BAB II

LANDASAN TEORI

2.1 Konsep Dasar Program Aplikasi

Program adalah ekspresi pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman, sehingga dapat dieksekusi oleh program.

Aplikasi merupakan penerapan, penyimpanan sesuatu hal, data, permasalahan pekerja ke dalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan suatu hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan. Jadi, dalam hal ini hanya berbentuk tampilan data yang berubah, sedangkan isi yang termuat dalam data tersebut tidak mengalami perubahan.

Program Aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer agar dapat melakukan pekerjaan sesuai dengan keinginan *programmer* atau *user*.

2.2 Pengertian Internet

Internet merupakan jaringan global yang terdiri dari berbagai komputer yang saling berhubungan dan bekerjasama dengan cara berbagai informasi dan data. Media penghubung tersebut bisa melalui kabel, kanal satelit maupun frekuensi radio.

Setiap komputer yang terhubung dengan jaringan tersebut, diberikan sebuah nomor yang unik, dan berkomunikasi satu sama lainnya dengan bahasa komunikasi yang sama. Bahasa komunikasi yang sama ini disebut protokol. Protokol yang digunakan di internet adalah TCP/IP (*Transmission Control Protocol* / *Internet Protocol*).

2.3 Sejarah Internet

Internet pertama kali dikembangkan oleh ARPANet, suatu proyek yang dimulai dari Departemen Pertahanan Amerika Serikat (*US Departement of Defense–DOD*) pada tahun 1969, suatu percobaan dalam *reliable networking* (jaringan secara terpercaya) untuk menghubungkan antara DOD dengan kontraktor penelitian militer, termasuk sejumlah besar universitas yang melakukan penelitian dengan dana militer. Pada konsepnya sebenarnya internet merupakan suatu jaringan, yang mana suatu paket informasi dapat dikirim dari suatu komputer ke komputer yang lain. Hal ini dimaksudkan untuk mempermudah pemindahan berbagai data penting apabila terjadi perang

ARPA merupakan singkatan dari *Advanced Research Projects Agency*, cabang dari *Defence* (Pertahanan) yang mempunyai kewajiban membagi-bagikan uang *grant* sehingga menjadi DARPA (*Defence-ARPA*). ARPANet mulai dengan 3 komputer kecil yang dikoneksi di California digabungkan dengan satu di Utah, tetapi secara cepat berkembang di seluruh kontinen. Internet ini mulai tumbuh pesat pada dekade 1990.

2.4 Kegunaan Internet

Kegunaan internet yang utama antara lain :

1. Fungsi komunikasi

Internet adalah alat komunikasi, kegunaan yang sangat penting dari internet adalah pertukaran pesan dengan menggunakan *electronic mail* (*e-mail*).

2. Fungsi *Resource Sharing*

Dengan internet, kita dapat mencari *software*, essay, data dan program dari ribuan titik distribusi di seluruh dunia.

3. Fungsi *Resource Discovery*

Navigasi untuk mencari file tertentu, dokumen, *host* atau orang diantara jutaan host.

4. Fungsi Komunitas

Masyarakat pengguna internet dapat berhubungan dan membuat perkumpulan/komunitas tertentu.

2.5 Perkembangan Internet

Jumlah pengguna internet yang besar dan semakin berkembang, telah mewujudkan budaya internet. Internet juga mempunyai pengaruh yang besar atas ilmu, dan pandangan dunia. Dengan hanya menggunakan mesin pencari seperti *Google*, pengguna di seluruh dunia mempunyai akses yang mudah atas bermacam-macam informasi. Dibanding dengan buku dan perpustakaan, internet melambangkan penyebaran (*decentralization*) informasi dan data secara ekstrim.

Perkembangan internet juga telah mempengaruhi perkembangan ekonomi. Berbagai transaksi jual beli yang sebelumnya hanya bisa dilakukan dengan cara

tatap muka (dan sebagian sangat kecil melalui pos atau telepon), kini sangat mudah dan sering dilakukan melalui internet. Transaksi melalui internet ini dikenal dengan nama *e-commerce*.

Terkait dengan teknologi pembuatan website sebagai konten dari teknologi internet itu sendiri memicu berkembangan teknik pembuatan website yang semakin canggih dengan fitur-fitur yang sangat beragam tidak hanya terbatas pada teks dan gambar saja, akan tetapi sudah merambah kedalam dunia multimedia dan telkomunikasi visual. Bahkan dengan munculnya teknologi konten manajemen sistem pengguna yang awam dengan kemampuan teknis yang sedikipun bisa membuat website sesuai dengan keinginannya tanpa harus berhubungan dengan *webmaster* dengan konten yang dinamis dan selalu *uptodate*.

2.6 Pengertian Web 2.0

Web 2.0 merupakan perkembangan dari teknologi *web* saat ini. Teknologi ini berupa layanan di *web* yang lebih menekankan pada *social network* atau jalinan sosial antara penggunanya dan memungkinkan pengguna dapat saling berkolaborasi dan berbagi informasi secara *online*. *Web* 2.0 juga merupakan istilah yang digunakan untuk menunjukkan suatu aplikasi web yang mempunyai interaktivitas dengan penggunanya sama seperti halnya dengan aplikasi *desktop*.

2.7 Pengertian Sistem

Pengertian sistem adalah kumpulan dari unsur atau elemen-elemen yang saling berkaitan atau berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. Sebagai contoh, sistem komputer terdiri dari: *Software, Hardware* dan *Brainware*. [1]

Menurut Jerry FithGerald, Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau mePnyelesaikan suatu sasaran tertentu. [1]

Menurut Ludwig Von Bartalanfy, Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan. [1]

Menurut Anatol Raporot, Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain. [1]

Menurut L. Ackof, Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya. [1]

Suatu sistem memiliki beberapa syarat – syarat, antara lain: [1]

a. Sistem harus dibentuk untuk menyelesaikan tujuan.
b. Elemen sistem harus mempunyai rencana yang ditetapkan.
c. Adanya hubungan diantara elemen sistem.
d. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting daripada elemen sistem.
e. Tujuan organisasi lebih penting dari pada tujuan elemen.

2.8 Konsep Dasar Sistem

Didalam mendefinisikan sistem, terdapat dua pendekatan yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut :

"Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersam-sama unutk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu". [2]

Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut :

"Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu ". [2]

2.9 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, antara lain akan dijelaskan dibawah ini. [1]

1. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-komponen atau ***subsistem-subsistem***. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yangdisebut ***supra sistem***, misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalaudipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistemdipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas darisubsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Untuk membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain output dari suatu subsistem akan menjadi input dari subsistem yang lainnya.

5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untukdidapatkan keluaran. Sebagai contoh didalam sistem

komputer, program adalah maintanance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem. Meliputi : Keluaran yang berguna, contohnya Informasi yang dikeluarkan oleh komputer. Dan Keluaran yang tidak berguna yang dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

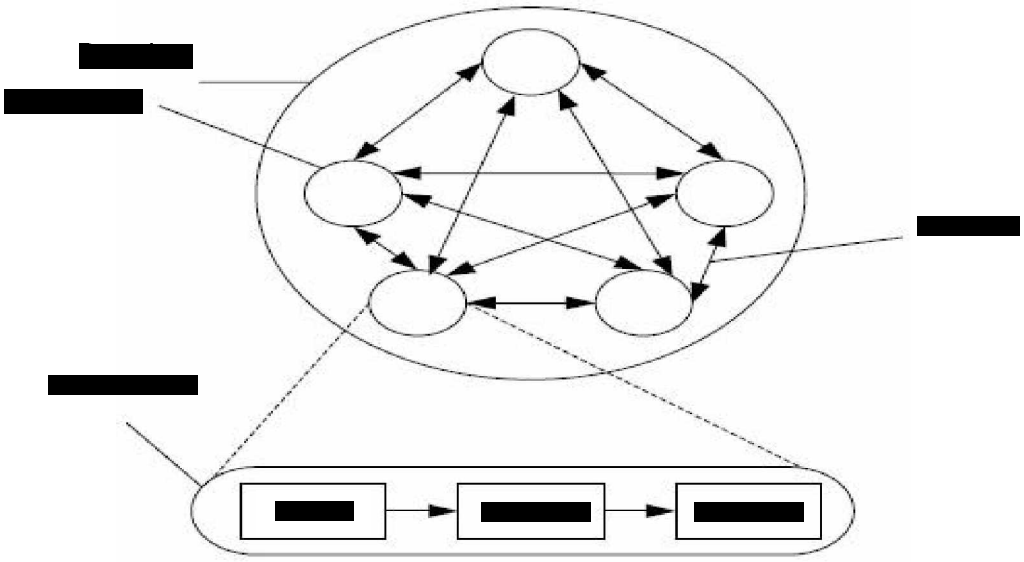
7. Pengolahan Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh CPU pada Komputer, Bagian Produksi yang mengubah bahan baku menjadi barang jadi, Bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

8. Tujuan Sistem (Goal)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

Karakteristik atau sifat-sifat suatu sistem dapat dilihat pada gambar 2.1.



Gambar 2.1 Karakteristik Sistem [6]

2.10 Manajemen

Manajemen berasal dari kata "*to manage*" yang berarti mengatur, mengurus, atau mengelola. Banyak definisi yang telah diberikan oleh para ahli terhadap istilah manajemen ini. Namun dari sekian banyak definisi tersebut ada satu yang kiranya dapat dijadikan pegangan dalam memahami pengertian manajemen tersebut, yaitu : Manjemen adalah suatu proses yang terdiri dari rangkaian kegiatan, seperti perencanaan, pengorganisasian, penggerakan dan pengendalian atau pengawasan, yang dilakukan untuk menentukan dan mencapai tujuan yang telah ditetapkan melalui pemanfaatan sumber daya manusia dan sumber daya lainnya.[3]

Pengertian manajemen lainnya menurut para ahli diantaranya adalah : [3]

1. Menurut Horold Koontz dan Cyril O'donnel

Manajemen adalah usaha untuk mencapai suatu tujuan tertentu melalui kegiatan orang lain.

2. Menurut R. Terry

Manajemen merupakan suatu proses khas yang terdiri dari tindakan-tindakan perencanaan, pengorganisasian, penggerakan dan pengendalian yang dilakukan untuk menentukan serta mencapai sasaran yang telah ditentukan melalui pemanfaatan sumber daya manusia dan sumberdaya lainnya.

3. Menurut James A.F. Stoner

Manajemen adalah suatu proses perencanaan, kepemimpinan dan pengendalian upayadari anggota organisasi serta penggunaan semua sumber daya yang ada pada organisasi yang telah ditetapkan sebelumnya. [4]

4. Menurut Mary Parker Follet

Manajemen adalah suatu seni, karena untuk melakukan suatu pekerjaan melalui orang lain dibutuhkan keterampilan khusus. [4]

2.11 Konten

Konten berasal dari bahasa inggris "*content*" yang artinya isi atau kandungan. Jenis atau ragam konten berdasarkan ruang lingkupnya dapat diartikan sebagai berikut : [5]

1. Berdasarkan Multimedia

   a. *Text-Based Content*, konten berbasis teks. Konten berbasis teks lebih cepat dibuat dan dipublikasikan melalui internet karena relatif secara ukuran *file* lebih kecil.

b. *Multimedia-Based Content,* konten berbasis multimedia baik itu multimedia *linier* (seperti video) maupun multimedia interaktif. Konten berbasis multimedia relatif lebih memerlukan waktu dan *cost* dalam pembuatan maupun publikasinya di internet dikarenakan ukuran *file*nya yang relatif lebih besar.

2. Berdasarkan Tingkat Kemanfaatan

a. Data, adalah sesuatu yang tidak membawa arti, bersifat mentah dan merupakan kumpulan dari fakta-fakta tentang suatu kejadian. Bisa juga merupakan catatan terstruktur dari suatu transaksi dan boleh dikatakan materi penting dalam membentuk informasi.

b. Informasi, merupakan kompilasi dari data. Informasi memiliki arti, relevansi dan juga tujuan. Transformasi data menjadi informasi adalah data yang telah diolah menjadi sesuatu yang membawa arti.

c. Pengetahuan, merupakan gabungan dari suatu pengalaman, nilai, informasi, kontekstual, dan juga pandangan pakar yang memberikan suatu *framework* untuk mengevaluasi dan menciptakan pengalaman baru. Bisa berupa solusi pemecahan suatu masalah, petunjuk suatu pekerjaan, dan ini bisa ditingkatkan nilainnya, dipelajari dan diajarkan kepada yang lain.

3. Berdasarkan Lisensi

a. Open Content, segala jenis hasil kerja kreatif yang dipublikasikan dalam suatu format (lisensi) yang memungkinkan pihak lain (individu, perusahaan, organisasi) untuk memperbanyak dan memodifikasi

informasi didalamnya. Hak cipta dan *open content* tetap ada, tetapi lisensi memungkinkan orang lain bebas untuk menggunakan dan memodifikasinya. Misalnya adalah GNU *free documentation* (GFDL), *Creative Common License, Open Content License* (OPL) dan lainnya.

b. Proprietary Content, segala jenis hasil karya kreatif yang dipublikasikan dalam suatu format (lisensi) dengan berbagai batasan dalam penggunaan, modifikasi atau memperbanyak. lisensi *proprietary content* biasanya dalam bentuk royalty (uang) ke pemilik hak ciptanya.

2.12 Sistem Manajemen Konten (CMS)

Sistem manajemen konten atau yang sering disebut CMS yang merupakan singkatan dari *Content Management System* yaitu sebuah aplikasi yang dikembangkan untuk memudahkan mengelola *website* (*updating, maintenance*) sehingga bisa dilakukan dengan efektif, efisien untuk mengelola *website* agar di *update* tiap hari, jam bahkan tiap detik maka perlu rancangan *website* yang dinamis. CMS biasa dikembangkan dengan menggunakan bahasa pemrograman *server side* seperti PHP, Perl, ASP dll dengan menggunakan *database* Mysql, SQL, PostgreSQL dll.

CMS merupakan Sebuah sistem yang memberikan kemudahan kepada para penggunanya dalam mengelola dan mengadakan perubahan isi sebuah *website* dinamis tanpa sebelumnya dibekali pengetahuan tentang hal-hal yang bersifat teknis. Dengan demikian, setiap orang, penulis maupun editor, setiap saat dapat menggunakannya secara leluasa untuk membuat, menghapus atau bahkan memperbaharui isi *website* tanpa campur tangan langsung dari pihak *webmaster*.

CMS secara umum dapat diartikan sebagai sebuah sistem yang memberi kemudahan para para penggunanya dalam mengelola dan melakukan perubahan isi sebuah *website* dinamis tanpa harus dibekali pengetahun tentang hal-hal yang bersifat teknis. Artinya setiap orang baik itu masyarakat umum atau pengunjung website kita dapat leluasa untuk membuat atau bahkan memperbaharui isi website layaknya seorang web *master* / sistem *administrator*.

CMS pertama kali muncul sebagai jawaban atau solusi dari kebutuhan manusia akan penyediaan informasi yang sangat cepat. Masih segar dalam ingatan kita, betapa sederhananya sebuah website di era tahun 90-an. Dengan hanya mengandalkan bahasa pemograman HTML dan beberapa gambar serta informasi yang statis, sebuah perusahaan berusaha sebaik mungkin menampilkan informasi secukupnya kepada para pengunjung. Setiap kali ada perubahan informasi dalam perusahaan, pihak manajeman mau tak mau haruslah berhubungan terlebih dahulu dengan pihak Humas sebelum akhirnya semua bahan diserahkan kepada pihak *webmaster*. Pihak inilah yang nantinya akan mengadakan perubahan terhadap isi situs web. Dapat dibayangkan bila hal yang sama terjadi terus-menerus, berulang kali dan dalam kuantitas yang besar, seberapa banyak waktu dan tenaga yang dibutuhkan untuk memproses semuanya.

Tentu saja situasi seperti ini tidak diinginkan oleh setiap orang. Dapat dikatakan sebuah metode atau sistem yang dapat meningkatkan tingkat produktivitas dan efisiensi dalam pengembangan website sangatlah dibutuhkan. Dan salah satu solusi yang tepat untuk ini adalah dengan menerapkan *Content Management System* atau CMS.

2.13 Pengertian Sistem Manajemen Konten (CMS)

CMS secara sederhana dapat diartikan sebagai berikut: “Sebuah sistem yang memberikan kemudahan kepada para penggunanya dalam mengelola dan mengadakan perubahan isi sebuah website dinamis tanpa sebelumnya dibekali pengetahuan tentang hal-hal yang bersifat teknis. Dengan demikian, setiap orang, penyusun maupun editor, setiap saat dapat menggunakannya secara leluasa untuk membuat, menghapus atau bahkan memperbaharui isi website tanpa campur tangan langsung dari pihak *webmaster*” . Bukankah ini suatu hal yang efisien ?

Karena CMS memisahkan antara isi dan desain, konsistensi tampilan dapat senantiasa dijaga dengan baik. Setiap bagian dari website dapat memiliki isi dan tampilan yang berbeda-beda, tanpa harus khawatir kehilangan identitas dari website secara keseluruhan. Oleh karena semua data disimpan dalam satu tempat, pemanfaatan kembali dari informasi yang ada untuk berbagai keperluan dapat dengan mudah dilakukan.

CMS juga memberikan kefleksibelan dalam mengatur alur kerja atau ‘*workflow*’ dan hak akses, sehingga memperbesar kesempatan berpartisipasi dari pengguna dalam pengembangan website. Hal ini akan sangat menguntungkan bila website yang dikelola memiliki kompleksitas yang tinggi dan mengalami kemajuan yang cukup pesat.

Pengertian lainnya, CMS adalah bagian perangkat lunak yang mengatur perkembangan website. Pengaturan CMS dapat membawa semua perubahan pada suatu *website*, merekam siapa yang mengubah, apa yang diubah dan kapan serta mengizinkan penambahan catatan. Penggunaan CMS dapat mengatasi banyak

masalah dan menghemat banyak waktu daripada membangun sebuah sendiri sebuah *website* dengan *programming* di HTML atau *Java Script*.

2.14 Ciri-ciri Sistem Manajemen Konten (CMS)

Berikut ini adalah ciri-ciri dari sebuah CMS :

1. CMS memisahkan isi dan desain sehingga konsistensi tampilan senantiasa terjaga dengan baik. Hal itu dikarenakan ada fungsi "aksi" sehingga perubahan tampilan halaman tidak merubah desain keseluruhan halaman, hanya modul atau bagian yang perlu dirubah saja, Setiap bagian website dapat memiliki isi dan tampilan berbeda, tanpa harus khawatir kehilangan identitas website secara keseluruhan.
2. Semua data disimpan dalam satu tempat sehingga mudah memanfaatkan kembali informasi untuk berbagai keperluan.
3. CMS membuat fleksibel pengaturan alur kerja dan hak akses sehingga memperbesar kesempatan user berpartisipasi mengembangkan website.

2.15 Manfaat Sistem Manajemen Konten (CMS)

Beberapa manfaat umum CMS dapat dijelaskan sebagai berikut :

1. Manajemen Data

Manajemen data merupakan fungsi utama CMS. Semua data dan informasi yang telah ditempilkan belum dapat diorganisasi dan disimpan dengan baik. Setiap saat data dan informasi dapat dipergunakan kembali sesuai dengan kebutuhan. CMS pun mendukung berbagai macam format data seperti XML, HTML, PDF dan lain-lain. Proses *indexing, searching, join* dan kontrol terhadap revisi data dan informasi dapat dilakukan dengan baik oleh CMS.

2. Pengaturan kelancaran website

Banyak CMS memberikan fasilitas kepada para penggunanya untuk mengelola bagian atau isi yang akan ditampilkan, masa atau waktu penampilan dan lokasi penampilan di *website*. Tak jarang sebelum ditampilkan ada fungsi untuk mereview dahulu oleh user agar validitas data terjamin.

3. Standarisasi Halaman

Setiap halaman website berasal dari "aksi" yang telah disediakan oleh CMS. Selain dapat menjaga konsistensi tampilan secara keseluruhan, para *user* dan *editor* dapat berkonsentrasi penuh dalam menyediakan isi *website*. Bila isi telah tersedia, maka proses publikasi ke halaman web mudah berjalan karena sudah ada aksi yang tersedia. Penetepan bagian *website* memberikan standarisasi kepada seluruh bagian *website*.

4. Personalisasi website

Sekali ditempatkan dalam CMS, isi dapat ditampilkan sesuai keinginan dan kebutuhan penggunanya. Terlebih lagi kelabihan CMS yang dapat memisahkan isi dan desain menyebabkan proses personalisasi berjalan mudah.

5. Sindikasi

Sindikasi memungkinkan sebuah website membagi isinya kepada *website-website* yang lain. Format data yang didukung pun cukup bervariasi mulai dari rss, rdx, xml, atom hingga "*backend scripting*". Sama halnya dengan personalisasi, sindikasi mudah pula dilakukan karena isi dan desain dibuat terpisah.

6. Akuntabilitas dan Akurasi

Karena CMS mendukung alur kerja dan hak akses yang jelas kepada penggunanya, data dan informasi dapat pula dipertanggungjawabkan dengan baik. Setiap *user* atau editor memilki tugas masing-masing dan hak akses yang berbeda-beda pula. Dengan demikian setiap perubahan di *website* dapat segera ditelusuri dan diperbaiki seperlunya.

2.16 Penggunaan Sistem Manajemen Konten (CMS)

CMS pada prinsipnya dapat diperguakan untuk berbagai macam keperluan dan dalam berbagai kondisi, seperti :

1. Forum
2. Mengelola *website* pribadi
3. Mengelola *website* sekolah
4. Portal atau *website* komunitas
5. Galeri foto
6. E-*Commerce*
7. E-*Learning*
8. E-*Goverment*, dll.

2.17 Model Sistem Manajemen Konten (CMS)

Model sistem manajemen konten terbagi kedalam dua model CMS, yaitu :

1. CMS Komersial

CMS Komersial dibuat dan dikembangkan oleh perusahaan – perusahaan *software* yang menjalankan usahanya dengan motif mencari keuntungan. CMS komersial pun menyediakan hampir semua fitur dan tidak tersedia secara gratis

karena *user*nya harus membeli lisensi dari perusahaan pembuatnya mulai dari harga termurah sampai harga termahal, baik untuk intranet maupun untuk internet.

2. CMS *Open Source*

CMS dibuat dan dikembangkan oleh sekelompok komunitas orang atau perusahaan yang intinya memberikan alternatif murah dan terjangkau kepada pengguna. CMS yang tersedia gratis ini dapat dipergunakan tanpa batasan dan memberikan hak akses kepada pengguna akan kode (*source*) program sehingga *user* dapat mudah memodifikasi CMS dimasa mendatang sesuai keinginannya.

2.18 Pengertian UML

Unified Modelling Language 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.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memilki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

2.19 Konsep Dasar UML

UML mendefinisikan diagram-diagram sebagai berikut : [9]

1. *Use Case Diagram*
2. *Class Diagram*
3. *Behaviour Diagram*
    a. *Activity Diagram*
    b. *Statechart Diagram*
    c. *Interaction Diagram*
        i. *Sequence Diagram*
        ii. *Collaboration Diagram*
4. *Implementation Diagram*
    a. Component Diagram
    b. Deployment Diagram

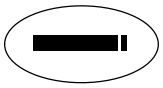
2.19.1 Diagram *Use Case* (*Use Case Diagram*)

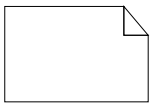
Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dan sistem. *Use case* merupakan sebuah pekerjaaan tertentu, misalnya *login* 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.

Diagram *use case* digunakan selama proses analisis untuk memperoleh kebutuhan sistem dan untuk memahami bagaimana sistem seharusnya bekerja.

Selama tahap desain, diagram *use case* menetapkan perilaku sistem saat diterapkan. Dalam sebuah model mungkin terdapat satu atau beberapa diagram *use case*. Tabel 2.1 di bawah ini memuat notasi simbol-simbol yang digunakan dalam diagram *use case.*

Tabel 2.1 Lambang-lambang dalam Diagram *Use Case*

No.	Lambang	Keterangan
1.		Aktor *(actor)* merupakan sebuah entitas yang berinteraksi dengan use case. Nama aktor dituliskan di bawah gambar tersebut. Aktor dapat berupa orang atau sistem lain diluar sistem yang tengah dianalis.
2.		*Use case* 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.
3.		Garis putus-putus dengan panah di salah satu ujungnya menggambarkan hubungan antar *use case*. Hubungan UseCase1 tersebut dapat berupa *include, extend* ataupun *generalization.* Hubungan *include* berarti *use case* yang ditunjukkan oleh garis ikut dikerjakan jika *use case* sumber dikerjakan. Hubungan extend berarti *use case* yang ditunjuk oleh *use case* dapat memanggil *use case* yang menunjuk jika persyaratannya terpenuhi. *Generalization* berarti menunjukkan tujuan bahwa *use case* yang ditunjuk merupakan bentuk umum dari *use case* yang menunjuk. Keterangan *include, extend* maupun *generalization* dituliskan dalam dua buah kurung sudut “<< >>”
4.		Hubungan, menggambarkan hubungan *association*. Garis ini digunakan untuk

		menghubungkan antara aktor dengan *use case*. Hubungan ini berarti aktor menggunakan *use case*.
5.	— — — — — —	Garis putus-putus. Menunjuk kepada objek yang dijelaskan oleh catatan.
6.		Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.

2.19.2 Diagram Kelas (*Class Diagram*)

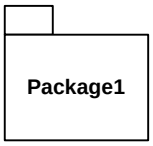
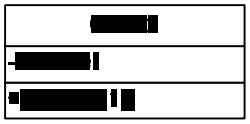
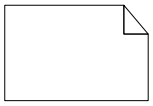
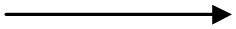
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

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, adalah sebagai beikut :

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

Tabel 2.2 Lambang-lambang dalam Diagram Kelas

No.	Lambang	Keterangan
1.		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.
2.		Kelas adalah bentuk umum dari objek-objek yang memilki struktur dan tingkah laku (atribut, operasi, relasi dan arti/definisi) yang sama. Kelas merupakan bentuk abstrak dari objek di dunia nyata.
3.		Catatan, berisi keterangan tambahan. Garis putusputus menunjukkan apa yang dijelaskan oleh catatan ini.
4.		Garis dengan mata panah tertutup pada salah satu ujungnya menggambarkan *generalization*. Kelas yang ditunjuk oleh mata panah merupakan kelas induk (*superclass*).
5.		Garis putus-putus dengan mata panah tertutup di salah satu ujungnya menggambarkan relasi.
6.		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
7.		terdapat jalur untuk berkomunikasi Pada ujung Garis putus-putus dari catatan menunjukkan objek apa yang dijelaskan oleh catatan

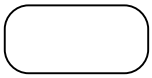
2.19.3 Diagram Kelakuan (*Behaviour Diagram*)

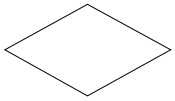
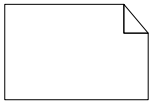
Diagram kelakuan terdiri dari diagram aktivitas dan diagram keadaan. Perbedaan dari kedua diagram tersebut adalah bahwa diagram aktivitas berorientasi proses sedangkan diagram keadaan berorientasi keadaan sesaat dari siklus hidup objek.

2.19.3.1 Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Diagram aktivitas juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram aktivitas juga dipergunakan untuk menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi. Lambang-lambang yang digunakan pada diagram aktivitas dapat dilihat pada tabel 2.3.

Tabel 2.3 Lambang-lambang dalam Diagram Kelas

No.	Lambang	Keterangan
1.		Awal (*Initial State*), menujukkan tempat dimulainya diagram. Lambang ini boleh diberi label dengan isi berupa nama kejadian (*event*) yang memanggilnya. Sebuah diagram aktivitas hanya boleh memiliki sebuah awal.
2.		Akhir (*Final State*) , menunjukkan tempat berakhirnya diagram. Lambang ini tidak memerlukan label. Diagram aktivitas dapat memiliki satu atau lebih akhir.
3.		Aktivitas (*activity*) adalah bentuk khusus dari keadaan (*state*) dimana aktivitas menggambarkan kegiatan yang dilakukan. Nama kegiatan

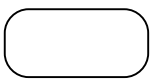
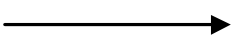
		dituliskan di tengah lambang.
4.		Keputusan (*decision*) menggambarkan keputusan atas sebuah kondisi, pertemuan dari jalur kondisi yang mungkin. Jika digunakan untuk menggambarkan keputusan, maka jalur masuk yang diijinkan hanya satu sedangkan jalur keluar sebanyak dua atau lebih. Jalur-jalur keluar harus diberikan nilai yang harus terpenuhi, dengan maksimal satu buah jalur keluar tidak diberi nilai sebagai kemungkinan lainnya (*else*). Sedangkan jika digunakan untuk pertemuan dari jalur kondisi yang mungkin maka jalur masuk sebanyak dua atau lebih sedangkan jalur keluar hanya satu.
5.		Penyelaras (*sync state*) digunakan untuk menyelaraskan lebih dari satu transisi baik itu masuk ataupun keluar. Sebuah jalur masuk dengan lebih dari satu jalur keluar atau beberapa jalur masuk dengan satu jalur keluar. Lambangnya dapat berupa batang vertikal maupun horizontal.
6.		Menyatakan alur aktivitas. Untuk alur yang memerlukan kondisi dituliskan kondisi yang harus dipenuhi dengan dibatasi oleh kurung siku “[dan “]”.
7.		Garis putus-putus. Menunjuk kepada objek yang dijelaskan oleh catatan.
8.		Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.

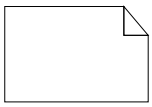
9.		Jalur unit (*swimlane*), menggambarkan unit organisasi dalam model bisnis. Nama dari unit tersebut ditempatkan di atas dari jalur. Setiap operasi/keadaan ditempatkan di dalam garis jalur sesuai dengan pemilik operasi/keadaan tersebut. Alur aktifitas diijinkan untuk melewati batas jalur untuk menuju unit lainnya.

2.19.3.2 Diagram Keadaan Sesaat (*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*). Diagram ini memodelkan tahap-tahap diskrit dari sebuah siklus objek.

Tabel 2.4 Lambang-lambang dalam Diagram Keadaan Sesaat

No.	Lambang	Keterangan
1.		Awal (*Initial State*), menujukkan tempat dimulainya diagram. Lambang ini boleh diberi label dengan isi berupa nama kejadian (*event*) yang memanggilnya
2.		Akhir (*Final State*) , menunjukkan tempat berakhirnya diagram. Lambang ini tidak memerlukan label.
3.		Aktivitas (*activity*) adalah bentuk khusus dari keadaan (*state*) dimana aktivitas menggambarkan kegiatan yang dilakukan. Nama kegiatan dituliskan di tengah lambang.
6.		Menyatakan alur aktifitas. Alur menghubungkan antara state awal, akhir maupun aktifitas. Alur dapat menunjuk dari state aktifitas ke dirinya.

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

2.19.3.3 Diagram Interaksi (*Interaction Diagram*)

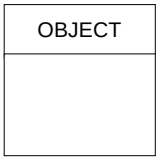
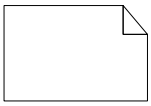
Diagram interaksi terdiri dari dua buah diagram yaitu diagram urutan (*sequence diagram*) dan diagram kolaborasi (*collaboration diagram*). Diagram urutan menekankan pada urutan kejadian, sedangkan diagram kolaborasi menekankan pada hubungan (*relationship*) antar objek.

2.19.3.3.1 Diagram Urutan (*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 atas 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* aktifitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

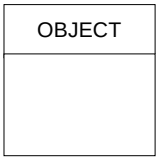
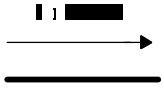
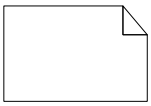
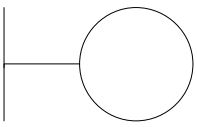
Tabel 2.5 Lambang-lambang dalam Diagram Urutan

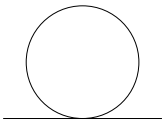
No.	Lambang	Keterangan
1.		Objek, dengan nama dari objek tersebut terdapat didalamnya
2.		Pesan, menunjukkan hubungan antara satu objek dengan objek lainnya. Arah pesan menunjukkan tujuan dari pesan tersebut.
3.		Objek aktor, merupakan objek yang beinteraksi sebagai pengguna sistem. Nama objek dituliskan di bawah gambar yang bersangkutan.
7.		Garis putus-putus. Menunjuk kepada objek yang dijelaskan oleh catatan.
8.		Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.

2.19.3.3.2 Diagram Kolaborasi (*Collaboration Diagram*)

Diagram Kolaborasi 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*, dimana *message* dari level tertinggi memiliki nomor 1. *Message* dari level yang sama memilki prefik yang sama.

Tabel 2.6 Lambang-lambang dalam Diagram Kolaborasi

No.	Lambang	Keterangan
1.		Objek, dengan nama dari objek tersebut terdapat di dalamnya. Objek memilki beberapa *stereotipe* yang dapat dilihat setelah ini.
2.		Pesan, menunjukkan hubungan antara satu objek dengan objek lainnya. Arah panah menunjukkan tujuan dari pesan tersebut. Nomor menunjukkan urutan dari pesan tersebut dalam keseluruhan proses.
3.		Hasil, menunjukkan nilai yang dikembalikan oleh sebuah objek. Arah tanda panah menunjukkan tujuan dari pengiriman hasil tersebut.
4.		Objek aktor, merupakan objek yang beinteraksi sebagai pengguna sistem. Nama objek dituliskan di bawah gambar yang bersangkutan.
5.		Garis putus-putus. Menunjuk kepada objek yang dijelaskan oleh catatan.
6.		Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.
7.		*Boundary*, menggambarkan antar muka antara sistem dengan entitas di luar sistem. *Boundary* merupakan *stereotype* dari kelas.

8.		Kelas Entitas. Digunakan untuk memodelkan informasi dan asosiasi yang harus disimpan. Entitas merupakan stereotipe dari kelas.

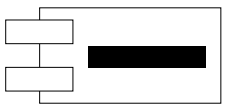
2.19.4 Diagram Implementasi (*Implementation Diagram*)

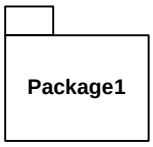
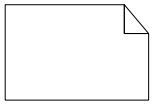
Diagram Komponen dan Diagram penyebaran merupakan bagian dari diagram implementasi. Diagram komponen menggambarkan desain dari perangkat lunak sedangkan diagram penyebaran lebih memfokuskan pada penerapan perangkat lunak tersebut ke dalam perangkat keras.

2.19.4.1 Diagram Komponen (*Component Diagram*)

Diagram komponen menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. 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.

Tabel 2.7 Lambang-lambang dalam Diagram Komponen

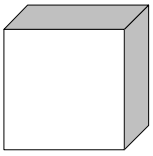
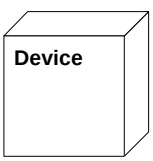
No.	Lambang	Keterangan
1.		Komponen, mewakili modul perangkat lunak (kode sumber, kode biner, program eksekusi, DLL, dan lain sebagainya)

2.		Paket komponen, mewakili tempat dari suatu komponen. Nama paket komponen dapat berisi nama dari direktori *file system.*
3.		Garis putus-putus. Menunjuk kepada objek yang dijelaskan oleh catatan.
4.		Catatan, berisi keterangan tambahan. Garis putus-putus menunjukkan apa yang dijelaskan oleh catatan ini.

2.19.4.2 Diagram Penyebaran (*Deployment Diagram*)

Diagram penyebaran menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, dimana 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.

Tabel 2.8 Lambang-lambang dalam Diagram Penyebaran

No.	Lambang	Keterangan
1.		Prosesor. Menggambarkan perangkat keras yang mampu mengeksekusi program. Setiap prosesor harus memiliki nama yang diletakkan di dalam lambang prosesor.
2.		Devais (*device*). Merupakan perangkat keras yang tidak memiliki kemampuan untuk melakukan penghitungan/ pemrosesan. Contoh device adalah modem, monitor dan juga speaker.

3.	———————	Penghubung, digambarkan oleh sebuah garis yang menghubungkan antara sebuah prosesor dengan prosesor, devais dengan devais, ataupun prosesor dengan devais. Sebuah penghubung boleh diberi nama.

2.20 Kelebihan dan Kekurangan UML

Kelebihan UML dibandingkan dengan bahasa pemodelan yang lain antara lain:

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk bahasa pemodelan.
5. Memadukan praktek-praktek terbaik di industri perangkat lunak menjadi terminologi dan notasi yang diterima luas.
6. Menyediakan kemampuan merepresentasikan semua konsep yang relevan untuk sistem perangkat lunak.
7. Menyediakan fleksibilitas yang diperlukan bagi konsep-konsep perangkat lunak yang baru.

Sedangkan kekurangan UML antara lain :

1. UML bukanlah bahasa pemrograman visual, melainkan bahasa pemodelan visual.
2. UML bukan spesifikasi dari *tool*, tapi spesifikasi bahasa pemodelan.

3. UML bukanlah proses, tapi yang memungkinkan proses-proses.

2.21 Apache

Server HTTP *Apache* atau *Server Web*/WWW *Apache* adalah *server web* yang dapat dijalankan dibanyak sistem operasi (Unix, BSD, Linux, Windows, Novotel Netware dan lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

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

Apache merupakan *software open source* dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*

2.22 PHP

Word Wide Web (WWW) telah berubah dengan cepat dengan berbagai cara Bahasa standar yang dipakai dalam WWW adalah bahasa HTML (*Hyper Text Language*). Dengan adanya teknologi ini, kita dapat melihat dokumen yang kadang-kadang berada dibagian lain di dunia ini dengan hanya sebuah program sederhana dan kita dapat banyak mendapatkan informasi yang dengan cepat dapat kita terima sesuai dengan yang kita perlukan.

Pada mulanya situs web yang ada hanya bersifat statis artinya konten atau isi dari website tersebut terbatas dan jarang di *update* serta tidak adanya interaksi

antara yang mempunyai website dengan pengunjung situs web tersebut. Sekarang kita mengharapkan lebih dari halaman situs web yang statis. Kita mengharapkan dan cepat sebuah situs web yang berisi dengan informasi yang banyak dengan reabilitas yang tinggi artinya selalu *uptodate* dan dalam tampilan yang menarik serta kita dapat mencari dokumen yang diperlukan dengan mudah. Untuk mencapai hal tersebut kita memerlukan suatu web yang dinamis, dengan pengelolaan yang mudah untuk mengubah dan mengatur kontennya serta dapat berinteraksi dengan pengunjung untuk mengetahui perkembangan situs web tersebut.

Berdasarkan hal tersebut diatas, berawal dari seorang yang bernama Rasmus Ledorf, PHP diciptakan. PHP merupakan bahasa *script* yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu baru. Semua *script* php dari pada *spesifikasi client*. Namun tetap diperhatikan bahwa halaman web yang dihasilkan tentunya harus dapat dibuka oleh *browser* pada *client.* Dalam hal ini versi dari html yang digunakan harus didukung oleh *browser client*.

PHP termasuk dalam *Open Source Product*. Jadi dapat merubah *source code* dan mendistribusikannya secara bebas. PHP juga diedarkan secara gratis.

2.23 MYSQL

MySQL merupakan salah satu *software database management system* (DBMS) yang berguna sebagai suatu *Database Server* yang cukup terkenal. Kepopulerannya seiring dengan pengguanan *script* PHP untuk web *programming*.

Database server itu sendiri merupakan suatu *software* yang bertugas untuk melayani permintaan (*request*) *query* dari *client.* MySQL sebagai suatu *database server* mempunyai beberapa kemampuan, salah satunya harus menyediakan suatu sistem manajemen *database* yang dapat mengatur bagaimana menyimpan, menambah, mengubah, menghapus mengakses data, memanipulasi dan transaksi-transaksi *database* lainnya. MySQL cepat sekali berkembang, karena MySQL merupakan suatu *software* yang *Open Source*. [7]

2.24 Macromedia Dreamweaver MX

Macromedia Dreamweaver adalah sebuah HTML editor professional untuk mendesain secara visual dan mengelola situs Web maupun halaman Web. *Macromedia Dreamweaver* merupakan *software* utama yang digunakan oleh *Web Designer* maupun *Web Progremer* guna mengembangkan situs Web. Ruang kerja, fasilitas dan kemampuan *Macromedia Dreamweaver* mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun situs Web. [8]