BAB 2

TINJAUAN PUSTAKA

2.1 Profil Balai Kesehatan Olahraga Masyarakat Bandung

Balai Kesehatan Olahraga Masyarakat Bandung merupakan unit pelaksanaan teknis di bidang kesehatan olahraga masyarakat di lingkungan Direktorat Jenderal Bina Kesehatan Masyarakat Departemen Kesehatan Repbulik Indonesia. Balai Kesehatan Olahraga Masyarakat (BKOM) Bandung berpusat di Jl. Merak No. 13 Telp : (022) – 2506207.

BKOM mempunyai tugas melaksanakan pelayanan dan fasilitas, pelatihan, penelitian, dan peningkatan kemitraan serta sosialisasi di bidang kesehatan olahraga. Dalam melaksanakan tugas sebagaimana dimaksud, BKOM menyelenggarakan fungsi :

a. Penyusunan rencana dan program pelayanan kesehatan olahraga masyarakat.
b. Pelayanan kesehatan olahraga masyarakat secara paripurna.
c. Pelatihan kesehatan olahraga masyarakat.
d. Penelitian dan pengembangan kesehatan olahraga masyarakat.
e. Pelaksaaan kemitraan dan sosialisasi kesehatan olahraga.
f. Evaluasi dan penyusunan laporan
g. Pelaksanaan urusan ketatausahaan dan kerumahtanggaan BKOM.

2.1.1 Visi Misi BKOM Bandung

BKOM Bandung mempunyai visi yaitu “memberikan pelayanan kesehatan olahraga secara paripurna dan bermutu dalam rangka meningkatkan derajat kesehatan masyarakat pada tahun 2015 “

Misi BKOM Bandung dalam mencapai visi adalah :

1. Memberikan pelayanan dan pusat rujukan kesehatan olahraga secara paripurna (promotif, preventif, kuratif, dan rehabilitatif) profesional dan terjangkau bagi seluruh lapisan masyarakat untuk mencapai Indonesia sehat tahun 2025.
2. Membuat masyarakat sehat dan bugar melalui pembudayaan aktivitas fisik/ latihan fisik dan atau olahraga secara baik, benar, terukur dan teratur.
3. Pelayanan yang diberikan mengutamakan kepuasan pelanggan baik internal, intermediate maupun eksternal.
4. Menjalin kemitraan dengan berbagai pihak yang saling menguntungkan dalam berbagai aspek yaitu sumber daya manusia, rujukan pelayanan dan rujukan keilmuan. Sebagai lahan penelitian, pelatihan dan pendidikan kesehatan olahraga dalam upaya pengembangan ilmu serta teknologi.
5. Mengembangkan sistem informasi manajamen kesehatan olahraga yang akurat dan valid.

2.1.2 Logo BKOM Bandung

Logo merupakan suatu bentuk gambar atau sketsa dengan arti tertentu dan mewakili suatu arti dari perusahaan, daerah, perkumpulan, produk, Negara dan hal-hal lainnya yang dianggap membutuhkan hal yang singkat dan mudah diingat sebagai pengganti dari nama sebenarnya.



Gambar 2.1 Logo BKOM Bandung

Logo BKOM Bandung mencerminkan beberapa hal, sebagai berikut :

1. Warna Dasar Biru

   Melambangkan keluhuran jiwa yang sehat, tenang, sabar, tangguh, serta berkepribadian yang luhur dari masyarakat Jabar khususnya dan masyarakat Indonesia umumnya, melalui pembinaan BKOM Jabar.

2. Warna Kuning

   Melambangkan pribadi karyawan BKOM Jabar yang selalu siap sedia memberikan pelayanan yang bermutu kepada masyarakat.

3. Bola Dunia

   Melambangkan pelayanan yang diberikan oleh BKOM Jabar ditujukan kepada seluruh lapisan masyarakat baik yang sehat maupun yang sakit.

4. Orang berlari

   Melambangkan masyarakat baik laki-laki maupun perempuan agar menjadi dengan melakukan olahraga teratur, terprogram, murah dan dapat dilakukan dimana saja dengan pengawasan BKOM Jabar.

5. Lingkaran merah berjumlah 5 buah saling berkaitan

   Melambangkan bahwa dengan olahraga dapat menggalang persahabatan diantara masyarakat melalui pembinaan secara berkesinambungan oleh BKOM Jabar.

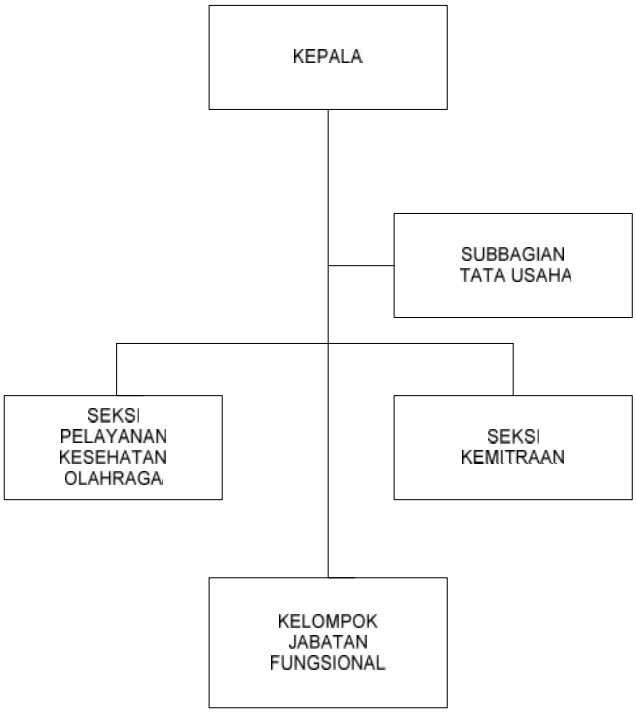
2.1.3 Badan Hukum Instansi

Keputusan Menteri Kesehatan Republik Indonesia Nomor : 127/MENKES/SK/II/2004 tentang Organisasi dan Tata Kerja Balai Kesehatan Olahraga Masyarakat Bandung.

2.1.4 Struktur Organisasi

Struktur organisasi yang ada di Balai Kesehatan Olahraga (BKOM) Bandung terdiri dari :

1. Kepala.
2. Subbagian Tata Usaha.
3. Seksi Pelayanan Kesehatan Olahraga.
4. Seksi Kemitraan
5. Kelompok Jabatan Fungsional.



Gambar 2.2 Struktur Organisasi BKOM Bandung

Dibawah ini dijelaskan struktur organisasi di Balai Kesehatan Olahraga Masyarakat Bandung (BKOM) sebagaimana berikut :

1. BKOM Bandung dipimpin oleh seorang kepala yang berada dibawah dan bertanggung jawab kepada Direktur Jenderal Bina Kesehatan Masyarakat Depatemen Kesehatan dan secara teknis fungsional dibina oleh Direktur Kesehatan Komunitas.
2. Sub bagian Tata Usaha mempunyai tugas melakukan urusan persuratan, penyususan program dan pelaporan, keuangan, kepegawaian, kerumahtanggaan, dan perlengkapan.

3. Seksi Pelayanan Kesehatan Olahraga mempunyai tugas melakukan penyiapan bahan koordinasi pelayanan kesehatan olahraga dan kebugaran jasmani, serta fasilitas penelitian dan pengembangan di bidang kesehatan olahraga.
4. Seksi Kemitraan mempunyai tugas melakukan penyiapan bahan koordinasi kerjasama kemitraan dan sosialisai, serta fasilitas pelatihan di bidang kesehatan olahraga.
5. Kelompok Jabatan Fungsional mempunyai tugas melakukan kegiatan sesuai dengan jabatan fungsional masing-masing berdasarkan peraturan perundang-undangan yang berlaku.

2.2 Landasan Teori

Pada landasan teori akan dijelaskan beberapa teori-teori yang berhubungan dengan proses pembuatan sistem pakar ini yang akan dijelaskan sebagai berikut.

2.2.1 Kecerdasan Buatan

Kecerdasan buatan berasal dari bahasa inggris "*Artificial Intelligence*" atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan yang dimaksud di sini merujuk pada mesin yang mampu berfikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan sepeti yang dilakukan oleh manusia [1].

Berikut adalah tujuan dari kecerdasan buatan) [1] :

1. Membuat mesin menjadi lebih pintar (tujuan utama).
2. Memahami apa itu kecerdasan (tujuan ilmiah).
3. Membuat mesin lebih bermanfaat (tujuan *entrepreneurial*)

Berdasarkan definisi ini, maka kecerdasan buatan menawarkan media ataupun uji teori tentang kecerdasan. Teori-teori ini nantinya dapat dinyatakan dalam bahasa pemrograman dan eksekusinya dapat dibuktikan pada komputer nyata.

2.2.1.1 Sejarah Kecerdasan Buatan

Istilah AI pertama kali dikemukakan pada tahun 1956 di konferensi Dartmouth. Sejak saat itu, AI terus dikembangkan sebab berbagai penelitian mengenai teori-teori dan prinsip-prinsipnya juga terus berkembang. Meskipun istilah AI baru muncul tahun 1956, tetapi teori-teori yang mengarah ke AI sudah muncul sejak 1941. Secara lengkap, ini berikut tahapan-tahapan sejarah perkembangan AI [7].

Pada awal abad 17, René Descartes mengemukakan bahwa tubuh hewan bukanlah apa-apa melainkan hanya mesin-mesin yang rumit. Blaise Pascal menciptakan mesin penghitung digital mekanis pertama pada 1642. Pada 19, Charles Babbage dan Ada Lovelace bekerja pada mesin penghitung mekanis yang dapat diprogram.

Bertrand Russell dan Alfred North Whitehead menerbitkan *Principia Mathematica*, yang merombak logika formal. Warren McCulloch dan Walter Pitts menerbitkan "Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas " pada 1943 yang meletakkan pondasi untuk jaringan syaraf.

Tahun 1950-an adalah periode usaha aktif dalam AI. Program AI pertama yang bekerja ditulis pada 1951 untuk menjalankan mesin Ferranti Mark I di *University of Manchester* (UK), sebuah program permainan naskah yang ditulis

oleh Christopher Strachey dan program permainan catur yang ditulis oleh Dietrich Prinz. John McCarthy membuat istilah "kecerdasan buatan " pada konferensi pertama yang disediakan untuk pokok persoalan ini, pada 1956. Dia juga menemukan bahasa pemrograman Lisp. Alan Turing memperkenalkan "Turing test" sebagai sebuah cara untuk mengoperasionalkan test perilaku cerdas. Joseph Weizenbaum membangun ELIZA, sebuah chatterbot yang menerapkan psikoterapi Rogerian.

Selama tahun 1960-an dan 1970-an, Joel Moses mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah di dalam program Macsyma, program berbasis pengetahuan yang sukses pertama kali dalam bidang matematika. Marvin Minsky dan Seymour Papert menerbitkan *Perceptrons*, yang mendemostrasikan batas jaringan syaraf sederhana dan Alain Colmerauer mengembangkan bahasa komputer Prolog. Ted Shortliffe mendemonstrasikan kekuatan sistem berbasis aturan untuk representasi pengetahuan dan inferensi dalam diagnosa dan terapi medis yang kadangkala disebut sebagai sistem pakar pertama. Hans Moravec mengembangkan kendaraan terkendali komputer pertama untuk mengatasi jalan berintang yang kusut secara mandiri.

Pada tahun 1980-an, jaringan syaraf digunakan secara meluas dengan algoritma perambatan balik, pertama kali diterangkan oleh Paul John Werbos pada 1974. Tahun 1990-an ditandai perolehan besar dalam berbagai bidang AI dan demonstrasi berbagai macam aplikasi. Lebih khusus Deep Blue, sebuah komputer permainan catur, mengalahkan Garry Kasparov dalam sebuah

pertandingan 6 game yang terkenal pada tahun 1997. *Defense Advanced Research Projects Agancy* (DARPA) menyatakan bahwa biaya yang disimpan melalui penerapan metode AI untuk unit penjadwalan dalam Perang Teluk pertama telah mengganti seluruh investasi dalam penelitian AI sejak tahun 1950 pada pemerintah Amerikat Serikat.

2.2.1.2 Subdisiplin Ilmu Dalam Kecerdasan Buatan

Kecerdasan buatan dibagi kedalam beberapa subdisiplin sebagai berikut[1].

1. Sistem Pakar (*Expert System*)

   Komputer sebagai sarana untuk menyimpan pengetahuan para pakar sehingga komputer memiliki keahlian menyelesaikan permasalahan dengan meniru keahlian yang dimiliki pakar.

2. Pengolahan Bahasa Alami (*Natural Language Processing*)

   Pemrosesan bahasa alami adalah suatu bidang kecerdasan buatan yang mempelajari tata bahasa manusia. Untuk itu komputer dibuat agar mengerti bahasa manusia, dengan demikian komputer dapat memberikan tanggapan terhadap masukan dari manusia dengan jawaban yang sesuai dengan struktur bahasa yang baik.

3. Pengenalan Ucapan (*Speech Recognition*)

   Pengenalan ucapan, atau yang sering disebut dengan *Automatic Speech Recognition* (ASR), adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan.

4. Robotika dan Sistem Sensor (*Robotics and Sensory Systems*)

   Robotika adalah suatu bidang rekayasa yang mencurahkan perhatiannya ke bidang duplikasi kemampuan fisik manusia, yaitu suatu komplemen alami dalam bidang artificial intelligence yang berusaha menirukan kemampuan mental manusia. Robot adalah mesin atau manipulator yang mampu melaksanakan fungsi fisik manusia secara terbatas.

5. *Computer vision*

   Menginterpretasikan gambar atau objek-objek tampak melalui computer.

6. *Intelligent Computer-Aided Instruction*

   Komputer dapat digunakan sebagai tutor yang dapat melatih & mengajar.

7. *Game Playing*

   Kebanyakan permainan dilakukan dengan menggunakan sekumpulan aturan. Dalam permainan digunakan apa yang disebut dengan pencarian ruang. Teknik untuk menentukan alternatif dalam menyimak problema ruang merupakan sesuatu yang rumit. Teknik tersebut disebut dengan HEURISTIC. Permainan merupakan bidang yang menarik dalam studi heuristic.

2.2.2 Sistem Pakar

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan

pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar [1].

Untuk memahami aplikasi sistem pakar, selain memahami definisinya, kita juga harus mengetahi tujuan dari sistem pakar, komponen-komponennya, semua domain, dan contoh-contoh aplikasinya, stakeholders, dan alasan digunkannya sistem ini.

2.2.2.1 Manfaat Dan Kerugian Sistem Pakar

Sistem pakar menjadi sangat popular karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya [1]:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seseorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberikan nasehat konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi dilingkungan berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal. Sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem konvensional, sistem pakar dapat bekerja dengan informasi

yang tidak lengkap. Penggguna dapat merespon dengan: “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar akan tetap memberikan jawabannya.

10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelasan yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sisstem pakar mengambil sumber pengetahuan dari banayak pakar.

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya:

1. Biaya yang sangat mahal membuat dan memeliharanya.
2. Sulit di kembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar

2.2.2.2 Ciri-Ciri Sistem Pakar

Ciri-ciri sistem pakar adalah sebagaimana berikut [1]:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/rule tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.

8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntut oleh dialog dengan pengguna.

2.2.2.3 Area Permasalahan Aplikasi Sistem Pakar

Biasanya aplikasi sistem pakar menyentuh beberapa area permasalahan berikut [1]:

1. Interpretasi, yaitu menghasilkan deskripsi situasi berdasarkan masukan-masukan.
2. Prediksi, yaitu memperkirakan akibat yang mungkin terjadi dari situasi yang ada.
3. Diagnosis, yaitu menyimpulkan suatu keadaan berdasarkan gejala-gejala yang diberikan.
4. Desain, yaitu melakukan perancangan berdasarkan kendala-kendala yang diberikan
5. Planning, yaitu merencanakan tindakan-tindakan yang akan dilakukan.
6. *Monitoring*, yaitu membandingkan hasil pengamatan dengan proses perencanaan..
7. *Debugging*, yaitu menentukan penyelesaian dari suatu kesalahan sistem.
8. Reparasi, melaksanakn rencana perbaikan.
9. *Instruction*, yaitu melakukan instruksi untuk diagnosis, *debugging* dan per-baikan kinerja.

10. Kontrol, yaitu mengatur tingkah laku suatu environment yang kompleks seperti kontrol terhadap interpretasi, prediksi, perbaikan, dan *monitoring* kelakuan sistem.

2.2.3 Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar meliputi enam hal sebagai berikut [1]:

1. Kepakaran (*Expertise*)
2. Pakar (*Expert*)
3. Pemindahan kepakaran (*Transfering Expertise*)
4. Inferensi (*Inferencing)*
5. Aturan-aturan (*Rule*)
6. Kemampuan Menjelaskan (*Explanation Capability*)

2.2.3.1 Kepakaran (*Expertise*)

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatih, membaca dan pengalaman. Kepakaran ini lah yang memungkin parah ahli dapat mengambil keputusan lebih cepat dan lebih baik dari pada seseorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan tentang [1]:

1. Fakta-fakta tentang bidang permasalahan tertentu.
2. Teori-teori tentang bidang permasalahan tertentu.
3. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya.

4. Aturan *heuristic* yang harus dikerjakan dalam situasi tertentu
5. Strategi global untuk memecahkan permasalahan.
6. Pengetahuan tentang pengetahuan (*meta knowledge*)

2.2.3.2 Pakar (*Expert*)

Pakar adalah seorang yang memiliki keahlian tentang suatu hal dalam tingkatan tertentu, ahli dapat menggunakan suatu permasalahan yang ditetapkan dengan beberapa cara yang berubah-ubah dan merubahnya kedalam bentuk yang dapat dipergunakan oleh dirinya sendiri dengan cepat dan cara pemecahan yang mengesankan. Jadi seorang pakar harus mampu melakukan kegiatan-kegiatan berikut [1]:

1. Mengenali dan memformulasikan masalah.
2. Memecahlan masalah dengan cepat dan tegas.
3. Menerangkan pemecahannya.
4. Belajar dari pengalaman.
5. Merekstrukturisasi pengetahuan.
6. Memecahkan aturan-aturan.
7. Menentukan relevansi.

2.2.3.3 Pemindahan kepakaran (*Transfering Expertise*)

Tujuan dari sistem pakar adalah untuk memindahkan kemampuan (*transferring expertise*) dari seorang ahli atau sumber keahlian yang lain ke dalam komputer dan kemudian memindahkannya dari komputer kepada pemakai yang tidak ahli (bukan pakar). Proses ini meliputi empat aktivitas yaitu [1]:

1. Akuisi pengetahuan (*knowledge acquisition*) yaitu kegiatan mencari dan mengumpulkan pengetahuan dari para ahli atau sumber keahlian yang lain.
2. Representasi pengetahuan (*knowledge representation*) adalah kegiatan menyimpan dan mengatur penyimpanan pengetahuan yang diperoleh dalam komputer. Pengetahuan berupa fakta dan aturan disimpan dalam komputer sebagai sebuah komponen yang disebut basis pengetahuan
3. Inferensi pengetahuan (*knowledge inferencing*) adalah kegiatan melakukan inferensi berdasarkan pengetahuan yang telah disimpan didalam komputer.
4. Pemindahan pengetahuan (*knowledge transfer*) adalah kegiatan pemindahan pengetahuan dari komputer ke pemakai yang tidak ahli.

2.2.3.4 Inferensi (*Inferencing)*

Inferensi merupakan suatu proses untuk menghasilkan informasi dari fakta yang diketahui. Inferensi adalah konklusi logis atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar, proses inferensi dialakukan dalam suatu modul yang disebut *inference engine*. Ketika representasi pengetahaun pada bagian basis pengetahuan telah lengkap, atau paling tidak telah

berada pada level yang cukup akurat, maka representasi pengetahuan tersebut telah siap digunakan [1]:

2.2.3.5 Aturan-aturan (*Rule*)

Kebanyakan software sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based systems*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur maslah [1].

2.2.3.6 Kemampuan Menjelaskan (*Explanation Capability*)

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsitem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinnnya sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya.

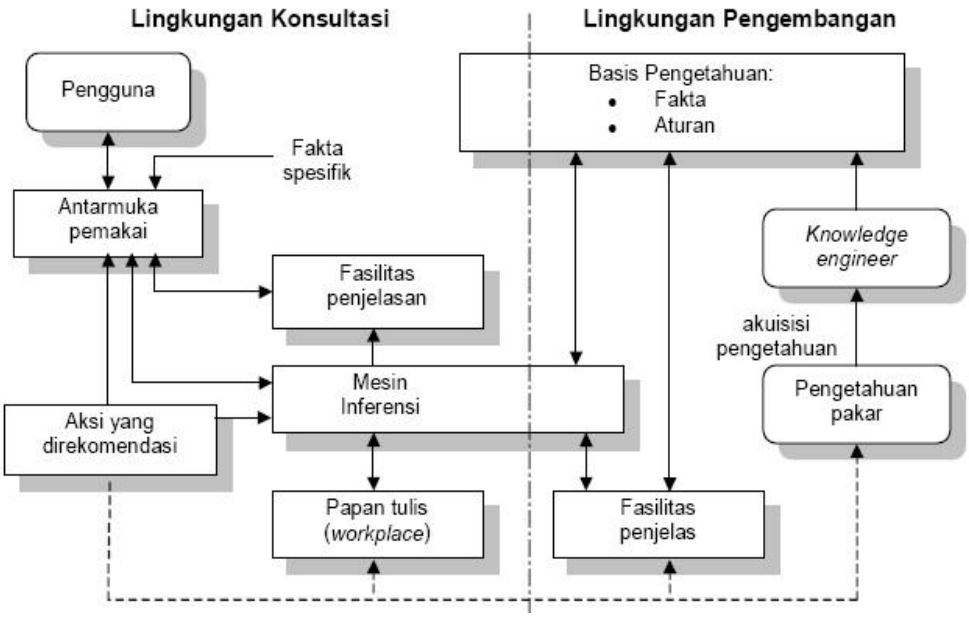
Sistem Konvensional	Sistem Pakar
Informasi dan pemrosessannya biasanya digabungkan dalam satu program	Basis pengetahuan dipisahkan secara jelas dengan mekanisme inferensi.
Program tidak membuat kesalahan(yang membuat kesalahan : program atau pengguna).	Program dapat berbuat kesalahan
Biasanya tidak menjelaskan mengapa data masukan diperlukan atau bagaimana output dihasilkan.	Penjelasn merupakan bagian terpenting dari semua sistem pakar.
Perubahan program sangat menyulitkan	Perubahan dalam aturan-aturan mudah dilakukan.
Sistem hanya bias beroperasi setelah lengkap atau selesai	Sistem dapat beroperasi hanya dengan aturan-aturan yang sedikit 9sebagai prototipe awal).
Eksekusi dilakukan langkah demi langkah.	Eksekusi dilakukan dengan menggunakan heuristic dan logika pada seluruh basis pengetahuan.
Perlu informasi lengkap agar lebih bias beroperasi	Dapat beroperasi dengan informasi yang tidak lengkap atau mengandung ketidakpastian.
Manipulasi efektif dari basis data yang besar.	Manipulasi efektif dari basis data yang besar.
Menggunakan data	Menggunakan pengetahuan.

Tujuan utama : efisiensi	Tujuan utama : efektivitas.
Mudah berurusan dengan data kuantitatif.	Mudah berurusan dengan data kualitatif.
Menangkap, menambah, dan mendistribusikan akses ke data numeric atau informasi.	Menangkap, menambah, dan mendistribusikan akses ke pertimbangan dan pengetahuan.

Tabel 2.1 Perbandingan Antara Sistem Konvensional Dengan Sistem Pakar [9].

2.2.4 Struktur Sistem Pakar

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. Gambar 2.3 menunjukan komponen-komponen yang penting dalam sebuah sistem pakar [1].



Gambar 2.3 Komponen-komponen Yang Penting Dalam Sebuah Sistem Pakar [9]

Berikut adalah penjelasan komponen-komponen yang penting dalam sebuah sistem pakar berdasarkan gambar diatas.

1. Akuisisi Pengetahuan

   Akuisisi Pengetahuan merupakan penerimaan atau perolehan penetahuan yang dapat diperoleh dari seorang pakar, buku teks, laporan penelitian dengan dukungan dari seorang *knowledge engineer*.

2. Basis Pengetahuan (*Knowledge Base*)

   Basis Pengetahuan, terdiri dari dua jenis, yaitu fakta (situasi dan teori) dan rule atau aturan.

3. Mesin Inferensi (*Inference Engine*)

   Mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, dapat dikatakan sebagai mesin pemikir (*thinking machine*). Pada prinsipnya mesin\ inferensi inilah yang akan mencari solusi dari suatu permasalahan. Konsep yang biasanya digunakan untuk mesin inferensi adalah runut balik (*backward chaining*) dan menggunakan runut maju (*forward chaining*).

4. Daerah Kerja

   Workplace merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil dan kesimpulan yang dicapai. Ada 3 tipe keputusan yang dapat direkam yaitu :

a. Rencana, bagaimana menghadapi masalah.

b. Agenda, aksi-aksi potensial yang sedang menunggu untuk dieksekusi.

c. Solusi, calon aksi yang akan dibangkitkan.

5. Antarmuka Pengguna (*User Interface*)

   Antarmuka (*Interface*) yaitu sistem pakar menggantikan seorang pakar dalam suatu situasi tertentu, sistem harus menyediakan pendukung yang diperlukan oleh pemakai yang tidak memahami masalah teknis. Sistem pakar juga menyediakan komunikasi antara sistem dan pemakainya, yang disebut sebagai antarmuka. Antarmuka yang efektif dan ramah pengguna (*user-friendly*) penting sekali terutama bagi pemakai yang tidak ahli dalam bidang yang diterapkan pada sistem pakar.

6. Subsistem Penjelasan (*Explanation Subsystem/Justifier*)

   Fasilitas Penjelasan yaitu proses menentukan keputusan yang dilakukan oleh mesin inferensi selama sesi konsultasi mencerminkan proses penalaran seorang pakar. Karena pemakai kadangkala bukanlah ahli dalam bidang tersebut, maka dibuatlah fasilitas penjelasan. Fasilitas penjelasan inilah yang dapat memberikan informasi kepada pemakai mengenai jalannya penalaran sehingga dihasilkan suatu keputusan. Bentuk penjelasannya dapat berupa keterangan yang diberikan setelah suatu pertanyaan diajukan, yaitu penjelasan atas pertanyaan mengapa, atau penjelasan atas pertanyaan bagaimana sistem mencapai konklusi.

7. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)

   Perbaikan Pengetahuan yakni dapat menganalisis pengetahuannya sendiri dan kegunaannya, belajar darinya, dan meningkatkannya untuk konsultasi mendatang.

8. *Knowledge Engineer*

   Knowledge Engineer yaitu seorang spesialis sistem yang menerjemahkan pengetahuan yang dimiliki seorang pakar menjadi pengetahuan yang akann tersimpan dalam basis pengetahuan pada sebuah sistem pakar.

9. Pengetahuan pakar

   Pakar merupakan seseorang yang ahli di bidang tertentu.

10. Pengguna (*User)*

    Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert)* yang membutuhkan solusi, saran ,dan pelatihan (*training*) dari berbagai permasalahan yang ada.

2.2.5 Representasi Pengetahuan

Data, informasi, pengetahuan, yang terekam dalam ingatan seseorang, dipelajari dan ditiru untuk ditata ulang dalam memori komputer sehingga dengan aplikasi perangkat lunak tertentu, data yang tersimpan biasa menghasilkan informasi yang bermanfaat. Dalam perkembangannya, memori komputer dapat dimanfaatkan untuk menyimpan pengetahuan seseorang [1].

Representasi pengetahuan (knowledge representation) adalah cara untuk menyajikan pengetahuan yang diperoleh kedalam suatu skema/diagram tertentu sehingga dapat diketahui relasi antara satu pengetahuan dengan pengetahuan yang

lain dan dapat dipakai untuk menguji kebenaran penalarannya. Format representasi harus mudah dipahami sehingga seorang programmer mampu mengekspresikan pengetahuan untuk mendapatkan solusi suatu masalah.

Banyak cara untuk merepresentasikan pengetahuan (fakta), namun semua cara tersebut harus mengacu pada dua entitas berikut.

1. Fakta, yaitu kejadian sebenarnya. Fakta inilah yang akan kita representasikan.
2. Representasi dari fakta. Berdasarkan representasi inilah kita dapat mengolah fakta.

Secara teknik, pada bagian ini kita akan membahas representasi pengetahuan menjadi lima kelompok, yaitu Representasi Logika, jaringan Semantik, *Frame*, *Script* (Naskah), dan aturan produksi [1].

1. Representasi Logika

   Untuk merepresentasikan pengetahuan (fakta), biasanya teknik representasi logika menggunakan ekspresi-ekspresi dalam logika formal. Ekspresi-ekspresi inilah yang nantinya digunakan sebagai proses untuk membentuk kesimpulan atau menarik suatu inferensi berasarkan fakta yang ada. Representasi logika dibagi menjadi dua jenis, yaitu logika proposisi dan logika predikat.

   a. Logika proposisi

      Proposisi (pernyataan) adalah suatu kalimat deklaratif yang bernilai benar saja atau salah saja, tetapi tidak sekaligus benar dan salah. Benar atau salahnya sebuah proposisi disebut nilai kebenaran proposisi itu. Proposisi

dilambangkan dengan huruf kecil p, q, r, … dan disebut sebagai proposisi atomic.

b. Logika Predikat

Pada logika predikat proposisi dibedakan menjadi argument (obyek) dan perdikat (keterangan). Secara umum penulisan proposisi dalam logika predikat dapat dinyatakan sebagai berikut.

Proposisi :”Bu Astuti mencintai Pak Agus Winarno”

Dalam logika predikat dinyatakan sebagai :

Mencintai (Bu Astuti, Pak Agus Winarno)

Dalam hal ini,

Predikat = mencintai

Argumen 1 = Bu Astuti

Argumen 2 = Pak Agus Winarno

2. Jaringan semantik

Pada dasarnya jaringan semantik merupakan representasi pengetahuan yang digunakan untuk menggambarkan data dan informasi, yang menunjukan hubungan antara berbagai objek. Jaringan semantik merupakan grafik, yang terdiri dari simpul (*nodes*), yang merepresentasikan objek, dan busur (*arc*), yang menunjukan relasi antara objek-objek tersebut.

3. Bingkai (*Frame*)

Bingkai berupa kumpulan slot-slot yang berisi atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berupa kejadian,

lokasi, situasi ataupun elemen-elemen lainnya. Bingkai digunakan untuk representasi pengetahuan deklaratif.

4. Kaidah Produksi

   Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dalam bentuk implikasi yaitu jika–maka (IF–THEN). Kaidah IF-THEN menghubungkan antesenden (antecedent) dengan konsekuensi yang diakibatkannya. Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Berbagai struktur kaidah IF–THEN yang menghubungkan objek atau atribut sebagai berikut :

   IF [kondisi] THEN [aksi]

   Contoh :

   IF [demam AND kelelahan AND sesak nafas]

   THEN [ambil aspirin AND istirahat]

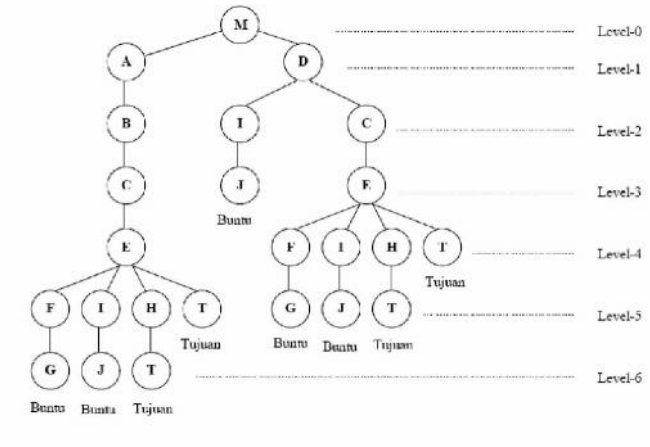
5. Sript (Naskah)

   Script mula-mula dirancang oleh Roger Schank dan Robert P. Abelson dan kelompok riset mereka sebagai alat pengorganisasi struktur-struktur *Conceptual Dependency* (ketergantungan konseptual) menjadi deskripsi khusus. *Conceptual Dependency* adalah teori tentang bagaimana merepresentasikan pengetahuan tentang *event* (kejadian) yang biasanya terkandung dalam kalimat bahasa natural.

2.2.6 Pohon Keputusan

Pohon keputusan (*Tree*) adalah suatu hirarki struktur yang terdiri dari node (simbol) yang menyimpan informasi atau pengetahuan dan cabang yang menghubungkan node. Cabang disebut juga *link* atau *edge* dan *node* disebut juga vertek [5].

Struktur *tree* digunakan untuk menggambarkan keadaan secara hirarkis. Node yg terletak pada level-o disebut ’akar’. *Node* akar : menunjukkan keadaan awal & memiliki beberapa percabangan yang terdiri atas beberapa *node* yg disebut ’anak’. *Node-node* yg tidak memiliki anak disebut ’daun’ menunjukkan akhir dari suatu pencarian, dapat berupa tujuan yang diharapkan (*goal*) atau jalan buntu (*dead end*). Gambar 2.4 menunjukkan *tree* pencarian untuk *graph* keadaan dengan 6 level.



Gambar 2.4 Struktur *Tree* [5]

2.2.6.1 Metode Pencarian

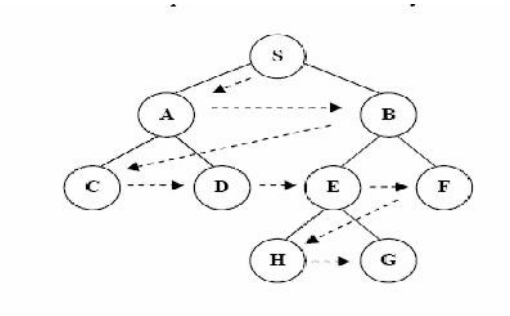
Hal penting dalam menentukan keberhasilan sistem cerdas adalah kesuksesan dalam pencarian. Pada dasarnnya ada dua teknik pencarian yaitu [1] :

1. Pencarian Buta (*Blind Search*)

   Pencarian Buta (*Blind Search*) dibagi kedalam beberapa pencarian yaitu:

   a. Breadth-First Search

   Semua *node* pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi *node-node* pada level n+1. pencarian dimulai dari *node* akar terus ke level 1 dari kiri ke kanan, kemudian berpindah ke level berikutnya dari kiri ke kanan hingga solusi ditemukan.



Gambar 2.5 *Breadth_First Search* [1]

Keuntungan Metode *Breadth -First Search* :

1) Tidak akan menemui jalan buntu

2) Jika ada 1 solusi, maka *breadth – first search* solusi, maka solusi minimum akan ditemukan.

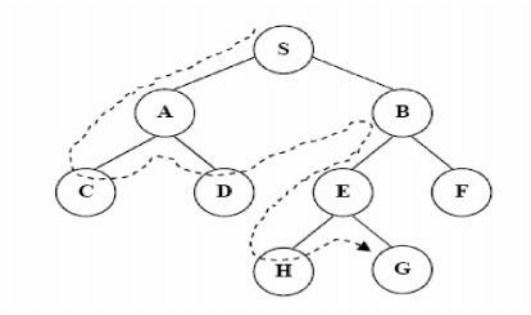
Kelemahan Metode *Breadth -First Search* :

1) Membutuhkan memori yang cukup banyak

2) Membutuhkan waktu yang cukup lama.

b. *Depth-First Search*

Proses pencarian dilakukan pada semua anaknya sebelum dilakukan pencarian ke *node-node* yang selevel. Pencarian dimulai dari *node* akar ke level yang lebih tinggi. Proses diulangi terus hingga ditemukan solusi.



Gambar 2.6 *Depth-First Search* [1]

Keuntungan dari Metode *Depth – First Search*:

1) Membutuhkan memori relatif kecil, karena hanya *node-node* pada lintasan yang aktif saja yang disimpan
2) Secara kebetulan, akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan.

Kelemahan Metode *Depth – First Search* :

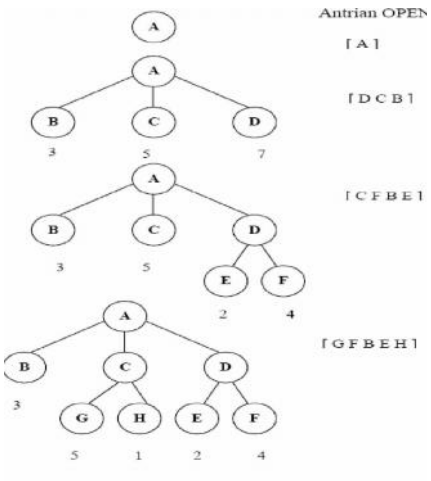
1) Memungkinkan tidak ditemukannya tujuan yang diharapkan
2) Hanya mendapat 1 solusi pada setiap pencarian .

2. *Heuristic Search*

Pencarian buta tidak selalu dapat diterapkan dengan baik, hal ini disebabkan waktu aksesnya yang cukup lama & besarnya memori yang diperlukan. Kelemahan ini dapat diatasi jika ada informasi tambahan dari domain yang bersangkutan.

a. *Best First Search*

Metode *best first search* merupakan kombinasi dari metode *depth first search* dan *breadth first search* dengan mengambil kelebihan dari kedua metode tersebut. *Hill climbing* tidak diperbolehkan untuk kembali ke *node* pada lebih rendah meskipun *node* tersebut memiliki nilai heuristik lebih baik. Pada *best first search*, pencarian diperbolehkan mengunjungi *node* di lebih rendah, jika ternyata *node* di level lebih tinggi memiliki nilai heuristik lebih buruk. Untuk mengimplementasikan metode ini, dibutuhkan 2 antrian yang berisi *node-node*, yaitu : *OPEN* : berisi *node-node* yang sudah dibangkitkan, sudah memiliki fungsi heuristik namun belum diuji. Umumnya berupa antrian berprioritas yang berisi elemen-elemen dengan nilai heuristik tertinggi. *CLOSED* : berisi *node-node* yang sudah diuji.



Gambar 2.7 *Best First Search* [1]

Diasumsikan *node* dengan nilai yang lebih besar memiliki nilai evaluasi yang lebih baik. Pada keadaan awal, antrian berisi A. Pengujian dilakukan di level pertama, *node* D memiliki nilai terbaik, sehingga menempati antrian pertama, disusul dengan C dan B. Node D memiliki cabang E dan F yang masing-masing bernilai 2 & 4. Dengan demikian C merupakan pilihan terbaik dengan menempati antrian pertama. Demikian seterusnya.

2.2.6.2 Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini akan menganalisa suatu masalah dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik [1]. Mesin inferensi mempunyai dua fungsi yaitu inferensi dan kendali. Inferensi adalah proses menalar, sedangkan kendali berfungsi mengendalikan eksekusi. Inferensi melibatkan proses pencocokkan (*watching*) dan penggabungan (*unification*). Proses tersebut berdasarkan pada

suatu basis data yang berisi fakta-fakta biasanya tersimpan dalam berkas khusus dan dapat juga diperoleh dari konsultasi dan dipakai dalam proses pengujian aturan-aturan yang diisyaratkan dari basis pengetahuan. Dua teknik inferensi yaitu:

1. Pelacakan ke depan (*forward chaining*).
2. Pelacakan ke belakang (*backward chaining*).

Pada dasarnya ada dua jenis teknik inferensi yaitu sebagaimana berikut :

1. *Forward Chaining*

Forward chaining merupakan perunutan yang dimulai dengan menampilkan kumpulan data atau fakta yang menyakinkan menuju konklusi akhir. *Forward chaining* dimulai dari premis – premis atau informasi masukan (*IF*) dahulu kemudian menuju konklusi atau *derived information* (*THEN*) atau dapat dimodelkan sebagai berikut [9]:

IF (informasi masukan)

THEN (konklusi)

Informasi masukan dapat berupa data, bukti, temuan atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan atau diagnosa. Sehingga jalannya *forward chaining* maju dapat dimulai dari data menuju tujuan dan bukti menuju hipotesa, dari temuan menuju penjelasan, atau dari pengamatan menuju diagnosa.

Contoh : IF Lampu 1 dinyalakan

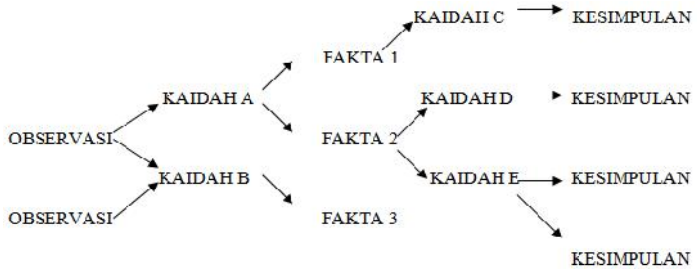
AND Lampu 1 tidak menyala

AND Lampu 1 dihubungkan dengan sekering

AND Sekering masih utuh

THEN Lampu 1 rusak

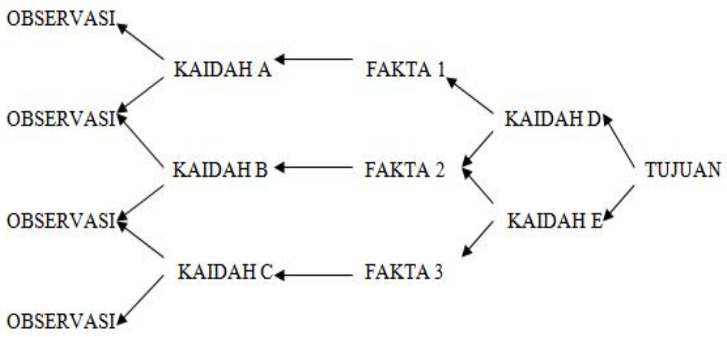
Strategi dari sistem ini adalah dimulai dari inputan beberapa fakta, kemudian menurunkan beberapa fakta dari aturan-aturan yang cocok pada *knowledge base* dan melanjutkan prosesnya sampai jawaban sesuai. *Forward chaining* dapat dikatakan sebagai penelusuran deduktif.



Gambar 2.8 Diagram *Forward Chaining* [9]

2. *Backward Chaining*

Strategi penarikan keputusan yang didasarkan dari hipotesa atau dugaan yang didapat dari informasi yang ada. Ciri dari strategi ini adalah pertanyaan user. Memperoleh fakta biasanya diajukan dalam bentuk “YA” atau “TIDAK”, proses ini berdampak dengan diterima atau tidaknya hipotesis [9].



Gambar 2.9 Diagram *Backward Chaining* [9]

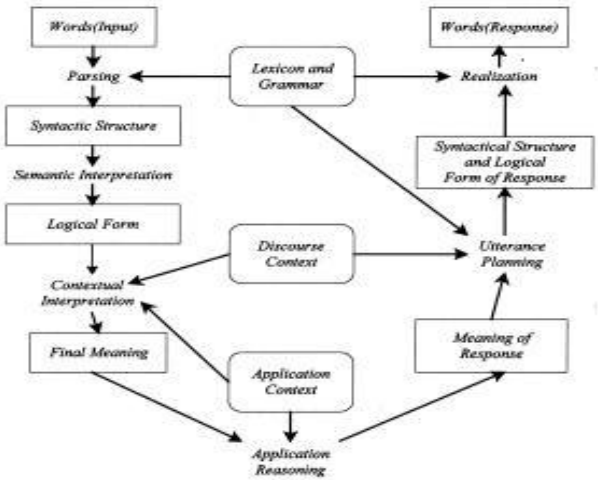
Ada empat faktor metode menentukan mana arah yang lebih baik digunakan dari dua arah penelusuran yaitu [9]:

1. Jumlah keadaan awal dan keadaan akhir akan lebih muda bergerak dari kumpulan keadaan yang lebih sedikit ke kumpulan yang lebih banyak.
2. Besar kecilnya faktor percabangan lebih baik menuju ke arah yang faktor percabangannya sedikit.
3. Proses penalaran program sangatlah penting untuk menuju kearah yang lebih condong dengan cara pemikiran pemakai.
4. Kejadian yang memicu rangkaian tindakan pemecahan masalah. Jika kejadian ini adalah kedatangan fakta baru, maka dipilih *forward chaining*, tetapi jika kejadian ini adalah suatu pertanyaan yang membutuhkan tanggapan, akan lebih baik jika dipilih *backward chaining*

2.2.7 *Natural Language Processing* (Pengolahan Bahasa Alami)

Natural Languange Prosessing (pengolahan bahasa alami) adalah pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia. Bentuk utama representasinya adalah berupa suara/ucapan (*spoken language*), tetapi sering pula dinyatakan dalam bentuk tulisan [1].

Untuk mencapai tujuan ini dibutuhkan tahapan proses. Proses yang pertama ialah *parsing* atau analisa sintaksis yang memeriksa kebenaran struktur kalimat berdasarkan suatu *grammar* (tata bahasa) dan *lexicon* (kosa kata) tertentu. Proses kedua ialah *semantic interpretation* atau interpretasi semantik disebut juga analisis semantik yang bertujuan untuk merepresentasikan arti dari kalimat secara *context-independent* untuk keperluan lebih lanjut. Gambaran organisasi dari sebuah sistem NLP yang lengkap ditunjukkan pada gambar berikut.



Gambar 2.10 Organisasi Sebuah Sistem NLP [6]

2.2.7.1 Grammar and Lexicon

Grammar ialah aturan-aturan pembentukan suatu kalimat dalam sebuah bahasa, atau yang biasa disebut tata bahasa. Dengan adanya grammar, parsing dapat dilakukan secara cepat dengan hanya melakukan proses *searching*. Parser akan mencari aturan-aturan yang tepat untuk membentuk struktur suatu kalimat. Salah satu contoh grammar yang sederhana untuk bahasa Inggris ialah sebagai berikut [6]:

Sentence → noun phrase + verb phrase

noun phrase → determiner + noun

verb phrase → verb + noun phrase

determiner → the, a

noun → dog, dogs, cat

verb → chased, see, sees

Grammar ini akan mampu menghasilkan sekumpulan kalimat bahasa Inggris yang sederhana, seperti :

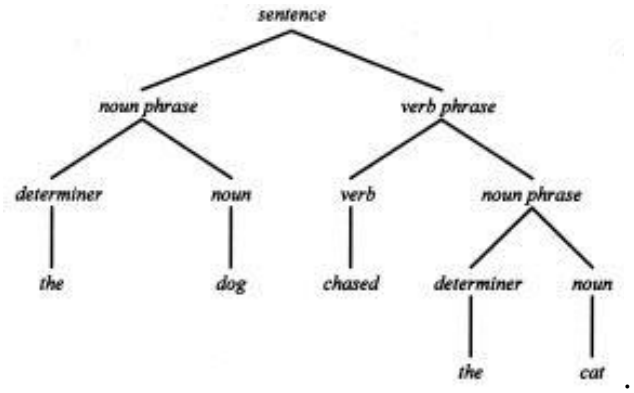
The dog chased a cat

The dog sees a cat

The dogs sees a cat (nonvalid)

A dogs chased a cat (nonvalid)

Proses pembentukan kalimat dari grammar disebut *derivation*. Contoh derivation untuk menghasilkan kalimat *"The dog chased the cat"* dari grammar di atas ditunjukkan pada gambar berikut



Gambar 2.11 Parse Tree untuk Kalimat "*The dog chased the cat"* [6]

Untuk membuat sebuah aplikasi NLP yang handal, dibutuhkan kamus atau kosa kata yang handal pula. Seperti halnya seorang manusia, semakin lengkap kosa kata dalam sebuah sistem NLP, semakin baik sistem tersebut dapat berkomunikasi. Kosa kata dalam komputer disebut lexicon. Faktor yang penting menyangkut lexicon ialah penyimpanannya, karena umumnya lexicon memiliki ukuran yang sangat besar. Karena itu umumnya lexicon hanya menyimpan bentuk dasar dari kata-kata yang ada, sedangkan untuk bentuk-bentuk turunannya (misalnya *kicks* dari *kick*) dapat diperoleh dengan menerapkan analisa *morphology* (proses perubahan bentuk kata).

2.2.7.2 Parsing (Analisis Sintaksis)

Parser atau *syntactic analyzer* pada kompilator bahasa pemrograman berfungsi untuk memeriksa kebenaran kemunculan setiap *token*. Pada sistem yang akan dibuat, fungsi dari *parser* ini agak berbeda karena *token* yang akan diolah semua memiliki tipe yang sama yaitu berupa kata (word). Urutan kemunculan *token* yang berupa kata-kata tersebut akan diolah dengan mengacu pada bank data

agar didapatkan makna kalimat yang sesungguhnya. Dengan kata lain, tahap analisa semantik terjadi di bagian bank data. Kemampuan dari *parser* untuk mengolah *token* dan bekerja sama dengan bank data inilah yang paling menentukan tingkat kecerdasan dari sistem yang akan dibuat.

Proses parsing dapat dibagi menjadi dua bagian besar metode yaitu *Top Down parsing* dan *Bottom Up parsing* [6].

1. *Top Down parsing*

   Memulai pemeriksaan dari simbol awal S dan mencoba untuk mencari bentuk simbol terminal berikutnya yang sesuai dengan jenis kata dari kalimat masukan. Ada 2 kelas metoda parsing *top-down*, yaitu kelas metoda *dengan backup* dan kelas metoda *tanpa backup*. Contoh metoda kelas *dengan backup* adalah metoda *Brute-Force*, sedangkan contoh metoda kelas *tanpa backup* adalah metoda *recursive descent*.

   a. *Backtrack/backup: Brute Force*

      Metode ini akan memilih aturan produksi mulai dari paling kiri, dan melakukan *expand* semua non terminal pada aturan produksi sampai yang tertinggal adalah simbol terminal. Kemungkinan pertama *string* masukan sukses di-*parsing*, bisa juga bila terjadi expansi yang salah untuk suatu simbol variabel maka akan dilakukan *backtrack*.

   b. *No backtrack: Recursive Descent Parser*

      Recursive descent parser adalah salah satu cara untuk mengaplikasikan Bahasa Bebas konteks untuk melakukan analisa sintaksis suatu *source code*. Di sini simbol terminal maupun simbol variabelnya

sudah bukan sebuah karakter, tetapi berupa besaran leksik sebagai simbol terminalnya dan besaran sintaks sebagai simbol variabelnya. Ciri dari *recursive descent parser* yang menonjol adalah secara rekursif menurunkan semua variabel dari awal sampai bertemu terminal dan tidak pernah mengambil *token* secara mundur (*no backtrack*).

2. *Bottom Up parsing*

   Metode ini melakukan penelusuran dari *leaf*/daun menuju ke *root*/puncak. Gramatika yang dipakai akan lebih banyak bercabang ke arah simbol non-terminal. Hal lain yang juga berkaitan erat dengan proses parsing adalah kamus atau leksikon yang digunakan. Dalam leksikon disimpan daftar kata yang dapat dikenali sebagai simbol terminal dalam grammar dan informasi yang diperlukan untuk tiap kata tersebut untuk proses parsing yang bersangkutan. *Bottom Up parser* yaitu mencari dari simbol – simbol terminal menuju ke arah pembentukan simbol awal S.

2.2.7.3 Interpretasi Semantik

Analisis semantik merupakan kelanjutan dari proses *scanning* dan *parsing*. Fungsi dari analisis semantik adalah untuk menentukan makna dari serangkaian instruksi yang terdapat dalam program sumber atau masukan dari penguna. Penganalisa semantik harus mampu menentukan aksi atau respon apa yang yang akan dilakukan terhadap instruksi yang diberikan [6].

2.2.8 Question Answering System (QA System)

QA system atau *question-answering system* memberikan kemampuan pada sebuah mesin untuk menginterpretasikan bahasa alami untuk melakukan dialog dengan pengguna hampir seperti dialog antara dua orang manusia dalam bahasa sehari-hari [4].

Dalam tingkatan bahasa yang dikemukakan oleh Noam Chomsky, *QA system* ini dapat digolongkan ke tingkatan bahasa tipe 0. Tingkatan bahasa itu sendiri terdiri dari empat buah tipe sebagaimana pada *hirarki Chomsky* berikut ini:

Tabel 2.2 Hirarki Chomsky [8]

Bahasa	Mesin otomata	Batasan aturan Produksi
Regular / Tipe 3	*Finite State Automata* (FSA) meliputi *Deterministic Finite Automata* (DFA) dan *Non-deterministic Finite Automata* (NFA)	α adalah sebuah simbol variabel. B maksimal memiliki sebuah simbol variabel yang bila ada terletak di posisi paling kanan
Bebas Konteks / *Context Free* / Tipe 2	*Push Down Automata* (PDA)	α berupa sebuah simbol variabel
Context Sensitive / Tipe 1	*Linier Bounded Automata*	$\|\alpha\| \leqslant \|\beta\|$
Unrestricted / *Phase Structure* / *Natural Language* / Tipe 0	Mesin Turing	Tidak ada batasan

2.2.9 Chat Bot

Chat bot adalah sebuah simulator percakapan yang berupa program komputer yang dapat berdialog dengan penggunanya dalam bahasa alami. Karena *chat bot* hanya sebuah program, dan bukan robot (*chat bot* tidak memiliki tubuh dan tidak memiliki mulut sehingga tidak dapat berbicara seperti manusia), maka yang dimaksud dengan dialog antara manusia sebagai pengguna dengan *chat bot* dilakukan dengan cara mengetik apa yang akan dibicarakan dan *chat bot* akan

memberikan respon. Orang yang membuat dan mengembangkan program *chat bot* disebut *bot master* [2].

Chat bot yang pertama adalah ELIZA yang dibuat pada tahun 1964 sampai 1966 oleh Professor Joseph Weizenbaum di MIT (*Massachusetts Institute of Technology*), dengan tujuan untuk mempelajari komunikasi *natural language* antara manusia dengan mesin. Eliza bertindak seolah-olah dia adalah seorang psikolog yang dapat menjawab pertanyaan-pertanyaan dari pasien dengan jawaban yang cukup masuk akal atau menjawabnya dengan pertanyaan balik.

2.2.9.1 Komponen Utama Chat Bot

Chat bot terdiri dari dua komponen utama yakni *bot program* dan *brain file*. *Bot program* merupakan program utama pada *chat bot* yang akan mengakses input dari pengguna, melakukan *parsing* dan kemudian membawanya ke *brain file* untuk kemudian diberikan respon. Adapun *bot program* sendiri terdiri dari komponen *scanner* dan *parser* [3].

Brain file merupakan otak dari *chat bot* itu sendiri yang menentukan bagaimana cara *chat bot* berpikir dan akan memberikan respon. *Brain file* biasanya berupa file *plain text*. *Brain file* berfungsi sebagaimana tabel informasi pada kompilator bahasa pemrograman tingkat tinggi. Di dalam *brain file* inilah disimpan semua kosakata, kepribadian, dan pengetahuan (*knowledge*) dari *chat bot*. Semakin banyak pengetahuan yang dimiliki *chat bot* maka akan semakin besar ukuran file dari *brain file* tersebut.

2.2.10 Basis Data (*Database*)

2.2.10.1 Pengertian Basis Data

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut [13].

2.2.10.2 Bahasa Basis Data

Bahasa Basis Data dibagi menjadi tiga yaitu :

1. Data *Definition Language* (DDL)

Struktur atau skema basis data yang menggambarkan atau mewakili desain basis data secara keseluruhan disepesifikasikan dengan bahasa khusus yaitu DDL. Dengan bahasa ini kita dapat membuat tabel *(Create Table)* baru, indeks, mengubah *table,* menentukan struktue penyimpanan *table* dan lainnya. Hasil dari kompilasi perintah DDL adalah kumpulan *table* yang disimpan dalam *file* khusus yang disebut kamus data *(data dictionary).*

2. *Data Manipulation Language* (DML)

Merupakan bahasa basis data yang berguna untuk melakukan proses modifikasi dan pengambilan data pada suatu basis data. Modifikasi yang dapat dilakukan adalah penambahan *(insert),* pembaharuan *(update),* Penghapusan *(delete).* DML merupakan bahasa yang bertujuan memudahkan pengguna untuk mengakses data sebagaimana direpresentasikan oleh model data. Ada dua jenis DML adalah sebagai berikut :

1. *Prosedural*

   Yang mengisyaratkan agar pengguna menentukan data apa yang diinginkan serta bagaimana cara mendapatkannya.

2. *Non-Prosedural*

   Yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

3. *Data Control Language* (DCL)

Merupakan sub bahasa SQL yang digunakan untuk melakukan pengontrolan data dan *server database*-nya.

2.2.10.3 *Database Management System (DBMS)*

Database management system adalah perangkat lunak yang menangani mekanisme pengaksesan database. Tujuan dari DBMS adalah menyediakan lingkungan yang mudah dan aman untuk pengguna dan perawatan *database*.

Fungsi DBMS terbagi kedalam enam bagian, adalah sebagai berikut :

1. Data *Definition*

   DBMS harus dapat mengolah pendefinisian data.

2. Data *Manipulation*

   DBMS harus dapat menangani permintaan dari pemakai untuk mengakses data.

3. Data *Security* dan *Integrity*

   DBMS harus dapat memeriksa security dan integritas data yang didefinisikan oleh DBA.

4. Data *Recovery* dan *Concurency*

   DBMS harus dapat menangani kegagalan-kegagalan pengaksesan database yang dapat disebabkan oleh kesalahan sistem, kerusakan disk dan lain sebagainya.

5. Data *Dictionary*

   DBMS harus dapat menyediakan data *dictionary* (kamus data).

6. *Performance*

   DBMS harus dapat menangani unjuk kerja dari semua fungsi seefisein mungkin.

2.2.11 Alat Pemodelan Sistem

2.2.11.1 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Diagram konteks merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan [12]. Proses tersebut diberi nomor nol. Semua entitas *eksternal* yang ditunjukkan pada diagram konteks beserta aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data (*storage*) dan tampak sederahana untuk diciptakan, begitu *entitaseksternal* serta aliran data-aliran data menuju dan dari sistem diketahui penganalisis dari wawancara dengan *user* dan sebagai hasil analisis dokumen.

2.2.11.2 *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut dan asal, tujuan dan penyimpanan dari data tersebut.Dalam mengembangkan suatu aliran data atau proses yang terjadi didalam sistem data flow diagram menggunakan simbol-simbol yang memiliki arti tersendiri dalam menerangkan. Berikut arti dari simbol-simbol pada data flow diagram [13]:

1. Eksternal Entity

   Eksternal Entity merupakan kesatuan (*Entity*) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang memberikan input output dari sistem.

2. Data Flow

   Data Flow mengatur diantara proses, simpan data dan kesatuan luar. Data flow menunjukan arus data yang dapat berupa masukan sistem atau hasil proses sistem.

3. Proses

   Untuk Physical Data Flow Diagram (PDFD) data dilakukan oleh orang, mesin atau komputer. Sedangkan untuk logical data flow diagram (LDFD) suatu proses hanya menunjukan proses dari komputer.

4. Penyimpanan Data

   Penyimpanan data (Data Store) merupakan tempat penyimpanan data. simpanan data dari DFD disimbolkan dengan sepasang garis horizontal parallel

2.2.11.3 Diagram E-R (*Entity Relationship Diagram*)

Entity Relationship Diagram adalah sebuah diagram yang menggambarkan model relasi antar rancangan data tersimpan. Model relasi ini diperlukan untuk menggambarkan struktur data dari relasi antar data (yang mungkin sangat komplek). Model data ERD dibentuk dari empat komponen dasar yaitu [13]:

a. Entitas

   Entitas adalah segala sesuatu yang ada dan yang dapat dibedakan. Entitas dapat berupa orang, tempat, benda, peristiwa atau konsep yang bisa memberikan atau mengandung informasi.

b. Atribut

   Setiap entitas memiliki atribut-atribut atau elemen data yang mencirikan entitas tersebut.

c. Kunci (*Key*)

   Sebuah atribut atau set atribut yang nilainya mengidentifikasikan entitas secara unik dalam set entitas.

d. Relasi

Relasi menunjukan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Hubungan antara entitas atau beberapa entitas jenis relasi yang ada di dalam basis data adalah sebagai berikut :

a. Relasi banyak ke satu (n-1)

Misalkan direpresentasikan dalam pencantuman atribut kunci dari entitas B (berderajat 1) ke himpunan entitas A (berderajat n), ini berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan B, tetapi tidak sebaliknya.

b. Relasi banyak ke banyak (n-m)

Artinya setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B dan demikian juga sebaliknya.

c. Relasi satu ke banyak (1-n)

Misalkan di representasikan dalam bentuk pencantuman atribut kunci dari himpunan entitas A (berderajat 1) ke himpunan entitas B (berderajat n), ini berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya.

d. Relasi satu ke satu (1-1)

Ini berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B dan begitu juga sebaliknya.

2.2.12 Internet

Internet dapat diartikan sebagai jaringan komputer luas dan besar yang mendunia, yaitu menghubungkan pemakai komputer dari suatu negara ke negara lain di seluruh dunia, dimana di dalamnya terdapat berbagai sumber daya informasi dari mulai yang statis hingga yang dinamis dan interaktif.

2.2.12.1 Fasilitas Internet

Berikut adalah layanan yang disediakan oleh internet yang berbasis pada protokol TCP/IP :

a. WWW (*World Wide Web*)

Layanan ini adalah layanan yang sering kita gunakan. Layanan WWW ini menggunakan protokol yang dinamakan protokol HTTP (*HyperText Tranfer Protocol*).

b. *File Transfer* (Pengiriman File)

FTP ini memungkinkan kita untuk mengirimkan atau menerima file ke atau dari komputer jaringan. Oleh karena masalah keamanan data, FTP seringkali memerlukan username dan password tertentu, meskipun banyak juga FTP server yang dapat melalui anonymous (pengguna biasa dan tanpa password, permintaan password dapat diisi dengan alamat email).

c. *Remote Login* (Telnet)

Telnet memungkinkan pengguna komputer dapat melakukan login ke dalam suatu komputer di dalam jaringan. Ketika kita melakukan telnet, secara tidak langsung kita telah menjadi pengguna yang sah dari komputer tersebut.

d. *Electronic Mail (Email)*

Layanan yang digunakan untuk mengirim dan menerima email. Untuk pengiriman email digunakan protokol SMTP *(Simple Mail Transfer Protocol*) dan untuk membaca *email* digunakan protokol POP3 (*Post Office Protocol* 3).

e. NFS (*Network File System*)

Sebuah pelayanan akses file-file jarak jauh yang memungkinkan klien-klien untuk mengakses file- file pada jaringan jarak jauh.

f. *Domain Name Server*

Daftar nama alamat yang digunakan pada internet untuk menjelaskan mengenai alamat asli host internet (berupa nomor IP mesin).

g. *Remote Execution*

Memungkinkan pengguna komputer untuk menjalankan suatu program di komputer yang lain.

2.2.13 Perangkat Lunak Yang Digunakan

2.2.13.1 *Apache Web Server*

Pada awal mulanya, *Apache* merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari *server web Netscape* (sekarang dikenal sebagai Sun *Java System Web Server*). Sejak April 1996 Apache menjadi *server web* terpopuler di internet. Pada Mei 1999, Apache digunakan di 57% dari semua web server di dunia. Pada November 2005 persentase ini naik menjadi 71%.

Apache adalah komponen server web dari paket perangkat lunak LAMP (Linux, Apache, MySQL, PHP/Perl/Bahasa Pemrograman Phython).

2.2.13.2 *HypeText 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. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML, HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web dan HTML kini merupakan standar Internet yang saat ini dikendalikan 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.

2.2.13.3 *Cascading Style Sheet (CSS)*

CSS (*Cascading Style Sheet)* adalah sebuah bahasa *style sheet* (lembar gaya) yang digunakan untuk mengatur tampilan dokumen yang ditulis dalam bahasa *markup*. *CSS* Level 1 (*CSS*1) diperkenalkan pada tahun 1995 oleh *World Wide Web Consortium* (W3C) dan dimaksudkan untuk mengatasi keterbatasan *HTML* dalam hal keleluasaan pengaturan desain dan tampilan pada sebuah dokumen *HTML*.

Setahun kemudian *CSS*1 diberi status rekomendasi penuh oleh W3C yang juga mengatur spesifikasi *CSS*. Saat ini ada tiga level *CSS*, yaitu *CSS* Level 1 (*Recommendation*), *CSS* Level 2 (*Recommendation*), dan *CSS Level 2 Revision 1 (Candidate Recommendation)*. Penggunaan *CSS* paling banyak untuk memformat halaman *web* yang ditulis dengan *HTML* dan X*HTML*. Walau demikian, *CSS* dapat dipergunakan untuk bahasa *markup* lain seperti *SVG* dan *XUL*.

2.2.13.4 PHP

PHP adalah bahasa pemrograman yang berfungsi untuk membuat *website* dinamis maupun aplikasi *web*. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan *database*, *file* dan *folder*, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah *website* . *Blog*, Toko *Online*, CMS, Forum dan *Website Social Networking*adalah contoh aplikasi *web* yang bisa dibuat oleh PHP [14].

PHP adalah bahasa *scripting*, bukan bahasa *tag-based* seperti HTML. PHP termasuk bahasa yang *cross-platform*, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam *file plain text* (teks biasa).

Database yang didukung PHP adalah Oracle, Adabas-D, Sybase, FilePro, mSQL, Velocis, MySQL, Informix,Solid, dBase, ODBC, Unix dbm dan PostgreSQL. Untuk mencoba PHP , tidak perlu menggunakan komputer berkelas *server*. Hanya dengan sebuah komputer biasa, bisa mempelajari dan mempraktikkan PHP. Kelebihan PHP dibandingkan dengan pemrograman lainnya antara lain :

1. Bahasa pemrograman PHP merupakan sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Server web yang mendukung PHP dapat ditemukan di manamana, mulai dari IIS sampai dengan apache, 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 referensinya banyak

2.2.13.5 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (DBMS) yang *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL) [15].

Terdapat beberapa API tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk dapat mengakses basis data MySQL antara lain: bahasa pemrograman C, C++, C#, bahasa pemrograman Eiffel, bahasa pemrograman Smalltalk, bahasa pemrograman Java, bahasa pemrograman Lisp, Perl, PHP, bahasa pemrograman Python, Ruby, REALbasic dan Tcl. Sebuah antarmuka ODBC memanggil MyODBC yang memungkinkan setiap bahasa pemrograman yang mendukung ODBC untuk berkomunikasi dengan basis data MySQL. Kebanyakan kode sumber MySQL dalam ANSI C.

Untuk melakukan administrasi dalam basis data MySQL, dapat menggunakan modul yang sudah termasuk yaitu command-line (perintah: mysql dan mysqladmin). Juga dapat diunduh dari situs MySQL yaitu sebuah modul berbasis grafik (GUI): MySQL Administrator dan MySQL Query Browser. Selain itu terdapat juga sebuah perangkat lunak gratis untuk administrasi basis data MySQL berbasis web yang sangat populer yaitu phpMyAdmin. Untuk perangkat lunak untuk administrasi basis data MySQL yang dijual secara komersial antara lain: MySQL front, Navicat dan EMS SQL Manager for MySQL.

2.2.13.6 Keistimewaan *MySQL*

MySQL memiliki beberapa keistimewaan, antara lain [15]:

1. *Portabilitas* MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka, MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multiuser*, MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data, MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi, MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan, MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan, MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) 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. Konektivitas, MySQL dapat melakukan koneksi dengan *clients* menggunakan protokol TCP/IP, Unix soket (UNIX), atau *Named Pipes* (NT).
10. Lokalisasi, MySQL dapat mendeteksi pesan kesalahan pada clients dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka, MySQL memiliki antar muka *(interface)* terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API *(Application Programming Interface*).
12. Clients dan Peralatan, MySQL dilengkapi dengan berbagai peralatan (*tool*)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. Struktur tabel, MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.2.13.7 *Adobe Macromedia Dreamveaver*

Adobe Macromedia Dreamweaver merupakan salah satu *software* dari kelompok macromedia yang banyak digunakan untuk mendesain website. Adapun *Adobe Macromedia Dreamweave* itu sendiri adalah sebuah HTML editor professional untuk mendesain secara visual dalam mengolah halaman *website*. *Dreamweaver* memiliki performa yang lebih baik dan memiliki tampilan yang memudahkan anda untuk membuat website, baik dalam jendela desain maupun

dalam jendela kode rumus, *Dreamwaver* didukung dengan pemakaian yang praktis dan standar dan juga didukung untuk pengembangan pengguna *CSS, XML dan RSS*, dan kemudahan-kemudahan lain yang diperlukan [16].

Adobe Macromedia Dreamweaver merupakan software yang digunakan oleh desainer maupun web programmer dalam mengembangkan websitenya, hal ini disebabkan ruang kerja, fasilitas dan kemampuan *Macromedia Dreamweaver* yang mampu mengingkatkan produktivitas dan efektivitas dalam mendesain maupun membangun sebuah *website.*

2.2.14 *Body Shaping*

Body shaping atau pembentukan tubuh adalah program pembentukan badan yang dilakukan dengan cara latihan beban, pengaturan pola makan, suplementasi dan istirahat yang cukup untuk mendapatkan bentuk tubuh padat berotot tanpa lemak sesuai dengan yang diinginkan. Bagi para wanita yang menjalankan program ini akan mendapatkan tubuh yang kencang dan langsing karena perbedaan hormon yang tidak memungkinkan untuk wanita menjadi berotot dan kekar seperti pria.

Dalam pembentukan program *Body Shaping*, Anda perlu memperhatikan 4 poin [11].

1. Pola Makan yang disiplin, artinya makan 5-6x sehari, tinggi protein, rendah karbohidrat dan rendah lemak. Tujuannya untuk meningkatkan metabolisme tubuh dan menajamkan massa otot kita.

2. Pola Latihan yang rutin, artinya menu latihan dalam seminggu rata untuk semua bagian otot kita. Contohnya kita ingin membentuk otot perut tapi tetap perlu memperhatikan latihan kaki. Selain itu, usahakan selalu mendahulukan otot besar (dada, punggung, bahu, kaki) baru otot kecil (lengan, perut dan betis) saat melatih dalam 1 sesi.
3. Pola Istirahat yang cukup, artinya kita merusak otot dalam sesi latihan dan istirahat memperbaiki otot kita menjadi lebih besar dan kuat. Sehingga kualitas dan kuantitas istirahat yang tinggi akan sangat membantu kita mencapai tujuan.
4. Pola Suplemen yang jitu, artinya kita mesti jeli memilih jenis suplemen yang akan membantu program kita. Contohnya dalam program *Body Shaping*, suplemen yang dibutuhkan adalah Protein (bisa Amino ataupun Whey Protein) dan Creatine Ethyl Esther (Kre Alkalyn / Beta K)