BAB II

LANDASAN TEORI

2.1 *Artificial Intelligence*

2.1.1 Sejarah Singkat

Awal pekerjaan dipusatkan pada seperti *game playing* (misalnya: audio dengan kecerdasan dan permainan catur (*chess player*), pembuktian teorema (theorem proving) pada Tugas-tugas formal (*Formal Tasks*).

Samual (1963) menulis sebuah program yang diberi nama check-er-playing program, yang tidak hanya untuk bermain *game*, tetapi digunakan juga pengalamannya pada permainan untuk mendukung kemampuan sebelumnya.

Catur juga diterima, karena banyak sekali perhatian terhadap permaianan catur yang merupakan permainan yang lengkap atau kompleks, program catur di sini situasinya harus jelas dan rule atau ketentuannya harus seperti dunia nyata.

Kandidat Artificial Intelligence harus mampu menangani masalah-masalah yang sulit. *Logic theorist* diawal percobaan untuk membuktikan teorema matematika. Ia mampu membuktikan beberapa teorema dari bab 1 Prinsip Matematika Whiteheat dan Russell. *Theorema Gelernter* (1963) membuktikan pencarian area yang lain dari matematika yaitu geometri.

Pada tahun 1963, pemecahan masalah umum menggunakan *object*, pembuktian dengan atraksi (eksternal).

Dari awal pekerjaan *Artificial Intelligence* ini memindahkan lebih khusus tugas yang sering berguna antara lain:

1. Tugas biasa/keduniaan (*Mundane Tasks*)
   - · Persepsi : - *vision*
     - *speech*
   - · *Natural Language :* - *understanding*
     - *generation*
     - *translation*
   - · *Commonsence Reasoning:* - *robot control*

2. Tugas Formil (*Formal Tasks*)
   - · Games : - *chess*
     - *checkers*
   - · *Math:* - *geometri*
     - *logic*
     - *proving properties of programs*

3. Tugas Ahli (*Expert Tasks*)
   - · Teknik : - *Design*
     - *Fault Diagnosis*
     - *Planning*

· Scientific Analysis

· Medical : - *Diagnosis & Theraphy.*

2.1.2 Pengertian *Artificial Intelligence*

Definisi *Artificial Intelligence* merupakan proses di mana peralatan mekanik dapat melaksanakan kejadian-kejadian dengan menggunakan pemikiran atau kecerdasan seperti manusia.

Menurut beberapa ahli kecerdasan buatan didefinisikan sebagai berikut :

1. H. A.Simon [1987] :

   "Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang -dalam pandangan manusia adalah- cerdas "

2. Rich and Knight [1991]:

   "Kecerdasan Buatan (*artificial intelligence*) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia."

3. Encyclopedia Britannica:

   "Kecerdasan Buatan (*artificial intelligence*) merupakan cabang dari ilmu komputer yang dalam merepresentasi pengetahuan lebih banyak menggunakan bentuk simbol-simbol daripada bilangan, dan memproses informasi berdasarkan metode heuristic atau dengan berdasarkan sejumlah aturan"

Artificial Intelligence dapat ditinjau dari dua pendekatan :

1. Pendekatan Ilmiah (*A Scientific Approach*)

   Pendekatan dasar ilmiah timbul sebelum invansi ke komputer, ini tidak sama dengan kasus mesin uap. Pendekatan ilmiah melihat batas sementara dari komputer, dan dapat diatasi dengan perkembangan teknologi lanjutan. Mereka tidak mengakibatkan tingkatan pada konsep.

2. Pendekatan Teknik (*An Engineering Approach*)

   Usaha untuk menghindari definisi AI, tetapi ingin mengatasi atau memecahkan persoalan-persoalan dunia nyata (*real world problem*).

2.1.3 Tujuan *Artificial Intelligence*

Menurut Lenat dan Feigenbaum (1992), terdapat sembilan tujuan *Artificial Intelligence*, yaitu :

1. Memahami kognisi manusia

   Mencoba untuk mendapatkan pengetahuan ingatan manusia yang mendalam, kemampuan problem solving, belajar, membuat keputusan, dll.

2. Otomatisasi biaya-efektif

   Menggantikan manusia dalam tugas-tugas intelegensi, mempunyai program yang performa-nya sebaik manusia dalam mengerjakan pekerjaan.

3. Penguatan intelegensi biaya-efektif

   Membangun sistem untuk membantu manusia berpikir lebih baik, lebih cepat, lebih dalam, dan lain-lain. Contoh : sistem untuk membantu diagnosa penyakit.

4. Intelegensi manusia super

   Membangun program yang mempunyai kemampuan untuk melebihi intelegensi manusia.

5. Problem-solving umum

   Sistem penyelesaian berbagai masalah yang luas, sistem ini mempunyai kelebaran pikiran.

6. Wacana koheren

   Komunikasi dengan manusia menggunakan bahasa alami, contoh : dialog cerdas yang ada dalam *Turing Test*.

7. Belajar (induksi)

   Sistem sebaiknya dapat untuk memperoleh data sendiri dan tahu bagaimana memperolehnya, sistem dapat menyamaratakan, membuat hipotesis, penerapan atau pembelajaran secara heuristik, membuat alasan dengan analogi.

8. Otonomi

   Mempunyai sistem intelegensi yang beraksi atas inisiatif sendiri dan harus bereaksi dengan dunia nyata.

9. Informasi

   Simpan informasi dan mengetahui cara untuk mengambil informasi

2.2 Pengertian Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggeraknya yaitu rakyat yang berada dinegara tersebut.

Kata "sistem" banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.

2.2.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu:

1. Komponen (Elemen)

   Komponen dari suatu sistem dikenal sebagai subsistem.

2. Batasan (*Boundary*)

   Daerah yang membatasi antara sistem yang satu dengan yang lainnya atau dengan lingkungan luar.

3. Lingkungan luar sistem (*environment*)

   Segala sesuatu di luar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Contoh : Vendor, Pelanggan, Pemilik, Pemerintah, Bank, Pesaing, dan lain-lain.

4. Penghubung Sistem (*interface*)

   Suatu media penghubung antara 1 subsistem dengan subsistem lainnya.

5. Masukan (*Input*)

   Energi yang dimasukkan ke dalam sistem. Pada sistem informasi, masukan dapat berupa : Data transaksi, Data non transaksi (contoh : surat pemberitahuan), dan Instruksi

6. Keluaran (*Output*)

   - Hasil dari pemrosesan, dapat berupa keluaran yang berguna (informasi, produk) atau keluaran yang tidak berguna (limbah).
   - Pada sistem informasi, keluaran dapat berupa : Informasi, Saran, dan Cetakan laporan

7. Sasaran sistem (*objective*)

   Suatu tujuan yang ingin dicapai oleh suatu sistem.

2.3 Pengertian Pakar

Pakar atau ahli ialah seseorang yang banyak dianggap sebagai sumber terpercaya atas teknik maupun keahlian tertentu yang bakatnya untuk menilai dan memutuskan sesuatu dengan benar, baik, maupun adal sesuai dengan aturan dan status oleh sesamanya ataupun khayalak dalam bidang khusus tertentu. Lebih umumnya, seorang pakar ialah seseorang yang memiliki pengetahuan ataupun kemampuan luas dalam bidang studi tertentu. Para pakar dimintai nasihat dalam bidang terkait mereka, namun mereka tidak selalu setuju dalam kekhususan bidang studi. Melalui pelatihan, pendidikan, profesi, publikasi, maupun pengalaman, seoran pakar dipercaya memiliki pengetahuan khusus dalam bidangnya di atas rata-rata orang, di mana orang lain bisa secara resmi (dan sah) mengandalkan pendapat pribadi.

2.4 Sistem Pakar

2.4.1 Pengertian Sistem Pakar

Sistem pakar adalah suatu program komputer yang mengandung pengetahuan dari satu atau lebih pakar manusia mengenai suatu bidang spesifik. Jenis program ini pertama kali dikembangkan oleh periset kecerdasan buatan pada dasawarsa 1960-an dan 1970-an dan diterapkan secara komersial selama 1980-an.

Bentuk umum sistem pakar adalah suatu program yang dibuat berdasarkan suatu set aturan yang menganalisis informasi (biasanya diberikan oleh pengguna suatu sistem) mengenai suatu kelas masalah spesifik serta analisis matematis dari masalah tersebut. Tergantung dari desainnya, sistem pakar juga mampu merekomendasikan suatu rangkaian tindakan pengguna untuk dapat menerapkan koreksi. Sistem ini memanfaatkan kapabilitas penalaran untuk mencapai suatu simpulan.

2.4.2 Konsep Dasar Sistem Pakar

Menurut Efraim Turban (1995) konsep dasar sistem pakar mengandung beberapa unsur atau elemen, yaitu keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan. Keahlian merupakan suatu penguasaan pengetahuan di bidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Contoh bentuk pengetahuan yang termasuk keahlian:

1. Fakta-fakta pada lingkup permasalahan tertentu.
2. Teori-teori pada lingkup permasalahan tertentu.
3. Prosedur atau aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
4. Strategi-strategi global untuk menyelesaikan masalah.
5. *Meta–knowledge* (pengetahuan tentang pengetahuan).

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: tambahan pengetahuan (dari para ahli atau

sumber-sumber lainnya), representasi pengetahuan (ke komputer), inferensi pengetahuan, dan pengalihan pengetahuan ke user, pengetahuan yang disimpan di komputer disebut basis pengetahuan yaitu: fakta dan 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 basis data, maka komputer harus dapat diprogram untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk motor inferensi (*inference engine*).

2.4.3 Tujuan Sistem Pakar

Tujuan dari sebuah Sistem Pakar adalah mentransfer kepakaran yang dimiliki seorang pakar kedalam komputer dan kemudian kepada orang lain (non *expert*). Aktivitas pemindahan kepakaran adalah :

1. *Knowledge Acquisition* (dari pakar atau sumber lain) yaitu kegiatan mencari dan mengumpulkan pengetahuan dari para ahli atau sumber keahlian yang lain.
2. *Knowledge Representation* (ke dalam komputer) adalah kegiatan menyimpan mengatur penyimpanan pengetahuan yang diperoleh dalam komputer. Pengetahuan berupa fakta dan aturan disimpan dalam komputer sebagai sebuah komponen yang disebut basis pengetahuan.

3. *Knowledge Inferencing* adalah kegiatan melakukan inferensi berdasarkan pengetahuan yang telah disimpan didalam komputer.
4. *Knowledge Transfering* adalah kegiatan pemindahan pengetahuan dari komputer ke pemakai yang tidak ahli.

2.4.4 Keuntungan Sistem Pakar

Mengapa Sistem Pakar menjadi sangat populer? Hal ini disebabkan oleh sangat banyaknya kemampuan dan manfaat yang diberikan oleh Sistem Pakar, di antaranya:

1. Meningkatkan output dan produktivitas, karena Sistem Pakar dapat bekerja lebih cepat dari manusia.
2. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
3. Mampu menangkap kepakaran yang sangat terbatas.
4. Dapat beroperasi di lingkungan yang berbahaya.
5. Memudahkan akses ke pengetahuan.
6. Handal. Sistem Pakar tidak pernah menjadi bosan dan kelelahan atau sakit. Sistem Pakar juga secara konsisten melihat semua detil dan tidak akan melewatkan informasi yang relevan dan solusi yang potensial.
7. Meningkatkan kapabilitas sistem terkomputerisasi yang lain. Integrasi Sistem Pakar dengan sistem komputer lain membuat lebih efektif, dan mencakup lebih banyak aplikasi .

8. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional, Sistem Pakar dapat bekerja dengan inofrmasi yang tidak lengkap. Pengguna dapat merespon dengan: "tidak tahu" atau "tidak yakin" pada satu atau lebih pertanyaan selama konsultasi, dan Sistem Pakar tetap akan memberikan jawabannya.
9. Mampu menyediakan pelatihan. Pengguna pemula yang bekerja dengan Sistem Pakar akan menjadi lebih berpengalaman. Fasilitas penjelas dapat berfungsi sebagai guru.
10. Meningkatkan kemampuan *problem solving*, karena mengambil sumber pengetahuan dari banyak pakar.
11. Meniadakan kebutuhan perangkat yang mahal.
12. Fleksibel.

2.4.5 Kelemahan Sistem Pakar

Metodologi Sistem Pakar yang ada tidak selalu mudah, sederhana dan efektif. Berikut adalah keterbatasan yang menghambat perkembangan Sistem Pakar:

1. Pengetahuan yang hendak diambil tidak selalu tersedia.
2. Kepakaran sangat sulit diekstrak dari manusia.
3. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.

4. Adalah sangat sulit bagi seorang pakar untuk mengabstraksi atau menjelaskan langkah mereka dalam menangani masalah
5. Pengguna Sistem Pakar mempunyai batas kognitif alami, sehingga mungkin tidak bisa memanfaatkan sistem secara maksimal.
6. Sistem Pakar bekerja baik untuk suatu bidang yang sempit.
7. Banyak pakar yang tidak mempunyai jalan untuk mencek apakah kesimpulan mereka benar dan masuk akal.
8. Istilah dan jargon yang dipakai oleh pakar dalam mengekspresikan fakta seringkali terbatas dan tidak mudah dimengerti oleh orang lain.
9. Pengembangan Sistem Pakar seringkali membutuhkan perekayasa pengetahuan (*knowledge engineer*) yang langka dan mahal.
10. Kurangnya rasa percaya pengguna menghalangi pemakaian Sistem Pakar.
11. Transfer pengetahuan dapat bersifat subyektif dan bias.

2.4.6 Elemen manusia yang terkait dalam penggunaan dan pengembangan Sistem Pakar

1. Pakar

Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.

2. Perekayasa pengetahuan

Perekayasa pengetahuan adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan *counter example* dan menerangkan kesulitan-kesulitan konseptual.

3. Pemakai

a. Pemakai awam merupakan sistem pakar bertindak sebagai konsultan untuk memberikan saran dan solusi kepada pemakai.
b. Pelajar yang ingin belajar merupakan sistem pakar bertindak sebagai instruktur.
c. Pembuat sistem pakar merupakan sistem pakar sebagai partner dalam pengembangan basis pengetahuan.
d. Pakar merupakan sistem pakar bertindak sebagai mitra kerja atau asisten

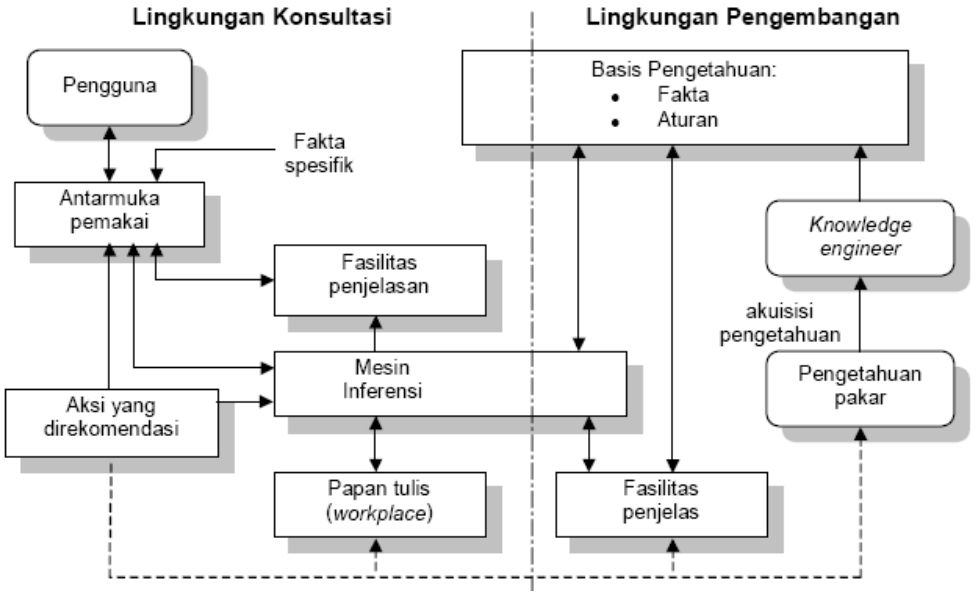
2.4.7 Skema Struktur Dan Komponen Sistem Pakar

2.4.7.1 Skema Struktur

Sistem pakar terdiri dari dua bagian pokok, yaitu lingkungan pengembangan (*development environment)* dan lingkungan konsultasi (*consultation environment*).

Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan

konsultasi digunakan oleh seorang yang bukan ahli untuk berkonsultasi (lihat Gambar 2.1 Struktur Sistem Pakar).



Gambar 2.1 Struktur Sistem Pakar

2.4.7.2 Komponen Sistem Pakar

1. **Basis Pengetahuan,** berisi pengetahuan yang dibutuhkan untuk memahami, memformulasi, dan memecahkan masalah. Basis pengetahuan tersusun atas 2 elemen dasar:
   a. Fakta, misalnya: situasi, kondisi, dan kenyataan dari permasalahan yang ada, serta teori dalam bidang itu
   b. Aturan, yang mengarahkan penggunaan pengetahuan untuk memecahkan masalah yang spesifik dalam bidang yang khusus Mesin Inferensi (*Inference Engine*), merupakan otak dari Sistem Pakar. Juga dikenal sebagai penerjemah aturan (*rule interpreter*). Komponen ini berupa

program komputer yang menyediakan suatu metodologi untuk memikirkan (*reasoning*) dan memformulasi kesimpulan.

2. **Mesin inferensi,** program yang berisi metodologi yang diguanakan untuk melakukan penalaran terhadap informasi di dalam basis pengetahuan dan *blackboard*, serta digunakan untuk memformulasikan konklusi. Terdapat tiga elemen utama, yaitu:
   a. *Interpreter*: mengeksekusi item item yang terpilih
   b. *Scheduler*: mengontrol agenda
   c. *Consistensi Enforcer*: memelihara konsistensi dalam merepresentasikan solusi yang bersifat darurat

   Yang meliputi:

   a. Menentukan aturan mana akan dipakai
   b. Menyajikan pertanyaan kepada pemakai, ketika diperlukan.
   c. Menambahkan jawaban ke dalam memori Sistem Pakar.
   d. Menyimpulkan fakta baru dari sebuah aturan
   e. Menambahkan fakta tadi ke dalam memori.

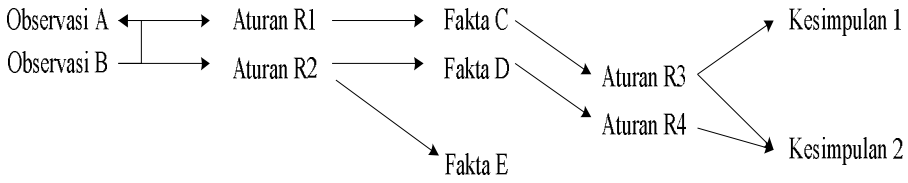
   Ada 2 macam tipe metode atau teknik inferensi, yaitu :

a. Forward Chaining (Pelacakan ke Depan)

Pendekatan yang dimotori oleh data (data *driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan dan selanjutnya mencoba menggambarkan kesimpulan. Aturan dalam sistem merepresentasikan aksi-aksi

yang harus diambil apabila terdapat suatu kondisi khusus pada item-item dalam memori kerja yang disebut himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil. Aktivitas sistem dilakukan berdasarkan siklus mengenal-beraksi (*recognize-act*).

Pelacakan ke depan mencari fakta yang sesuai dari bagian IF dari aturan IF-THEN (Gambar 2.2 menunjukkan proses *forward chaining*).



Gambar 2.2 Proses *Forward Chaining*

b. Backward Chaining

Runut balik atau *backward chaining* merupakan strategi pencarian yang arahnya kebalikan dari runut maju. Proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang kesimpulannya merupakan solusi yang ingin dicapai, kemudian dari kaidah-kaidah yang diperoleh, masing-masing kesimpulan dirunut balik jalur yang mengarah ke kesimpulan tersebut. Jika informasi-informasi atau nilai dari atribut-atribut yang mengarah ke kesimpulan tersebut sesuai dengan data yang diberikan maka kesimpulan tersebut merupakan solusi yang dicari, jika tidak sesuai maka kesimpulan tersebut bukan

merupakan solusi yang dicari. Runut balik memulai proses pencarian dengan suatu tujuan sehingga strategi ini disebut juga *goal-driven*.

Algoritma Runut Balik

Penggunaan strategi pencarian runut balik untuk membangun mesin inferensi memerlukan suatu algoritma tertentu sehingga bisa diimplementasikan menggunakan bahasa pemrograman. Salah satu algoritma runut balik yang diambil dari buku *Introduction To Expert Systems : The Development and Implementation Of Rule-Based Expert Systems* (Ignizio, 1991) adalah sebagai berikut :

1. Inisialisasi

a) Buat 3 tabel yaitu :

1) Tabel Working Memory, untuk menyimpan pasangan atribut-nilai hasil dari proses inferensi.

2) Tabel Goal, untuk menyimpan atribut yang nilainya sedang dicari.

3) Tabel Rule/Premise Status, untuk menyimpan nomor kaidah, status dari kaidah yang bersesuaian, pasangan atribut-nilai dari klausa premis kaidah-kaidah dalam basis pengetahuan, nomor klausa premis dalam kaidah yang bersesuaian, dan status dari klausa premis tersebut.

b) Semua klausa premis dalam tabel Rule/Premise Status diberi status *free*.

c) Semua kaidah dalam tabel Rule/Premise Status diberi status *active*.

2. Mulai inferensi

a) Sebutkan kesimpulan akhir (solusi dari permasalahan) yang ingin dicapai.

b) Letakkan atribut dari klausa kesimpulan yang merupakan solusi dari permasalahan pada puncak tabel Goal.

3. Pengecekan kaidah

a) Jika tabel Goal kosong maka STOP.

b) Jika tabel Goal tidak kosong maka cari kaidah-kaidah berstatus *active* yang atribut dari klausa kesimpulannya bersesuaian dengan atribut yang berada pada puncak tabel Goal.

1) Jika hanya 1 kaidah yang ditemukan, lakukan langkah 6. Jika ada beberapa kaidah yang ditemukan, cari yang berstatus *triggered*, lakukan langkah 6. Jika tidak ada kaidah yang berstatus *triggered*, pilih salah satu kaidah lalu proses dengan langkah 6.

2) Jika tidak ada kaidah yang ditemukan, lakukan langkah 4.

4. Query

Ambil salah satu data dari sekumpulan data yang diberikan oleh pemakai yang bersesuaian dengan atribut yang berada pada puncak tabel Goal.

a) Jika tidak ada maka STOP.

b) Jika ada maka pindahkan atribut yang berada pada puncak tabel Goal lalu tempatkan pada tabel Working Memory beserta nilainya, yaitu data yang diambil tadi.

5. Pembaharuan status kaidah/premis

a) Gunakan isi dari tabel Working Memory untuk membaharui tabel Rule/Premise Status.

b) Jika ada klausa premis yang berstatus *false* pada suatu kaidah maka beri status *discard* pada kaidah tersebut. Tetapi jika seluruh klausa premis pada suatu kaidah semuanya berstatus *true* maka beri status *triggered* pada kaidah tersebut.

c) Kembali ke langkah 3.

6. Evaluasi kaidah

a) Jika kaidah berstatus *triggered* maka pindahkan atribut yang berada pada puncak tabel Goal ke tabel Working Memory beserta nilainya yang

terdapat dalam klausa kesimpulan pada kaidah tersebut. Ubah status kaidah tersebut menjadi *fired*. Kembali ke langkah 5.

b) Jika kaidah tidak berstatus *triggered* maka pilih atribut dari klausa premis pertama yang berstatus *free* lalu tempatkan pada puncak tabel Goal. Kembali ke langkah 3.

Algoritma runut balik di atas mempunyai 2 keadaan yang menyebabkan proses inferensi berhenti, yaitu :

1. Pada saat tabel Goal kosong, berarti kesimpulan yang merupakan solusi dari permasalahan sudah diperoleh.
2. Pada saat data dari pemakai yang diminta oleh sistem pakar untuk memenuhi prompt kaidah tidak ada, berarti kesimpulan yang merupakan solusi dari permasalahan tidak ditemukan.

3. **Antarmuka Pemakai (*User Interface*)**, Sistem Pakar mengatur komunikasi antara pengguna dan komputer. Komunikasi ini paling baik berupa bahasa alami, biasanya disajikan dalam bentuk tanya jawab dan kadang ditampilkan dalam bentuk gambar/grafik. Antarmuka yang lebih canggih dilengkapi dengan percakapan (*voice communication*).
4. **Akuisisi Pengetahuan,** Dalam proses akuisisi pengetahuan, seorang perekayasa pengetahuan menjembatani antara pakar dengan basis pengetahuan. Perekayasa pengetahuan mendapatkan pengetahuan dari pakar, mengolahnya bersama pakar tersebut, dan menaruhnya dalam basis

pengetahuan, dengan format tertentu. Pengambilan pengetahuan dari pakar dapat dilakukan secara :

a. Manual, di mana perekayasa pengetahuan mendapatkan pengetahuan dari pakar (melalui wawancara) dan/atau sumber lain, kemudian mengkodekannya dalam basis pengetahuan. Proses ini biasanya berlangsung lambat, mahal, serta kadangkala tidak akurat.
b. Semi-otomatik, di mana terdapat peran komputer untuk: (1) mendukung pakar dengan mengijinkannya membangun basis pengetahuan tanpa (atau dengan sedikit) bantuan dari perekayasa pengetahuan, atau (2) membantu perekayasa pengetahuan sehingga kerjanya menjadi lebih efisien dan efektif.
c. Otomatik, di mana peran pakar, perekayasa pengetahuan, dan pembangun basis pengetahuan (*system builder*) digabung. Misalnya dapat dilakukan oleh seorang system analyst seperti pada metode induksi.

5. **Papan Tulis (*Blackboard/Workplace*)**, adalah memori/lokasi untuk bekerja dan menyimpan hasil sementara. Biasanya berupa sebuah basis data. Ada tiga keputusan yang dapat direkam, yaitu:
   a. Rencana : bagaimana menghadapi masalah
   b. Agenda : aksi-aksi potensial yang sedang menunggu untuk di eksekusi
   c. Solusi : calon aksi yang akan dibangkitkan
6. **Subsistem Penjelasan (*Explanation Facility*)**, Kemampuan untuk menjejak (*tracing*) bagaimana suatu kesimpulan dapat diambil merupakan hal yang sangat penting untuk transfer pengetahuan dan pemecahan masalah.

Komponen subsistem penjelasan harus dapat menyediakannya yang secara interaktif menjawab pertanyaan pengguna, misalnya:

a. “Mengapa pertanyaan tersebut anda tanyakan?”
b. “Seberapa yakin kesimpulan tersebut diambil?”
c. “Mengapa alternatif tersebut ditolak?”
d. “Apa yang akan dilakukan untuk mengambil suatu kesimpulan?”
e. “Fakta apalagi yang diperlukan untuk mengambil kesimpulan akhir?”

7. **Sistem Perbaikan Pengetahuan,** Sistem ini digunakan untuk mengevaluasi kinerja sistem pakar itu sendiri untuk melihat apakah pengetahuan pengetahuan yang ada masih cocok untuk digunaka di masa mendatang.

2.5 Basis Data

2.5.1 Pengertian basis data

Basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya

sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili denga menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Ada beberapa definisi basis data,diantaranya sebagai berikut:

1. Himpunan Kelompok Data (Arsip) yang saling berhubungan dan diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (Redundensi) yang tidak perlu.

3. Kumpulan File/Tabel/Arsip yang saling berhubungan yang disimpan daam media penyimpan Elektronik.

2.5.2 Tujuan Basis Data

1. Kemudahan dan kecepatan dalam pengambilan data (*speed*)
2. Efisiensi ruang penyimpanan (*space*)

   Mengurangi / menghilangkan redudansi data
3. Keakuratan (*Accuracy*)

   Pembentukan kode & relasi antar data berdasar aturan / batasan (*constraint*) tipe data, domain data, keunikan data, untuk menekan ketidakakuratan saat penyimpanan data.
4. Ketersediaan (*Avaibility*)

   Pemilahan data yang sifatnya pasif dari database aktif.
5. Kelengkapan (*Completeness*)

   Kompleksnya data menyebabkan perubahan struktur database.
6. Keamanan (*Security*)

   Memberikan keamanan atas hak akses data.
7. Kebersamaan pemakaian (*Sharability*)

   Bersifat multiuser.

2.5.3 Keuntungan Basis Data

1. Kebebasan data dan akses yang efisien
2. Mereduksi waktu pengembangan aplikasi
3. Integritas dan keamanan data
4. Administrasi keseragaman data
5. Akses bersamaan dan perbaikan dari terjadinya crashes (tabrakan dari proses serentak).

2.6 Metode Analisis Yang Digunakan

2.6.1 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

2.6.2 DFD (Data Flow Diagram)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

DFD memiliki beberapa komponen. Menurut Yourdan dan DeMarco komponen DFD digambarkan sebagai berikut:

1. Terminator

Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (*external entity*).

Terdapat dua jenis terminator :

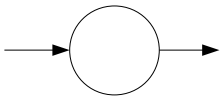
a. Terminator Sumber (*source*) : merupakan terminator yang menjadi sumber.
b. Terminator Tujuan (*sink*) : merupakan terminator yang menjadi tujuan data / informasi sistem.

Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.

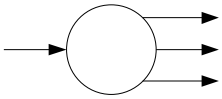
2. Proses

Proses diberi nama untuk menjelaskan proses/kegiatan apa yang sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan obyek).

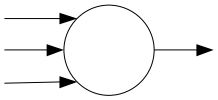
Ada empat kemungkinan yang dapat terjadi dalam proses sehubungan dengan input dan output :



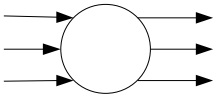
1 input dan 1 output



1 input dan banyak output



Banyak input dan 1 output



banyak input dan banyak output

Gambar 2.3 Proses *input output* DFD

3. Data Store

Komponen ini digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda jamak, misalnya Mahasiswa.

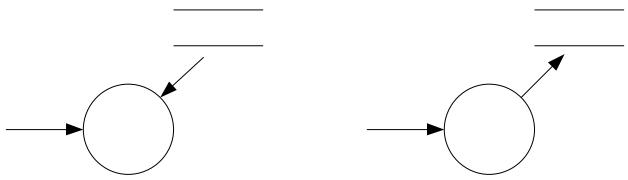
Data store ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file *harddisk*, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda.

Suatu data store dihubungkan dengan alur data hanya pada komponen proses, tidak dengan komponen DFD lainnya. Alur data yang menghubungkan data store dengan suatu proses mempunyai pengertian sebagai berikut :

a. Alur data dari data store yang berarti sebagai pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, sebagian dari satu paket

tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses (*lihat gambar 2.8 (a)*).

b. Alur data ke data store yang berarti sebagai pengupdatean data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/memodifikasi satu paket data atau lebih (*lihat gambar 2.8 (b)*).



(a) **(b)**

Gambar 2.4 Alur Data Store

4. Data Flow / Alur Data

Suatu data flow/alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya.

Selain menunjukkan arah, alur data pada model yang dibuat oleh profesional sistem dapat merepresentasikan bit, karakter, pesan, formulir, bilangan real, dan macam-macam informasi yang berkaitan dengan komputer. Alur data juga dapat merepresentasikan data/informasi yang tidak berkaitan dengan komputer.

2.6.3 Kamus data (*Data Dictionary)*

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga user dan analis sistem mempunyai pengertian yang sama tentang *input, output,* dan komponen data strore.

Kamus data ini sangat membantu analis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem.

Pada tahap analisis, kamus data merupakan alat komunikasi antara user dan analis sistem tentang data yang mengalir di dalam sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh user. Sementara itu, pada tahap perancangan sistem kamus data digunakan untuk merancang input, laporan dan database.

Pembentukan kamus data didasarkan atas alur data yang terdapat pada DFD. Alur data pada DFD ini bersifat global, dalam arti hanya menunjukan nama alur datanya tanpa menunjukan struktur dari alur data itu. Untuk menunjukan struktur dari alur data secara terinci maka dibentuklah kamus data yang didasarkan pada alur data di dalam DFD.

2.6.4 ERD (*Entity Relationship Diagram*)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi**.**

ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu :

1. Entiti

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999: 30). Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

2. Atribut

Setiap entitas pasti mempunyai elemen yang disebut *atribut* yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain.

3. Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

a. Satu ke satu (*One to one*)

   Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

b. Satu ke banyak (*One to many*)

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

c. Banyak ke banyak (*Many to many*)

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

2.7 Teori Tentang Sepeda Motor dan Kerusakannya

Sepeda motor adalah sebuah mesin yang terbuat dari ribuan komponen. Secara umum, pengguna offroad dengan sepeda motor tentu berharap bisa mengendarainya untuk sampai ke tujuan dan tidak mengalami masalah pada saat melalui beberapa *handicap* yang tidak jarang mampu membuat sepeda motor yang dikendarai mengalami permasalahan. Untuk menghindari potensi masalah yang mungkin terjadi, maka offroader sepeda motor harus memiliki kemampuan

penanganan kerusakan yang walaupun sifatnya sementara, tetapi tetap dapat membantu membawa keluar kendaraan dari areal offroad ke areal terbuka, pedesaan atau bengkel terdekat.

Berikut ini tips dan trik tentang permasalahan kerusakan sepeda motor :

1. Pada umumnya setiap kerusakan pasti akan terdapat tanda-tanda terlebih dahulu kecuali jika terjadi hal-hal yang menyimpang misalnya kecelakaan. Untuk mengantisipasinya maka perhatikan apabila ada gejala yang tidak normal / tidak seperti biasanya pada sepeda motor Anda. Sikap demikian akan membantu dan memudahkan Anda untuk mendeteksi kerusakan lebih dini.
2. Apabila terjadi kerusakan mesin maka perbaikan tidak boleh ditunda lebih lama dengan kata lain harus segera diperbaiki. Namun ingat, jika Anda tidak punya cukup keahlian jangan sekali-sekali membongkar dan memperbaikinya seorang diri. Karena disamping buang waktu dan tenaga maka kerusakan bisa jadi akan tambah parah.

Didalam dunia sepeda motor dikenal ada 3 (tiga) jenis mesin yang digunakan yaitu mesin 2 TAK, 4 TAK dan battere. Umumnya, kegiatan offroad sepeda motor saat ini dilakukan dengan menggunakan kendaraan sepeda motor dari jenis 2 TAK dan 4 TAK. Apa perbedaannya?

Secara harfiah, sebenarnya yang disebut dengan TAK adalah langkah atau dalam bahasa Inggrisnya disebut dengan *stroke*. Dengan kata lain, 2 TAK adalah

mesin 2 langkah, sementara mesin 4 TAK adalah mesin 4 langkah. Kembali kepada langkah tersebut, maka langkah disini merupakan proses. Untuk memudahkan pengertian terhadap hal tersebut, maka dapat dijelaskan bahwa proses yang terjadi pada mesin 4 langkah adalah sebagai berikut: *INTAKE – COMPRESSION – POWER – EXHAUST*.

2.8 XAMPP

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, Perl. Xampp merupakan alat yang menyediakan paket perangkat lunak ke dalam sebuah paket. Dalam paketnya sudah terdapat Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FTP server, phpMyAdmin dan berbagai pustaka bantu lainnya. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server apache, PHP, dan MySQL secara manual. XAMPP akan menginstall dan mengkonfigurasikannya secara otomatis untuk anda. XAMPP tersedia untuk Linux, Windows, Mac OS X maupun Solaris sehingga sangat memudahkan membuat web server multiplatform. Selain itu XAMPP adalah seratus persen *open source*, tersedia bebas dan legal.

2.9 Apache Web Server

Apache web server adalah sebuah perangkat lunak server web yang dapat djalankan pada banyak sistem operasi (Unix, BSD, Linux, Microsoft windows, dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik yang memungkinkan penanganan server menjadi mudah. Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang di bawah naungan *Apache Software Foundation*.

2.10 MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis. *MySQL* merupakan *database open source* yang saat ini cukup banyak digunakan pada berbagai aplikasi. Keandalanya dalam mengolah *database* ditunjang kecepatannya dalam mengakses perintah *query* serta banyaknya fitur-fitur yang dimilki menjadikannya sebagai database idola saat ini.

Perintah untuk mengelola *database* dibagi menjadi 3 (tiga) kelompok, diantaranya :

1. Perintah untuk mendefinisikan data/DDL (*Data Definition Language*).
2. Perintah untuk memanipulasi data/DML (*Data Manipulation Language*).
3. Perintah untuk mengendalikan data/DCL (*Data Control Language*).

2.11 PHP

PHP adalah singkatan dari (*Hypertext Preprocessor*), sebuah bahasa pemrograman yang lebih menitik beratkan kepada aplikasi web. PHP merupakan bahasa pemrogramman berbasis web yang memiliki kemampuan untuk memproses data dinamis.

PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web *browser*, tetapi prosesnya secara keseluruhan dijalankan di server.

Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki. PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain-lain.

Kelebihan PHP dari pemrograman lain adalah:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, nginx, 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 pemahamanan, 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.

2.12 PhpMyAdmin

Pengelolaan database dengan MYSQL harus dilakukan dengan mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika ingin membuat database, ketikkan baris perintah yang sesuai untuk membuat database. Jika ingin menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu cukup menyulitkan karena kita harus hafal dan mengetikkan perintahnya satu persatu.

Banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola data base dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin kita dapat membuat tabel, mengisi data dan lain-lain dengan mudah tanpa harus hafal perintahnya. Untuk mengaktifkan phpMyAdmin langkah-langkahnya adalah : yang pertama setelah XAMPP kita terinstall, kita harus mengaktifkan web server Apache dan MySQL dari control panel XAMPP. Yang kedua, jalankan browser (IE, Mozilla Firefox atau Opera) lalu mengetikkan alamat web berikut : http://localhost/phpmyadmin/ pada address bar lalu tekan

Enter. Langkah ketiga apabila telah nampak interface (tampilan antar muka) phpMyAdmin, kita bisa memulainya dengan mengetikkan nama database, nama tabel dan seterusnya.

2.13 Perl

PERL adalah bahasa pemrograman yang menggunakan tipe data dinamis, program PERL dapat langsung dieksekusi tanpa harus melalui proses kompilasi tersendiri ke format binaryyang dapat dieksekusi.

PERL banyak digunakan pada aplikasi manajemen sistem dan jaringan, pemrograman web, manipulasi teks, akses ke database. Berikut ini beberapa aplikasi yang menggunakan PERL yang cukup dikenal: mrtg, SQL Ledger (*aplikasi akunting*), Webmin (aplikasi admin), spam assasin (anti spam).

2.14 Macromedia Dreamwaever

Macromedia Dreamweaver adalah sebuah HTML editor profesional untuk mendesain secara visual dan mengelola situs web maupun halaman web. Bilamana kita menyukai untuk berurusan dengan kode-kode HTML secara manual atau lebih menyukai bekerja dengan lingkungan secara visual dalam melakukan editing, Dreamweaver mambuatnya menjadi lebih mudah dengan menyediakan tool-tool yang sangat berguna dalam peningkatan kemampuan dan pengalaman kita dalam mendesain web.

Dreamweaver MX dalam hal ini digunakan untuk web desain. Dreamweaver MX mengikutsertakan banyak tool untuk kode-kode dalam

halaman web beserta fasilitas-fasilitasnya, antara lain : Referensi HTML, CSS dan Javascript, Javascript debugger, dan editor kode (tampilan kode dan Code inspector) yang mengizinkan kita mengedit kode Javascript, XML, dan dokumen teks lain secara langsung dalam Dreamweaver. Teknologi Dreamweaver Roundtrip HTML mampu mengimpor dokumen HTML tanpa perlu memformat ulang kode tersebut dan kita dapat menggunakan Dreamweaver pula untuk membersihkan dan memformat ulang HTML bila kita menginginkannya.

Selain itu Dreamweaver juga dilengkapi kemampuan manajemen situs, yang memudahkan kita mengelola keseluruhan elemen yang ada dalam situs. Kita juga dapat melakukan evaluasi situs dengan melakukan pengecekan broken link, kompatibilitas browser, maupun perkiraan waktu download halaman web.