BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam mendefiniskan sistem, yaitu yang menekankan pada prosedur dan yang menekankan pada komponen atau elemen.

Pendekatan sistem yang lebih menekankan pada prosedur menurut Jerry F, Ardra F, Warren D mendefinisikan sistem sebagai berikut

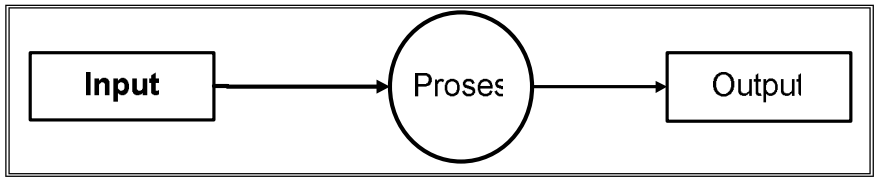
"*Suatu Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu*". (JOG [8]).

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

"*Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu*". (JOG [8]).

2.1.1 Bentuk Umum Sistem

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



Gambar 2.1 Bentuk umum Sistem

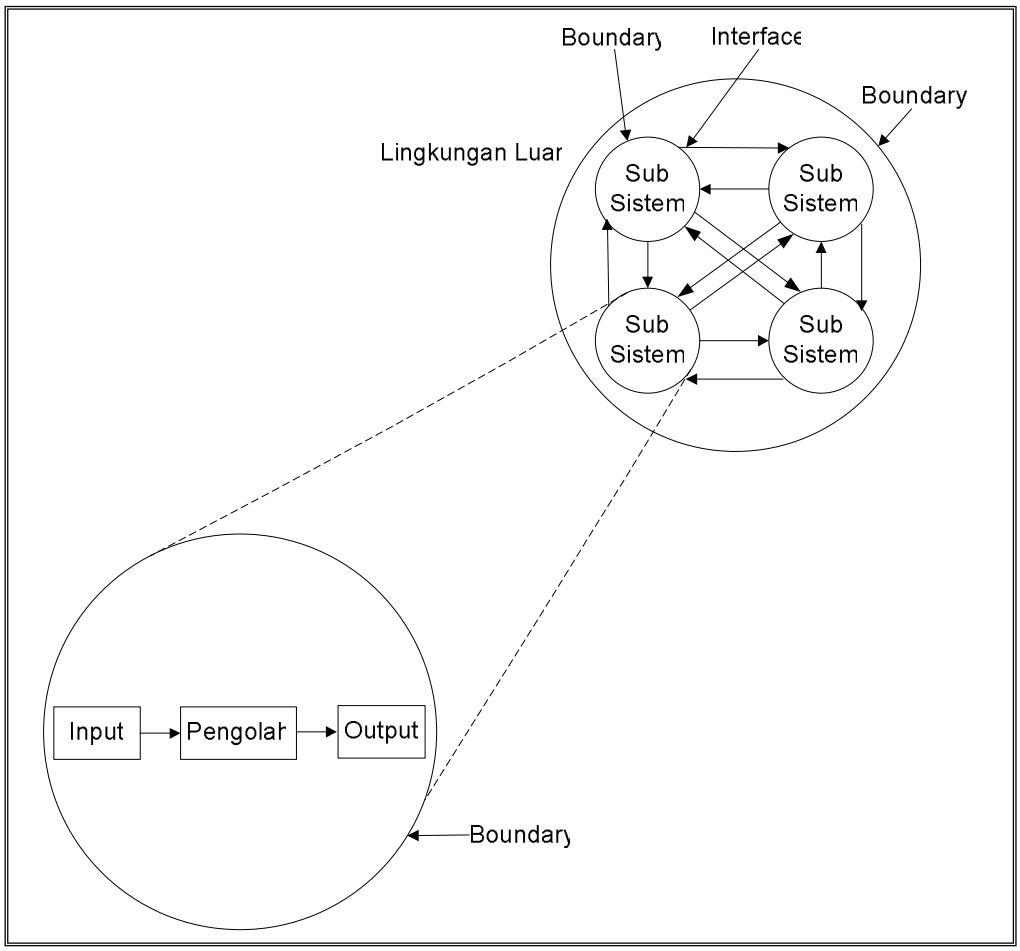
[Sumber : Hartono, JOG [8]]

2.1.2 Karakteristik Sistem

Dalam penggunaanya, suatu sistem mempunyai maksud tertentu. Maksud dari sistem adalah untuk mencapai tujuan dan sasaran yang telah disepakati bersama. Untuk mencapai tujuan dan sasaran tersebut menurut JOG [8] suatu sistem haruslah mempunyai karakteristik atau sifat-sifat tertentu yang saling berhubungan, antara lain sebagai berikut:

1. Komponen Sistem (*Component*), adalah suatu sub sistem atau bagian-bagian dari sistem yang saling berkerja sama membentuk satu kesatuan.
2. Batas Sistem (*Boundary*), merupakan ruang lingkup atau daerah yang membatasi antara suatu sistem dengan sistem yang lainnya dan dengan lingkungan luarnya.
3. Lingkungan Luar Sistem (*Environment*), adalah segala sesuatu yang diluar istilah tersebut sistem yang mempengaruhi operasi sistem baik itu bersifat menguntungkan ataupun merugikan, hal ini merupakan energi suatu sistem yang harus tetap dijaga dan dipelihara.
4. Penghubung Sistem (*Interface*), merupakan media penghubung antara satu sub sistem dengan sub sistem yang lainnya agar dapat berinteraksi dan membentuk kesatuan.

5. Masukan Sistem (*Input*), adalah suatu energi yang dimasukkan kedalam sistem supaya sistem tersebut dapat beroperasi dan menghasilkan keluaran.
6. Keluaran Sistem (*Output*), adalah merupakan hasil dari energi baik masukan untuk sistem yang diolah dan diklasifikasikan menjadi keluaran yang berguna.
7. Pengolahan Sistem (*Process*), suatu bagian yang akan merubah masukan menjadi keluaran.
8. Sasaran Sistem (*Objective*), sesuatu yang menjadi tujuan akhir dari sistem yang hendak dicapai.



Gambar 2.2 Karakteristik Suatu Sistem

[Sumber: JOG [8]]

2.2 Konsep Dasar Informasi

Informasi merupakan hal yang sangat penting di dalam organisasi, jika tidak ada informasi maka tidak ada penciptaan-penciptaan suatu teknologi yang baru yang nantinya akan berkembang pesat seiring waktu. Pentingnya informasi bagi suatu sistem menurut JOG [8] dapat didefinisikan sebagai berikut:

"*Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya*"

Sumber dari informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata pada saat yang tertentu. Jadi tidak ada data atau sumber maka informasi ini tidak akan jalan.

Menurut Joku Brunch dan Gary Grundnitski (JOG [8]) mengatakan bahwa kualitas dari informasi yang dihasilkan haruslah terdiri dari tiga hal, yaitu:

1. Informasi harus akurat

   Artinya informasi harus jelas dan bebas dari kesalahan-kesalahan.

2. Informasi harus tepat waktu

   Artinya informasi yang disampaikan pada penerima harus tepat waktu dan tidak boleh terlambat.

3. Informasi harus relevan

   Artinya informasi tersebut mempunyai manfaat untuk pemakaianya.

Suatu informasi dapat mempunyai nilai. Nilai dari informasi dapat ditentukan dengan dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaat lebih efektif dibandingkan dengan biaya mendapatkannya.

2.3 Konsep Dasar Sistem Informasi

Sistem informasi pada suatu sistem meliputi pemasukan data (*input*) yang kemudian di proses atau diolah, tujuanya untuk mendapatkan suatu keluaran. Hasil informasi atau keluaran tersebut akan ditangkap kembali sebagai suatu *input* dan seterusnya, sehingga membentuk siklus informasi yang dapat diperoleh dari sistem informasi tersebut.

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Pertanyaannya adalah darimana informasi tersebut bisa didapatkan?. Informasi dapat diperoleh dari sistem informasi (*information system*) atau disebut juga dengan *processing systems* atau *information processing system* atau *information-genering systems*.

Sistem informasi di definisikan oleh Robert A. Leitch dan K. Roscoe Davis (JOG [8]) sebagai berikut:

"*Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatannya strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan*".

Menurut Azhar Susanto sistem informasi mempunyai komponen atau faktor-faktor yang mempengaruhi siklus informasi. Komponen tersebut antara lain:

1. Perangkat keras (*Hardware*), merupakan perangkat keras yang dapat digunakan untuk mengumpulkan, memasukan, memperluas menyimpan dan mengeluarkan hasil pengolahan data dalam bentuk informasi.

2. Perangkat Lunak (*Software*), merupakan kumpulan dari program-program yang digunakan untuk menjalankan aplikasi tertentu pada kumpulan perangkat lunak.
3. Sumber Daya Manusia (*Brainware*), merupakan sumber daya yang terlibat dalam pembuatan sistem informasi, pengumpulan dan pengolahan data, pendistribusian dan pemanfaatan informasi yang dihasilkan oleh sistem informasi tersebut.
4. Prosedur (*Procedure*), merupakan rangkaian aktivitas atau kegiatan yang dilakukan secara berulang-ulang dengan cara yang sama.
5. Data, merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data terdiri dari tiga macam, yaitu *input* data, *output* data dan *database*.
6. Jaringan Komunikasi (*Communication Network*), merupakan penggunaan media elektronik atau cahaya untuk memindahkan data atau informasi dari satu lokasi kesatu atau beberapa lokasi lain yang berbeda dengan ruang lingkup yang kecil maupun luas.

2.4 Sistem Informasi Manajemen

Sistem informasi manajemen penerapan sistem di dalam organisasi mendukung informasi-informasi yang dibutuhkan oleh semua tingkatan manajemen. Sistem informasi manajemen ini sangat penting diterapkan karena sebagai penentu sukses tidaknya suatu organisasi bagaimana memanajemen alur sistem informasi yang akan disampaikan kepada orang lain.

Terdapat beberapa definisi dari sistem informasi manajemen diantaranya:

1. Menurut George M. Scott

   "*Sistem informasi manajemen adalah kumpulan dari interaksi-interaksi sistem-sistem informasi yang menyediakan informasi baik untuk kebutuhan manajerial maupun kebutuhan operasi*". (JOG [8]).

2. Menurut Barry E. Cushing

   "*Sistem informasi manajemen adalah kumpulan dari manusia dan sumber-sumber daya modal di dalam suatu organisasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menghasilkan informasi yang berguna untuk semua tingkatan manajemen di dalam kegiatan perencanaan dan pengendalian*". (JOG [8]).

3. Menurut Frederick H. Wu

   "*Sistem informasi adalah kumpulan-kumpulan dari sistem-sistem yang menyediakan informasi untuk mendukung manajemen*". (JOG [8]).

4. Menurut Gordon B. Davis

   "*Sistem informasi manajemen adalah sistem manusia atau mesin yang menyediakan informasi untuk mendukung operasi manajemen dan fungsi pengambilan keputusan dari suatu organisasi*". (JOG [8]).

Dari beberapa defisi tersebut diatas, dapat disimpulkan bahwa:

Menurut JOG [8] "*Sistem informasi manajemen adalah kumpulan dari interaksi sistem-sistem informasi dan menghasilkan informasi yang berguna untuk semua manajemen*".

Manajemen membutuhkan informasi untuk mendukung pengambilan keputusan yang dilakukan sumber informasi eksternal dan informasi internal.

Informasi dapat diperoleh dari sistem informasi yang dihasilkan dari operasi pengolahan data elektronik (PDE) dan informasi bukan pengolahan data elektronik (Non PDE).

2.5 Analisis Sistem

Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap perancangan sistem. Tahap analisis sistem merupakan tahap yang kritis dan sangat penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan ditahap berikutnya.

Menurut JOG [8], analisis sistem didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Langkah-langkah dasar dalam menganalisis sistem adalah sebagai berikut:

1. *Identify,* yaitu mengidentifikasi masalah.
2. *Understand,* yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report,* yaitu membuat laporan hasil analis.

2.6 Perancangan Sistem

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

Tujuan dari perancangan sistem adalah sebagai berikut:

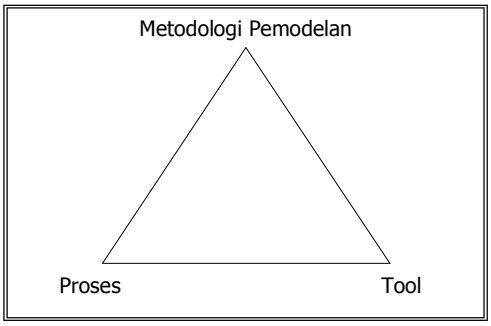
1. Untuk memenuhi kebutuhan pada pemakaian sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemograman komputer dan ahli-ahli teknik lainnya yang terlibat.

2.7 Pemodelan (Modeling)

Pemodelan (*modeling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model piranti lunak dapat dianalogikan seperti pembuatan *blueprint* pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem secara menyeluruh. Semakin kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin kompleks sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik.

Dengan menggunakan model, diharapkan pengembang piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk faktor-faktor seperti *scalability, robustness, security* dan sebagainya.

Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur, yang kemudian terkenal dengan sebutan segitiga sukses (*the triangle for success*). Ketiga unsur tersebut adalah metode pemodelan (*notation*), proses (*process*) dan *tool* yang digunakan. Memahami notasi pemodelan tanpa mengetahui cara pemakainan yang sebenarnya (proses) akan membuat proyek gagal. Dan pemahaman terhadap metode pemodelan dan proses disempurnakan dengan penggunaan *tool* yang tepat. (SRI [11]).



Gambar 2.3 Segitiga Sukses (*the triangle for success*)

[Sumber: SRI [11]]

2.8 Pengenalan Unified Modeling Language (UML)

UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi *object* dan *design* berorientasi *object* (OOA&D) ynga dimunculkan sekitar tahun 80-an dan awal 90-an.

UML merupakan gabungan dari metode *BOOCH*, *Rumbaugh* (OMT) dan *Jacobson*. Tetapi UML ini mencakup lebih luas dari pada OOA dan OOD. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagaian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

2.8.1 Pengertian Unified Modeling Language (UML)

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacs* (bagian dari informasi yang digunakan atau

dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi *object*. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelummnya, dibuat oleh Grady Booch OOD (*Object-Oriented Design*), James Rumbaugh OMT (*Object Modeling Technique*) dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*) di bawah bendera *Rational Software Corp*. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan.

2.8.2 Bagian-Bagian Dari UML

Bagian-bagian UML dibagi dengan dua kategori yakni *view* dan *diagram*

2.8.2.1 View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram.

Beberapa jenis *view* dalam UML antara lain: *use case view, logical view, component view, concurrency view* dan *deployment view.*

1. Use case View

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors. Actor* yang berinteraksi dengan sistem berupa *user* atau sistem lainnya.

View ini digambarkan dalam *use case diagrams* dan kadang-kadang dengan *activity diagrams. View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*depeloper*) dan penguji sistem (*tester*).

2. Logical View

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class, object* dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu.

View Ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state, sequence, collaboration* dan *activity diagram* untuk model dinamisnya. *View* ini digunakan untuk perancangan (*desinger*) dan pengembang (*developer*).

3. Component View

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungan juga alokasi sumber daya komponen dan informasi *administrative* lainnya. *View* ini digambarkan dalam *component view* dan digunakan untuk pengembang (*developer*).

4. Concurrency View

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state, sequence, collaboration* dan *activity diagrams*) dan

diagram implementasi (*component deplomen diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*) dan penguji (*tester*).

5. Deployment View

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya.

View ini digambarkan dalam *development diagrams* dan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*) dan penguji (*tester*).

2.8.2.2 Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain:

1. Use Case Diagram

Menggambarkan sejumlah *external actor*s dan hubunganya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams.*

Use case digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat *user*) dan bukan bagaimana fungsi yang ada di dalam sistem.

2. Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* mempersentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan

dengan yang lain melalui berbagai cara diantaranya *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung atau menggunakan *class* yang lain), *specialed* (satu *class* merupakan spesialisasi dari *class* yang lainnya) atau *package* (group bersama sebagai satu unit).

Sebuah sistem biasanya mempunyai beberapa *class diagram*.

3. **State Diagram**

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan.

State class tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

4. **Sequence Diagram**

Menggambarkan kolaborasi yang dinamis antara sejumlah *object*. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

5. **Collaboration Diagram**

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagrams*.

6. **Activity Diagram**

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi.

7. **Component Diagram**

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *source code*, komponen biner atau *excutable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view.*

8. **Deployment Diagram**

Menggambarkan arsitektur fisik dari perangkat keras dan perngkat lunak sistem, menunjukan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalan *nodes*, *executable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

2.8.3 Gambaran dari UML

1. UML sebagai bahasa pemodelan

UML merupakan bahasa pemodelan yang mememiliki pembendaharaan kata dan cara untuk mempresentasikan secara fokus pada konseptual dan fisik dari suatu sistem. Contoh untuk sistem software yang *intensive* membutuhkan bahasa yang menunjukan pandangan yang berbeda dari arsitektur sistem, ini sama seperti menyusun atau mengembangkan *software development life cycle*. Dengan UML akan memberitahukan kita bagaimana untuk membuat dan membaca bentuk

model yang baik, tetapi UML tidak dapat memberitahukan model apa yang akan dibangun dan kapan akan membangun model tersebut. Ini merupakan aturan dalam *software development process.*

2. UML sebagai bahasa untuk mengambarkan sistem (*visualizing*)

UML tidak hanya merupakan rangkaian simbol grafikal, cukup dengan tiap simbol pada notasi UML merupakan penetapan semantik yang baik. Dengan cara ini, satu pengembang dapat menulis model UML dan pengembang lain atau perangkat yang sama lainnya dapat mengartikan bahwa model tersebut tidak ambigu. Hal ini akan mengurangi *error* yang terjadi karena perbedaan bahasa dalam komunikasi model konseptual dengan model lainnya.

UML menggambarkan model yang dapat dimengerti dan dipresentasikan ke dalam model tekstual bahasa pemrograman. Contohnya kita dapat menduga suatu model dari sistem yang berbasis *web* tetapi tidak secara langsung dipegang dengan mempelajari *code* dari sistem. Dengan model UML maka kita dapat memodelkan suatu sistem *web* tersebut dan direpresentasiakan ke bahasa pemrograman.

UML merupakan suatu model eksplisit yang menggambarkan komunikasi informasi pada sistem. Sehingga kita tidak kehilangan informasi *code* implementasi yang hilang dikarenakan *developer* memotong *coding* dari implementasi.

3. UML sebagai bahasa untuk menspesifikasikan sistem (*specifying*)

Maksudnya membangun model yang sesuai, tidak ambigu dan lengkap. Pada faktanya UML menunjukkan semua spesifikasi keputusan analisis, desain

dan implementasi yang penting yang harus dibuat pada saat pengembangan dan penyebaran dari sistem software intensif.

4. UML sebagai bahasa untuk membangun sistem (*constructing*)

UML bukan bahasa pemograman visual, tetapi model UML dapat dikoneksikan secara langsung pada bahasa pemograman visual.

Maksudnya membangun model yang dapat di mapping ke bahasa pemograman seperti java, C++, VB atau tabel pada *database relational* atau penyimpanan tetap pada *database* berorientasi objek.

5. UML sebagai bahasa pendokumentasian sistem (*documenting*)

Maksudnya UML menunjukan dokumentasi dari arsitektur sistem dan detail dari semuanya. UML hanya memberikan bahasa untuk memperlihatkan permintaan dan untuk tes. UML menyediakan bahasa untuk memodelkan aktifitas dari perencanaan *project* dan manajemen pelepasan (*release management*).

2.9 Indikator Multiplicity

Multiplicity menentukan banyaknya *object* yang terlibat dalam relasi. *Multiplicity* menentukan banyaknya *object* yangterhubung satu dengan yang lainnya. Indikator *multiplicity* terdapat pada masing-masing akhir garis relasi, baik pada asosiasi maupun agregasi.

Beberapa contoh *multiplicity* adalah :

1	Tepat satu
0..*	Nol atau lebih
1..*	Satu atau lebih
0..1	Nol atau satu

5..8 range 5 s.d. 8

4..6,9 range 4 s.d. 6 dan 9

2.10 Pengertian Database

Database adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci tiap file yang ada. (HAR [8])

Database dapat juga diartikan sebagai kumpulan dari beberapa file yang sejenis. *Database* dalam *Database* Managemen Sistem mengandung arti "Sekumpulan data yang saling berhubungan dan berkaitan satu dengan yang lainnya digunakan oleh suatu organisasi".

Database merupakan "Kumpulan data-data yang mempunyai kaitan antara satu data dengan data lain sehingga membentuk satu bangunan data untuk menginformasikan suatu perusahaan/instansi dalam batasan tertentu", sedangkan program pengelolaannya disebut sebagai *Database* Managemen Sistem (DBMS).

2.10.1 Istilah Dalam Database

Ada beberapa istilah yang terdapat atau merupakan bagian dari *database* adalah sebagai berikut:

1. *Entity*, adalah orang, tempat, kejadian atau konsep yang informasinya direkan.
2. *Atribute*, sebagai elemen, data *field*, data item untuk mewakili suatu *entity*.
3. *Data Value* (nilai atau isi data), adalah data aktual atau informasi yang disimpan pada tiap data elemen atau *atribute*.

4. *Record*, ialah kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap.
5. *File*, merupakan kumpulan *record*-record yang mempunyai panjang elemen yang sama, *atribute* yang sama, namun berbeda-beda data *value*nya.

2.10.2 Tujuan Database

Tujuan utama dari *database* adalah Kemudahan dan kecepatan dalam proses akses data.

1. Kecepatan dan kemudahan (*speed*)

   Pemanfaatan *database* memungkinkan kita dapat dengan lebih cepat dan mudah menyimpan, memanipulasi dan menampilkan data.

2. Efisien Ruang Penyimpanan (*Space*)

   Dengan *database* dapat dilakukan penekanan redudansi data yang mungkin terjadi, baik dengan menerapkan sejumlah pengkodean maupun membuat relasi-relasi antar kelompok data yang berkaitan.

3. Keakuratan (*Accuracy*)

   Pemanfaatan pengkodean dan pembentukkan relasi antar data bersama penerapan aturan tipe, domain data, keunikan data dan sebagainya dapat menekan ketidak akuratan pemasukan atau penyimpanan data.

4. Ketersediaan (*Availability*)

   Pemanfaatan *database* memungkinkan kita menyimpan data pada banyak lokasi, termasuk dengan adanya jaringan komputer kita dapat mengakses banyak data yang kita butuhkan.

5. Kelengkapan (*Completeness*)

   Pemanfaatan *database* dapat mengamodasikan kebutuhan yang terus berkembang terhadap data, selain dapat menambahkan *record* namun juga dimungkinkan menambahkan *field-field* pada suatu tabel data.

6. Keamanan (*Security*)

   Dengan menerapkan sistem keamanan atas basis data dengan menentukan pihak mana saja yang berhak untuk mengakses data serta operasi apa saja yang boleh dilakukan.

7. Kebersamaan Pemakaian (*Sharebility*)

   Pemanfaatan *database* dengan menggunakan aplikasi atau sistem yang *multiuser* akan memenuhi kebutuhan banyak pemakai, namun tetap dengan menjaga atau menghindari munculnya inkonsistensi data.

2.11 Objek-Objek Yang Berhubungan Dalam Pembuatan W*eb*

Dalam pembuatan *web* terdapat banyak sekali objek-objek yang saling berhubungan mulai dari tempat desain tampilan, tempat untuk meletakkan *script* programnya, hingga fasilitas yang menghubungkan *web* dengan internet. Objek-objek tersebut antara lain internet, PHP, MySQL, HTML dan program aplikasi Macromedia Dreamweaver MX.

2.12 Internet

Pengertian internet secara umum adalah sebuah jaringan komputer yang terdiri dari berbagai macam ukuran jaringan komputer diseluruh dunia mulai dari sebuah PC, jaringan-jaringan lokal berskala kecil, jaringan-jaringan kelas

menengah hingga jaringan-jaringan utama yang menjadi tulang punggung internet.

Menurut Mac Bridge *Internet adalah jaringan komunikasi global yang terbuka dan menghubungkan ribuan jaringan komputer, telepon umum maupun pribadi (pemerintah maupun swasta)*. (SUG [9]).

Secara individual, jaringan komponennya dikelola oleh agen-agen pemerintah, universitas, organisasi komersial maupun suka relawan, internet muncul dari jaringan jarak jauh yang dikembangkan oleh ARPnet (*US Goverment Advance Research Project Agency Network* atau jaringan agen riset lanjutan dari pemerintah Amerika Serikat) di akhir tahun 1960-an.

2.12.1 Sejarah Internet

Sekitar tahun 1963, RAND *Coorporation,* organisasi pengendali perang dingin Amerika Serikat menghadapi masalah yang sangat pelik yaitu bagaimana Amerika Serikat tetap berkomunikasi dengan lancar setelah perang nuklir terjadi. Pemecahannya adalah dengan membangun sebuah jaringan yang menghubungkan semua tempat strategis di seluruh Amerika dan tetap memberikan *Request For Proposa*l (RFP) kepada UCLA *(University of California Los Angles).* Mereka yang terlibat antara lain Vinton Crf, Stefen Croackern, Jon Postel dan Robert Braden.

Pada tahun 1964 dikeluarkan proposal RAND yang intinya adalah bahwa jaringan yang akan dibentuk tidak berpusat pada suatu dan tetap berfungsi sekalipun dalam keadaan hancur. Proposal RAND ini juga diilhami oleh munculnya “NET” diawal tahun 1990-an yaitu suatu teknologi dimana suatu

pesan disampaikan dari tempat asalnya ke tempat tujuan dengan memfragmentasikan pesan tersebut menjadi bagian-bagian kecil yang disebut paket ditempat asalnya dan dirangkai kembali menjadi pesan semula ditempat tujuan. Teknologi ini dikenal sebagai *Packet Switching Network.*

Tahun 1969 empat buah IMP *(Interface Massage Processor)* dikirimkan kepada 4 perguruan tinggi yakni UCLA, SRI *(Standard Reseach Institute)*, UCSB *(University of California Santa Barbara)* dan *University of Utah.* Jaringan keempat tempat ini kemudian disebut sebagai ARPANET yang disponsori oleh DARPA *(Defense of Advanced Reseach Project Agency)* pentagon, yang sampai sekarang ARPANET merupakan pusat dari jaringan internet.

2.12.2 Fasilitas Yang Ada Di Internet

Fasilitas yang ada di *internet* adalah sebagai berikut:

1. *E-mail*, atau surat elektronik adalah fasilitas yang paling sederhana dan paling banyak digunakan di internet.
2. *Mailing List* atau kelompok diskusi, fasilitas ini di bangun dengan teknik yang sama dengan proses penyebaran surat elektronik. Dengan fasilitas ini berita atau file didistribusikan ke banyak pengguna sekaligus, sehingga pengguna dapat melakukan diskusi di internet.
3. *File Transfer*, digunakan untuk menyalin file dari komputer satu kekomputer yang lain.
4. *www*, digunakan untuk memudahkan dalam melakukan navigasi informasi di internet.
5. *Address* atau alamat adalah letak suatu *web* yang disebut juga dengan domain.

2.12.3 Peralatan Yang Diperlukan Untuk Koneksi Internet

Berbagai jenis komputer dapat digunakan untuk koneksi ke internet, mulai dari *Personal Computer* dengan *processor Intel, Workstation, Unix* sampai *aplle Macintosh*.

Untuk bergabung dalam jaringan internet beberapa perangkat tambahan selain komputer yaitu :

1. Perangkat Keras (*Hardware*)

Perangkat ini merupakan perangkat yang akan menghubungkan komputer kita kejaringan internet, contohnya adalah *modem*, untuk koneksi dengan menggunakan *dialup networking* yang menggunakan saluran komunikasi berupa kabel telepon.

Modem merupakan data komunikasi yang digunakan sebagai antar muka untuk mengirim dan menerima data dari satu komputer ke komputer lain melalui media gelombang elektromagnetik.

2. Perangkat Lunak (*Software*)

Perangkat lunak ini adalah perangkat yang memungkinkan kita dapat mengakses dalam internet, seperti *Netscape*, *Microsoft Internet Explorer*, *Opera*, *E-links*, *Mosaic* dan masih banyak lagi.

3. Penyedia Akses

Penyedia akses dapat dikatakan *provider* dimana kita mendaftar menjadi anggota, atau yang lebih kita kenal dengan nama ISP *(Internet Service Provider)*, seperti Centrin, Indonet, M-*web* dan masih banyak yang lainnya.

Provider merupakan tempat untuk mendaftarkan diri untuk menggabungkan komputer ke internet. *Provider* akan memberikan nama (*user name*) dan juga *password*.

2.12.4 Istilah-Istilah Dalam Internet

Istilah-istilah yang ada dalam internet adalah sebagai berikut:

1. *Browser*, merupakan *software* pelengkap yang digunakan untuk melihat *webpage*.
2. *Browsing*, adalah metode untuk menjelajahi internet.
3. *Hyperlink*, adalah bagian dari *webpage* yang saling berhubungan.
4. *Download*, adalah menyalin file dari komputer satu ke komputer yang lain.
5. *Homepage*, adalah informasi yang tersedia dalam bentuk teks, gambar maupun suara yang diletakkan di komputer.
6. *web*, adalah sistem pada internet yang ditampilkan dalam bentuk teks, gambar, suara.
7. *Website*, adalah lokasi pada *web*.

2.13 World Wide Web (WWW)

World Wide Web (WWW) yang lebih dikenal dengan *web*, merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. *World Wide Web* adalah sarana pembagian informasi antara pengguna jaringan komputer.

Web pada awalnya adalah ruang informasi dalam internet. Dengan menggunakan teknologi *hypertext*, pemakai dituntun untuk menemukan informasi

dengan mengikuti *link* yang disediakan dalam dokumen *web* yang ditampilkan dalam *browser web.*

Kini internet identik dengan *web* karena kepopuleran *web* sebagai standar antarmuka pada layanan-layanan yang ada di internet, dari awalnya sebagai penyedia informasi kini digunakan juga untuk komunikasi dari *e-mail* sampai dengan melakukan transaksi bisnis *(e-commerce).*

Kini *web* seakan lebih popular daripada *e-mail* walaupun secara *statistic e-mail* masih merupakan aplikasi terbanyak yang digunakan oleh pengguna internet. *Web* lebih popular bagi khalayak umum dan pemula terutama untuk tujuan pencarian infromasi dan melakukan komunikasi *e-mail* yang menggunakan *web* sebagai antarmuka.

Web memudahkan pengguna komputer untuk berinteraksi dengan pelaku internet lainnya dan menelusuri informasi di internet.

2.14 Aplikasi Web

Pada awalnya aplikasi *web* dibangun hanya dengan menggunakan bahasa yang disebut HTML (*Hypertext Markup Language*) dan *protocol* yang digunakan dinamakan HTTP (*Hypertext Transfer Protocol*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak skrip seperti itu, antara lain yaitu PHP dan ASP, sedangkan contoh yang berupa objek antara lain *applet* (*Java*).

Aplikasi *Web* sendiri dapat dibagi menjadi:

1. *Web* statis, dan
2. *Web* dinamis.

Web statis dibentuk dengan menggunakan HTML saja. Kekurangan apliksai seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model aplikasi *web* dinamis.

2.15 Teknologi Web

Dari sisi teknologi yang digunakan untuk membuat *Web* Dinamis, terdapat dua macam pengelompokan, yaitu:

1. Teknologi pada sisi klien (*client-side technology*), dan
2. Teknologi pada sisi server (*server-side technology*).

2.15.1 Teknologi Web Pada Sisi Klien

Teknologi *Web* pada sisi klien diimplementasikan dengan mengirimkan kode perluasan HTML atau program tersendiri dan HTML ke klien. Klienlah yang bertanggung jawab dalam melakukan proses terhadap seluruh kode yang diterima.

Kelemahan pendekatan seperti ini adalah terdapat kemungkinan bahwa *browser* pada klien tidak mendukung fitur kode perluasan HTML. Sebagai contoh, kode VB*Script* yang diletakkan pada kode HTML tidak akan berfungsi sekiranya *browser* yang digunakan klien tidak mendukungnya. Kelebihan teknologi pada sisi klien adalah memungkinkan penampilan yang bersifat dinamis, misalnya menampilkan jam yang terus-menerus berubah ataupun untuk membuat animasi gambar yang mengikuti gerakan petunjuk *mouse*.

Yang termasuk dalam teknologi pada sisi klien:

1. Kontrol ActiveX

Kontrol ActiveX adalah suatu komponen yang ditulis dengan menggunakan perangkat lunak seperti Visual C++, Visual BASIC atau Delphi. Jika komponen ini ditambahkan ke dokumen *web*, maka fungsi yang didukungnyaakan tersaji dalam halaman *web*. Misalnya, Kontrol ActiveX dapat digunakan untuk menampilkan grafik tiga dimensi atau bahkan untuk mengimplementasikan permainan (*game*) yang interaktif.

Di dalam dokumen HTML, control ActiveX diletakkan melalui tag <OBJECT>. Dalam hal ini server akan mengirimkan kode program yang melibatkan Applet dan HTML itu sendiri.

Sejauh ini ActiveX hanya berjalan di lingkungan *Windows*, dan hanya *browser* tertentu (misalnya *Internet Explorer*) yang dapat memprosesnya.

2. Java Applet

Applet adalah program yang ditulis dengan menggunakan bahasa pemograman Java. Program ini dapat diletakkan ke halaman *web*, melalui tag HTML bernama <APPLET> dan dapat diproses oleh *browser* yang mendukung Java (misalnya *Internet Explorer* dan *Netscape*). Dalam hal ini server akan mengirimkan kode Applet dan HTML

Berbeda dengan ActiveX, Applet bersifat *cross-platform*, artinya, dapat berjalan pada berbagai *platform*, asalkan platform tersebut mendukung Java.

3. Skrip sisi-klien

Skrip sisi-klien adalah kode-kode yang diletakkan menjadi satu dengan kode HTML dan skrip ini diproses di klien. Dua skrip di sisi klien yang terkenal adalah *JavaScript* dan VB*Script*. *JavaScript* merupakan skrip yang sangat popular

dan dapat berjalan pada hampir semua *browser* masa kini. Adapun VB*Script* hanya di *Internet Explorer*.

Saat ini skrip yang disebut CSS (*Cascading Style Sheets*) dan dikenal dengan sebutan *Dynamic* HTML (DHTML) mulai banyak digunakan. Skrip ini dapat digunakan untuk memformat halaman *web* dengan definisi skrip ditulis terpisah ataupun menyatu dengan HTML.

2.15.2 Teknologi Web Pada Sisi Server

Teknologi *Web* pada sisi server memungkinkan pemrossesan kode di dalam server sehingga kode yang sampai pada pemakai berbeda dengan kode asli pada server.

Keuntungan pengunaan teknologi pada sisi server adalah:

1. Mengurangi lalu-lintas jaringan dengan cara menghindari percakapan bolak-balik antara klien dan *server*.
2. Mengurangi waktu pemuatan kode, mengingat klien hanya mengambil kode HTML saja.
3. Mencegah masalah ketidakkompatibelan *browser*.
4. Klien dapat berinteraksi dengan data yang ada pada server.
5. Mencegah klien mengetahui rahasia kode (mengingat kode yang diberikan ke klien berbeda dengan kode asli pada *server*).

Beberapa contoh teknologi yang berjalan di *server* adalah:

1. *Common Gateway Interface* (CGI)

Pada awalnya, teknologi yang umum digunakan untuk menyajikan data yang bersifat dinamis di lingkungan *web* adalah CGI. Pada prinsipnya CGI dapat

ditulis dengan menggunakan bahasa apa saja. Namun, yang paling popular adalah *Perl*.

Kelemahan CGI terletak pada penciptaan proses sebanyak permintaan dari klien. Jika terjadi jumlah permintaan yang sangat banyak maka akan cenderung melumpuhkan *server*.

2. *Active Server Page* (ASP)

Asp merupakan teknologi di sisi *server* yang paling banyak digunakan dilingkungan *windows* saat ini Asp juga diusahakan untuk berjalan pada *platform* selain *windows*. Teknologi lanjutan dari ASP adalah ASP.net.

3. PHP

PHP merupakan skrip yang berjalan di *server* dan sangat popular di lingkungan *Linux*. Selain ini PHP dapat berjalan pada berbagai *platform*, dari *Unix* hingga *Windows*.

2.16 E-Goverment

E-goverment sangat populer sejalan dengan perkembangan teknologi informasi dan komunikasi. Berbagai negara dibelahan dunia mengimplementasikan *e-goverment* dengan strategi yang disesuaikan dengan kondisi sosial politik serta geografisnya masing-masing.

The Word Bank Group mendefinisikan *e-goverment* sebagai berikut:

"*E-Goverment merupakan penggunaan teknologi informasi (seperti web area network, internet dan mobile computing) oleh organisasi pemerintah yang mempunyai kemampuan membentuk hubungan dengan warga negara, bisnis dan organisasi lain dalam pemerintah*".

Menurut Zweers dan Plangue, 2001 *e-goverment* didefinisikan sebagai berikut:

"*E-Goverment adalah penyediaan informasi layanan atau produk yang disiapkan secara elektronik, dengan dan untuk pemerintah, tidak terbatas tempat dan waktu, menawarkan nilai lebih untuk partisipasi pada semua kalangan*".

Manfaat *E-Goverment* adalah sebagai berikut:

1. Proses Otomatisasi, artinya mengubah peran manusia dalam melakukan proses yang meliputi menambah, menyimpan, merubah data.
2. Proses Informasi, artinya mendukung peran manusia dalam menjalankan proses informasi.
3. Proses Transformasi, artinya membuat informasi komunikasi teknologi baru untuk menjalankan proses informasi.
4. Internal, menyediakan manfaat yang lebih baik untuk memotivasi status pemerintah dan kontrol politik yang baik.
5. Eksternal, dengan penyampaian yang baik, manfaat komunikasi teknologi informasi ini dapat dinikmati oleh populasi yang luas.

2.17 HTML

HTML (Hypertext Markup Language) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web. **(Bima Sunarfrihantono, *PHP dan MySQL untuk Web*, 2002, hal 1)**

HTTP (*Hypertext Transfer Protocol*) merupakan protokol yang digunakan untuk mentransfer data antara *web server ke web browser*. Protokol ini

mentransfer dokumen-dokumen *web* yang ditulis atau berformat HTML (*Hypertext Markup Language*). Dikatakan *Markup Language* karena HTML berfungsi untuk memformat file dokumen teks biasa yang disimpan dengan eksekusi *.htm* atau *.html*.

2.18 PHP

PHP (Hypertext Prepocessor) singkatan dari Personal Home Page tools adalah skrip bersifat server-side yang dapat diintegrasikan dan ditambahkan ke dalam HTML, sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi dinamis. (BIM [5]).

PHP adalah salah satu bahasa *server-side* didesain khusus untuk aplikasi *web*. PHP dapat disisipkan diantara bahasa HTML dan karena bahasa *Server side*, maka pengerjaan skrip akan dilakukan dan dieksekusi di server, baru kemudian hasilnya dikirimkan ke *browser* dalam bentuk HTML.

2.18.1 Sejarah Singkat PHP

PHP pertama kali dibuat pada musim gugur tahun 1994 oleh Rasmus Lerdoff (rasmus@php.net), awalnya digunakan pada *web*sitenya untuk mencatat siapa saja yang berkunjung dan melihat biodatanya. Versi pertama yang di-*release* tersedia pada tahun 1995, dikenal dengan *Personal Home Page*, yang terdir atas *engine parser* yang sangat sederhana yang hanya mengerti beberapa makro khusus dan sejumlah utilitas yang sering digunakan pada halaman-halaman *web*, seperti buku tamu, *counter* pengunjung dan lainnya. Parser diprogram ulang pada pertengahan 1995 dan diberi nama PHP/FI versi 2.0 FI berasal dari paket

Rasmus lainnya yang ditulis untuk menginterprestasi data dari form, yang kemudian dikombinasikan dengan *tool Personal Home Page* dan ditambahkan dukungan *database* mSQL (*mini* SQL).

Tahun 1995 ini dianggap sebagai tahun kelahiran dari PHP/FI yang kemudian membuat pertumbuhan aplikasi *web* yang pesat, dan banyak orang kemudian berkontribusi mengembangkan PHP/FI. Sulit untuk mendapatkan statistik yang tepat untuk memperkirakan penggunaan PHP/FI, tetapi diperkirakan pada akhir tahun 1996 telah digunakan oleh sedikitnya 15000 *website* di seluruh dunia. Dan pertengahan 1997 mencapai 50000 situs.

2.18.2 Keistimewaan PHP

Keunggulan dari bahasa PHP yang sifatnya *server-side* tersebut antara lain:

1. Tidak diperlukan kompatibilitas *browser* atau harus menggunakan *browser* tertentu, karena serverlah yang akan mengerjakan skrip PHP tersebut dan biasanya yang dikirim ke *browser* umumnya bersifat teks atau gambar saja sehingga pasti dikenal oleh *browser* apa pun.
2. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh *server*, misalnya koneksi ke *database*.
3. Skrip tidak dapat dilihat dengan menggunakan fasilitas *view* HTML *source*.

Kelebihan PHP yaitu dapat melakukan semua aplikasi program CGL, seperti mengambil nilai *form*, menghasilkan halaman *web* yang dinamis, serta mengirim dan menerima *cookie*. PHP juga dapat berkomunikasi dengan layanan-layanan yang menggunakan protokol IMAP, SNMP, NNTP, POP3,

HTTP, dan lain-lain. Namun kelebihan PHP yang paling signifikan adalah kemampuannya untuk melakukan koneksi dengan berbagai macam *database*. Saat ini *database* yang didukung oleh PHP adalah *Adabas D, dBase, Inter Base, Postger SQL, Front Base, Solid, Empress, MSQL, Sybase, File Pro, Direct MS-SQL, IBM DB2, Velocis, MySQL, Unix dan Informix*. Semua *database* yang mempunyai *provider ODBC, Ingres, Oracle (OC17 and OC18)*.

2.19 MySQL

MySQL adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakannya, tapi tidak boleh dijadikan produk turunan yang bersifat *Closed Source* atau komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structur Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *Optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibanding *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query MySQL* bisa sepuluh kali lipat lebih cepat dari *PostgreSQL* dan Lima kali lebih cepat dibanding *Interbase*.

2.19.1 Sejarah Singkat MySQL

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang *software* dan konsultan *database* bernama *MySQL* AB yang bertempat di Swedia. Waktu itu perusahaan tersebut masih bernama TcX Data *Konsult* AB, dan tujuan awal dikembangkannya *MySQL* adalah untuk mengembangkan aplikasi berbasis *web* pada *client*.

Awalnya Micheal Widenius "Monty", pengembang satu-satunya TcX, memiliki sebuah aplikasi *UNIREG* dan rutin *ISAM* buatannya sendiri dan sedang mencari antarmuka SQL yang cocok untuk diimplementasikan kedalamnya. Mula-mula Monty memakai *miniSQL* (mSQL) pada percobaannya itu, tetapi mSQL dirasa kurang sesuai, karena terlalu lambat dalam pemrosesan *query* data.

Akhirnya Monty menghubungi David Hughes, pembuat mSQL yang sedang merilis versi kedua dari mSQL. Kemudian Monty mencoba membuat sendiri mesin SQL yang memiliki antarmuka mirip dengan SQL, tetapi dengan kemampuan yang lebih sesuai, dan lahirlah *MySQL*.

2.19.2 Keistimewaan MySQL

Sebagai *database server* yang memiliki konsep *database modern*, *MySQL* memiliki banyak keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh *MySQL*.

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *Free*BSD, Mac OS X *Server*, dan lain-lain.

2. *Open Source*

MySQL didistribusikan secara *Open Source*, sehingga dapat digunakan secara bebas.

3. *Multi-user*

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

4. *Performance Tuning*

   MySQL memiliki kecepatan yang tinggi dalam menangani *query*, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column Types*

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

6. *Command* dan *Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

7. *Security*

   MySQL memiliki lapisan lapisan sekuritas seperti *level subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

8. *Scalability* dan *Limits*

   MySQL mampu menangani *database* dengan skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 juta tabel serta 5 miliar baris. Selain itu,

batas *indeks* yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, Unix Socet (UNIX), atau Named Pipes (NT).

10. *Localisation*

    Deteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa

11. *Interface*

    Interface terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*)

12. *Client* dan *Tools*

    Dilengkapi dengan berbagai *Tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

13. *Struktur Table*

    Struktur tabel yang lebih fleksibel dalam menangani *ALTER TABEL*, dibanding *database* lainnya semacam *Postgre*SQL ataupun *Oracle*.

2.20 Apache

Untuk menjalankan PHP dan *MySQL*, kita membutuhkan sebuah *webserver*. *Apache* merupakan salah satu *web server* yang ketangguhanya telah teruji. *Web server* adalah suatu program yang dijalankan pada komputer yang terkoneksi dengan internet, *web server* melihat hubungan internet dan menunggu

perintah atau permintaan dari *web browser* HTML atau dokumen PHP. Saat ini *Apache* merupakan *web server* yang paling popular.

2.21 Macromedia Dreamweaver MX

Macromedia Dreamweaver MX adalah sebuah *software web design* yang menawarkan cara mendesain dengan dua langkah sekaligus dalam satu waktu, yaitu mendesain dan memprogram.

Dreamweaver memiliki satu jendela mini yang disebut HTML *source* tempat meletakkan kode-kode HTML tertulis, setiap akan mendesain *website* seperti menulis kata-kata, meletakkan gambar, membuat tabel dan proses lainnya, tag-tag HTML akan ditulis secara langsung mengiringi proses pengaturan *website*. Artinya, selain untuk mendesain *website* sekaligus mengenal *tag-tag* HTML yang membangun *website* itu. Ataupun sebaliknya menulis *tag-tag* dan teks lain di jendela HTML *source* dan hasilnya bisa dilihat langsung dilayar.

2.22 Kesehatan

Kesehatan adalah kebutuhan dasar bagi semua orang dimanapun berada. Dahulu kala pelayanan kesehatan lebih perhatian kepada masalah sakit atau penyakit yang dialami oleh penderita untuk dilakukan penyembuhan sampai penderita menjadi sehat kembali. Tanpa memperhatikan lagi bahwa penyakit tersebut dapat kambuh kembali dikemudian hari.

Sebetulnya kesehatan dapat di promosikan secara baik di manapun. Tempat pelayanan kesehatan, petugas kesehatan dan peralatan yang canggih dan mahal dengan majunya perkembangan teknologi saat ini untuk menjalankan

prosedur pemeriksaan kesehatan mengakibatkan takutnya masyarakat ataupun yang membutuhkan untuk berobat terutama untuk sosial ekonomi yang rendah atau paspasan. Sementara pelayanan kesehatan yang bersifat pencegahan dengan teknologi yang sederhana masih dibutuhkan, untuk mengurangi kematian.

Masih banyak juga masyarakat kurang mampu menjangkau pelayanan kesehatan karena hambatan geogravis dan transportasi sehingga hal ini menyebabkan keterlambatan dalam pertolongan. Dengan kemajuan teknologi kedokteran saat ini seharusnya semua masyarakat Indonesia merasakan kemajuan tersebut, tetapi karena keterbatasan biaya dan pemerataan pelayanan maka teknologi tersebut baru dapat dirasakan sebagain kecil masyarakat saja. Keadaan tersebut diatas merupakan gambaran pembangunan kesehatan di Indonesia untuk itu sebaiknya kesehatan harus menjadi kebutuhan dasar dan merupakan juga hak azazi manusia dalam mencapai Indonesia sehat 2010. Prinsip ini merupakan visi dari pembangunan kesehatan Indonesia dimana semua masyarakat hendaknya derajat kesehatannya memungkinkan untuk produktif dalam bekerja dan berpartisipasi secara aktif dalam pencegahan dan pemeliharaan kesehatannya. Hal tersebut diatas tidak mungkin dapat dicapai melalui kesehatan saja tetapi juga harus melibatkan semua sektor yang terkait.