

BAB 2

LANDASAN TEORI

Pada bab ini diuraikan beberapa landasan teori dan konsep-konsep yang digunakan dalam penelitian ini. Teori yang digunakan adalah teori-teori *Certainty Factor* (Faktor Kepastian).

2.1. Intelegensia Semu

Istilah kecerdasan buatan berasal dari bahasa Inggris: “*Artificial Intelligence*”. Istilah ini didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu *entitas* buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. (http://id.wikipedia.org/wiki/Kecerdasan_buatan).

Menurut Turban dan Frenzel (1992, p3) kecerdasan buatan atau *artificial intelligence* merupakan bagian ilmu komputer yang digunakan untuk menciptakan perangkat lunak dan perangkat keras dengan tujuan untuk menghasilkan sesuatu seperti yang dihasilkan oleh manusia.

2.1.1 Latar Belakang Intelegensia Semu

Pada awal diciptakannya komputer, ia hanya difungsikan sebagai alat hitung. Seiring dengan perkembangan jaman, penggunaan komputer semakin mendominasi kehidupan manusia sehingga komputer tidak hanya digunakan

sebagai alat hitung saja, tetapi dapat menggantikan beberapa pekerjaan yang biasanya dilakukan oleh manusia.

Manusia menjadi pintar dalam menyelesaikan segala permasalahan yang dihadapi karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan didapatkan dari proses belajar, pengalaman didapatkan karena perjalanan waktu dan kehidupan yang dialami oleh manusia. Semakin banyak bekal pengetahuan dan pengalaman yang dimiliki oleh seseorang, diharapkan orang tersebut lebih mampu menyelesaikan masalah yang dihadapinya.

Namun bekal pengetahuan saja tidak cukup, manusia juga diberikan akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang dimilikinya. Tanpa memiliki kemampuan penalaran yang baik, tidak ada artinya manusia itu memiliki pengetahuan dan pengalaman sebanyak apapun. Demikian juga sebaliknya, walaupun seorang manusia memiliki kemampuan penalaran yang baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak dapat menyelesaikan masalahnya dengan baik.

Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan dan diberikan kemampuan untuk menalar. Untuk itu, *artificial intelligence* akan mencoba untuk memberikan beberapa metode untuk membekali computer dengan kedua komponen tersebut agar computer bisa menjadi mesin yang cerdas.

2.1.2 Definisi Intelegensia Semu

Menyatakan computer itu cerdas tentu tergantung sudut pandang dari orang yang memanfaatkan computer tersebut. Karena itulah sangat sulit untuk mendefinisikan dengan pasti apa yang dimaksud dengan kecerdasan buatan itu. Menurut John McCarthy, 1965, *Artificial Intelligence* adalah untuk mengetahui dan memodelkan proses – proses berpikir manusia dan mendesai mesin agar dapat menirukan perilaku manusia.

Menurut Russell dan Norvig (2003, p5) definisi tentang kecerdasan buatan dikembangkan berdasarkan empat kelompok kategori, yaitu :

- Sistem yang berpikir selayaknya manusia berfikir (*thinking humanly*).
- Sistem yang bertindak selayaknya manusia bertindak (*acting humanly*).
- Sistem yang berpikir secara rasional (*thinking rationally*).
- Sistem yang bertindak secara rasional (*acting rationally*).

Dari keempat perspektif diatas, pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain:

1. Sudut pandang kecerdasan, kecerdasan buatan akan membuat mesin menjadi cerdas, yaitu mampu berbuat seperti apa yang dilakukan oleh manusia.
2. Sudut pandang penelitian, kecerdasan buatan adalah suatu studi bagaimana membuat mesin atau komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia. Ada beberapa bidang (*domain*) yang sering dibahas oleh para peneliti meliputi:

a. Mundane Task

- Persepsi (*vision & speech*)
- Bahasa alami (*understanding, generation & translation*)
- Pemikiran yang bersifat *commonsense*
- Robot kontrol

b. Formuliral Task

- *Games* / permainan.
- Matematika (geometri, logika, kalkulus integral, pembuktian)

c. Expert Task

- Analisis finansial
- Analisis medikal
- Analisis ilmu pengetahuan
- Rekayasa (*desain*, pencarian kegagalan, perencanaan manufaktur)

3. Sudut pandang bisnis

Kecerdasan buatan adalah sekumpulan peralatan (*tools*) yang sangat *powerfull* dan metodologis dalam menyelesaikan masalah bisnis.

4. Sudut pandang pemrograman

Kecerdasan buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).

Untuk melakukan aplikasi kecerdasan buatan ada dua bagian utama yang sangat dibutuhkan, yaitu :

- a. Basis Pengetahuan (*Knowledge Base*), berisi fakta – fakta, teori, pemikiran dan hubungan antara satu dengan yang lainnya.

- b. Motor Inferensi (*Inference Engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.

2.1.3 Tujuan Intelegensia Semu

Adapun tujuan Intelegensia Semu adalah sebagai berikut:

1. Untuk mengembangkan metode dan sistem untuk menyelesaikan masalah, masalah yang biasa diselesaikan melalui aktifitas intelektual manusia, misalnya pengolahan citra, perencanaan, peramalan dan lain-lain, meningkatkan kinerja sistem informasi yang berbasis komputer.
2. Untuk meningkatkan pengertian/pemahaman kita pada bagaimana otak manusia bekerja

2.1.4 Beda Intelegensia Semu dan Kecerdasan Alami

Kelebihan kecerdasan buatan:

1. Lebih bersifat permanent

Kecerdasan alami bisa berubah karena sifat manusia pelupa. Kecerdasan buatan tidak berubah selama sistem computer dan program tidak mengubahnya.

2. Lebih mudah diduplikasi dan disebarkan

Mentransfer pengetahuan manusia dari 1 orang ke orang lain membutuhkan proses yang sangat lama dan keahlian tidak akan pernah dapat diduplikasi dengan lengkap. Jadi jika pengetahuan terletak pada suatu sistem computer, pengetahuan tersebut dapat disalin dari computer dan dapat dengan mudah dipindahkan ke computer yang lain

3. Lebih murah

Menyediakan layanan komputer akan lebih mudah dan murah dibandingkan mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.

4. Bersifat konsisten karena kecerdasan buatan adalah bagian dari teknologi computer sedangkan kecerdasan alami senantiasa berubah – ubah.

5. Dapat didokumentasi

Keputusan yang dibuat computer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit direproduksi.

6. Lebih cepat

7. Lebih baik

Kelebihan kecerdasan alami:

1. Kreatif: manusia memiliki kemampuan untuk menambah pengetahuan, sedangkan pada kecerdasan buatan untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun
2. Memungkinkan orang untuk menggunakan pengalaman secara langsung. Sedangkan pada kecerdasan buatan harus bekerja dengan input – input simbolik.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

2.2 Sistem Pakar Untuk diagnosa Penyakit

2.2.1 Definisi Sistem Pakar

Secara umum, sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli (Kusumadewi, 2003, p109). Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman.

Ada beberapa definisi tentang sistem pakar (Kusumadewi, 2003, p109), antara lain:

1. Menurut Durkin: Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan seorang pakar.
2. Menurut Ignizio: Sistem pakar adalah suatu model dan prosedur yang berkaitan, dalam suatu domain tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar.
3. Menurut Giarratano dan Riley: Sistem pakar adalah suatu sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar.

Sistem pakar pertama kali dikembangkan oleh komunitas AI pada pertengahan tahun 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose Problem Solver* (GPS) yang dikembangkan oleh Newel dan Simon. GPS (dan program-program yang serupa) ini mengalami kegagalan dikarenakan

cakupannya terlalu luas sehingga terkadang justru meninggalkan pengetahuan-pengetahuan penting yang seharusnya disediakan.

Sampai saat ini sudah banyak sistem pakar yang dibuat, beberapa contoh diantaranya terlihat pada Tabel 2.1.

Tabel 2.1. Sistem pakar yang terkenal

Sistem Pakar	Kegunaan
MYCIN	Diagnosa penyakit
DENDRAL	Mengidentifikasi struktur molekular campuran yang tak dikenal
XCON & XSEL	Membantu konfigurasi sistem komputer besar
SOPHIE	Analisis sirkit elektronik
Prospector	Digunakan di dalam geologi untuk membantu mencari dan menemukan deposit
FOLIO	Menbantu memberikan keutusan bagi seorang manajer dalam hal stok broker dan investasi
DELTA	Pemeliharaan lokomotif listrik disel

2.2.2 Keuntungan Sistem Pakar

Secara garis besar, manfaat yang dapat diambil dengan adanya sistem pakar antara lain:

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis.

3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Memiliki kemampuan untuk mengakses pengetahuan.
6. Memiliki reliabilitas.
7. Meningkatkan keabilitas sistem komputer.
8. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
9. Sebagai media pelengkap dalam penelitian.
10. Meningkatkan kapabilitas dalam penyelesaian masalah.
11. Menghemat waktu dalam pengambilan keputusan.

2.2.3 Kelemahan Sistem Pakar

Disamping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan, yaitu diantaranya:

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar dibidangnya.
3. Sistem Pakar tidak 100% bernilai benar, domain kepakarannya sangat terbatas.

2.2.4 Konsep Dasar Sistem Pakar

Menurut Efraim Turban (Kusumadewi, 2003, p111), konsep dasar sistem pakar mengandung: keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan.

Keahlian adalah suatu kelebihan penguasaan pengetahuan di bidang tertentu

yang diperoleh dari pelatihan, membaca atau pengalaman. Contoh bentuk pengetahuan yang termasuk keahlian adalah:

- Fakta-fakta pada lingkup permasalahan tertentu.
- Teori-teori pada lingkup permasalahan tertentu.
- Prosedur-prosedur dan aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
- Strategi-strategi global untuk menyelesaikan masalah.
- *Meta-knowledge* (pengetahuan tentang pengetahuan).

Bentuk-bentuk ini memungkinkan para ahli untuk dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan ahli.

Seorang ahli adalah seseorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan (domain), menyusun kembali pengetahuan jika dipandang perlu, memecah aturan-aturan jika dibutuhkan, dan menentukan relevan tidaknya keahlian mereka.

Pengalihan keahlian dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, merupakan tujuan utama dari sistem pakar.

Proses ini membutuhkan 4 aktivitas yaitu :

- Tambahan pengetahuan (dari para ahli atau sumber-sumber lainnya)
- Representasi pengetahuan (ke komputer)
- Inferensi pengetahuan,
- Pengalihan pengetahuan ke user.

Pengetahuan yang disimpan di komputer disebut dengan nama basis pengetahuan. Ada 2 tipe pengetahuan, yaitu :

- Fakta
- Prosedur (biasanya berupa aturan).

Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar. Jika keahlian-keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basisdata, maka komputer harus dapat diprogram untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk motor inferensi (*inference engine*).

Sebagian besar sistem pakar komersial dibuat dalam bentuk *rule-based systems*, yang mana pengetahuannya disimpan dalam bentuk aturan-aturan. Aturan tersebut biasanya berbentuk IF-THEN.

Fitur lainnya dari sistem pakar adalah kemampuan untuk merekomendasi. Kemampuan inilah yang membedakan sistem pakar dengan sistem konvensional. Perbandingan antara sistem pakar dengan sistem konvensional dapat dilihat pada Tabel 2.2.

Tabel 2.2. Sistem konvensional vs. sistem pakar.

<i>Sistem Konvensional</i>	<i>Sistem Pakar</i>
Informasi dan pemrosesannya biasanya jadi satu dengan program	Basis pengetahuan merupakan bagian dari mekanisme inferensi
Biasanya tidak bisa menjelaskan mengapa suatu input data itu dibutuhkan, atau bagaimana output itu	Penjelasan adalah bagian terpenting dari sistem pakar

<i>Sistem Konvensional</i>	<i>Sistem Pakar</i>
diperoleh	
Pengubahan program cukup sulit & membosankan	Pengubahan aturan dapat dilaksanakan dengan mudah
Sistem hanya akan beroperasi jika sistem tersebut sudah lengkap	Sistem dapat beroperasi hanya dengan beberapa aturan.
Eksekusi dilakukan langkah demi langkah	Eksekusi dilakukan pada keseluruhan basis pengetahuan
Menggunakan data	Menggunakan pengetahuan
Tujuan utamanya adalah efisiensi	Tujuan utamanya adalah efektivitas

2.2.5 Bentuk Sistem Pakar

Ada 4 bentuk sistem pakar, yaitu:

1. Berdiri sendiri. Sistem pakar jenis ini merupakan *software* yang berdiri-sendiri tidak tergantung dengan *software* yang lainnya. Semua contoh sistem pakar pada Tabel 2.1 Merupakan sistem pakar jenis ini.
2. Tergabung. Sistem pakar jenis ini merupakan bagian program yang terkandung didalam suatu algoritma (konvensional), atau merupakan program dimana didalamnya memanggil algoritma subrutin lain (konvensional).
3. Menghubungkan ke *software* lain . Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya DBMS.

4. Sistem Mengabdikan. Sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar.

2.2.6 Ciri-ciri Sistem Pakar

Sistem pakar yang baik harus memenuhi ciri-ciri sebagai berikut:

1. Memiliki fasilitas informasi yang handal.
2. Mudah dimodifikasi.
3. Dapat digunakan dalam berbagai jenis komputer.

2.2.7 Permasalahan Yang Disentuh Oleh Sistem Pakar

Ada beberapa masalah yang menjadi area luas aplikasi sistem pakar, antara lain:

1. Interpretasi. Pengambilan keputusan dari hasil observasi, termasuk diantaranya: pengawasan, pengenalan ucapan, analisis citra, interpretasi sinyal, dan beberapa analisis kecerdasan.
2. Prediksi. Termasuk diantaranya: peramalan, prediksi demografis, peramalan ekonomi, prediksi lalu lintas, estimasi hasil, militer, pemasaran, atau peramalan keuangan.
3. Diagnosis. Termasuk diantaranya: medis, elektronis, mekanis, dan diagnosis perangkat lunak.
4. Perancangan. Termasuk diantaranya: layout sirkuit dan perancangan bangunan.
5. Perencanaan. Termasuk diantaranya: perencanaan keuangan, komunikasi, militer, pengembangan produk, routing, dan manajemen proyek.

6. Monitoring. Misalnya: *Computer-Aided Monitoring Systems*.
7. Debugging, memberikan resep obat terhadap suatu kegagalan.
8. Perbaikan.
9. Instruksi. Melakukan instruksi untuk diagnosis, debugging, dan perbaikan kerja.
10. Kontrol. Melakukan kontrol terhadap interpretasi interpretasi, prediksi, perbaikan, dan monitoring kelakuan sistem.

2.3 *Certainty Factor* (Faktor Kepastian)

Certainty Factor (CF) menunjukkan ukuran kepastian terhadap suatu fakta atau aturan. Sistem pakar harus mampu bekerja dalam ketidakpastian (Giarattano dan Riley, 1994). Sejumlah teori telah ditemukan untuk menyelesaikan ketidakpastian, termasuk diantaranya probabilitas klasik (*classical probability*), probabilitas Bayes (*Bayesian probability*), teori Hartley berdasarkan himpunan klasik (*Hartley theory based on classical sets*), teori Shannon berdasarkan pada probabilitas (*Shanon theory based on probability*), teori Dempster-Shafer (*Dempster- Shafer theory*), teori fuzzy Zadeh (*Zadeh's fuzzy theory*) dan faktor kepastian (*Certainty Factor*).

Faktor kepastian merupakan cara dari penggabungan kepercayaan (*belief*) dan ketidakpercayaan (*unbelief*) dalam bilangan yang tunggal. Dalam *Certainty theory*, data-data kualitatif direpresentasikan sebagai derajat keyakinan (*degree of belief*).

Tahapan dalam merepresentasikan data-data kualitatif:

1. Kemampuan untuk mengekspresikan derajat keyakinan sesuai dengan metode yang sudah dibahas sebelumnya.
2. Kemampuan untuk menempatkan dan mengkombinasikan derajat keyakinan tersebut dalam sistem pakar.

Dalam mengekspresikan derajat keyakinan digunakan suatu nilai yang disebut *certain Factor (CF)* untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data.

Certainty Factor didefinisikan sebagai pada persamaan berikut (Giarattano dan Riley, 1994):

$$CF(H,E) = MB(H,E) - MD(H, E)$$

Dimana:

$CF(H,E)$: *Certainty Factor* dari hipotesis H yang dipengaruhi oleh gejala (*evidence*) E. Besarnya CF berkisar antara -1 sampai dengan 1. Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1 menunjukkan kepercayaan mutlak.

$MB(H,E)$: ukuran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

$MD(H,E)$: ukuran kenaikan ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

Bentuk dasar rumus *Certainty Factor* sebuah aturan JIKA E MAKA H adalah seperti ditunjukkan oleh persamaan berikut:

$$CF(H,e) = CF(E, e) * CF(H,E)$$

dimana

$CF(E,e)$: *Certainty Factor evidence* E yang dipengaruhi oleh *evidence* e

$CF(H,E)$: *Certainty Factor* hipotesis dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika $CF(E, e) = 1$

$CF(H,e)$: *Certainty Factor* hipotesis yang dipengaruhi oleh *evidence* e.

Jika semua *evidence* pada *antecedent* diketahui dengan pasti maka persamaannya akan menjadi:

$$CF(H, e) = CF(H, E)$$

Dalam aplikasinya, $CF(H,E)$ merupakan nilai kepastian yang diberikan oleh pakar terhadap suatu aturan, sedangkan $CF(E,e)$ merupakan nilai kepercayaan yang diberikan oleh pengguna terhadap gejala yang dialaminya. Sebagai contoh, berikut ini adalah sebuah aturan dengan CF yang diberikan oleh seorang pakar:

JIKA batuk

DAN demam

DAN sakit kepala

DAN bersin-bersin

MAKA influenza, CF: 0,7

2.4 Hypertext Markup Language (HTML)

HyperText Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser Internet (<http://id.wikipedia.org/wiki/HTML>). Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized*

Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium (W3C)*.

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan *browser web* seperti *Mozilla Firefox* atau *Microsoft Internet Explorer*. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan *browser*.

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: **TAMPIL TEBAL**, maka penulisannya dilakukan dengan cara: `TAMPIL TEBAL`. Tanda `` digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda `` untuk menonaktifkan cetak tebal tersebut.

Secara garis besar, terdapat 4 jenis elemen dari HTML:

- *Structural*. tanda yang menentukan level atau tingkatan dari sebuah teks (contoh, `<h1>Golf</h1>` akan memerintahkan browser untuk menampilkan "Golf" sebagai teks tebal besar yang menunjukkan sebagai Heading 1

- *Presentational*. tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, `boldface` akan menampilkan **bold**. Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks,
- *Hypertext*. tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain (contoh, `Wikipedia` akan menampilkan Wikipedia sebagai sebuah hyperlink ke URL tertentu),
- Elemen *widget* yang membuat objek-objek lain seperti tombol (`<button>`), list (``), dan garis horizontal (`<hr>`).

Selain markup *presentational*, markup yang lain tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan *tag* HTML untuk menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheets*.

2.5 Hypertext Preprocessor (PHP)

2.5.1 Definisi Hypertext Preprocessor (PHP)

Hypertext Preprocessor (PHP) adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memprogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain (<http://id.wikipedia.org/wiki/PHP>).

2.5.2 Sejarah Hypertext Preprocessor (PHP)

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Formulir *Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessing/Formulir Interpreter*. Dengan perilsan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

1.5.3 Kelebihan Hypertext Preprocessor (PHP)

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.6 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management sistem*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia

(<http://id.wikipedia.org/wiki/MySQL>). MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael “Monty” Widenius.

MySQL adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun 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 (*Structured 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 dibandingkan *database server* lainnya dalam *query data*. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain

(<http://id.wikipedia.org/wiki/MySQL>):

1. *Portability.*

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X *Server*, Solaris, Amiga, dan masih banyak lagi.

2. *Open Source.*

MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performulirance tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, 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, text, date, timestamp, dan lain-lain.

6. *Command dan functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

7. *Security*

MySQL memiliki beberapa 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* dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

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

10. *Localization*

MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Clients* dan *tools*

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

13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan *database* lainnya semacam PostgreSQL ataupun Oracle.