BAB II

TINJAUAN PUSTAKA

Teori merupakan dasar yang digunakan dalam penyusunan sebuah sistem. Pada bab ini akan dijelaskan mengenai tinjauan pustaka dan teori-teori yang menunjang proses pembangunan aplikasi e-learning berbasis web di LSI Baitul Mu'min Antapani.

2.1. Tinjauan Organisasi

2.1.1. Sejarah Singkat

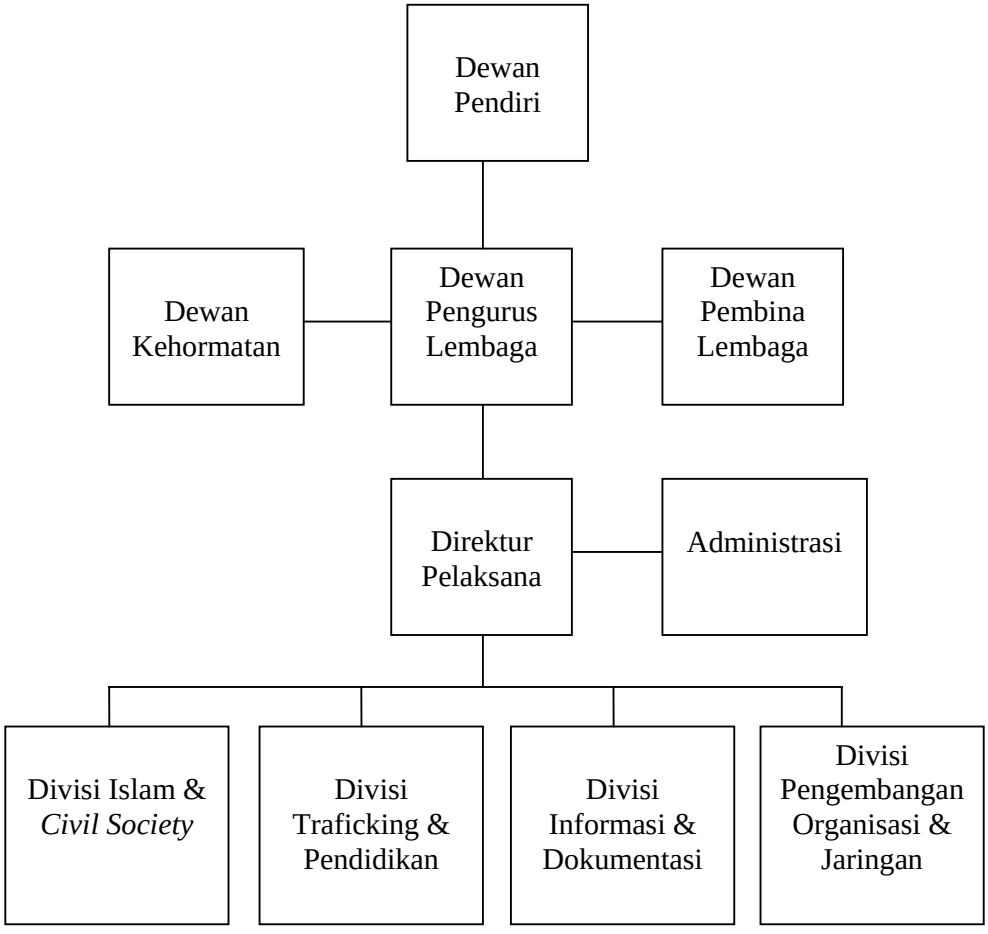
Pada pertengahan 2003 di Antapani (Bandung), terjadi diskusi kecil yang berawal dari adanya kegelisahan intelektual dalam melihat berbagai fenomena sosial, khususnya persoalan pembangunan di negara-negara berkembang. Kegelisahan itu berasal dari munculnya pandangan tentang model-model pembangunan yang diterapkan, di mana pada dirinya juga membawa nilai-nilai di mana model, strategi dan pendekatan pembangunan tersebut ditemukan.

Untuk mengkaji persoalan-persoalan itu muncullah sebuah gagasan untuk melihat pada nilai-nilai apa apa yang menjadi landasan suatu Negara dalam menerapkan sebuah model, strategi dan pendekatan-pendekatan di dalam membangun maysarakat, bangsa dan negara. Pada perkembangannya kemudian kegiatan diskusi informal tersebut untuk membentuk sebuah lembaga studi. Maka pada 16 Desember 2003 dibentuklah sebuah lembaga studi yang diberi nama Lembaga Studi Islam (LSI) Baitul Mu'min Antapani.

2.1.2. Struktur Organisasi

Struktur organisasi merupakan suatu kerangka yang menggambarkan hubungan antara departemen atau divisi yang terkait dalam suatu organisasi dan biasanya digambarkan dalam bentuk grafik.

Struktur organisasi LSI Baitul Mu'min berdasarkan Surat Keputusan Nomor 100/752.01/SO0001/01/2007, dapat dilihat pada gambar 2.1



Gambar 2.1 Struktur Organisasi LSI Baitul Mu'min

2.1.3. Deskripsi Pekerjaan

Lembaga Studi Islam (LSI) adalah suatu lembaga yang melakukan analisis tentang keislaman dan mendakwahkannya. Dalam lembaga ini dilakukan kegiatan studi keislaman, training dan dakwah. Dalam hal penyampaian materi, lembaga ini menggunakan metode *Quantum Learning*, yaitu metode yang bertujuan untuk mempermudah proses belajar dan penyampaian materi secara interaktif.

2.2. Landasan Teori

2.2.1. Pengertian E-Learning

E-learning terdiri dari dua bagian, yaitu ‘e’ yang merupakan singkatan dari *elektronic* dan *learning* yang berarti pembelajaran. Jadi e-learning berarti pembelajaran dengan menggunakan jasa bantuan perangkat elektronika, khususnya perangkat komputer. Karena itu, maka e-learning sering disebut pula dengan *‘online course’*. Dalam berbagai literatur, e-learning didefinisikan sebagai berikut :

E-learning is a generic term for all technologically supported learning using an array of teaching and learning tools as phone bridging, audio and videotapes, teleconferencing, satellite transmissions, and the more recognized web-based training or computer aided instruction also commonly referred to as online courses (Soekartawi, 2003).

Dengan demikian maka e-learning atau pembelajaran melalui online adalah pembelajaran yang pelaksaanya didukung oleh jasa teknologi seperti telepon, audio, videotape, transmisi satelit atau komputer.

Dalam perkembanganya, komputer dipakai sebagai alat bantu pembelajaran, karena itu dikenal dengan istilah *Computer Based Learning* (CBL) atau *Computer Assisted Learning* (CAL). Saat pertama kali komputer mulai diperkenalkan khususnya untuk pembelajaran, maka komputer menjadi popular dikalangan anak didik. Hal ini dapat dimengerti karena berbagai variasi teknik mengajar bisa dibuat dengan bantuan kompter tersebut. Maka setelah itu teknologi pembelajaran terus berkembang dan dikelompokan menjadi dua yaitu :

1. *Technology-based learning*
2. *Technology-based Web-learning*

Technology based-learning ini pada prinsipnya terdiri dari dua, yaitu *audio information technologies* (audio tape, radio, voice mail, telepon) dan *video information technologies* (video tape, nideo text, video messaging). Sedangkan *technology based web-learning* pada dasarnya adalah *data information technologies* (*bulletin board*, internet, email, *tele-collaboration*).

Dalam pelaksanaan pembelajaran sehari-hari, yang sering dijumpai adalah kombinasi dari teknologi yang dituliskan di atas (audio/data, video/data, audio/video). Teknologi ini juga sering dipakai pada pendidikan jarak jauh, dimaksudkan agar komunikasi antara murid dan guru bisa terjadi dengan keunggulan teknologi e-learning ini. Sedangkan interaksi antara guru dan murid bisa dilaksanakan melalui cara langsung (*synchronous*) atau tidak langsung, misalnya pesan direkam dahulu sebelum digunakan. Cara ini dikenal dengan nama e-synchronous.

Karakteristik e-learning antara lain adalah:

1. Memanfaatkan jasa teknologi elektronik; dimana guru dan siswa, siswa dan sesama siswa atau guru dan sesama guru dapat berkomunikasi dengan relatif mudah dengan tanpa dibatasi oleh hal-hal yang protokelor
2. Menggunakan bahan ajar bersifat mandiri (self learning materials) disimpan di komputer sehinga dapat diakses oleh guru dan siswa kapan saja dan dimana saja dan yang bersangkutan memerlukanya; dan
3. Memanfaatkan jadwal pembelajaran, kurikulum, hasil kemajuan belajar dan hal-hal yang berkaitan dengan administrasi pendidikan dapat dilihat setiap saat di komputer.
4. Menurut Miarso (2004), Pemanfaatan e-learning tidak terlepas dari jasa internet. Karena teknik pembelajaran yang tersedia di internet begitu lengkap, maka hal ini akan mempengaruhi terhadap tugas guru dalam proses pembelajaran. Dahulu, proses belajar-mengajar didominasi oleh peranan guru, karena itu disebut *the era of teacher*. Kini, proses belajar-mengajar, banyak didominasi oleh peran guru dan buku (*the era of teacher and book*) dan pada masa mendatang proses belajar mengajar akan didominasi oleh guru, buku, dan teknologi (*the era of teacher, book, and technology*).

2.2.2. Unified Modeling Language (UML)

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

Dengan menggunakan UML kita 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 ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

2.2.2.1. Sejarah UML

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi wirfs-brock [1], metodologi booch [2], metodologi coad [3], metodologi OOSE [4], metodologi OMT [5], metodologi shlaer-mellor [6], dsb. Masa itu terkenal dengan masa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa

notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – http://www.omg.org). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999 [7] [8] [9]. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

2.2.2.2. Konsepsi Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa dirangkum dalam tabel berikut ini :

Tabel 2.1 Abstraksi Konsep Dasar UML

Major Area	*View*	*Diagrams*	*Main Concepts*
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
	deployment view	deployment diagram	node, component, dependency, location
dynamic	state machine view	statechart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
		collaboration diagram	collaboration, interaction, collaboration role, message
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams*. *Main concepts* bisa kita pandang sebagai *term* yang akan muncul pada saat kita membuat diagram. Dan *view* adalah kategori dari diagaram tersebut.

2.2.2.3. Diagram-diagram UML

a. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

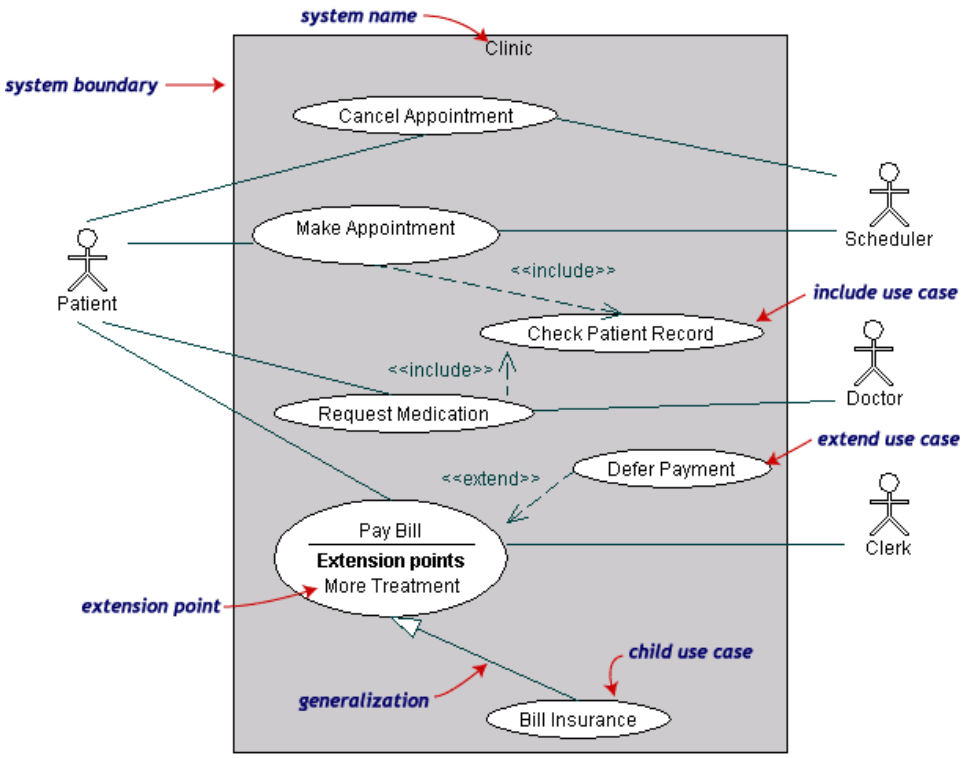
Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Contoh *use case diagram* :



Gambar 2.2. Contoh Use Case Diagram

b. 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/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

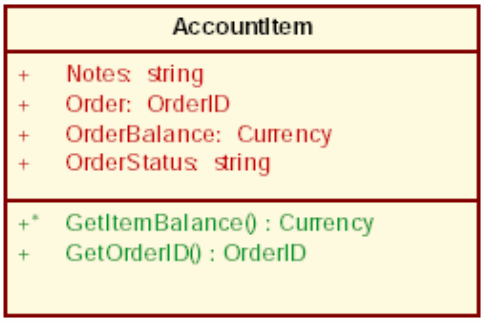
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

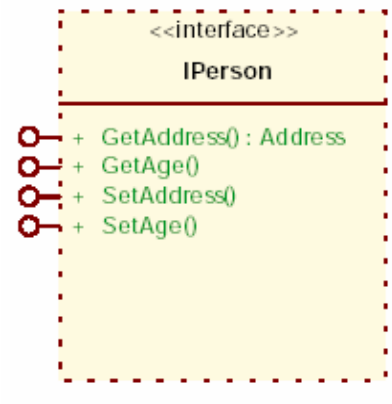
Atribut dan metoda dapat memiliki salah satu sifat berikut :

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



Gambar 2.3. Atribut dan Metoda dalam Class Diagram

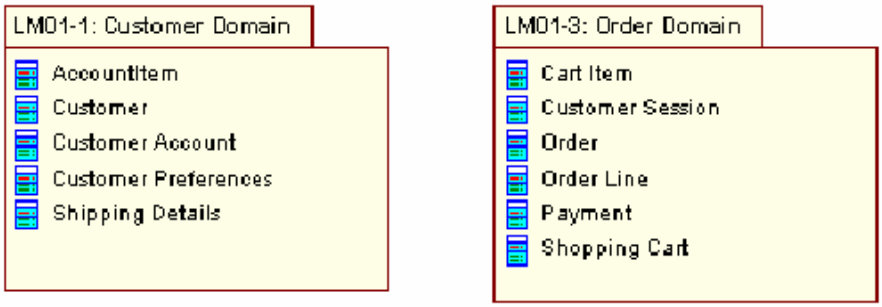
Class dapat merupakan implementasi dari sebuah *interface,* yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.



Gambar 2.4. Contoh Interface

Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*.

Contoh *Package* :



Gambar 2.5. Contoh Package

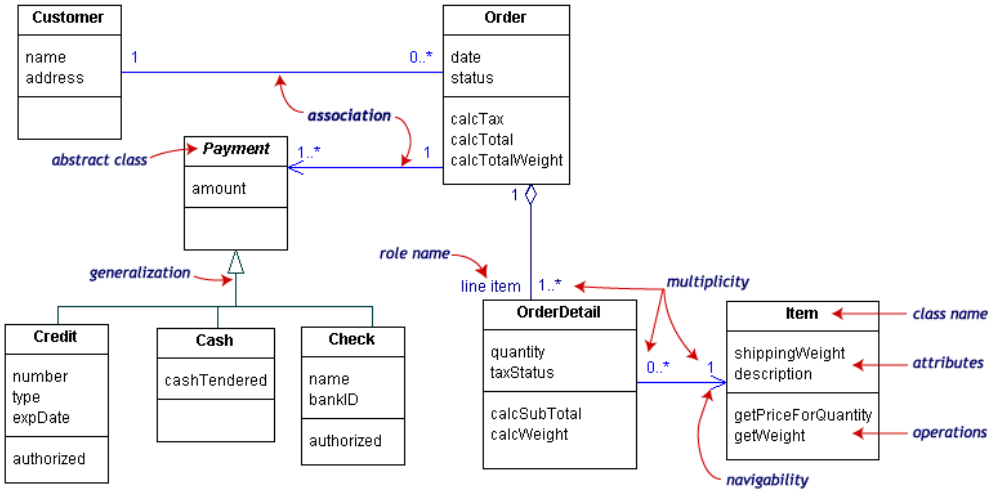
Hubungan Antar *Class* :

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan

menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

Contoh *class diagram* :



Gambar 2.6. Contoh Class Diagram

c. Statechart Diagram

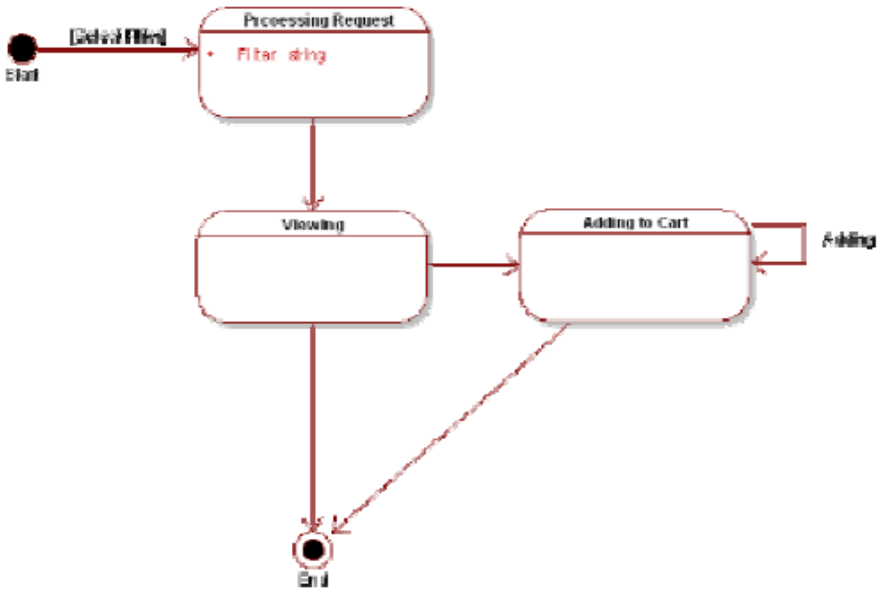
Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang

bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring.

Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

Contoh *statechart diagram* :



Gambar 2.7. Contoh Statechart Diagram

d. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

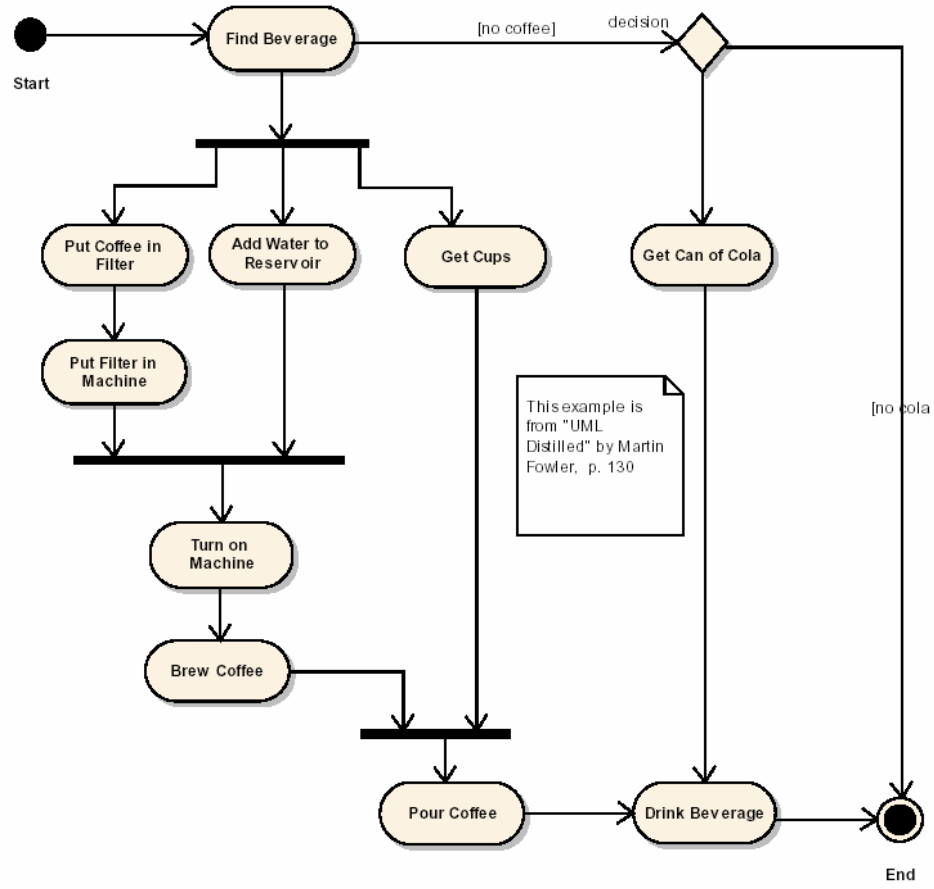
Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak

menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar 2.8. Contoh Activity Diagram Tanpa Swimlane

e. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas

tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

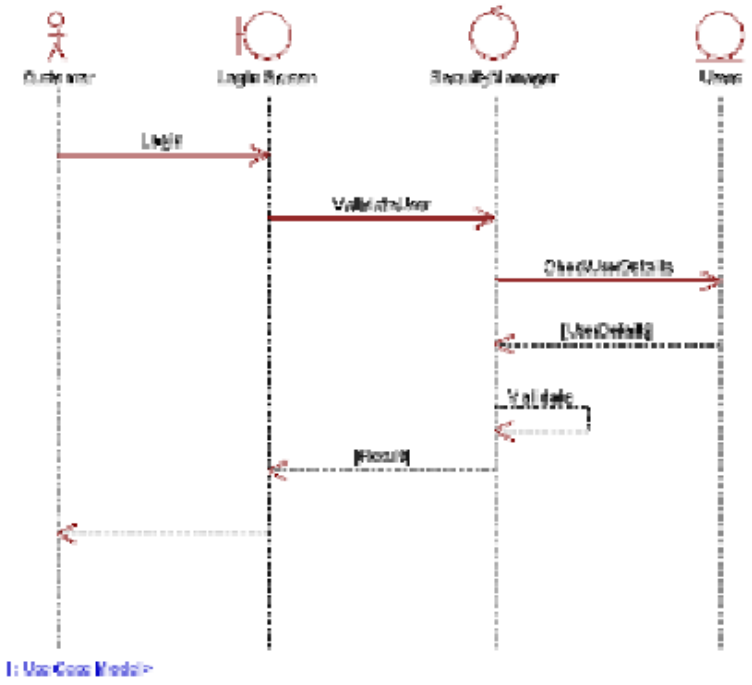
Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya

sebuah message.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

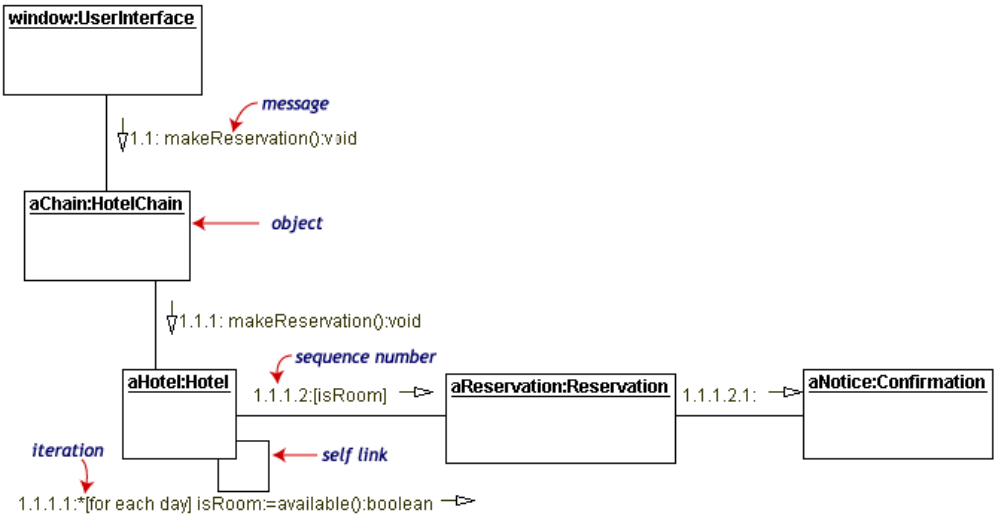
Contoh *sequence diagram* :



Gambar 2.9. Contoh Sequence Diagram

f. Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.



Gambar 2.10. Contoh Collaboration Diagram

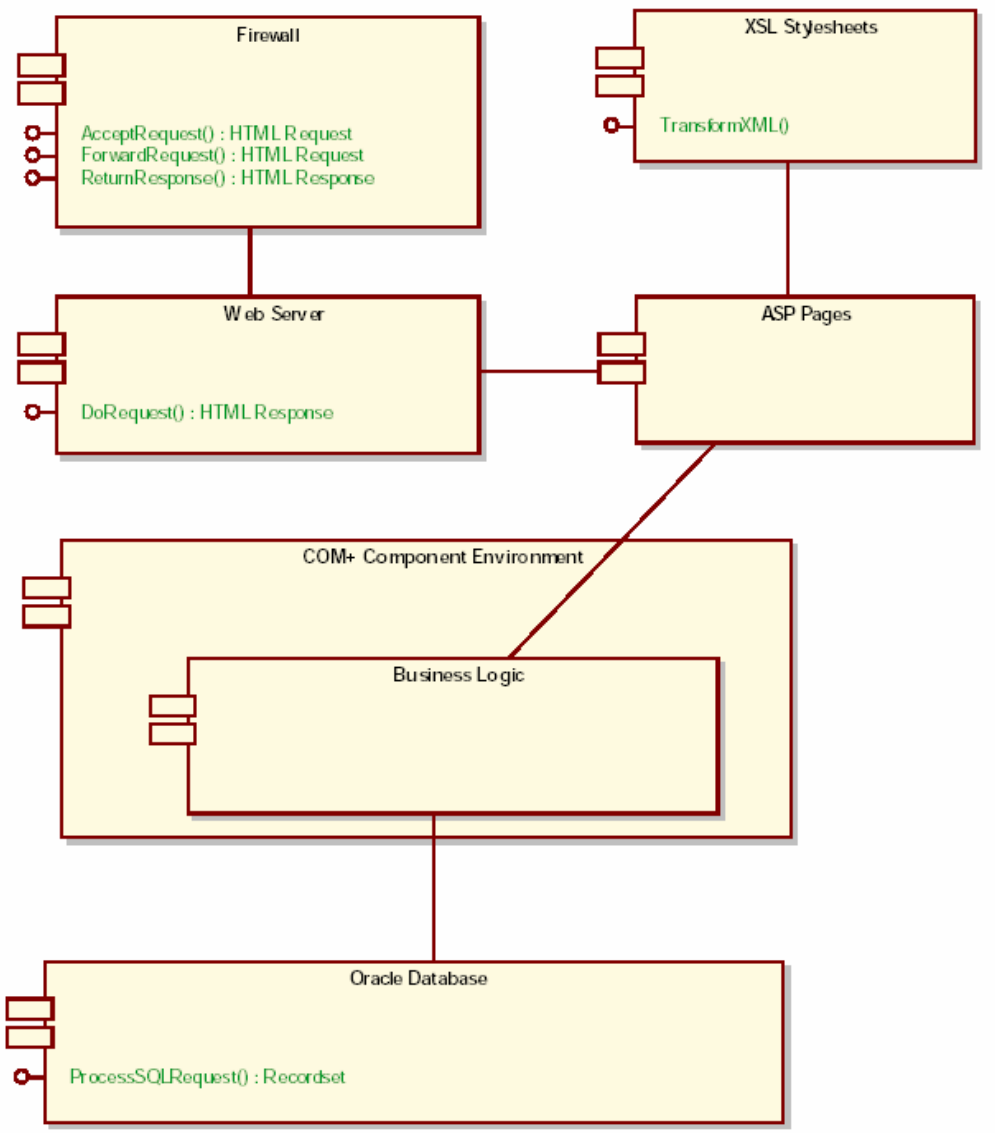
g. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.

Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang

lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

Contoh *component diagram*:



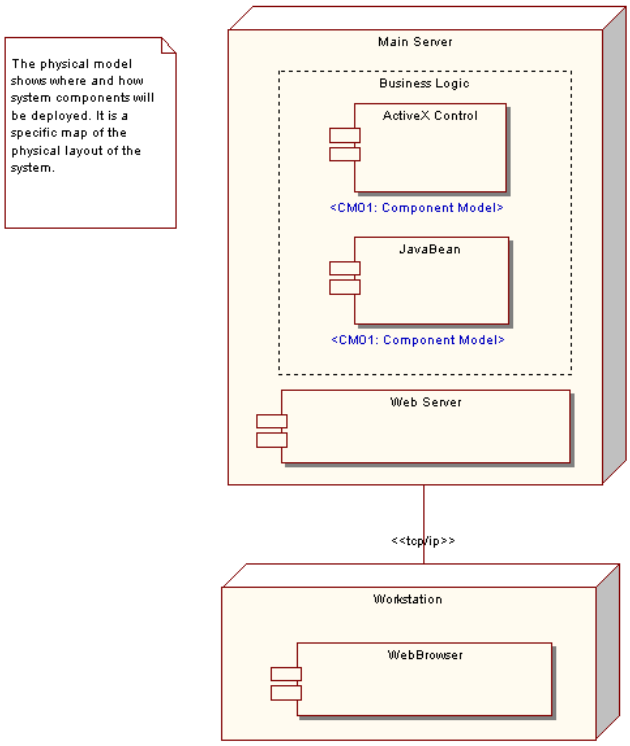
Gambar 2.11. Contoh Component Diagram

h. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal

Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

Contoh *deployment diagram* :



Gambar 2.12. Contoh Deployment Diagram

2.2.3. Web

2.2.3.1. Sejarah Web

Sejarah Web dimulai pada bulan Maret 1989 ketika Tim Berner-Lee yang bekerja di Laboratorium Fisika Partikel Eropa atau yang dikenal dengan nama *Consei European pour la Recherce Nuclaire* (CERN) yang berada di Genewa, Swiss, mengajukan protokol atau suatu tatacara untuk berkomunikasi sistem distribusi informasi internet yang digunakan untuk berbagai informasi diantara para fisikawan.

Protokol inilah yang yang selanjutnya dikenal sebagai protokol *World Wide Web* (WWW) dan dikembangkan oleh *World Wide Web Consortium* (W3C). Dimana W3C adalah konsorsium dari sejumlah organisasi yang berkepentingan dalam pengembangan berbagai standar yang berkaitan dengan Web.

W3C meletakan gabungan spesifikasi dalam standar web, berikut adalah hasil dari W3C:

1. Standar web yang paling mendasar adalah HTML, *Cascade Style Sheet* (CSS), dan *Extended Markup Language* (XML)
2. Standar HTML yang terakhir adalah *Extended Hypertext markup Language* 1.0 (XHTML 1.0)

Namun sekarang CERN telah kembali pada misi awalnya melakukan penelitian pada fisika partikel, dan yang berkaitan dengan Web telah diserahkan kepada W3C, sekelompok akademis dan organisasi komersial yang mengabdikan pada kemajuan dari web.

2.2.3.2. Aplikasi Web

Aplikasi web pada awalnya dibangun hanya dengan menggunakan bahasa yang disebut dengan HTML dan protokol yang digunakan yang dinamakan *Hyper Text Transfer Protocol* (HTTP). Pada perkembangan berik utnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML, antara lain yaitu PHP dan *Active Server Pages* (ASP), sedangkan contoh yang berupa objek antara lain adalah applet (Java).

Aplikasi Web sendiri dapat dibagi menjadi dua, yaitu:

1. Web Statis
2. Web Dinamis

Web statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus-menerus untuk mengikuti setiap perubahan yang terjadi, karena suatu web dapat cepat sekali populer dikalangan pengguna internet apabila terdapat kemudahan yang disediakan untuk pengguna web dalam melakukan penelusuran, penjelajahan dan pencarian informasi (surfing) dan juga suatu web akan dikenal dengan cepat apabila informasi yang disajikan selalu up to date dan atau lengkap, semua kelemahan ini dapat diatasi dengan model aplikasi Web Dinamis yang tidak memerlukan pemeliharaan program secara terus menerus untuk mengikuti perubahan yang terjadi. Salah satu bahasa yang sering digunakan untuk aplikasi web dinamis adalah PHP, CSS dan banyak lagi.

2.2.4. Hyper Text Markup Language (HTML)

HTML adalah bahasa yang digunakan untuk mendisain dan memformat halaman web. Kita mungkin sudah mendengar tentang bahasa program seperti C, C++, Java, dan Visual basic. Masing-Masing bahasa ini terdiri dari perintah sintak dan programming. Sintak ini adalah yang programmer sering digunakan untuk memanggil kode. Sangatlah penting mengetahui bagaimana cara menulis kode menggunakan bahasa yang relevan. Lebih dari itu, kita harus konvensional dengan aturan menyangkut bahasa tertentu. Didalam HTML, sintak ini disebut tag.Tag ditulis dengan tanda-kurung bersudut < dan >.Ada kelompok tag yang sudah dikenal didalam HTML, yang mana digunakan untuk berbagai tujuan.Sebagai contoh, dalam rangka memodifikasi satu baris teks ke dalam Bold, kita menerapkan tag bold dengan suatu tag , kemudian tulis beberapa teks atau suatu paragraph yang berisi beberapa teks, dan tutup tag menggunakan tag seperti dalam contoh listing 1.

Semua tag didalam HTML harus ditutup menggunakan sintak </>.Tetapi ada beberapa perkecualian pada aturan ini.Kita memakai tag ini sebab HTML bukanlah bahasa yang sensitip seperti C++ dan Java.

Penerapan Fasilitas Pada HTML

Berikut Fasilitas yang disediakan HTML:

1. Penerapan Gambar : Teknik yang digunakan untuk menyisipkan gambar pada halaman HTML adalah dengan menggunakan tag <IMG> yang kemudian diikuti dengan perintah SRC yang bertujuan untuk menerapkan alur dari file gambar tersebut.

2. Pengisian Backgroud dengan warna dan gambar : Kita dapat membuat background dari halaman web kita dengan suatu warna yang ditetapkan dgn menggunakan bgcolor didalam tag <body>. Dengan cara yang sama, jika memerlukan suatu gambar untuk mengubah backgrounddari halaman web, atribut background menggunakan tag <body> .Sedangkan untuk pewarnaan juga bisa menggunakan kode hexadesimal 6 digit dan diawali dengan Pagar:#FF0000 = akan menampilkan warna merah (2 digit awal bernilai maksimal),#00FF00 = akan menampilkan warna hijau (2 digit tengah bernilai maksimal),#0000FF = akan menampilkan warna biru (2 digit terakhir bernilai maksimal),nilai minimal adalah 0 (nol) dan maksimal adalah F.
3. Tabel : Salah satu corak HTML yang unik yaitu kemampuannya menciptakan tabel. Developer dengan mudah mengkonversi data yang diperlukan ke dalam suatu format menggunakan tag <TABEL>, <TR>, <TH> dan <TD>.untuk membuat tabel

   Berikut beberapa perintah yang digunakan untuk memodifikasi tabel :

   a. Border : Mengbah garis tepi dari tabel. Default border 1.

   b. Bgcolor : Mengubah warna latar belakang dari tabel.

   c. align : Baris tabel untuk left, center, right. Default align left.

   d. Cellpadding : Menandai ruang;spasi antar isi dan perbatasan dari sel.

   e. Cellspacing : Menandai ruang;spasi antar dua sel.
4. Frames : Frames memberikan kemampuan untuk membuat jendela browser baru ke dalam dua atau lebih dengan membingkai. penggunaan tag

<FRAMESET>. Tag ini membantu untuk menetap kandivisi, baris(ROWS Atribut) atau kolom (COLS Atribut) dari WINDOWS.

Penggunaan Atribut dengan menggunakan tag Frameset :

1. Cols : Untuk membagi jendela ke dalam kolom.
2. Bordercolor : mengUbah batas mewarnai dari batas bingkai.
3. Frameborder : Nilai-Nilai 0 atau 1. batasan bingkai akan jadi tak kelihatan

5. Teks Field : Unsur ini digunakan untuk teks seperti usernames dan lain lain. atribut utama adalah type (value= teks), size (value= “berupa angka”), Maxlength (value= “berupa angka”).
6. Tombol : Salah satu dari unsur form yang paling utama adalah tombol. Tombol melayani bagian integral didalam suatu form HTML.Sebagian besar digunakan untuk mengambil tindakan. Terdapat 2 jenis tombol, Submit dan Reset. Tombol submit membawa keseluruhan data form pada server sedang tombol reset membersihkan semua masukan sebelumnya.
6. Combo box : Kotak ini memungkinkan kita untuk mengambil suatu item dari satu set nilai. Diterapkan penggunaannya dengan TAG <SELECT>.
7. checkbox : checkbox memungkinkan untuk memilih daftar materi. Dengan pilihan lebih dari 1.
8. RadioButton : Radio button mempunyai fungsi yang hampir sama dengan checkbox akan tetapi pemilihan materi hanya memungkinkan untuk memilih satu pilihan.
9. Pengaturan Teks : Penggunaan tag <PRE></PRE> ke text form mewakili Teks Preformatted. Apapun juga yang kita masukkan pada tag ini akan

diperlihatkan seperti halnya pada browser. Tag ini digunakan untuk penulisan programming kode.

10. Teks Berjalan : Teks dan Gambar dapat dipindahkan dengan bantuan tag <MARQUEE> </MARQUEE>. Tag ini menerima banyak atribut ditunjukkan di bawah ini:
    a. Behavior (Perilaku) : Atribut ini berubah arah dari objek bergerak. Nilai-Nilainya dapat bernilai left, right, top dan bottom.
    b. Scroll delay : Menandai adanya kecepatan dari gerakkan obyek. Yang secara khas, nilainya ditetapkan didalam seperseribu detik. Dan bahwa 1 detik/second sama dengan 1000 seperseribu detik.
    c. Loop (Pengulangan) : Loop (Pengulangan) menetapkan banyaknya, obyek atau teks tertentu perlu pindah/bergerak dengan bantuan dari atribut ini. Suatu nilai -1 menandai adanya gerakkan teks.
11. Addresstag : Pemberian informasi kontak pada suatu halaman web, dapat menggunakan tag <ADDRESS></ADDRESS>. Menampilkan teks didalamnya. Karenanya, tidak ada yang dibutuhkan untuk form teks secara terpisah
12. Font : Merubah penampilan dari teks menggunakan tag. Atributnya adalah face, size dan color. Atribut face menerima font seperti Arial, Courier, Verdana dan lain-lain. Size menerima nilai-nilai dari 1 sampai 7 sedangkan color menetapkan warna font.

2.2.5. PHP

PHP terkenal dengan empat kelebihannya :

1. Practical/Praktis : PHP adalah bahasa pemrograman yang sangat longgar dalam penulisan, dan ini meningkatkan kepraktisan buat para penggunanya. Programmer tidak diharuskan untuk menuliskan atau menghapus variabel. Walaupun terkadang juga tidak bisa mengatakan dengan mudah yang kemudian dipanggil dalam sebuah skrip, menebak formula terbaik dalam penetapan variabel secara otomatis kemudian menghapus variabel dan dan mengembalikan resource ke sistem setelah skrip berhasil di eksekusi. Pada akhirnya, PHP mampu membuat programmer lebih berpikir pada tujuan akhir dari project yang akan dibuat.
2. Power : PHP mampu membuat halaman dinamis, memanipulasi form, dan dapat dihubungkan dengan database. Selain yang disebutkan tadi, ternyata PHP juga mampu melakukan hal-hal dibawah ini :
   a. Membuat dan memanipulasi file Macromedia Flash, gambar, dan *Portabel Document Format* (PDF).
   b. Mengevaluasi sebuah password. Misal: Apakah sebuah password yang dibuat, gampang ditebak atau tidak.
   c. Berkomunikasi dengan LDAP.
   d. Menguraikan string, bahkan yang kompleks sekalipun, dengan menggunakan POSIX dan Perl Based Regular Expression Library.
   e. Berkomunikasi dengan banyak protokol, termasuk IMAP, POP3 dan NNTP.

3. Possibility : Jarang ada developer PHP yang terikat pada suatu implementasi pemecahan masalah. Dilain sisi, ada banyak pilihan yang ditawarkan oleh PHP. Contohnya ada pada database yang didukung oleh PHP. Kurang lebih semuanya ada 25 database, termasuk Adabas D, dBase, Empress, FilePro, FrontBase, Hyperwave, IBM DB2, Informix, Ingres, Interbase, mSQL, direct MS-SQL, MySQL, Oracle, Ovrimos, PostgreSQL, Solid, Sybase, Unix dbm, dan Velocis. Kemampuan string-parsing juga bisa dianggap sebagai banyaknya kemungkinan yang ditawarkan oleh PHP. Dalam hal in, PHP memiliki lebih dari 85 function untuk memanipulasi string. Kelebihan ini tak hanya akan menawarkan keleluasaan untuk melakukan operasi string yang kompleks, namun juga menjembatani program yang memiliki fungsionalitas yang sama (seperti Python dan Perl) lewat PHP.

4. Price : PHP merupakan salah satu open source software, yang dapat diartikan sebagai berikut :

a. PHP dapat dimodifikasi, didistribusikan, dan diintegrasikan dengan produk lain oleh penggunanya.
b. Pengembangan dan auditing yang dilakukan secara terbuka. Semua orang bebas berpastisipasi.

2.2.6. CSS

CSS memungkinkan web developer untuk memisahkan HTML dari aturan-aturan untuk membentuk tampilan sebuah website. CSS (Cascading Style Sheet) digunakan untuk melengkapi file HTML, dan tugas utamanya adalah menetapkan aturan tampilan/style yang akan digunakan pada sebuah website.

CSS diperkenalkan untuk pengembangan website pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi style yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan parent-child pada setiap style,Setelah CSS distandarisasikan, Internet Explorer dan Netscape melepas browser terbaru yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS.

CSS adalah sebuah dokumen yang berdiri sendiri dan dapat dimasukkan dalam kode HTML atau sekedar mejadi rujukan oleh HTML dalam pendefinisian style. CSS menggunakan kode2 yang tersusun untuk menetapkan style pda elemen HTML atau dapat juga digunakan membuat style baru yang biasa disebut class.

CSS dapat mengubah besar kecilnya text, mengganti warna background pada sebuah halaman, atau dapat pula mengubah warna border pada tabel, dan masih banyak lagi hal yang dapat dilakukan oleh CSS. Singkatnya, CSS digunakan untuk mengatur susunan tampilan pada halaman HTML.

CSS dapat digunakan untuk menggantikan <font>, <b>, <u> dan <u>, dikarenakan hal berikut:sebuah file css dapat menjadi rujukan banyak halaman HTML. Hanya dibutuhkan satu baris kode untuk melakukan hal tersebut. Ini berarti akan meminimalkan file-file HTML yang akan dibuat. Jika ingin mengubah tampilan website yang telah dibuat, maka yang perlu dilakukan hanya mengganti baris-baris kode pada css nya saja, tanpa perlu mengutak-atik file-file HTML nya, CSS dapat mengatur banyak atribut pada sebuah halaman secara

mudah. Misalnya: warna background, border, shadow, yang berbeda pada masing-masing tag yang digunakan.

2.2.7. JavaScript

JavaScript adalah bahasa script yang berdasar pada object yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML dimana obyek tersebut dapat berupa suatu window, frame, URL, dokumen, form, button, atau item yang lain. Semuanya itu memiliki property yang saling berhubungan dan masing – masing memiliki nama, lokasi, warna, nilai dan atribut lain, contoh struktur bahasa Java Script:

```
<SCRIPT LANGUAGE="JavaScript">
<!—Begin
…
…   Syntax programme
…
//  End -->
</script>
```

2.2.8. Asynchronous Javascript and XML (Ajax)

Ajax merupakan singkatan dari “Asynchronous JavaScript and XML”, sebuah teknik dalam mengembangkan aplikasi web yang interaktif. Ajax sendiri bukan merupakan teknologi yang baru ditemukan, lebih tepatnya Ajax menggunakan gabungan beberapa teknologi dan pemrograman yang telah dikenal, seperti HTML, XML, JavaScript, DOM (Document Object Model), dan XMLHttpRequest. Ajax bekerja saat sebuah halaman web merespon suatu aksi, maka halaman tersebut harus melakukan pertukaran data dengan server. Hal

tersebut yang menyebabkan jeda waktu, dan kurang interaktifnya sebuah halaman website. Ajax mengatasinya dengan melakukan pertukaran data yang kecil dan dilakukan *behind the scene*, atau di belakang layer sehingga tidak mengganggu proses pertukaran data tersebut.

2.2.8.1. Pemrograman Ajax

Ajax sendiri bukanlah sebuah bahasa pemrograman yang baru, tetapi lebih merupakan sebuah teknik yang menggunakan standar yang telah ada, khususnya dalam hal ini adalah JavaScript dan HTTP request. Diperlukan pemahaman terhadap dasar pemrograman JavaScript untuk memulai Ajax. Di bawah ini adalah contoh sebuah script dari JavaScript: Script di atas adalah sebuah contoh function JavaScript yang sederhana. Script di atas menghasilkan sebuah kotak pesan “halo” pada browser, di mana script tersebut dapat digabungkan pada file HTML, dan dipanggil dengan suatu event. Misalnya event OnClick, OnLoad, dan lain sebagainya. Apabila telah mengenal HTML dan JavaScript, tentunya telah terbiasa menggunakan event-event pada elemen HTML, dan function-function JavaScript.

Sesuai dengan namanya, Ajax atau “Asynchronous JavaScript and XML”, pertukaran data antara server dan browser dilakukan secara asynchronous, di mana data yang dikirimkan dapat berupa sebuah plain text ataupun dalam format XML. Mengapa penting untuk mengetahui JavaScript? Karena di dalam JavaScript tersebut, akan memanggil object XMLHttpRequest, yang merupakan object inti dari Ajax. Object XMLHttpRequest inilah yang membangun koneksi antara browser dengan server, dan dilakukan behind the scene untuk men-

download data. Object XMLHttpRequest harus diinisiasi terlebih dahulu oleh web browser melalui JavaScript.

Dalam pembuatan object XMLHttpRequest, perlu dibedakan antara web browser yang digunakan, karena terdapat perbedaan dalam pembuatan object-nya. Web browser yang digunakan dapat dikategorikan menjadi 2 bagian, yaitu browser buatan Microsoft (Internet Explorer) dan browser yang non-Microsoft, seperti Mozilla Firefox, Opera, Netscape dan lain sebagainya. Di bawah ini adalah potongan kode Java- Script yang digunakan sebagai salah satu cara untuk melakukan penciptaan Object XMLHttpRequest. Tetapi cikal bakal kelahiran Ajax, tidak lain adalah pada saat dirilisnya Microsoft Internet Explorer 5. Microsoft memperkenalkan object XMLHttpRequest, yang merupakan object inti yang digunakan dalam Ajax.

Mungkin Anda penasaran, apa saja kemampuan yang ditawarkan Ajax? Dari sisi interface, banyak sekali yang dapat Anda lakukan dengan Ajax. Untuk melihat contoh live, Anda dapat mencari melalui search engine. Beragam website yang menggunakan Ajax. Secara acak, penulis mencari beberapa website yang mendemonstrasikan kemampuan Ajax, satu contoh dapat dilihat pada alamat http://www.mathertel.de/AJAXEngine/ S03_AJAXControls/TreeView.aspx, di mana terdapat hierarki data menyerupai Windows Explorer, untuk menunjukkan kota-kota di Amerika Serikat. Contoh lainnya terdapat pada salah satu variasi search engine Google, yang dikenal dengan Google Suggest yang beralamatkan di URL http://www.google.com/webhp?complete=1&hl=en.

Setiap mengetikkan sebuah huruf, Google Suggest secara otomatis akan menampilkan daftar kata yang kemungkinan ingin dicari, beserta dengan jumlah result-nya. Bisa dibilang, Google Suggest inilah yang mengawali pesatnya penggunaan Ajax Secara keseluruhan, contoh di atas hanyalah beberapa contoh bagaimana dengan Ajax, website Anda akan semakin interaktif dan menarik, menyerupai aplikasi desktop. Point yang penting adalah walau bisa jadi tetap terdapat waktu tunggu atau delay akibat proses pertukaran data, tetapi proses dilakukan behind the scene, sehingga tidak mengganggu kenyamanan.

2.2.8.2. Keuntungan dan Kerugian Ajax

a. Keuntungan Ajax

Adanya proses refresh pada halaman web tradisional mengindikasikan sebuah proses yang berat. Berbeda jika Anda menggunakan Ajax, di mana halaman dapat di-update secara otomatis, tanpa melakukan reload/refresh. Hal ini tentu saja mempercepat respon halaman terhadap interaksi dari pengguna.

```
var XMLHttpRequestObject = false; if (window.XMLHttpRequest) {
XMLHttpRequestObject = new XMLHttpRequest();
} else if (window.ActiveXObject) {
XMLHttpRequestObject = new ActiveXObject(“Microsoft.XMLHTTP”);
}
```

Kondisi if (window.XMLHttpRequest) akan bernilai true, jika web browser yang Anda gunakan adalah browser non-Microsoft, sementara kondisi if (window. ActiveXObject) akan bernilai true jika Anda menggunakan browser Microsoft Internet Explorer. Perbedaannya hanyalah pada cara pembuatan object

XMLHttpRequest. Anda harus meletakkan script di atas pada awal program, dan di luar dari semua function JavaScript, karena kode tersebut harus dieksekusi terlebih dahulu sebelum objectnya digunakan. Anda dapat menyimpan program Anda ke dalam fi le dengan ekstensi fi le, umumnya seperti *.html, *.asp, *.aspx, *.php, dan sebagainya. Sebagaimana umumnya sebuah object, object XMLHttpRequest yang Anda ciptakan memiliki *property* dan *method*. Terdapat dua property penting, yaitu property readyState dan status. Property ini menunjukkan keadaan object yang direpresentasikan dengan nilai numerik.

Kondisi-kondisi di atas perlu diketahui, untuk menentukan langkah-langkah selanjutnya pada program. Terutama pada saat property readyState bernilai 4 (*Complete*) dan property status bernilai 200 (*OK*), yang merupakan kondisi di mana object XMLHttpRequest berjalan sebagaimana diharapkan sehingga Anda dapat melanjutkan proses yang diinginkan.

Property di atas hanya digunakan untuk mendeteksi status dan keadaan object XMLHttpRequest. Untuk melakukan proses selanjutnya, Anda dapat menggunakan method yang disediakan oleh object XMLHttpRequest. XMLHttpRequest menyediakan methodmethod yang dapat Anda manfaatkan, seperti method open digunakan untuk membuka file pada lokasi URL yang ditentukan, di mana file tersebut merupakan file text yang berisikan data text biasa, XML, atau bahkan script dalam bahasa yang dikenal oleh web server Anda, misalnya ASP atau PHP. Contoh method lain yang disediakan oleh XMLHttpRequest adalah abort, yang digunakan untuk membatalkan *request* yang aktif. juga dikenal method send, yang berfungsi untuk mengirimkan request.

Script ASP, PHP ataupun bahasa pemrograman lainnya, dapat Anda gunakan dalam Ajax untuk keperluan menghasilkan data yang dibutuhkan oleh object XMLHttpRequest, contohnya script Anda dapat mengakses *table* negara, dan menghasilkan data dalam format XML, yang akan dibaca oleh object XMLHttpRequest. Di sini Anda dapat melihat, teknik Ajax didukung oleh beragam teknologi dan bahasa pemrograman yang bekerja sama.

Penulisan kode HTML, JavaScript, dan script pemrograman seperti PHP atau ASP dalam satu file seperti pada teknik Ajax ini, tentunya bukan hal yang baru, bagi Anda yang telah terbiasa melakukan web programming. Karena itu secara programming, teknik Ajax hanyalah sedikit modifi kasi, dan sama sekali tidak mengubah *syntax* atau paradigma apapun. Tetapi jika Anda baru memulai web programming, memang diperlukan pengenalan dan latihan yang lebih mendalam untuk mengimplementasikan Ajax, yang tentunya bukan hal yang mustahil tercapai dalam waktu yang relatif singkat jika kita memiliki motivasi dan minat.

Penggunaan Ajax cukup diminati hingga saat ini. Terbukti dari sekian banyak framework yang ditawarkan, dan perkembanganperkembangan yang berbasis teknologi Ajax.

Framework yang dikembangkan tidak jarang berbasis *open source*. Salah satu pengembangan lain dari teknologi Ajax dalam dunia open source adalah pengembangan AppML (*Application Markup Language*) dari W3Schools. AppML menggunakan Ajax dalam pengembangannya, dan merupakan platform independen. Daftar di atas dapat bertambah dengan lebih banyak lagi referensi

dan website yang dapat Anda gunakan, dan tidak tertutup kemungkinan Anda sendiri yang akan mengembangkan Ajax dalam karya Anda.

b. Kerugian Ajax

1. **Integrasi Browser.**

   Browser memiliki engine tersendiri untuk mengolah *history*, sehingga Anda dapat kembali ke halaman sebelumnya dengan mudah, dengan mengeksekusi tombol “Back”. Dengan menggunakan Ajax, halaman yang diciptakan secara dinamis tidak ter-*register* di dalam engine browser, sehingga mungkin navigasi (seperti contoh tombol “Back”) akan tidak berjalan sesuai keinginan Anda. Walaupun demikian, bukannya tidak adajalan atau trik pemrograman yang dapat mengatasi hal ini. Anda dapat mencari artikel-artikel Ajax yang memberikan tip untuk mengatasi hal ini, setidaknya hal ini memerlukan pemrograman yang relative rumit.

2. **Waktu Tunggu.**

   Dengan Ajax, memang tidak terdapat lagi refresh keseluruhan halaman. Tetapi menimbulkan masalah baru, yaitu waktu tunggu yang terjadi karena pertukaran data antara request dari pengguna dengan *response* server, menjadi lebih terasa dan terkadang menyesatkan pengguna yang masih awam. Untuk itu, memberikan progress selama waktu tunggu, terlebih untuk data yang relatif cukup besar, akan dapat membantu permasalahan tersebut.

3. Optimasi Search Engine.

Sebuah website yang statik mudah untuk di-*index* di dalam search engine, tidak demikian dengan halaman yang dibuat dengan Ajax. Masalah serupa dihadapi oleh semua website yang dinamis, di mana search engine tidak dapat mengenali content website Anda.

4. Kompatibilitas Browser.

JavaScript merupakan bahasa pemrograman utama di dalam Ajax. Permasalahannya adalah masing-masing browser (juga browser yang sama tetapi dengan versi yang berbeda), mungkin mengimplementasikan JavaScript berbeda satu dengan yang lainnya. Untuk menghindarinya, JavaScript yang Anda gunakan perlu dites dengan berbagaibrowser. Sebaiknya Anda menghindari solusi menuliskan JavaScript yang berbeda untuk browser yang berbeda, untuk mendukung kompatibilitas browser. Cara ini dapat memperumit kode program Anda dan menjadi tidak efi sien.

2.2.8.3. Ajax Framework

Teknik Ajax terus berkembang seiring dengan kemampuannya yang mengagumkan untuk menciptakan website yang interaktif,dan dinamis. Untuk lebih memudahkan proses pemrograman bagi developer, maka dibuat Ajax Framework. Ajax Framework terdiri dari kumpulankumpulan fungsi dan *library* yang siap digunakan. Dengan adanya *framework*, developer yang menggunakan Ajax terbantu dari dua sisi, yaitu sisi client dan sisi server. Pada sisi client, framework menyediakan functionfunction JavaScript untuk mengirimkanrequest ke server. Pada sisi server, framework melakukan proses terhadap request yang

diterima, mencari data yang diinginkan, dan mengirimkannya kembali pada web browser.

Contoh framework yang dapat Anda gunakan adalah AjaxLib, yang beralamatkan

di *http://karaszewki.com/tools/ajaxlib/*. Anda dapat men-download library JavaScript yang disediakan, dan menggunakannya untuk mendevelop website Anda. AjaxLib menyediakan function-function untuk keperluan sisi client. AjaxLib hanyalah satu contoh Ajax Framework yang dapat Anda temukan di Internet. Masih banyak pilihan Ajax Framework yang tersedia dan siap pakai, baik dari sisi client ataupun server.

Ajax Framework dapat juga ditemui pada platform dan bahasa lainnya, misalnya pada platform Windows .NET yang dikenal dengan ASP.NET AJAX (sebelumnya dikenal dengan nama Microsoft Atlas), yang menyediakan komponen-komponen yang dapat Anda gunakan dengan mudah. Beberapa contoh komponen/control yang diambil dari ASP.NET AJAX berikut ini dapat memperkaya kreativitas Anda untuk menghasilkanweb dinamis dan interaktif :

1. **AlwaysVisible.**

   Sebuah control yang mengijinkan Anda meletakkan sebuah informasi (misalnya pengumuman, tanggal dan jam, dan seterusnya) pada suatu posisi yang tetap di halaman web, walaupun halaman tersebut di-*scroll* atau di-*resize*.

2. **Animation.**

   Untuk membuat animasi pada suatu event tertentu, seperti OnClick, OnLoad, OnMouseOver, dan lain-lain.

3. **CascadingDropDown.**

   Teknik yang cukup umum digunakan dalam Ajax, yaitu memuat data dinamis.

4. **NumericUpDown.**

   Banyak cara untuk memilih sebuah angka. Pada aplikasi desktop, Anda tentu telah terbiasa memilih angka dengan cara mengklik tombol *up* untuk menaikkan nilai pada sebuah *textbox*, atau klik tombol *down* untuk menurunkan nilai. Contohnya saat menentukan *margin* pada aplikasi Microsoft Word. Pada halaman web tradisional, Anda dapat menentukan angka dengan sebuah dropdown, ataupun mengetikkan secara manual angka tersebut. Jika Anda ingin berkreasi, *control* NumericUpDown ini membuat Anda dapat menciptakan tombol up dan down yang serupa dengan aplikasi desktop.

5. **UpdateProgress.**

   Salah satu control untuk mengatasi permasalahan waktu tunggu pada Ajax. Saat proses behind the scene terjadi, mungkin Anda perlu menampilkan sebuah status *progress* yang terjadi, sehingga pengguna menyadari bahwa halaman web sedang bekerja dan memanggil data.

2.2.9. Web Browser

Browser Web merupakan software yang digunakan untuk menampilkan informasi dari server web.informasi yang disajikan dengan web menggunakan konsep multimedia ,informasi yang disajikan dengan menggunakan banyak media (teks,gambar,animasi,suara,(audio)dan atau film). Pada awalnya pada browser web hanya dapat menampilkan informasi dalam bentuk text sehingga menyulitkan user dalam pengaksesan data. Perangkat lunak ini kini telah dikembangkan dengan menggunakan *Grafis User interface* (GUI), sehingga pemakai dapat melakukan 'point and click' untuk berpindah dari satu dokumen ke dokumen yang lainnya serta memudahkan user dalam melakukan kegiatannya.

Pada saat ini hanya ada dua browser web GUI yang populer yaitu Internet Explorer dan Netscape Navigator, kedua browser ini bersaing untuk menarik pemakai dengan berusaha mendekati standar spesifikasi dokumen HTML yang direkomendasikan oleh W3C.

2.2.10. Aplikasi Pendukung Perancangan Web

Beberapa aplikasi yang dianggap sangat membantu dalam proses design web adalah :

1. Macromedia Dreamweaver Pro 2008 : Macromedia Dreamweaver Pro 2006 merupakan perangkat lunak yang dikenal sebagai *web authoring tool*, yaitu perangkat lunak yang digunakan untuk desain tampilan halaman web. Macromedia Dreamweaver Pro 2006 mempunyai kemampuan untuk editing kode pada aplikasi web dengan bahasa pemrograman yang berbeda-beda, seperti PHP, ASP, *Java Server Pages* (JSP) dan lainnya.

2. Adobe Photoshop CS 2 : Adobe Photoshop CS 2 banyak sekali digunakan dalam pengolahan gambar, teks maupun dalam pembuatan button, karena dapat memberikan efek-efek yang dinamis secara praktis. Hasil pengolahan dari photoshop banyak digunakan dalam halaman-halaman web.
3. Corel Draw X3 : Merupakan salah satu Software grafis berbasis vector dengan kemampuan yang besar yang bisa dijadikan pilihan yang akan membawa sebuah ide fantsi yang ada di kepala menuju sebuah karya nyata.
4. Macromedia Flash Pro 2008 : Macromedia Flash Pro 2006 merupakan teknologi pilihan dalam membuat animasi-animasi yang dinamis dan interaktif, baik untuk pembuatan isi multimedia maupun untuk pembuatan animasi yang dibutuhkan dalam halaman web. Ada beberapa faktor yang mempunyai kontribusi besar pada teknologi yang membuat animasi flash menjadi populer, yaitu format .swf yang dapat mengompres file image dan suara, dengan ukuran file yang sangat kecil.

2.2.11. Sistem

2.2.11.1. Analisis Sistem

Analisis sistem *(sistem analis)* merupakan tahapan yang sangat kritis dan penting karena kesalahan dalam tahap ini dapat mempengaruhi tahapan selanjutnya, Analisa sistem didefinisikan sebagai berikut :

"Penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya".

Dalam tahap analisis terdapat langkah-langkah sebagai berikut:

a. *Identify* yaitu mengidentifikasi masalah.

b. *Understand* yaitu memahami kerja dari sistem yang ada.

c. *Analyze* yaitu menganalisis sistem.

d. *Report* yaitu membuat laporan hasil analisis.

2.2.11.2. Desain Sistem

Desain sistem didefinisikan sebagai berikut :

"Penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi".

a. Tujuan Desain Sistem.

Desain sistem memunyai maksud dan tujuan sebagai berikut:

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik yang terlibat.

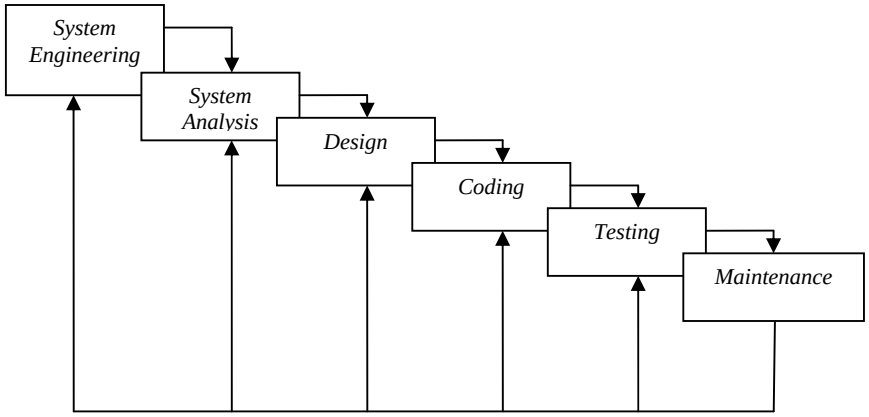
Sasaran-sasaran yang harus dicapai agar desain sistem mencapai tujuan:

1. Desain sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
2. Desain sistem harus dapat mendukung tujuan utama perusahaan.
3. Desain harus efektif dan efisien mendukung pengolahan transaksi yang, pelaporan manajemen dan mendukung keputusan yang akan dilakukan oleh manajemen, termasuk tugas-tugas lainnya yang tidak dapat dilakukan oleh komputer.

4. Desain sistem harus dapat mempersiapkan rancang bangun yang terinci untuk masing-masing komponen yang meliputi data dan informasi, simpanan data, metode-metode, prosedur-prosedur, orang-orang, perangkat keras, perangkat lunak dan pengendalian intern.

2.2.11.3. Model Perancangan Sistem

Langkah awal yang dilakukan dalam membangun sistem adalah dengan menentukan model sistem yang akan digunakan. Berikut ini akan dijelaskan mengenai model sistem *waterfall*. Dalam penelitian ini model sistem yang digunakan adalah model sistem *waterfall* dengan struktur seperti pada gambar 2.5 Berikut ini :



Gambar 2.13. Model Sistem Waterfall

Model sistem ini menuntut cara yang teratur dari suatu rangkaian yang mendekati perkembangan *software*, yang dimulai dengan suatu tingkatan kemajuan-kemajuan melalui analisa, perancangan, pengkodean, pengujian dan pemeliharaan. Model tersebut meliputi kegiatan aktifitas sebagai berikut :

a. *System/Information Engineering*

Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem dan mengalokasikannya kedalam pembentukan perangkat lunak.

b. *System Analysis*

Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.

c. *Design*

Perancangan berdasarkan analisis data kedalam bentuk gambaran aplikasi yang dibuat.

d. *Coding*

Menerapkan hasil perancangan dari aplikasi yang dibangun kedalam bahasa pemrograman tertentu, yaitu menggunakan tools Macromedia Flash 8.

e. *Testing*

Pengujian terhadap aplikasi yang dibangun.

f. *Maintenance*

Tahap akhir dimana suatu perangkat lunakyang sudah selesai dapat mengalami perubahan-perubahan atau penambahan sesuai dengan permintaan *user*.

2.2.11.4. Alat Pengembangan Sistem

Untuk dapat melakukan langkah-langkah sesuai dengan yang diberikan oleh metodologi pengembangan sistem yang terstruktur, maka dibutuhkan alat

untuk melaksanakannya. Alat-alat yang digunakan dalam suatu metodologi umumnya berupa suatu gambar, diagram, atau grafik. Selain berbentuk gambar, alat-alat yang digunakan juga ada yang tidak berupa gambar atau grafik, seperti misalnya kamus data (*data dictionary*), *pseudocode*, serta formulir-formulir untuk mencatat dan menyajikan data, seperti :

1. **Bagan Alir Informasi**, bagian yang menggambarkan arus informasi berupa laporan, formulir, dan dokumen yang keluar atau masuk dari bagian – bagian tertentu dalam sistem.
2. **Bagan Alir Sistem (*System Flowchart*)**, merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem, yang menjelaskan urutan – urutan dari prosedur – prosedur yang ada didalam sistem.
3. **Bagan Hubungan Database (*Relational Database*)**, dimaksudkan untuk mengidentifikasikan kebutuhan dan hubungan dari file – file database yang digunakan.

2.2.12. Konsep Dasar Basis Data

Basis Data (*Database)* dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis Data menunjukkan suatu kumpulan data yang dipakai dalam suatu lingkungan perusahaan atau instansi-instansi. Penerapan basis data dalam sistem informasi disebut sistem basis data (*database system)*

.

2.2.12.1. Pengertian Basis Data

Basis Data terdiri dari kata Basis dan Data. **Basis** dapat diartikan gudang atau tempat bersarang dan **Data** yang berarti representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

Jadi dapat disimpulkan bahwa Basis Data merupakan kumpulan data yang (arsip) yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Atau bisa diartikan sebagai kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

a. *Database Manajemen Sistem (DBMS*)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak atau sistem yang khusus atau spesifik. Sistem ini yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali Disamping itu sistem ini juga menerapkan mekanisme pengaman data, pemakaian data secara bersama, pemaksaan keakuratan atau konsistensi data dan sebagainya. Perangkat lunak yang termasuk DBMS misalnya dBase II+, dBase IV, FoxBase, RBase, MS-*Access* dan *Borland-Paradox* atau *Borland-Interbase*, MS-SQL *Server*, CA-*Open Ingres*, *Oracle*, *Informix* dan *Sybase*.

b. Tujuan Basis Data

Basis Data pada prinsipnya ditujukan untuk pengaturan data agar terdapat kemudahan dalam pengambilan kembali data tersebut. Berikut ini terdapat beberapa tujuan dari Basis Data diantaranya yaitu :

1. Kecepatan dan kemudahan (*Speed)*
2. Efisiensi ruang penyimpanan (*Space)*
3. Keakuratan (*Accuracy)*
4. Ketersediaan (*Avaibility)*
5. Kelengkapan (*Completeness)*
6. Keamanan (*Security*)
7. Kebersamaan (*Sharability*)

c. Pengguna Basis Data

Pengguna basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem diantaranya yaitu :

1. Programmer Aplikasi (*Application Programmer*)

   Pemakai yang berinteraksi dengan basis data dengan menggunakan *Data Manipulation Language* (DML) untuk membuat aplikasi dengan menggunakan bahasa pemrograman.

2. *User* Khusus (*Specialized User*)

   Pemakai yang membuat program aplikasi basis data untuk keperluan tertentu atau khusus.

3. *User* Mahir (*Casual User*)

   Pemakai yang berinteraksi dengan sistem tanpa membuat modul program.

4. *User* Umum (*Naïve User*)

   Pemakai berinteraksi dengan aplikasi basis data yang telah dibuat atau disediakan oleh sistem.

2.2.12.2. Perancangan Basis Data

Perancangan basis data dibutuhkan agar diperoleh suatu sistem yang sesuai dengan apa yang diinginkan, dimana dapat melalui tahapan berikut :

a. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram atau bisa dikenal dengan diagram E-R secara grafis menggambarkan isi sebuah database. Diagram ini memiliki dua komponen utama yaitu *entity* dan relasi. Untuk melambangkan fungsi diatas maka digunakan simbol - simbol yang bisa dilihat pada daftar simbol.

Elemen-elemen *Entity Relationship Diagram* adalah sebagai berikut :

1. Entity

Pada *E-R* diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan. Entitas diberi nama dengan kata benda dan dapat dikelompokan dalam empat jenis nama, yaitu: orang, benda, lokasi kejadian (terdapat unsur waktu didalamnya).

2. Relationship

Pada *E-R diagram, Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara

entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau dengan kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

3. *Relationship Degree*

   Relationship degree atau derajat relasi adalah jumlah entitas yang berpartisipasi dalam suatu *relationship*.

4. Atribut

   Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship*, sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan *relationship*.

5. Kardinalitas

   Kardinalitas relasi menunjukan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antar entitas, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu :

   a. One to One

      Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

b. One to Many atau Many to One

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua . Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

c. Many to Many

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

b. Normalisasi

Normalisasi merupakan tahapan perancangan dalam membangun basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Adapun langkah-langkah untuk menghasilkan struktur tabel yang normal, diantaranya yaitu :

1. Bentuk *Unnormallized*

   Bentuk unnormal adalh bentuk table dengan mencantumkan semua field data yang ada

2. Bentuk Normal Pertama / 1 NF (*First Normal Form*)

Bentuk Normal pertama terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*Multivalued Atribute)* atau lebih dari satu atribut dengan domain nilai yang sama.

3. Bentuk Normal Kedua / 2 NF (*Second Normal Form*)

   Bentuk Normal Kedua terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *key primer* memiliki ketergantungan fungsional pada *key primer* secara utuh.

4. Bentuk Normal Ketiga / 3 NF (*Third Normal Form*)

   Bentuk Normal Ketiga terpenuhi jika dan hanya jika semua atribut bukan kunci memiliki dependensi transitif terhadap kunci primer.

5. *Boyce-Codd Normal Form* (BCNF)

   Suatu tabel dikatakan berada pada BCNF jika dan hanya jika semua ketergantungan fungsional dengan notasi X $\longrightarrow$ Y, maka X harus merupakan *superkey* pada tabel tersebut.

6. Bentuk Normal Keempat / 4 NF (*Fourth Normal Form*)

   Suatu tabel dikatakan berada pada normal keempat jila tidak mengandung dua atribut atau lebih yang bernilai banyak.

7. Bentuk Normal Kelima / 5 NF (*Fifth Normal Form*)

   Bentuk Normal Kelima berkaitan dengan ketergantungan relasi antar tabel (*Join Dependency*).

c. Relasi Tabel

Relasi tabel menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Dari adanya relasi tabel tersebut

terdapat kardinalitas relasi yang menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan enttas yang lain. Terdapat empat kemungkinan kombinasi kardinalitas yaitu : satu ke satu (*One To One)*, satu ke banyak *(One To Many)*,banyak ke satu (*Many To One) dan* banyak ke banyak (*Many To Many*).