

IMPLEMENTASI *RULE-BASED EXPERT SYSTEMS* DALAM MENDETEKSI KERUSAKAN SISTEM JARINGAN KOMPUTER DENGAN METODE *BACKWARD CHAINING*

Saiful Nur Arif

Program Studi Sistem Komputer, STMIK Triguna Dharma

saiful.nurarif@gmail.com

ABSTRAK: Perkembangan teknologi jaringan komputer dewasa ini semakin pesat seiring dengan kebutuhan masyarakat akan layanan yang memanfaatkan jaringan komputer. Pada saat ini manfaat dari jaringan komputer sudah sangat banyak dirasakan, apalagi dalam dunia komunikasi yang serba cepat ini, jaringan komputer sangat berperan dalam kegiatan pendistribusian informasi yang cepat tersebut. Seluruh komponen yang tergabung dalam jaringan komputer tersebut mampu saling mendukung untuk menghasilkan suatu sistem yang kokoh dan handal untuk melayani setiap permintaan informasi yang dibutuhkan oleh pengguna.

Mendiagnosis permasalahan perangkat yang tersambung jaringan merupakan pekerjaan yang harus dilakukan oleh administrator jaringan. Pekerjaan ini memerlukan ketelitian dan kesabaran yang tinggi agar di dapat hasil yang baik.

Untuk membantu menyelesaikan masalah di atas, maka diperlukan kehadiran sistem pakar yang diyakini mampu mendeteksi kerusakan jaringan komputer secara tepat, cepat dan akurat. Dengan mengimplementasikan *rule-based expert systems* dalam mendeteksi kerusakan sistem jaringan komputer dengan *metode backward Chaining* diharapkan dapat membantu para teknisi dalam mendeteksi kerusakan jaringan komputer.

Kata Kunci: *expert system, backward chaining, networking*

A. PENDAHULUAN

Dewasa ini kemajuan di bidang ilmu jaringan komputer melaju sangat pesat dan telah memberikan manfaat dan kontribusi bagi kehidupan manusia di berbagai bidang, di antaranya untuk sosialisasi maupun transfer data informasi akademik, Sharing Jaringan WIFI Internet, mentransfer uang dari mesin ATM di tempat yang jauh (*telebanking*), pembayaran rekening air, listrik dan telephone (*telebanking*), Internet (surat elektronik/e-mail, pertukaran file/ftp) dan sebagainya. Dalam pemanfaatan jaringan tersebut, tentunya haruslah diiringi dengan berbagai asesories lain yang mampu mendukung kehandalan dari seluruh fasilitas

yang ada. Kehandalan dari seluruh fasilitas yang ada tentunya juga seiring dengan segala kemungkinan-kemungkinan kesalahan dan kerusakan yang dapat mengganggu dari kelancaran pemakaian serta kesinambungan dari pemakaian jaringan tersebut.

Mendiagnosis permasalahan perangkat yang tersambung jaringan merupakan pekerjaan yang harus dilakukan oleh administrator jaringan. Pekerjaan ini memerlukan ketelitian dan kesabaran yang tinggi agar di dapat hasil yang baik. Komputer yang terhubung jaringan sering kali mengalami gangguan maupun kerusakan baik dari sisi hardware atau software. Hal ini disebabkan oleh banyaknya pengguna komputer yang terhubung dalam sistem

jaringan. Jaringan komputer sangat rawan terhadap gangguan atau kerusakan dikarenakan banyak sekali faktor yang dapat menyebabkan terjadi gangguan atau kerusakan pada jaringan tersebut.

B. SISTEM PAKAR (*EXPERT SYSTEM*)

Salah satu bagian dari ilmu komputer yang berkembang saat ini adalah kecerdasan buatan (*artificial intelligence*) atau disingkat AI. Kecerdasan buatan diciptakan agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia, untuk itu komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Berbagai bidang yang menggunakan kecerdasan buatan antara lain sistem pakar (*expert system*), permainan komputer (*games*), jaringan saraf tiruan (*artificial neural network*) dan robotika.

Sistem Pakar (*Expert System*) merupakan salah satu contoh kecerdasan buatan yang memanfaatkan komputer sebagai basis pengetahuan yang mempermudah dalam menghasilkan informasi (Kusrini, 2006).

Berbagai masalah yang dapat dipecahkan melalui sistem pakar pada umumnya berupa masalah yang sulit atau bahkan tidak dapat dipecahkan melalui pemrograman konvensional, karena komputasi konvensional pemrosesan datanya berdasarkan algoritmik dan sifat inputnya harus lengkap serta pencarian data biasanya didasarkan kepada algoritma. Sedangkan komputasi kecerdasan buatan pemrosesan data mengandung konsep-konsep simbolik dan sifat inputnya boleh tidak lengkap serta pencarian data kebanyakan bersifat *heuristik*. Untuk itu dalam penyelesaian masalah yang sulit sangat dibutuhkan sistem pakar.

Pemrograman dari sistem pakar dibuat berdasarkan suatu himpunan dan aturan yang menganalisis informasi mengenai suatu masalah spesifik serta analisis matematis dari masalah tersebut, tergantung dari disain sistem pakar yang akan dipakai. Sistem pakar juga mampu merekombinasikan suatu rangkaian tindakan

pengguna untuk dapat menerapkan koreksi. Sistem pakar memanfaatkan kemampuan penalaran (*reasoning*) untuk mendapatkan suatu kesimpulan yang ingin dicapai. Dengan memahami cara kerja sistem pakar yang meniru cara manusia dalam memecahkan suatu masalah spesifik. Dari dasar pemikiran tersebut di atas maka timbul suatu ide untuk membuat suatu sistem pakar yang diharapkan dapat membantu memecahkan masalah.

Sistem pakar adalah sebuah program komputer yang didesain untuk memodelkan/membuat simulasi kemampuan seorang *expert* dalam memecahkan suatu masalah. Dua kapabilitas penting dari seorang *expert* yang dicoba untuk dimodelkan pada *expert system* adalah pengetahuan (*knowledge*) dan konsep berpikir (*reasoning*) dari sang *expert*. Sistem pakar memiliki dua modal penting yaitu *knowledge base* dan *inference engine* yang digunakan untuk berinteraksi dengan *user* (*user interface*) untuk dapat menghasilkan kedua kapabilitas tersebut. *Knowledge base* berisi pengetahuan sangat spesifik yang disediakan oleh seorang pakar untuk memecahkan masalah tertentu. Contohnya: *knowledge* dari seorang ahli dibidang teknologi jaringan komputer untuk menganalisa kerusakan jaringan komputer tertentu. *Knowledge planning* disediakan oleh seorang konsultan IT. *Inference engine* adalah "engine" pemroses *knowledge* yang dimodelkan berdasarkan konsep berpikir dari *expert* penyedia *knowledge*. *Inference engine* beserta informasi yang didapat dari sebuah masalah, berpasangan dengan *knowledge* yang disimpan pada *knowledge base*, berusaha untuk mencari/menarik kesimpulan, jawaban dan rekomendasi guna memecahkan masalah tersebut. Tujuan membangun sebuah *expert system* adalah menggantikan seorang *expert* dan Membantu seorang *expert* dalam mengerjakan pekerjaan rutin dan tugas-tugas yang sulit (Rolston, 1988).

Saat ini jaringan komputer semakin dibutuhkan karena tidak dapat dipungkiri lagi dapat memberikan manfaat dan kontribusi yang cukup besar diberbagai bidang. Tetapi pengguna jaringan komputer sering dihadapkan dengan

berbagai permasalahan yang mengakibatkan terhambatnya suatu pekerjaan terutama yang berhubungan langsung dengan komunikasi data dan transfer data, hal ini disebabkan jaringan komputer yang digunakan mengalami *trouble* atau rusak, yang mungkin kerusakan tersebut berasal dari komponen *hardware* atau pun *software*-nya. Dan yang menjadi masalah lagi adalah penanganan jaringan komputer oleh bagian teknisi atau *maintenance* (pemeliharaan) sangat lambat, hal ini mengakibatkan suatu pekerjaan tidak selesai tepat pada waktunya sehingga tujuan perusahaan atau instansi tidak tercapai.

Untuk membantu menyelesaikan masalah di atas, maka diperlukan kehadiran sistem pakar yang diyakini mampu mendeteksi kerusakan jaringan komputer secara tepat, cepat dan akurat. Dengan mengimplementasikan ***rule-based expert systems*** dalam mendeteksi kerusakan sistem jaringan komputer dengan metode ***backward Chaining*** diharapkan dapat membantu para teknisi dalam mendeteksi kerusakan jaringan komputer.

Sistem pakar merupakan bagian dari *Artificial Intelligence* yang dalam keberadaannya memiliki atribut atau ciri-ciri yang mencerminkan sistem tersebut. Adapun atribut atau ciri-ciri dari sistem pakar adalah sebagai berikut :

1. Berpikir dan bernalar
2. Memakai penalaran untuk memecahkan persoalan
3. Menyerap dan menerapkan pengetahuan
4. Memperlihatkan kreativitas dan imajinasi
5. Bekerja dengan situasi yang kompleks dan membingungkan
6. Melakukan tanggapan dengan cepat dan berhasil terhadap situasi baru
7. Mengenali elemen-elemen yang relatif penting dalam suatu situasi
8. Menangani informasi yang rancu, tak lengkap atau salah.

Kelebihan Sistem Pakar

Kelebihan sistem pakar terhadap manusia adalah:

1. Tidak perlu istirahat

2. Dapat diperbanyak (semaunya)
3. Tidak mati dengan membawa keahliannya
4. Tidak terpengaruh oleh kelelahan
5. Tidak subyektif

Kelemahan Sistem Pakar

Di samping memiliki kelebihan, sistem pakar juga memiliki kelemahan di antaranya :

1. Biaya yang diperlukan untuk membuat dan memelihara cukup mahal.
2. Sulit dikembangkan, hal ini tentu saja erat hubungannya dengan ketersediaan pakar dibidangnya.
3. Sistem pakar tidak 100% bernilai benar.

Sistem pakar dikembangkan lebih lanjut dengan alasan :

1. Dapat menyediakan kepakaran setiap waktu dan diberbagai lokasi.
2. Secara otomatis mengerjakan tugas rutin membutuhkan seorang pakar.
3. Seorang pakar mahal.

Manfaat Sistem Pakar

Dengan banyaknya kemampuan dan manfaat yang diberikan oleh Sistem Pakar, di antaranya (*Diktat Mata Kuliah Kecerdasan Buatan Ir. Balza Achmad, M.Sc.E. 8*)

- a. Meningkatkan output dan produktivitas, karena Sistem Pakar dapat bekerja lebih cepat dari manusia.
- b. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
- c. Mampu menangkap kepakaran yang sangat terbatas.
- d. Dapat beroperasi di lingkungan yang berbahaya.
- e. Memudahkan akses ke pengetahuan.
- f. 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.
- g. Meningkatkan kapabilitas sistem terkomputerisasi yang lain. Integrasi Sistem Pakar dengan sistem komputer lain

- membuat lebih efektif, dan mencakup lebih banyak aplikasi .
- h. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional, Sistem Pakar dapat bekerja dengan informasi 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.
 - i. Mampu menyediakan pelatihan. Pengguna pemula yang bekerja dengan Sistem Pakar akan menjadi lebih berpengalaman. Fasilitas penjelas dapat berfungsi sebagai guru.
 - j. Meningkatkan kemampuan problem solving, karena mengambil sumber pengetahuan dari banyak pakar.
 - k. Meniadakan kebutuhan perangkat yang mahal.
 - l. Fleksibel.

Keterbatasan Sistem Pakar

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

- a. Pengetahuan yang hendak diambil tidak selalu tersedia.
- b. Kepakaran sangat sulit diekstrak dari manusia.
- c. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.
- d. Adalah sangat sulit bagi seorang pakar untuk mengabstraksi atau menjelaskan langkah mereka dalam menangani masalah
- e. Pengguna Sistem Pakar mempunyai batas kognitif alami, sehingga mungkin tidak bisa memanfaatkan sistem secara maksimal.
- f. Sistem Pakar bekerja baik untuk suatu bidang yang sempit.
- g. Banyak pakar yang tidak mempunyai jalan untuk mengecek apakah kesimpulan mereka benar dan masuk akal.
- h. Istilah dan jargon yang dipakai oleh pakar dalam mengekspresikan fakta seringkali

terbatas dan tidak mudah dimengerti oleh orang lain.

- i. Pengembangan Sistem Pakar seringkali membutuhkan perekayasa pengetahuan (knowledge engineer) yang langka dan mahal.

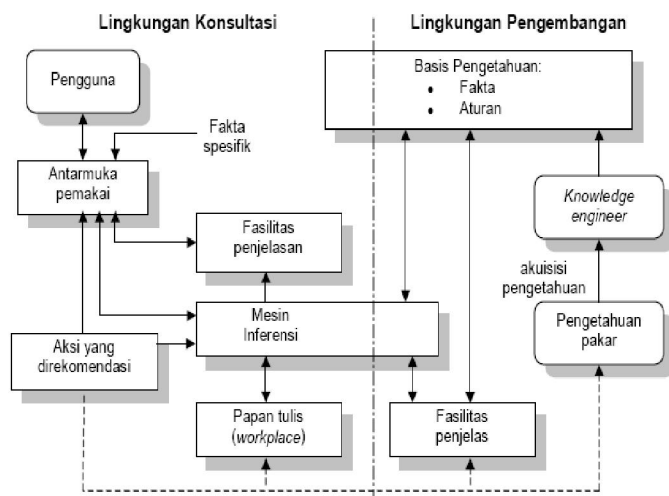
C. KOMPONEN SISTEM PAKAR

Secara umum, Sistem Pakar biasanya terdiri atas beberapa komponen yang masing-masing berhubungan seperti terlihat pada Gambar 1 Basis Pengetahuan, berisi pengetahuan yang dibutuhkan untuk memahami, memformulasi, dan memecahkan masalah. Basis pengetahuan tersusun atas 2 elemen dasar:

1. Fakta, misalnya: situasi, kondisi, dan kenyataan dari permasalahan yang ada, serta teori dalam bidang itu.
2. 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. Kerja mesin inferensi meliputi:

1. Menentukan aturan mana akan dipakai.
2. Menyajikan pertanyaan kepada pemakai, ketika diperlukan.
3. Menambahkan jawaban ke dalam memori Sistem Pakar.
4. Menyimpulkan fakta baru dari sebuah aturan.
5. Menambahkan fakta tadi ke dalam memori.



Gambar 1. Struktur Skematis sebuah Sistem Pakar

Papan Tulis (*Blackboard/Workplace*), adalah memori/lokasi untuk bekerja dan menyimpan hasil sementara. Biasanya berupa sebuah basis data.

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*).

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:

1. "Mengapa pertanyaan tersebut anda tanyakan?"
2. "Seberapa yakin kesimpulan tersebut diambil?"
3. "Mengapa alternatif tersebut ditolak?"
4. "Apa yang akan dilakukan untuk mengambil suatu kesimpulan?"
5. "Fakta apalagi yang diperlukan untuk mengambil kesimpulan akhir?"

Sistem Penghalusan Pengetahuan (*Knowledge Refining System*). Seorang pakar mempunyai sistem penghalusan pengetahuan,

artinya, mereka bisa menganalisa sendiri performa mereka, belajar dari pengalaman, serta meningkatkan pengetahuannya untuk konsultasi berikutnya. Pada Sistem Pakar, swa-evaluasi ini penting sehingga dapat menganalisa alasan keberhasilan atau kegagalan pengambilan kesimpulan, serta memperbaiki basis pengetahuannya.

D. SISTEM PAKAR MELAKUKAN INFERENSI

Ada dua strategi pencarian dasar yang bisa digunakan oleh mesin inferensi dalam mencari kesimpulan untuk mendapatkan solusi bagi permasalahan yang dihadapi sistem pakar, yaitu runut maju (*forward chaining*) dan runut balik (*backward chaining*). Berikut ini penjelasan mengenai kedua strategi pencarian tersebut :

1. Metode Runut Maju (*forward chaining*)

Runut maju merupakan strategi pencarian yang memulai proses pencarian dari sekumpulan data atau fakta, dari data-data tersebut dicari suatu kesimpulan yang menjadi solusi dari permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang premisnya sesuai dengan data-data tersebut, kemudian dari kaidah-kaidah tersebut diperoleh suatu kesimpulan. Runut maju memulai proses pencarian dengan data sehingga strategi ini disebut juga *data-driven*.

Dicontohkan pada Tabel 1 di bawah ini terlihat 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan hanya *A & F* (artinya *A* dan *F* bernilai benar). Ingin dibuktikan apakah *K* bernilai benar (hipotesis : *K*) ?

Tabel 1. Contoh Sistem Perantaraan Maju

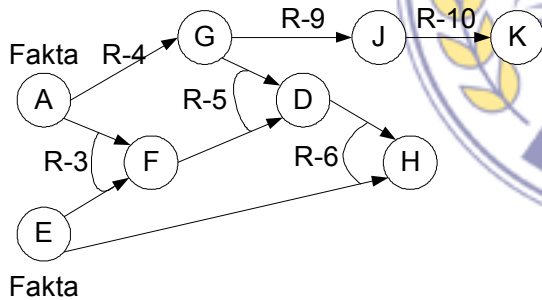
NO	ATURAN
R-1	IF A & B THEN C
R-2	IF C THEN D
R-3	IF A & E THEN F
R-4	IF A THEN G
R-5	IF F & G THEN D
R-6	IF G & E THEN H
R-7	IF C & H THEN I
R-8	IF I & A THEN J
R-9	IF G THEN J
R-10	IF J THEN K

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*. Contoh sistem runut balik diberikan pada Tabel 2.

Tabel 2. Contoh Sistem Perantaraan Mundur

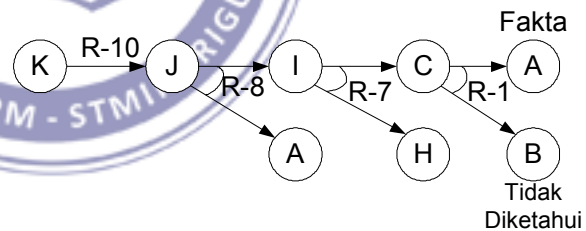
Aturan	Fakta Baru
R-3	F
R-4	G
R-5	D
R-6	H
R-9	J
R-10	K (terbukti)

Penyelesaian dengan *Forward Chaining* pada Gambar 2 berikut.

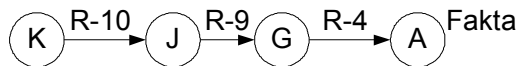


Gambar 2. Penyelesaian Forward Chaining

Penyelesaian dengan *Backward Chaining* pada Gambar 3 berikut.



Pertama : Gagal



Kedua : Sukses

Gambar 3. Penyelesaian Backward Chaining

2. Runut Balik (*backward chaining*)

Runut balik 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,

3. Algoritma Runut Balik

Penggunaan strategi pencarian runut balik untuk membangun mesin inferensi memerlukan suatu algoritma tertentu sehingga bisa diimplementasikan menggunakan bahasa

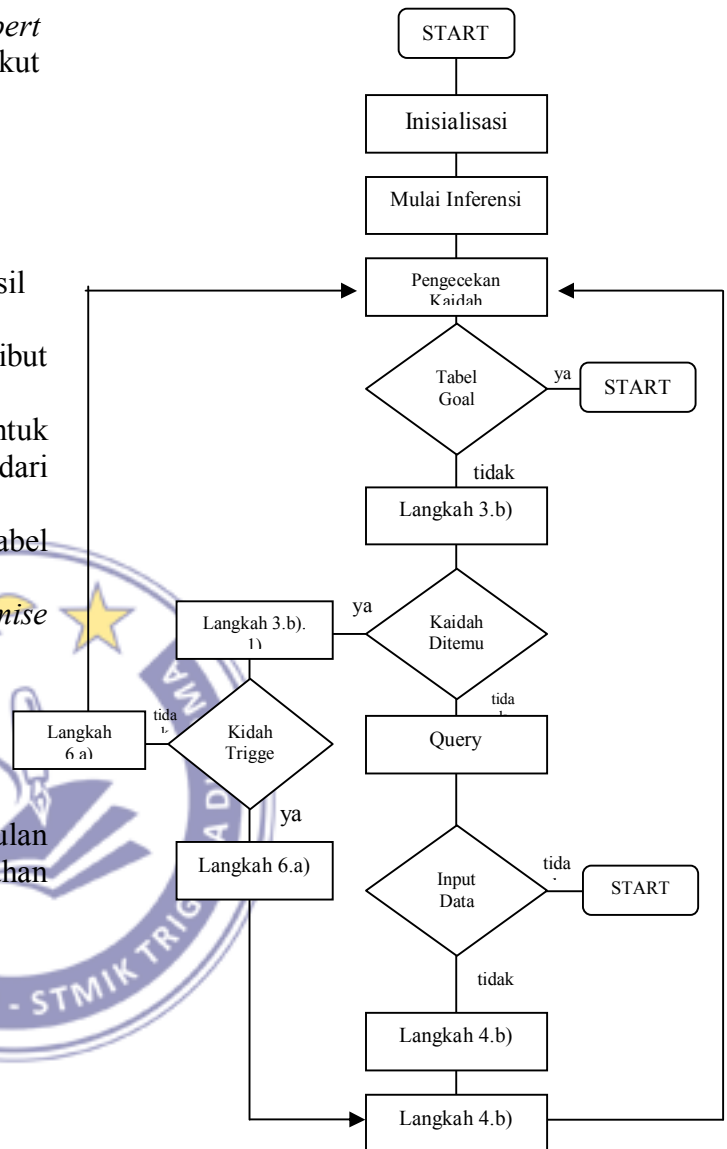
pemrograman. Salah satu algoritma runut balik yang diambil dari buku *Introduction To Expert Systems : The Development and Implementation Of Rule-Based Expert Systems (Ignizio, 1991)* adalah sebagai berikut (Gambar 4):

3.1. 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.
- b) Semua klausa premis dalam tabel *Rule/Premise Status* diberi status *free*.
- c) Semua kaidah dalam tabel *Rule/Premise Status* diberi status *active*.

3.2. 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*.



Gambar 4. Algoritma Backward Chaining dalam Bentuk Diagram

E. PENYELESAIAN MASALAH

1. Analisa Masalah

Untuk memahami permasalahan yang dihadapi maka sebelumnya dilakukan proses awal yaitu dengan melakukan analisa permasalahan dengan menentukan permasalahan. Tahapan ini sangat dibutuhkan

untuk keakuratan analisa dalam pengembangan sistem pakar tersebut. Berdasarkan permasalahan yang telah dirumuskan sebelumnya, maka dapat ditentukan beberapa pendekatan untuk menyelesaikan permasalahan tersebut. Solusi untuk menyelesaikan permasalahan tersebut di atas dapat diuraikan dengan pendekatan-pendekatan sebagai berikut :

- a. Mempelajari dan melakukan studi kasus mengenai kerusakan jaringan jaringan komputer yang sering terjadi, mulai dari jenis kerusakan, gejala-gejala yang terjadi hingga ke solusi penyelesaian masalah kerusakan jaringan dan pemeliharaan jaringan komputer yang penting dilakukan.
- b. Memahami sistem kerja jaringan komputer mulai dari mengaktifkan komputer, perakitan dan pemasangan komponen (*device*) yang terhubung ke jaringan komputer, instalasi *software* sehingga jaringan komputer dapat digunakan.
- c. Membangun sistem informasi berupa sistem pakar yang menyediakan *referensi* mengenai tata cara melakukan pemeliharaan jaringan dan perawatan jaringan komputer untuk mengurangi terjadinya kerusakan pada jaringan komputer.
- d. Membangun sistem pakar yang dapat mendeteksi permasalahan pada jaringan komputer sekaligus memberikan solusi penyelesaian masalah jaringan, yang dapat membantu para teknisi maupun *user* dalam melakukan penanganan berbagai permasalahan kerusakan jaringan komputer.
- e. Pengembangan sistem pakar yang dapat membantu memberikan keputusan yang secara tepat dan akurat, sehingga penanganan kerusakan jaringan komputer akan lebih efektif.

2. Analisa Sistem

Kajian analisis yang diperoleh dari beberapa penjelasan dengan berbagai permasalahan dalam mengidentifikasi masalah yang dikombinasikan dengan pendekatan solusi, maka dapat diperoleh beberapa kesimpulan sebagai berikut :

- a. Pengembangan Sistem pakar yang harus menyediakan *referensi* tentang bagaimana tata-cara merawat dan memelihara jaringan komputer serta memberikan petunjuk bagaimana cara memperbaiki jaringan komputer bila terjadi kerusakan.
- b. Diperlukannya pengembangan sistem pakar untuk dapat membantu para teknisi dalam menyelesaikan kerusakan jaringan komputer atau pun pengguna jaringan komputer lainnya, sehingga dapat memperbaiki jaringan komputer sendiri tanpa harus memanggil teknisi atau membawanya ke tempat service komputer.
- c. Proses pengambilan kesimpulan, sistem pakar akan memberikan beberapa pertanyaan mengenai berbagai permasalahan yang dihadapi dan kemudian pengguna sistem memberikan jawaban dari pertanyaan tersebut. Dari hasil jawaban tersebut dapat diidentifikasi solusi langkah-langkah apa saja yang harus dilakukan untuk mengetahui kerusakan pada jaringan komputer.
- d. Setelah ditemukan adanya permasalahan dari kerusakan jaringan komputer, maka sistem pakar akan memberikan solusi bagaimana cara memperbaiki kerusakan jaringan komputer tersebut.

3. Perancangan Secara Umum

Di dalam merancang suatu perangkat lunak diperlukan tahapan-tahapan untuk menyelesaikan permasalahan yang dihadapi. Dan juga harus diketahui bagaimana kita memberikan solusi yang baik sesuai dengan permasalahan atau keluhan yang akan diberikan.

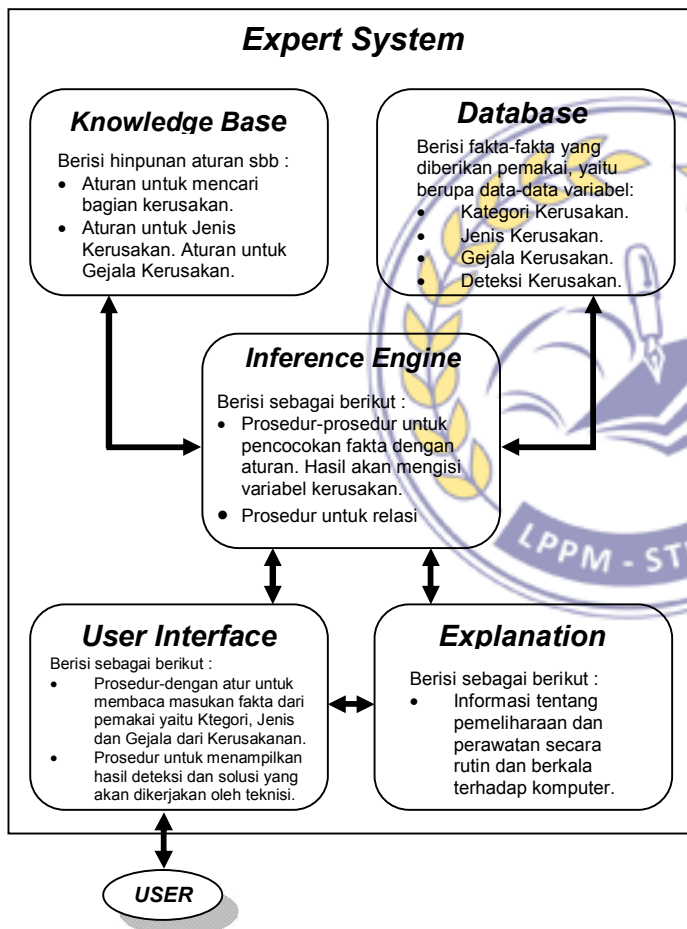
User interface merupakan bagian dari suatu software yang menyediakan sarana untuk pemakai agar dapat berkomunikasi dengan sistem. *User interface* akan menghasilkan pertanyaan-pertanyaan yang berhubungan dengan sepeda motor, selain itu juga menyediakan menu pilihan untuk memasukkan informasi kerusakan serta solusi yang lain kedalam sistem.

Penerapan metode *backward chaining* dimulai dari macam-macam kerusakan, jenis

kerusakan, gejala kerusakan, ciri kerusakan dan diagnosa. Dengan menggunakan metode ini akan ditelusuri dengan cara penalaran mundur mulai dari macam kerusakan sampai diagnosa kerusakan tersebut.

4. Desain Arsitektur Sistem

Dengan mengacu pada sistem pakar berbasis aturan yang telah dibahas pada bab sebelumnya dan dengan melakukan beberapa penyederhanaan pada beberapa komponen, maka arsitektur sistem didisain seperti terlihat pada Gambar 5.



Gambar 5. Disain Arsitektur Deteksi Kerusakan Jaringan

5. Knowledge Based

Knowledge base berisi himpunan aturan atau rule-rule untuk mencari aturan, mencari macam-macam kerusakan, jenis-jenis

kerusakan, gejala kerusakan, cirri-ciri kerusakan dan diagnosa kerusakan.

Contoh aturan-aturan adalah sebagai berikut :

IF Kategori Kerusakan :
Pengkabelan (Media Komunikasi)

AND Jenis kerusakan :
Jaringan Tidak Terdeteksi

AND Gejala :
Icon Network disudut kanan bawah layar bertanda silang dan Muncul pesan "A Network Cable is Unplugged" sehingga tidak dapat melakukan *sharing* data

AND Ciri kerusakan :
Kabel Jaringan Short

THEN Diagnosanya :
Lepaskan kabel yang terhubung antara *Hub* dan *CPU* selanjutnya gunakan alat *Tester Jaringan* yaitu dengan cara memasukkan ke dua buah ujung konektor pada alat tersebut dan jika hasil alat tester tersebut menyatakan kabel tersebut terbakar maka segera ganti kabel tersebut. Selanjutnya lakukan pemasangan ulang dengan mengukur panjang kabel dengan memasangkan konektor dikedua ujung kabel tersebut. Pada saat pemasangan Kabel UTP dan konektor RJ-45, yang perlu diperhatikan adalah pemasangan wayar kabel pada konektor RJ-45 harus disesuaikan dengan warna wayar dengan urutan pin pada konektor RJ-45. Setelah selesai melakukan installasi kabel lakukan uji coba kabel dengan menggunakan alat tester jaringan yang berfungsi untuk menguji kehandalan instalasi kabel dan konektor tersebut.

Aturan atau rule diatas menunjukkan bahwa **JIKA** Kategori kerusakan pada Media Komunikasi Jaringan yaitu pengkabelan **DAN** jenis kerusakan jaringan yang jaringan tidak

terdeteksi **DAN** Icon Network disudut kanan bawah layar bertanda silang dan Muncul pesan "A Network Cable is Unplugged" **DAN** ciri kerusakan adalah kabel terbakar **MAKA** diagnosa kerusakan adalah Lepaskan kabel yang terhubung antara *Hub* dan *CPU* selanjutnya gunakan alat *Tester Jaringan* yaitu dengan cara memasukkan ke dua buah ujung konektor pada alat tersebut dan jika hasil alat tester tersebut menyatakan kabel tersebut terbakar maka segera ganti kabel tersebut. Selanjutnya lakukan pemasangan ulang dengan mengukur panjang kabel dengan memasang konektor di kedua ujung kabel tersebut. Pada saat pemasangan Kabel UTP dan konektor RJ-45, yang perlu diperhatikan adalah pemasangan wayar kabel pada konektor RJ-45 harus disesuaikan dengan warna wayar dengan urutan pin pada konektor RJ-45. Setelah selesai melakukan instalasi kabel lakukan uji coba kabel dengan menggunakan alat tester jaringan yang berfungsi untuk menguji kehandalan instalasi kabel dan konektor tersebut.

6. Data Base

Dalam perancangan suatu sistem dibutuhkan suatu database yang digunakan untuk menyimpan data dan informasi lain yang diperlukan dalam sistem. Dalam hal ini nama database yang dirancang adalah **dbExpertSystem.mdb**.

Data base berisi tentang fakta-fakta yang dibutuhkan pemakai yaitu berupa data-data variabel dari : tabel password, tabel macam-macam kerusakan, tabel jenis kerusakan, tabel gejala kerusakan, tabel ciri-ciri kerusakan dan tabel diagnosa kerusakan. Adapun contoh tabel data base adalah sebagai berikut (Tabel 3 dan 4).

Tabel 3. Tabel Kategori Kerusakan

Field Nama	Type	Width	Description
NoKategori	Text	5	Nomor Kategori Device
Kategori	Text	70	Kategori Kerusakan

Tabel 3 (Tabel Kategori) didefinisikan dua buah *field* yaitu NoKategori yang memuat 5 karakter dan Kategori yang memuat 70 karakter, data yang diinputkan ke dalam tabel ini adalah bagian-bagian *device* atau sistem jaringan komputer yang sering mengalami kerusakan.

Tabel 4. Tabel Jenis Kerusakan

Field Nama	Type	Width	Description
NoJenis	Text	5	Nomor Jenis Kerusakan
Jenis	Text	70	Jenis Kerusakan
Gejala	Memo	-	Uraian Gejala Kerusakan

Tabel 4 (Tabel Jenis) didefinisikan tiga buah *field* yaitu NoJenis yang memuat 5 karakter, Jenis 7 karakter dan Gejala merupakan *field* memo yang berisi uraian mengenai gejala-gejala kerusakan jaringan komputer.

7. Inference Engine

Inference Engine merupakan bagian dari sistem pakar yang bertugas sebagai otak dalam menemukan solusi yang tepat dari banyaknya solusi yang ada. Kesesuaian fakta atau pernyataan pada inferensi *backward chaining* dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari dari fakta-fakta yang ada dalam basis pengetahuan.

Konsep *backward chaining* ini diterapkan pada bahasa pemrograman Microsoft Visual Basic 6.0. Proses ini dimulai dari pencarian solusi dari kesimpulan kemudian menelusuri fakta-fakta yang ada hingga menemukan solusi yang sesuai dengan fakta-fakta yang diberikan *user*. Metode *backward chaining* melaksanakan penalaran dengan pendekatan *goal driven* memulai titik pendekatannya dari *goal* yang akan dicari nilainya kemudian bergerak untuk mencari informasi yang mendukung *goal* tersebut.

8. Disain Rule (Aturan)

Perancangan *rule* pada sistem ini menggunakan metode *backward chaining*, metode ini memulai inferensi *goal* (tujuan). Fakta tentang aturan kategori, jenis dan ciri kerusakan komputer diperoleh dari database dan pengguna sistem memilih komponen komputer yang bermasalah dengan memasukkan jenis dan ciri kerusakan pada *interface* (antarmuka) pengguna. Di dalam sistem atau aplikasi ini data masing-masing komponen dan bagian-bagian yang bermasalah baik itu *software* maupun *hardware* (sistem komputer) disajikan dalam bentuk tipe struktur data yang diimplementasikan menggunakan *Development Tool Visual Basic*. Daftar aturan yang disajikan pada tabel berikut ini sebagai rancangan dari sistem ini hanya diperlihatkan sebagian dari rule yang ada, karena jumlahnya cukup banyak. Sebagai sampel dari komponen-komponen dari jaringan komputer yang akan dilakukan perbaikan dan perawatan dapat dilihat pada contoh Tabel 5 berikut ini.

Tabel 5. Tabel Kategori Kerusakan

Masalah/Gejala	Penyebab	Langkah pemeriksaan
Jaringan Komputer tidak terhubung yang disebabkan oleh kabel	Kabel tidak dihubungkan/kendor	Hubungkan/Kencangkan kabel ke HUB dan CPU
	Konektor RJ-45 Rusak	Lakukan penggantian Konektor RJ-45 dengan yang baru
	Kabel Jaringan Rusak/Short	Lakukan penggantian kabel baru dan selanjutnya lakukan pemasangan kembali
	Kesalahan pemasangan konektor RJ-45	Lakukan pemisahan wayar pada kabel dan selanjutnya pasang ke konektor sesuai posisi pin konektor.
	Plug/Socket rusak	Pindahkan ke socket HUB yang lain

9. Teknik Inferensi Engine

Berdasarkan teknik *inferensi* yang digunakan yaitu *Bakward chaining* atau *Goal Driven*, kinerja sistem dalam pencapaian kesimpulan didasarkan pada langkah berikut :

1. Memulai penalaran dari sekumpulan hotesa.
2. Menemukan ciri-ciri yang sesuai hipotesa menuju fakta-fakta yang mendukung hipotesa tersebut.
3. Begitu selanjutnya sampai pada kesimpulan atau *goal* tercapai.

Proses implementasi sistem *inferensi* yang menggunakan *Bakward chaining* dapat diuraikan pada Tabel 6 berikut ini dengan inisialisasi awal **Jaringan komputer tidak terkoneksi.**

Tabel 6. Database yang Mengandung Fakta

VARIABEL	FAKTA-FAKTA
A	Kabel tidak dihubungkan/kendor
B	Konektor RJ-45 Rusak
C	Kabel Jaringan Rusak/Short
D	Kesalahan pemasangan konektor RJ-45
E	Port Hub Rusak
W	Kabel Jaringan tidak berfungsi

Berdasarkan variabel-variabel yang telah dikelompokkan maka dapat diterapkan ke dalam aturan-aturan (*knowledge base*) yang ditampilkan pada Tabel 7 berikut ini.

Tabel 7. Database fakta Setelah Eksekusi

NOMOR	ATURAN
R-1	IF E THEN D
R-2	IF D AND B THEN A
R-3	IF A AND C THEN W
R-4	IF W THEN X

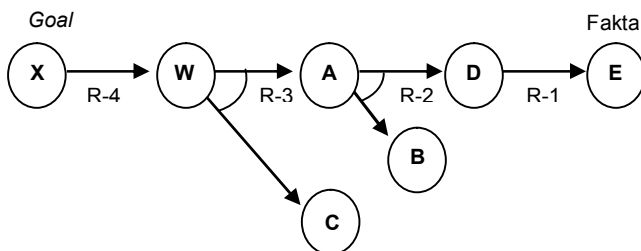
Pada tabel 7 terdapat 4 aturan yang tersimpan di dalam *knowledge base* (basis pengetahuan). Fakta awal yang diberikan hanya: *E* (artinya : *E* bernilai benar). Fakta awal yang akan dibuktikan apakah *X* bernilai benar (hipotesis : *X*) ?. Langkah-langkah *inferensi* yang harus dilakukan adalah sebagai berikut :

- a. Pertama sekali cari terlebih dahulu dari *R1*, aturan mana yang memiliki konsekuen *X*, yaitu bagian yang menyatakan suatu

tindakan tertentu atau konklusi yang diterapkan jika situasi atau *premise* bernilai benar (Pernyataan berawalan THEN). Ternyata setelah ditelusuri, aturan dengan konsekuen *X* baru ditemukan pada *R4*. Untuk membuktikan bahwa *X* benar, maka perlu dibuktikan bahwa jawaban *W* benar.

- b. Kemudian cari aturan yang memiliki konsekuen *W*. Pencarian dimulai dari *R1*, dan ternyata baru akan ditemukan aturan dengan konsekuen *W* pada *R3*. Untuk membuktikan kebenaran *W* benar, maka perlu dicari aturan dengan konsekuen *A* dan *C* benar. Dan ternyata *A* bernilai benar ternyata ada pada *R2*.
- c. Kemudian cari aturan yang memiliki konsekuen *A*. Pencarian dimulai dari *R1*, dan ternyata baru akan ditemukan aturan dengan konsekuen *A* pada *R2*. Untuk membuktikan kebenaran *A* benar, maka perlu dicari aturan dengan konsekuen *D* dan *B* benar. Dan ternyata *D* bernilai benar ternyata ada pada *R1*.
- d. Kemudian cari aturan yang memiliki konsekuen *D*. Pencarian dimulai dari *R1*, dan ternyata ditemukan aturan dengan konsekuen *E* jelas benar karena *E* merupakan fakta. Dengan demikian berdasarkan penalaran ini bisa dibuktikan bahwa *X* bernilai benar.

Dari hasil eksekusi inferensi *backward chaining* dapat digambarkan dengan pola pemodelan yang dinyatakan oleh Gambar 6.



Gambar 6. Alur Inferensi Backward Chaining

Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN). Dengan kata lain penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut dicari fakta-fakta yang ada dalam *knowledge base*. Fakta awal yang diberikan

adalah *A* (artinya : *A* bernilai benar). Setelah dilakukan eksekusi dapat dibuktikan bahwa (**hipotesis : *W***) adalah bernilai benar.

10. Disain Implementasi *Inferensi Engine*

Berikut ini pengimplementasian sistem *inferensi* menggunakan *backward chaining*. Langkah pertama dalam membangun *inference engine* adalah penelusuran kerusakan mesin dengan mengembangkan tampilan sebagai berikut :

Pelacakan Kerusakan Jaringan Komputer

<p>Kategori Kerusakan</p> <div style="border: 1px solid gray; padding: 2px; display: flex; justify-content: space-between; align-items: center;"> Pengkabelan ▼ </div> <p>Jenis Kerusakan</p> <ul style="list-style-type: none"> Jaringan tidak terdeteksi Konektor longgar Kabel terbuka Kesalahan pada pemasangan konektor RJ-45 <p>Gejala Kerusakan</p> <div style="border: 1px solid gray; padding: 5px; font-size: small;"> Icon Network disudut kanan bawah layar bertanda silang dan Muncul pesan "A Network Cable is Unplugged" </div> <p style="text-align: center; margin-top: 10px;">Proses</p>	<p style="text-align: center; font-weight: bold; margin: 0;">Solusi Masalah</p> <p style="font-size: x-small; margin-top: 5px;"> Lepaskan kabel yang terhubung antara <i>Hub</i> dan <i>CPU</i> selanjutnya gunakan alat <i>Tester Jaringan</i> yaitu dengan cara memasukkan ke dua buah ujung konektor pada alat tersebut dan jika hasil alat tester tersebut menyatakan kabel tersebut terbakar maka segera ganti kabel tersebut. Selanjutnya lakukan pemasangan ulang dengan mengukur panjang kabel dengan memasangkan konektor di kedua ujung kabel tersebut. Pada saat pemasangan Kabel UTP dan konektor RJ-45, yang perlu diperhatikan adalah pemasangan wayar kabel pada konektor RJ-45 harus disesuaikan dengan warna wayar dengan urutan pin pada konektor RJ-45. Setelah selesai melakukan instalasi kabel lakukan uji coba kabel dengan menggunakan alat tester jaringan yang berfungsi untuk menguji kehandalan instalasi kabel dan konektor tersebut. </p>
--	---

F. SIMPULAN

Berdasarkan teknik *inferensi* yang digunakan yaitu *Bakward channing* atau *Goal Driven*, kinerja sistem dalam pencapaian kesimpulan didasarkan pada langkah berikut :

1. Memulai penalaran dari sekumpulan hotesa.
2. Menemukan ciri-ciri yang sesuai hipotesa menuju fakta-fakta yang mendukung hipotesa tersebut.

3. Begitu selanjutnya sampai pada kesimpulan atau *goal* tercapai.

G. DAFTAR PUSTAKA

- Anjik, Rianto. 2008. *Jaringan Komputer*. Yogyakarta: Penerbit Andi Yogyakarta.
- Arhami, Muhammad. 2003. *Konsep Dasar Sistem Pakar*. Yogyakarta: Penerbit ANDI
- Brandon, Bill. 2003. "Learning Developers Journal", *The Power of Backward Chaining*.
- Hermawan, Arief, (2006), *Jaringan Saraf Tiruan (Teori dan Aplikasinya)*, Yogyakarta : Penerbit ANDI.
- Jong Jek Siang, Drs, M.Sc (2005), *Jaringan Saraf Tiruan & Pemrogramannya Menggunakan Matlab*, Yogyakarta : Penerbit ANDI.
- Kusumadewi, Sri (2003), *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta : Graha Ilmu.
- Kristanto, Andri (2004), *Kecerdasan Buatan*, Yogyakarta : Graha Ilmu.
- Kuswadi, Son (2007), *Kendali Cerdas (Teori dan Aplikasi Praktisnya)*, Yogyakarta : Penerbit ANDI.
- Puspitaningrum, Diyah, (2006), *Pengantar Jaringan Saraf Tiruan*, Yogyakarta : Penerbit ANDI.

