

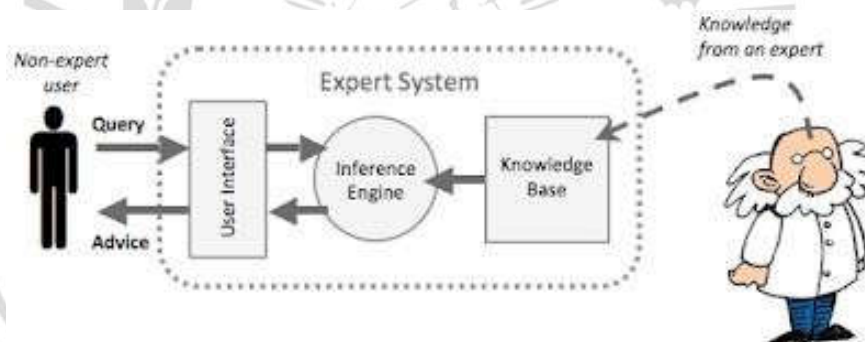
BAB II

LANDASAN TEORI

2.1 Sistem Pakar

Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli, dan sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli .

Sistem pakar pertama kali dikembangkan oleh komunitas AI pada pertengahan tahun 1960. Sistem pakar yang muncul pertama kali adalah *General Purpose Problem Solver* (GPS) yang dikembangkan oleh Newel & Simon. Sistem pakar adalah suatu sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan orang awam. Contohnya dokter, mekanik, psikolog, dan lain-lain.



Gambar 2.1 Ilustrasi Sistem Pakar

2.1.1 Tujuan Sistem Pakar

Sistem pakar (*expert system*) sendiri merupakan paket perangkat lunak atau paket program komputer yang ditujukan sebagai penyedia nasihat dan sarana bantu dalam memecahkan masalah di bidang-bidang spesialisasi tertentu seperti

sains, perekayasaan, matematika, kedokteran, pendidikan dan sebagainya. Sistem pakar merupakan subset dari *Artificial Intelligence* (Arhami, 2005). Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan masalah yang dimaksud seperti:

1. Interpretasi. Membuat kesimpulan atau deskripsi dari sekumpulan data mentah. Pengambilan keputusan dari hasil observasi, termasuk pengenalan ucapan, analisis citra, interpretasi sinyal, dll.
2. Prediksi. Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu. Contoh: prediksi demografi, prediksi ekonomi, dll.
3. Diagnosis. Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati diagnosis medis, elektronis, mekanis, dll.
4. Perancangan (desain). Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu. Contoh: perancangan layout sirkuit, bangunan.
5. Perencanaan. Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu. Contoh: perencanaan keuangan, militer, dll.
6. Monitoring. Membandingkan hasil pengamatan dengan kondisi yang diharapkan. Contoh: *computer aided monitoring system*.
7. Debugging. Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi. Contoh: memberikan resep obat terhadap kegagalan.
8. Instruksi. Mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subjek. Contoh: melakukan instruksi untuk diagnosis dan debugging.
9. Kontrol. Mengatur tingkah laku suatu *environment* yang kompleks. Contoh: melakukan kontrol terhadap interpretasi, prediksi, perbaikan dan monitoring kelakuan sistem.

2.1.2 komponen 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 pembangunan komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi³. Komponen-komponen yang terdapat dalam arsitektur/struktur sistem pakar pada gambar di atas dijelaskan sebagai berikut:

a. Antarmuka Pengguna (*User Interface*)

Antarmuka merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

b. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah.

c. Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, laporan penelitian, dan pengalaman pemakai.

d. Mesin/Motor Inferensi (*Inference Engine*)

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin *inferensi* adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan.

e. *Workplace/Blackboard*

Workplace merupakan area dari sekumpulan memori kerja (*working memory*), digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara

f. Fasilitas Penjelasan

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar, digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan.

g. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya dan juga mengevaluasi apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa mendatang

2.2 Pengertian *Hardware* (Perangkat Keras)

Pengertian dari *hardware* atau dalam bahasa Indonesia-nya disebut juga dengan nama “perangkat keras” adalah salah satu komponen dari sebuah komputer yang sifat alatnya bisa dilihat dan diraba secara langsung atau yang berbentuk nyata, yang berfungsi untuk mendukung proses komputerisasi. *Hardware* dapat bekerja berdasarkan perintah yang telah ditentukan ada padanya, atau yang juga disebut dengan istilah *instruction set*. Dengan adanya perintah yang dapat dimengerti oleh *hardware* tersebut, maka *hardware* tersebut dapat melakukan berbagai kegiatan yang telah ditentukan oleh pemberi perintah. Secara fisik, Komputer terdiri dari beberapa komponen yang merupakan suatu sistem. Sistem adalah komponen-komponen yang saling bekerja sama membentuk suatu kesatuan. Apabila salah satu komponen tidak berfungsi, akan mengakibatkan tidak berfungsinya proses-proses yang ada komputer dengan baik. Komponen komputer ini termasuk dalam kategori elemen perangkat keras (*hardware*). Berdasarkan fungsinya, perangkat keras komputer di bagi menjadi :

1. *input device* (unit masukan)
2. *Process device* (unit Pemrosesan)
3. *Output device* (unit keluaran)
4. *Backing Storage* (unit penyimpanan)
5. *Periferal* (unit tambahan)

Komponen dasar pada komputer terdiri dari *input*, *process*, *output* dan *storage*. *Input device* terdiri dari *keyboard* dan *mouse*, *Process device* adalah *microprocessor* (*ALU*, *Internal Communication*, *Registers* dan *control section*), *Output device* terdiri dari monitor dan printer, *Storage external memory* terdiri dari *harddisk*, *Floppy drive*, *CD ROM*, *Magnetic tape*. *Storage internal memory* terdiri dari *RAM* dan *ROM*. Sedangkan komponen *Peripheral Device* merupakan komponen tambahan atau sebagai komponen yang belum ada atau tidak ada sebelumnya. Komponen *Peripheral* ini contohnya : *TV Tuner Card*, *Modem*, *Capture Card*.

2.3 *Forward Chaining*

Forward Chaining merupakan metode pencarian yang memulai prosesnya dari sekumpulan data atau fakta dan dari fakta-fakta tersebut didapatkan kesimpulan yang dijadikan solusi dari permasalahan. *Forward chaining* merupakan metode *inferensi* yang melakukan penalaran dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai *TRUE*), maka proses akan menyatakan konklusi. *Forward chaining* adalah *data-driven* karena *inferensi* dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan *tree* yang lebar dan tidak dalam, maka gunakan *forward chaining*.

Forward Chaining merupakan suatu penalaran yang dimulai dari fakta untuk mendapatkan kesimpulan (*conclusion*) dari fakta tersebut. *Forward chaining* bisa dikatakan sebagai strategi *inference* yang bermula dari sejumlah fakta yang diketahui. Pencarian dilakukan dengan menggunakan *rules* yang premisnya cocok dengan fakta yang diketahui tersebut untuk memperoleh fakta baru dan melanjutkan proses hingga goal dicapai atau hingga sudah tidak ada *rules* lagi yang premisnya cocok dengan fakta yang diketahui maupun fakta yang diperoleh. *Forward chaining* bisa disebut juga runut maju atau pencarian yang dimotori data (*data driven search*). Jadi pencarian dimulai dari premis-premis atau informasi masukan (*if*) dahulu kemudian menuju konklusi atau *derived information* (*then*). *Forward Chaining* berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan atau dengan menambahkan data memori kerja untuk diproses agar ditemukan suatu hasil.

Forward Chaining digunakan jika :

1. Banyak aturan berbeda yang dapat memberikan kesimpulan yang sama.
2. Banyak cara untuk mendapatkan sedikit konklusi.
3. Benar-benar sudah mendapatkan berbagai fakta, dan ingin mendapatkan konklusi dari fakta-fakta tersebut.

Adapun tipe sistem yang dapat menggunakan teknik pelacakan *forward chaining*, yakni :

1. Sistem yang direpresentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari *rule-rule* dalam *knowledge base* untuk *rule-rule* yang berkorespondensi dengan kondisi dalam bagian *if*.
3. Setiap *rule* dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian *then*. Kondisi baru ini dapat ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang di tambahkan ke sistem akan diproses. Jika ditemui suatu kondisi, sistem akan kembali kelangkah 2 dan mencari *rule-rule* dalam *knowledge base* Jika tidak ada konklusi baru, sesi ini berakhir.

Jika klausa premis sesuai dengan situasi (bernilai *true*), maka proses akan meng-assert konklusi. *Forward chaining* juga digunakan jika suatu aplikasi menghasilkan *tree* yang lebar dan tidak dalam. Pada metode *forward chaining*, ada 2 cara yang dapat dilakukan untuk melakukan pencarian, yaitu :

1. Dengan memasukkan semua data yang tersedia kedalam sistem pakar pada satu kesempatan dalam sesi konsultasi. Cara ini banyak berguna pada sistem pakar yang termasuk dalam proses terautomatisasi dan menerima data langsung dari komputer yang menyimpan *data base*, atau dari satu set sensor.
2. Dengan hanya memberikan elemen spesifik dari data yang diperoleh selama sesi konsultasi kepada sistem pakar. Cara ini mengurangi jumlah data yang diminta, sehingga data yang diminta hanyalah data-data yang benar-benar dibutuhkan oleh sistem pakar dalam mengambil kesimpulan.

Dalam *Forward Chaining* pencarian dimulai dengan fakta yang diketahui dan mengambil fakta baru menggunakan aturan yang telah diketahui pada sisi Jika (*if*). Karena diketahui A dan B benar, sistem pakar mulai dengan mengambil fakta baru menggunakan aturan yang memiliki A dan B pada sisi Jika (*if*). Dengan menggunakan R4, sistem pakar mengambil fakta baru C dan menambahkannya kedalam *assertion base* sebagai benar. Sekarang R1 *fire* (karena A dan C benar) dan menyatakan E sebagai benar dalam *assertion base* sebagai benar. Karena B dan E keduanya benar (berada dalam *assertion base*), R3 *fire* dan menetapkan F sebagai benar dalam *assertion base*. Sekarang R5 *fire* (karena F berada dalam sisi Jika), yang menetapkan G sebagai benar, jadi hasilnya adalah G.

Contoh : Terdapat 10 aturan yang tersimpan dalam basis pengetahuan yaitu :

R1 : *if A and B then C*

R2 : *if C then D*

R3 : *if A and E then F*

R4 : *if A then G*

R5 : *if F and G then D*

R6 : *if G and E then H*

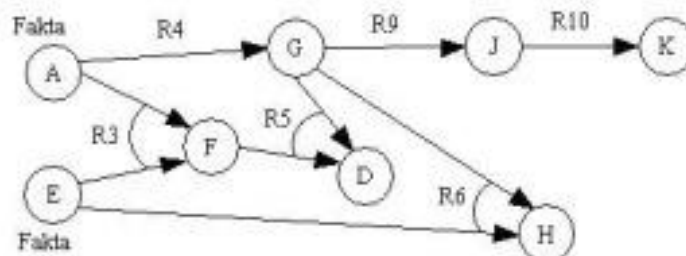
R7 : *if C and H then I*

R8 : *if I and A then J*

R9 : *if G then J*

R10 : *if J then K*

Fakta awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar. Proses penalaran *forward chaining* terlihat pada gambar dibawah



Gambar 2.2 Alur *Forward Chaining*

2.4 Penelitian Sebelumnya

Berikut Beberapa referensi pembelajaran dari beberapa contoh kasus yang hampir sama dengan permasalahan yang dihadapi, antara lain :

1. Gideon Abram Filando Suwarso, 2011, dengan judul "*Sistem Pakar untuk Penyakit Anak Menggunakan Metode Forward Chaining*". Pembahasan utama dalam Skripsi ini adalah perancangan dan pembuatan sistem pakar dengan metode *forward chaining* dan *backward chaining*. Dalam skripsi ini menggunakan subjek penyakit anak karena kekebalan tubuh pada anak kecil tidak sebaik dan sesempurna kekebalan tubuh orang dewasa. Hal inilah yang membuat anak kecil mudah terjangkit virus penyakit. Metode yang digunakan dalam pembuatan sistem pakar ini adalah *forward chaining* dan *backward chaining*, di mana sistem digerakkan oleh fakta-fakta yang ada. Fakta tersebut diperoleh dari kondisi fisik pasien, atau disebut dengan gejala. Dalam pembuatan sistem pakar ini diperlukan penggalan *knowledge* oleh engineer yang bersumber pada pakar, dalam hal ini adalah dokter. Proses penggalan *knowledge* dengan cara wawancara. Berdasarkan analisis, sistem pakar ini memiliki prosentase kelayakan program untuk digunakan oleh orang awam adalah sebesar 85.4%. Selain itu berdasarkan hasil kuisioner, kemudahan program ini juga mendapatkan prosentase yang besar juga.
2. MITA FULJANA, 2017, dengan judul "*SISTEM PAKAR DIAGNOSIS PENYAKIT TANAMAN CABAI MENGGUNAKAN METODE FORWARD CHAINING BERBASIS ANDROID*". Penelitian ini dilakukan untuk membuat suatu sistem pakar yang mampu mendiagnosis penyakit pada tanaman cabai berdasarkan pengetahuan yang diberikan langsung dari pakar/ahlinya. Penelitian ini menggunakan metode perhitungan probabilitas klasik dalam menghitung presentase diagnosis dan dibuat pada *mobile device platform Android*. Pada penelitian ini terdiri dari 37 data gejala, 10 data penyakit cabai yang disebabkan oleh jamur, dan 10 data aturan. Sistem pakar ini menggunakan metode *inferensi forward chaining*.

Hasil pengujian menunjukkan bahwa: (1) Pengujian fungsional dengan menggunakan metode *Black Box Equivalence Partitioning* (EP) mendapatkan hasil sesuai dengan yang diharapkan pada scenario uji di setiap kelas uji. (2) Pengujian kepakaran dengan membandingkan hasil perhitungan manual dan sistem sudah sesuai dan berjalan baik. (3) Pengujian kuesioner dengan 53 responden yang dibagi menjadi empat Kelompok responden menunjukkan; kelompok responden pertama yang terdiri dari pakar penyakit cabai mendapatkan rata-rata nilai sebesar 85,14 % (dikategorikan sangat baik), kelompok responden kedua yang terdiri mahasiswa jurusan Pertanian 84,13 % (dikategorikan sangat baik), kelompok responden ketiga yang terdiri dari mahasiswa jurusan Ilmu Komputer 84,28 % (dikategorikan sangat baik), dan kelompok responden keempat yang terdiri dari petani cabai 86 % (dikategorikan sangat baik).

