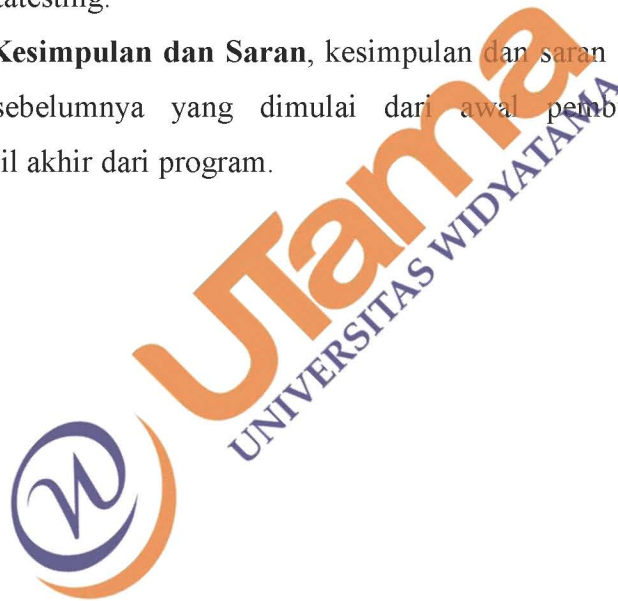
Bab II Landasan Teori, bab ini menjelaskan teori-teori yang berkaitan dengan pembangunan aplikasi dan menjelaskan secara singkat mengenai bahasa pemrograman yang digunakan.

Bab III Analisis, bab ini menjelaskan tahapan mulai dari Analisis sistem yang sedang berjalan, identifikasi permasalahan, Analisis pemrosesan sistem yang sedang berjalan, mendeskripsikan kebutuhan, dan hasil Analisis sistem.

Bab IV Perancangan, meliputi perancangan sistem metaphor dengan uml, tipe informasi, struktur navigasi dengan storyboard dan link program , dan kontrol sistem.

Bab V Implementasi, meliputi lingkungan implementasi, yaitu perancangan prototype dan betatesting.

Bab VI Kesimpulan dan Saran, kesimpulan dan saran dari beberapa bab yang telah disusun sebelumnya yang dimulai dari awal pembuatan sampai dengan mendapatkan hasil akhir dari program.



BAB II LANDASAN TEORI

Pada bab ini penulis akan menguraikan tentang teori-teori yang melandasi penulisan Laporan Tugas Akhir ini.

2.1 Definisi Aplikasi

Application atau biasa disebut dengan Aplikasi memiliki makna yang banyak, dalam bidang pendidikan aplikasi dikenal dengan suatu sarana atau alat bantu untuk mengimplementasikan suatu hal tertentu[1]. Aplikasi yang dimaksud disini adalah definisi aplikasi dalam sistem komputer.

2.2 Multimedia

Multimedia adalah output berdasarkan sistem *computer* yang mengintegrasikan atau mengkonvergensi teks, suara, gambar bergerak dan atau animasi. Termasuk didalamnya konten yang dimuat di media internet, *games computer*, film animasi, klip video, file musik, desain grafis *computer* dan piranti lunak aplikasi (*application software*) [2].

Multimedia berasal dari kata:

1. Multi [latin nouns]: banyak bermacam-macam
2. Medium [latin]: sesuatu yang dipakai untuk menyampaikan atau membawa sesuatu medium [American Heritage Electronic Dictionary, 1991]: alat untuk mendistribusikan dan mempresentasikan informasi.

Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk *text*, *audio*, grafik, animasi, dan video.

Beberapa definisi multimedia menurut beberapa ahli:

1. Kombinasi dari *computer* dan video (Rosch, 1996)
2. Kombinasi dari tiga elemen: suara, gambar, dan teks (McComick, 1996)
3. Kombinasi dari paling sedikit dua media input atau output. Media ini dapat berupa *audio* (suara, musik), animasi, *video*, teks, grafik, dan gambar (Turban dan kawan-kawan, 2002).

4. Alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, *audio* dan *video* (Robin dan Linda, 2001)
5. Multimedia dalam konteks computer menurut Hofstetter 2001 adalah: pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, *audio*, *video*, dengan menggunakan tool yang memungkinkan pemakai berinteraksi, berkreasi, dan berkomunikasi.

Sistem multimedia dapat dibagi menjadi:

1. Sistem *Multimedia Stand Alone*
Sistem ini berarti merupakan sistem *computer* multimedia yang memiliki minimal storage (*harddisk*, CD-ROM/DVD-ROM/CD-RW/DVD-RW), alat *input* (*keyboard*, *mouse*, *scanner*, mic), dan *output* (*speaker*, *monitor*, LCD Proyektor), *VGA* dan *Soundcard*.
2. Sistem Multimedia berbasis jaringan.
Sistem ini harus terhubung melalui jaringan yang mempunyai *bandwidth* yang besar. Perbedaan adalah adanya *sharing system* dan pengaksesan terhadap sumber daya yang sama. Contoh: *video converance* dan *video broadcast*. Permasalahan: bila *bandwidth* kecil, maka akan terjadi kemacetan jaringan, *delay* dan masalah infrastruktur yang belum siap.

2.3 Definisi Kebun Binatang atau Taman Margasatwa

Kebun binatang atau taman margasatwa adalah tempat hewan dipelihara dalam lingkungan buatan, dan dipertunjukkan kepada publik. Selain sebagai tempat rekreasi, kebun binatang berfungsi sebagai tempat pendidikan, riset, dan tempat konservasi untuk satwa terancam punah. Binatang yang dipelihara di kebun binatang sebagian besar adalah hewan yang hidup di darat, sedangkan satwa air dipelihara di akuarium [6].

Kebun binatang yang memungkinkan pengunjung masuk dengan mobil atau bus disebut taman safari. Binatang dilepas di kawasan luas dan terbuka, serta tidak dikurung di kandang-kandang sempit, melainkan dibatasi dengan pagar atau parit. Kebun binatang sering dilengkapi dengan kebun binatang anak untuk mempertontonkan hewan ternak atau spesies satwa liar yang belum dewasa dan jinak untuk dipegang-pegang atau diberi makan, termasuk oleh anak-anak. Ada pula taman bermain yang bertemakan hewan, misalnya SeaWorld dan Disney's Animal Kingdom [6].

Kebun binatang tertua di Indonesia adalah Taman Margasatwa Ragunan Jakarta yang didirikan tahun 1864 di Cikini sebagai Planten- en Dierentuin Batavia (Kebun Raya dan Botani Batavia). Pengelolanya bernama Perhimpunan Penyayang Flora dan Fauna Batavia [13].

2.4 Taman Margasatwa Tamansari Bandung

Objek wisata di Kota Bandung ini oleh masyarakat Sunda tempo dulu populer dengan disebut “derenten” (dari bahasa Belanda “dierentuin”) artinya kebun binatang.

Terletak di jalan Tamansari, sebelah barat kampus ITB. Luas arealnya +- 14 ha. Didirikan tahun 1930 oleh Bandungse Zoological Park (BZP). Pelopornya seorang bangsa Belanda bernama HOOGLAND. Direktur Bank Denis mendapat pengesahan Gubernur Jendral Hindia Belanda Tanggal 12 April 1933 No.32. Setelah terlantar selama Penduduk Jepang dan revolusi kemerdekaan, tahun 1948 direhabilitasi. Tahun 1956 BZP sebagai pengelola dibubarkan sebagai gantinya didirikan Yayasan Margasatwa Tamansari (1957) atas inisiatif B. EMA BRATAKOESOEMA kini memiliki tidak kurang dari 218 jenis satwa dengan jumlah 1.135 ekor. Binatang-binatangnya antara lain gajah, harimau, singa, komodo, buaya, orang utan, beruang, unta, kuda nil, kapibara dan lain lain.

Menyajikan pula :

1. Museum Zoologi yang memamerkan berbagai jenis satwa yang telah diawetkan aquarium yang berisi aneka macam ikan hias dari dalam dan luar negeri yang dapat menimbulkan ketenangan jiwa pengunjungnya.
2. Pertunjukan kesenian tradisional Sunda seperti ketuk tilu, kendang penca atau calung, diselenggarakan setiap hari Minggu.
3. Arena permainan anak-anak seperti korsel, Kereta Api Mini, Mobil-mobilan, Scaller dll
4. Perahu dayung sepeda air yang dapat dinaiki keliling kolam
5. Gajah Tunggang, Unta tunggang; empat ekor gajah dan empat ekor unta terlatih dapat membawa anda keliling kompleks kebun binatang

Fungsi Yayasan Margasatwa Tamansari Bandung :

1. Fungsi Kebudayaan : menanamkan dan menumbuhkan kesadaran dan rasa cinta tanah air melalui pengamatan dan pemahaman kekayaan alam berupa fauna dan flora yang dilestarikan.

2. Fungsi Pendidikan dan Ilmu Pengetahuan : menanamkan dan menumbuhkan kesadaran dan rasa sayang terhadap makhluk hidup , terutama satwa di kalangan anak-anak, membentuk pekerti yang baik agar kelak tidak bertindak termena-mena terhadap isi bumi ini, melainkan selalu menjaga keseimbangan kehidupan alam semesta; flora dan fauna koleksi kebun binatang dapat menjadi objek studi pelajar mahasiswa, peneliti, dan kalangan cendikiawan lainnya.
3. Fungsi Perlindungan dan Pelestarian Kekayaan Alam : Perkembangan ilmu dan teknologi dewasa ini antara lain berdampak makin mempersempit ruang kehidupan satwa yang dampak selanjutnya mengancam keberadaan satwa itu; kebun binatang justru berupaya melindungi dan melestarikan aneka macam satwa dan kepunahan melalui pengumpulan, pemeliharaan dan pengembangbiakkan satwa terutama satwa-satwa langka.
4. Fungsi Rekreasi : dalam kehidupan manusia yang semakin kompleks permasalahannya dewasa ini, kebun binatang melalui sajian tontonan satwa yang aneka warna penampilan dan perilakunya sehingga sangat menarik dapat menghilangkan kejenuhan, kepenatan dan tekanan mental (stress), menjaga kestabilan aktivitas kerja serta memulihkan kembali kebugaran jasmani dan kesegaran rohani para pengunjungnya.

2.5 Informasi

Informasi adalah pengetahuan yang didapatkan dari pembelajaran, pengalaman, atau instruksi. Namun demikian istilah ini memiliki banyak arti bergantung pada konteksnya, dan secara umum berhubungan erat dengan konsep seperti arti, pengetahuan, negentropy, komunikasi, kebenaran, representasi, dan rangsangan mental [5].

Kata informasi berasal dari kata Perancis kuno *informacion* (tahun 1387) yang diambil dari bahasa latin *informationem* yang berarti “garis besar, konsep, ide”. Informasi merupakan kata benda dari *informare* yang berarti aktivitas dalam “pengetahuan yang dikomunikasikan” [5].

Banyak orang menggunakan istilah "era informasi", "masyarakat informasi," dan teknologi informasi, dalam bidang ilmu informasi dan ilmu komputer yang sering disorot, namun kata "informasi" sering dipakai tanpa pertimbangan yang cermat mengenai berbagai arti yang dimilikinya [5].

Informasi merupakan fungsi penting untuk membantu mengurangi rasa cemas seseorang. Menurut Notoatmodjo (2008) bahwa semakin banyak informasi dapat

mempengaruhi atau menambah pengetahuan seseorang dan dengan pengetahuan menimbulkan kesadaran yang akhirnya seseorang akan berperilaku sesuai dengan pengetahuan yang dimilikinya [5].

2.6 Perangkat lunak Pendukung (software)

Perangkat lunak yang akan digunakan dalam membangun aplikasi multimedia informasi YMTB ini adalah sebagai berikut :

2.6.1 Macromedia Flash

Adobe Flash (dahulu bernama Macromedia Flash) adalah salah satu perangkat lunak komputer yang merupakan produk unggulan Adobe Systems. Software yang dirancang untuk membuat sebuah design/gambar animasi, banner, game dan lain-lain pada awalnya software ini dikembangkan oleh perusahaan yang bernama macromedia dengan produk yang terkenal yaitu macromedia flash tetapi kini licencinya telah di beli oleh perusahaan pengembang software design yang terkenal yaitu adobe, sehingga munculah nama adobe flash, setelah flash jatuh ke tangan adobe, adobe terus mengembangkan software ini hingga saat ini. Berkas yang dihasilkan dari perangkat lunak ini mempunyai file extension .swf dan dapat diputar di penjelajah web yang telah dipasang Adobe Flash Player. Flash menggunakan bahasa pemrograman bernama ActionScript yang muncul pertama kalinya pada Flash 5 [7].

2.6.1.1 Kelebihan Macromedia Flash

Macromedia Flash merupakan salah satu software aplikasi design grafis yang sangat populer saat ini terutama untuk membuat aplikasi animasi dalam efek yang spektakuler. Kesederhanaan tool yang disediakan serta kemampuan yang luas menjadikan Flash semakin digemari.

Beberapa alasan memilih Flash yaitu [7] :

1. Hasil akhir Flash memiliki ukuran yang lebih kecil (setelah dipublish)
2. Flash dapat mengimpor hampir semua gambar dan file-file audio sehingga dapat lebih hidup.
3. Animasi dapat dibentuk, dijalankan dan dikontrol
4. Gambar Flash tidak akan pecah meskipun di zoom beberapa kali karena gambar flash bersifat gambar vektor.

5. Hasil akhir dapat disimpan dalam berbagai macam bentuk seperti *.avi, *.gif, *.mov, maupun file dengan format lain.

2.6.1.2 Tipe-Tipe File Yang Didukung Macromedia Flash

Dibawah ini adalah daftar file-file yang didukung oleh software Macromedia Flash :

Tabel 2.1 Tipe-Tipe File Yang Didukung Macromedia Flash [7]

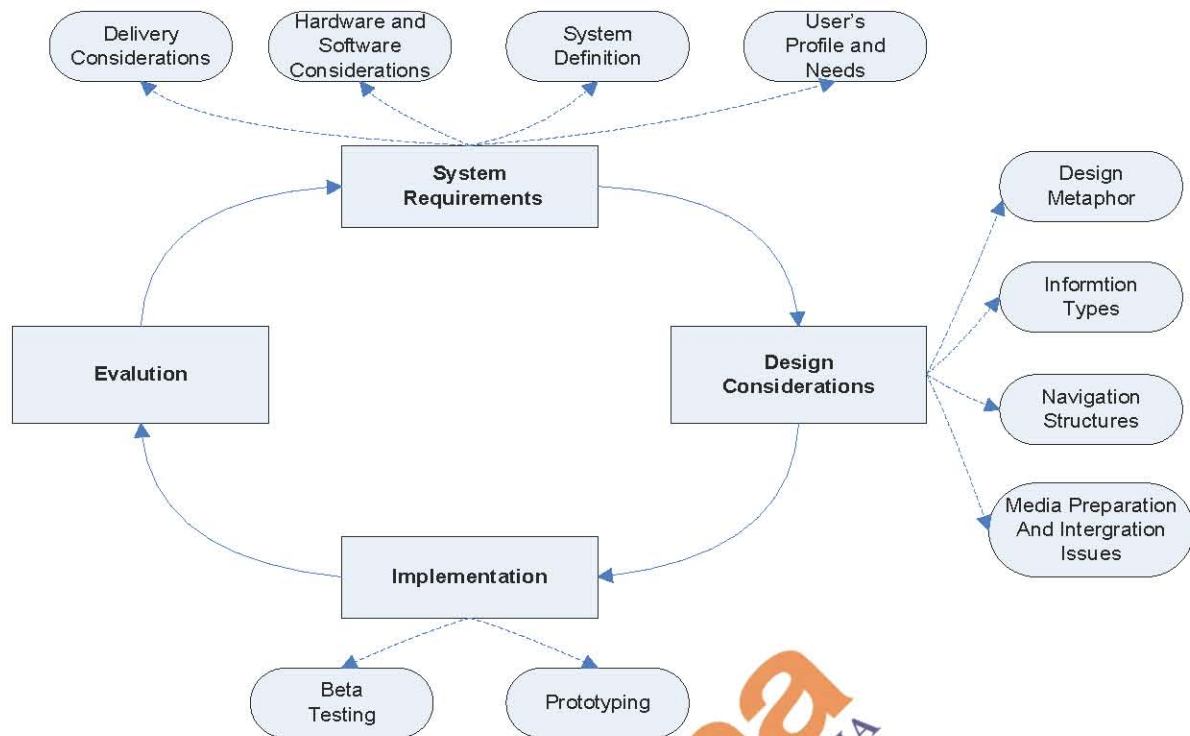
Tipe File	Format File yang didukung
Gambar	BMP, GIF, JPG, LRG (xRest), File Photoshop, MacPaint, PNG, TIFF, PICT, dll
Suara	AIFF, WAV, MP3, dll
Video	AVI(video for windows),MOV,MPEG, dll
Text	RTF, ASSCI, DOC, TXT, dll

2.6.2 Adobe Photoshop CS 2

Software keluaran perusahaan Adobe ini merupakan software pengolah foto dan manipulasi objek yang sangat baik untuk membuat desain grafis dalam bentuk bitmap. Software ini digunakan penulis untuk membuat dan memanipulasi objek yang digunakan untuk dijadikan objek yang representatif dan menarik, yang kemudian di ekspor kedalam stage di Macromedia Flash 8.

2.7 Metodologi Pengembangan Perangkat Lunak

Untuk menyelesaikan masalah aktual dalam sebuah rekayasa perangkat lunak diperlukannya strategi untuk pengembangan yang melengkapi lapisan proses dan metode. Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya yaitu Perancangan dan Pembangunan Sistem Multimedia Interaktif [3].



Gambar 2.1 Siklus Perancangan dan Pembangunan Sistem Multimedia Interaktif Datbaz[3]

2.7.1 Tahap-tahap pada siklus IMSDD (*Interactive Multimedia System Design & Development*)

Tahap-tahap yang terdapat dalam siklus perancangan dan pengembangan IMS yaitu [3] :

a. Kebutuhan Sistem (system Requirements)

Tahap ini sama dengan tahap spesifikasi kebutuhan (*Requirement specification*) yang terdapat dalam model *Waterfall* dan didalamnya terdapat elemen-elemen seperti *feasibility* dan *hardware selection* yang juga terdapat dalam model RMM (*The Relationship Management Methodology*). Pada tahap ini mempunyai fungsi utama, diantaranya :

1. Untuk menyajikan definisi sistem yang mencakup garis besar dan tujuan dari sistem.
2. Untuk menjelaskan pengguna (*user*) mana saja yang akan menggunakan sistem dan juga menjelaskan kebutuhan-kebutuhan khusus yang digunakan dalam pertimbangan. Sebagai contoh jika kita akan melakukan perancangan untuk mengajar bahasa isyarat bagi pengguna yang memiliki kekurangan dalam pendengaran dengan menggunakan audio, yang merupakan cara penyampaian informasi yang tepat. Oleh sebab itu kita

harus memberikan perhatian khusus pada kegiatan mengelompokkan informasi yang akan digunakan dan pendekatan perancangan yang akan kita ambil untuk penyajian informasi.

3. Untuk mengevaluasi kebutuhan Hardware dengan *platform-platform software* yang digunakan, sehingga dapat dibuat keputusan yang tepat.
4. Untuk mempertimbangkan dengan baik, platform yang dibutuhkan untuk sistem pada kenyataannya membangun sistem multimedia interaktif yang terdistribusi yang dapat dijalankan pada jaringan (LAN/WAN) membutuhkan pendekatan yang berbeda dibandingkan dengan tipe sistem CD-ROM yang stand alone terutama dibagian perancangan dan pembangunan.

b. Pertimbangan Perancangan (*design considerations*)

Tujuan dari tahap ini yaitu untuk menyusun pedoman mengenai rincian perancangan. Dalam hal ini, tahap ini sama dengan tahap-tahap perancangan arsitektual (*arsitectural design*) dan perincian perancangan (*detailed design*) pada model waterfall atau tahap Perancangan (*design*) pada siklus perancangan antarmuka pengguna (*user interface design cycle*) yang dikemukakan oleh Preece (1993). Tahap ini bertujuan untuk mengemukakan hal-hal mengenai:

1. Perancangan Metaphor (*design Metaphor*)

Melakukan pemilihan model yang sesuai dengan keadaan dilapangan (*real word mental mode*) yang akan digunakan sebagai solusi perancangan antarmuka bagi sistem (contoh : sebuah film, buku, permainan, dll.)

2. Format dan tipe Informasi (*Information types and formats*)

Untuk mendefinisikan tipe informasi yang ingin diintegrasikan ke dalam sistem (contoh: teks/tulisan, grafik, suara, video dan animasi), sebagai contoh sebuah sistem multimedia interaktif untuk film dan bioskop akan menunjukkan bahwa isi dari tipe video yang akan digunakan kemungkinan dibutuhkan dalam skala yang besar. Sedangkan sebuah sistem ensiklopedia akan membutuhkan campuran isi yang seimbang dengan memberikan penekanan pada tipe text/isi dari informasi.

3. Struktur Navigasi (*Navigational Structures*)

Untuk menjelaskan strategi dari alat navigasi yang akan digunakan termasuk didalamnya struktur link dan fitur-fitur yang berhubungan dengan sistem.

4. Kontrol Sistem (*System Control*)

Untuk menjelaskan fitur-fitur dan tipe dari kontrol *tools* yang dibutuhkan sistem. Termasuk didalamnya pencarian *tools*, suara, video, dan animasi kontrol, fasilitas penanda buku, dll.

c. Implementasi (*implementation*)

Ketika fitur perancangan telah di definisikan, tahap implementasi pada sistem akan dimulai dengan menggunakan multimedia-*authoring tools*. Tahap implementasi terdiri atas:

- a. Membuat *prototype* sistem
- b. Melakukan *beta test* pada *prototype* untuk mengetahui rancangan yang digunakan dan kontrol pada setiap permasalahan.

Tahap ini sama dengan tahap *coding, integration, unit testing* pada model *waterfall* atau tahap implementasi pada siklus perancangan antarmuka pengguna (*user interface design cycle*), tahap implementasi pada model perancangan OOHDM (*The Object Oriented Hypermedia Design Model*) dan tahap *construction* pada model perancangan RMM (*The relationship Management Methodology*).

d. Evaluasi (*Evaluation*)

Pada tahap ini sistem akan dinilai berdasarkan tujuan awal yang telah direncanakan. Terdapat dua jenis pendekatan yang biasa digunakan dalam evaluasi seperti *formative* atau *summative*. Evaluasi *formative* berkaitan dengan pencapaian tujuan suatu inovasi selama pelaksanaannya. Sedangkan evaluasi *summative* berkaitan dengan efektivitas suatu inovasi pada penyelesaian dengan tujuan yang lain (Thorpe, 1988). Dengan kata lain evaluasi *formative* sangat cocok untuk memastikan apakah produk memenuhi kebutuhan pengguna, sedangkan evaluasi *summative* paling tepat dalam menentukan kesesuaian produk dibandingkan dengan produk lain atau serupa yang sudah ada.

2.7.2 Panduan Perancangan Antarmuka pada IMS [3]

- a. Menggunakan *metaphor* (bayangan/imajinasi) yang tepat.

Bayangan/imajinasi yang baik akan menciptakan suasana yang nyaman bagi pengguna sehingga dengan cepat dapat mempelajari atau mengenali sistem.

- b. Kesederhanaan dan kenyamanan dalam penggunaan merupakan hal yang utama.

Antarmuka yang bagus dapat membuat pengguna langsung menjalankan sistem tanpa harus mempelajari petunjuk pemakaian terlebih dahulu.

- c. Konsistensi dalam perancangan merupakan hal yang sangat penting.

Dengan adanya konsistensi dalam perancangan akan membuat pengguna merasa nyaman dalam menggunakan sistem. Penggunaan icon dan fitur alat navigasi yang konsisten akan membantu mengurangi kompleksitas pada antarmuka sistem multimedia interaktif.

- d. Kebutuhan akan panduan yang dapat membantu user.

Penyediaan panduan informasi (seperti keterangan yang muncul ketika user menggerakkan *mouse* pada sebuah *icon*) dan panduan manual akan membantu user yang masih awam agar dapat mempelajari sistem lebih jauh lagi dan menguasainya secara mendalam.

- e. Menyediakan mekanisme untuk menangani kesalahan yang mungkin dilakukan oleh *user*.

Suatu hal penting yang harus diperhatikan oleh seorang *designer* dalam IMS yaitu adanya fitur control yang dapat membuat *user* memperbaiki kesalahan yang telah dibuat dan mengulang kembali proses yang telah mereka jalani dengan kurang hati-hati.

2.8 Pemrograman Berorientasi Objek Menggunakan *Unified Modeling Language* (UML)[8]

2.8.1 Pemrograman Berorientasi Objek (*Object Oriented Programming*)

Dalam program berbasis pada objek, sebuah program dibagi menjadi bagian-bagian kecil yang disebut dengan objek. Setiap objek memiliki entiti yang terpisah dengan entiti objek-objek lain dalam lingkungannya. Objek-objek yang terpisah ini dapat diolah sendiri-sendiri dan setiap objek memiliki sekumpulan sifat dan metode yang melakukan fungsi tertentu sesuai dengan yang telah kita programkan kepadanya.

Setiap objek mengandung tiga hal utama seperti dibawah ini:

1. Properti atau Atribut, properti atau atribut adalah karakteristik atau sifat dari sebuah objek.
2. Metode, metode adalah serangkaian prosedur yang dimiliki oleh suatu objek yang akan dijalankan sesuai dengan respon yang diberikan oleh suatu perintah atau kejadian.
3. Event, event adalah kejadian atau segala sesuatu yang dapat dialami oleh sebuah objek.

Objek-objek dibuat secara terpisah dan masing-masing memiliki properti serta metode sendiri-sendiri. Setiap objek bisa memiliki metode dan properti yang berbeda satu dengan yang lainnya. Tetapi ada pula dua atau lebih objek yang memiliki metode yang sama. Objek-objek seperti ini harus dibuat dari satu kelas yang sama. Dengan memakai fasilitas ini, pembuatan objek yang bersifat sama tidak perlu dilakukan berulang-ulang.

Pemrograman berorientasi objek memiliki beberapa kelebihan dari pemrograman linier, diantaranya seperti:

1. Lebih cepat dari pemrograman linier karena tidak perlu menyetikkan kode program untuk setiap objek.
2. Resiko kesalahan kecil karena melakukan pengetikan lebih sedikit dan beberapa objek sudah disediakan di dalam menu pilihan yang tinggal dipilih sesuai kebutuhan.
3. Bisa dilakukan daur ulang. Setiap objek dapat digunakan secara berulang-ulang dalam program yang sama maupun program yang berlainan.

2.8.2 *Unified Modeling Language* (UML)[2]

Unified Modeling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem yang terkait dengan objek (Whitten L. Jeffery et al, 2004). Bahasa pemodelan UML lebih cocok untuk pembuatan perangkat lunak dalam bahasa pemrograman berorientasi objek (C+, Java, VB.NET), namun demikian tetap dapat digunakan pada bahasa pemrograman prosedural (Ziga Turck, 2007).

UML biasa digunakan untuk (Henderi, 2007:11) :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan aktor.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.

3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model *behavior* ”yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan/memperluas *fungsi* dengan *stereotypes* (Ziga Turck, 2007).

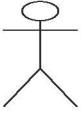

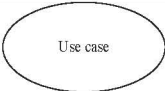



Pemodelan menggunakan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan menggunakan UML merupakan pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah.

Penjelasan mengenai berbagai diagram UML serta tujuannya dapat dijelaskan sebagai berikut:

1. Model Use Case Diagram

Use case diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *use case diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Use case* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi. *Use case diagram* dibuat untuk memvisualisasikan/menggambarkan hubungan antara aktor dan *use case*. Aktor adalah para pengguna (*users*) dari sebuah sistem/seseorang atau sesuatu yang harus berinteraksi dengan sistem atau sistem yang dibangun/dikembangkan. Kadangkala sebuah sistem merupakan aktor bagi sistem yang lain, beri nama aktor sistem tersebut dengan stereotipe (bentuk klise/tiruan) aktor. Notasi simbol-simbol yang digunakan dalam diagram *use case* dapat dilihat pada tabel 2.2 berikut ini:

Tabel 2.2 Notasi Simbol dalam Diagram *Use Case*

Lambang	Keterangan
	Aktor (<i>actor</i>) merupakan sebuah entitas yang berinteraksi dengan <i>use case</i> . Nama aktor dituliskan di bawah gambar tersebut. Aktor dapat berupa orang atau sistem lain diluar sistem yang tengah dianalisis.
	Garis putus-putus dengan panah di salah satu ujungnya menggambarkan hubungan antar <i>use case</i> . Hubungan tersebut dapat berupa <i>include</i> , <i>extend</i> ataupun <i>generalization</i> . Hubungan <i>include</i> berarti <i>use case</i> yang ditunjukkan oleh garis ikut dikerjakan jika <i>use case</i> sumber dikerjakan. Hubungan <i>extend</i> berarti <i>use case</i> yang ditunjuk oleh <i>use case</i> dapat memanggil <i>use case</i> yang menunjuk jika persyaratannya terpenuhi. <i>Generalization</i> berarti menunjukkan tujuan bahwa <i>use case</i> yang ditunjuk merupakan bentuk umum dari <i>use case</i> yang menunjuk. Keterangan <i>include</i> , <i>extend</i> maupun <i>generalization</i> dituliskan dalam dua buah kurung sudut "<< >>".
	<i>Use case</i> menggambarkan sebuah fungsi tertentu yang disediakan oleh sistem, sebuah subsistem atau urutan pertukaran pesan antar anggota sistem dan satu atau lebih aktor melakukan aksi yang dikerjakan oleh sistem.
	Hubungan, menggambarkan hubungan <i>association</i> . Garis ini digunakan untuk menghubungkan antara aktor dengan <i>use case</i> . Hubungan ini berarti aktor menggunakan <i>use case</i> .
	Garis putus-putus, menunjuk kepada objek yang dijelaskan oleh catatan.
	Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.








2. Diagram Struktur Statis

UML menawarkan dua diagram untuk memodelkan struktur statis sistem informasi, yaitu:

- a. *Class Diagram*: menggambarkan struktur *object* sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut. *Class* merepresentasikan sebuah abstraksi dari entitas-entitas dengan sifat-sifat atau

karakteristik yang bersifat umum. Notasi simbol-simbol yang digunakan dalam *class diagram* dapat dilihat pada tabel 2.3 berikut ini:

Tabel 2.3 Notasi Simbol dalam *Class Diagram*

Lambang	Keterangan
	Paket merupakan elemen yang dapat digunakan untuk berbagai keperluan. Fungsi paket adalah untuk mengorganisasikan elemen-elemen model ke dalam sebuah grup. Sebuah sistem dapat dianggap sebagai sebuah paket.
	Kelas adalah bentuk umum dari objek-objek yang memiliki struktur dan tingkah laku (atribut, operasi, relasi dan arti/definisi) yang sama. Kelas merupakan bentuk abstrak dari objek di dunia nyata.
	Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.
	Garis dengan mata panah tertutup pada salah satu ujungnya menggambarkan <i>generalization</i> . Kelas yang ditunjuk oleh mata panah merupakan kelas induk (<i>superclass</i>).
	Garis putus-putus dengan mata panah tertutup di salah satu ujungnya menggambarkan relasi.
	Garis solid tanpa mata panah menggambarkan asosiasi. Asosiasi menggambarkan bahwa garis dapat dituliskan kardinalitas dari kelas dihubungkan. Kardinalitas tersebut dapat berupa n,0,1,0..n, 1..n, 0..1
	terdapat jalur untuk berkomunikasi Pada ujung garis putus-putus dari catatan menunjukkan objek apa yang dijelaskan.

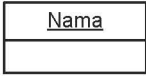

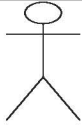



- b. *Object Diagram*: serupa dengan *class diagram*, tetapi *object diagram* memodelkan *instance object actual* dengan menunjukkan nilai-nilai saat ini dari atribut *instance*. *Object diagram* menyajikan “*snapshot*/potret” tentang objek sistem pada point waktu tertentu. Diagram ini tidak digunakan sesering *class diagram*, tetapi saat digunakan dapat membantu seorang developer memahami struktur sistem secara lebih baik.

3. Diagram Interaksi

Diagram interaksi memodelkan sebuah interaksi, terdiri dari satu set objek, hubungan-hubungannya, dan pesan yang terkirim di antara objek. Model diagram ini memodelkan *behavior* (kelakuan) sistem yang dinamis dan UML memiliki dua diagram untuk tujuan ini, yaitu:

a. Diagram rangkaian/*Sequence Diagram*: secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah use case atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi atau timing apa. Notasi simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel 2.4 berikut ini:

Tabel 2.4 Notasi Simbol dalam *Sequence Diagram*

Lambang	Keterangan
	Objek, dengan nama dari objek tersebut terdapat di dalamnya.
	Pesan, menunjukkan hubungan antara satu objek dengan objek lainnya. Arah pesan menunjukkan tujuan dari pesan tersebut.
	Objek aktor, merupakan objek yang berinteraksi sebagai pengguna sistem. Nama objek dituliskan di bawah gambar yang bersangkutan.
	Garis putus-putus, menunjuk kepada objek yang dijelaskan oleh catatan.
	Pesan respon, menunjukkan hubungan antara satu objek dengan objek lainnya. Arah pesan menunjukkan tujuan dari pesan tersebut.
	Catatan, berisi keterangan tambahan. Garis putusputus menunjukkan apa yang dijelaskan pada catatan ini.

b. Diagram kolaborasi/*Collaboration Diagram*: serupa dengan diagram rangkaian/sekuensi, tetapi tidak fokus pada timing atau sekuensi pesan. Diagram ini justru menggambarkan interaksi (atau kolaborasi) antara objek dalam sebuah format jaringan.





Diagram rangkaian maupun diagram kolaborasi merupakan *isomorphic* artinya kita dapat mengubah dari satu diagram ke diagram lain.




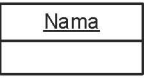
4. Diagram State/State Diagram

UML memiliki sebuah diagram untuk memodelkan *behavior* objek khusus yang kompleks (*statechart*) dan sebuah diagram untuk memodelkan *behavior* dari sebuah *use case* atau sebuah metode, yaitu:

- a. *Diagram statechart*: digunakan untuk memodelkan *behavior* objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan *event-event* (kejadian) yang menyebabkan objek beralih dari satu *state* ke *state* lain.
- b. *Diagram aktivitas/Activity Diagram*: secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut. Notasi simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel 2.5 berikut ini:

Tabel 2.5 Notasi Simbol dalam *Activity Diagram*

Lambang	Keterangan
	Awal (<i>Initial State</i>), menunjukkan tempat dimulainya diagram. Lambang ini boleh diberi label dengan isi berupa nama kejadian (<i>event</i>) yang memanggilnya. Sebuah diagram aktivitas hanya boleh memiliki sebuah awal.
	Akhir (<i>Final State</i>), menunjukkan tempat berakhirnya diagram. Lambang ini tidak memerlukan label. Diagram aktivitas dapat memiliki satu atau lebih akhir.
	Aktivitas (<i>activity</i>) adalah bentuk khusus dari keadaan (<i>state</i>) dimana aktivitas menggambarkan kegiatan yang dilakukan. Nama kegiatan dituliskan di tengah lambang.
	Keputusan (<i>decision</i>) menggambarkan keputusan atas sebuah kondisi, pertemuan dari jalur kondisi yang mungkin. Jika digunakan untuk menggambarkan keputusan, maka jalur masuk yang diijinkan hanya

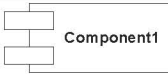

	satu sedangkan jalur keluar sebanyak dua atau lebih. Jalur-jalur keluar harus diberikan nilai yang harus terpenuhi, dengan maksimal satu buah jalur keluar tidak diberi nilai sebagai kemungkinan lainnya (<i>else</i>). Sedangkan jika digunakan untuk pertemuan dari jalur kondisi yang mungkin maka jalur masuk sebanyak dua atau lebih sedangkan jalur keluar hanya satu.
	Menyatakan alur aktivitas. Untuk alur yang memerlukan kondisi dituliskan kondisi yang harus dipenuhi dengan dibatasi oleh kurung siku “[dan]”.
	Garis putus-putus, menunjuk kepada objek yang dijelaskan oleh catatan.
	Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.
	Jalur unit (<i>swimlane</i>), menggambarkan unit organisasi dalam model bisnis. Setiap operasi/keadaan ditempatkan di dalam garis jalur sesuai dengan pemilik operasi/keadaan tersebut. Alur aktifitas diijinkan untuk melewati batas jalur untuk menuju unit lainnya.


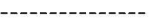
5. Diagram Implementasi

Diagram implementasi juga memodelkan struktur sistem informasi, yaitu:

- Diagram komponen *Component Diagram*: digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen *software* sistem. Komponen diagram dapat digunakan untuk menunjukkan bagaimana kode pemrograman dibagi menjadi modul-modul (komponen). Notasi simbol-simbol yang digunakan dalam *component diagram* dapat dilihat pada tabel 2.6 berikut ini:

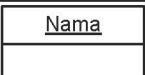

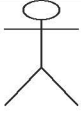

Tabel 2.6 Notasi Simbol dalam *Component Diagram*

Lambang	Keterangan
 Component1	Komponen, mewakili modul perangkat lunak (kode sumber, kode biner, program eksekusi, DLL, dan lain sebagainya).
 Package1	Paket komponen, mewakili tempat dari suatu komponen. Nama paket komponen dapat berisi nama dari direktori <i>file system</i> .

	Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.
	Garis putus-putus. Menunjuk kepada objek yang dijelaskan pada catatan.

- b. Diagram penguraian/*Deployment*: digunakan untuk mendeskripsikan arsitektur fisik dalam istilah "node" untuk *hardware* dan *software* dalam sistem. Diagram ini menggambarkan konfigurasi komponen-komponen *software real-time*, prosesor, dan peralatan yang membentuk arsitektur sistem. Notasi simbol-simbol yang digunakan dalam *deployment diagram* dapat dilihat pada tabel 2.7 berikut ini:

Tabel 2.7 Notasi Simbol dalam *Deployment Diagram*

Lambang	Keterangan
	Prosesor, menggambarkan perangkat keras yang mampu mengeksekusi program. Setiap prosesor harus memiliki nama yang diletakkan di dalam lambang prosesor.
	<i>Device</i> , merupakan perangkat keras yang tidak memiliki kemampuan untuk melakukan penghitungan/pemrosesan. Contoh device adalah modem, monitor dan juga speaker.
	Penghubung, digambarkan oleh sebuah garis yang menghubungkan antara sebuah prosesor dengan prosesor, ataupun prosesor dengan <i>device</i> . Sebuah penghubung boleh diberi nama.
	Garis putus-putus, menunjuk kepada objek yang dijelaskan oleh catatan.