BAB II

LANDASAN TEORI

2.1 Sistem Pakar

Aplikasi yang mungkin berkembang saat ini yaitu sistem pakar (*expert system*). Pada dasarnya *expert sistem* terdiri atas banyak *database* dan seperangkat aturan yang dapat mencari sendiri dari *database* tersebut menjadi solusi terbaik atas suatu masalah. Sistem pakar diambil dari istilah *knowledge base expert system*. *knowledge base expert system* dibentuk dari *knowledge base system* yang merupakan hasil dari proses *knowledge engineering*, (Turban, 1992).

Perkembangan komputer dewasa ini telah mengalami banyak perubahan yang sangat pesat, seiring dengan kebutuhan manusia yang semakin banyak dan kompleks. Komputer yang pada awalnya hanya digunakan oleh para akademisi dan militer, kini telah digunakan secara luas di berbagai bidang, misalnya: Bisnis, Kesehatan, Pendidikan, Psikologi, Permainan dan sebagainya. Hal ini mendorong para ahli untuk semakin mengembangkan komputer agar dapat membantu kerja manusia atau bahkan melebihi kemampuan kerja manusia.

Kecerdasan buatan atau *artificial intelligence* merupakan bagian dari ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Sistem cerdas (*intelligent system*) adalah sistem yang dibangun dengan menggunakan teknik-teknik *artificial intelligence*.

Sistem Pakar *(Expert System)* adalah program berbasis pengetahuan yang menyediakan solusi-solusi dengan kualitas pakar untuk problema-problema dalam suatu *domain* yang spesifik. Sistem pakar merupakan program komputer yang meniru proses pemikiran dan pengetahuan pakar dalam menyelesaikan suatu masalah tertentu. Implementasi sistem pakar banyak digunakan dalam bidang psikologi karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar pada bidang tertentu dalam program komputer sehingga keputusan dapat diberikan dalam melakukan penalaran secara cerdas. Umumnya pengetahuannya diambil dari seorang manusia yang pakar dalam domain tersebut dan sistem pakar itu berusaha meniru metodelogi dan kinerjanya *(performance),* (Kusumadewi, 2003).

2.1.1 Fitur-fitur Sistem Pakar

Sistem pakar harus memiliki fitur file berikut, (Kusumadewi, 2003):

1. **Keahlian**

   Sistem pakar harus memiliki keahlian yang memungkinkan sistem membuat keputusan tingkat pakar. Sistem harus menampilkan performa pakar dan kekuatan yang cukup.

2. Pertimbangan Simbolik

Pengetahuan harus direpresentasikan secara simbolik, dan mekanisme pertimbangan primer juga harus simbolik. Mekanisme pertimbangan simbolik biasanya menyertakan *backward chaining* dan *forward chaining*, yang akan dideskripsikan pada bagian selanjutnya.

3. *Deep knowledge* (Kedalaman Pengetahuan)

Basis pengetahuan harus berbasis pengetahuan yang kompleks yang tidak mudah diperoleh dari non pakar.

4. *Self-knowledge*

Sistem pakar harus dapat menganalisis pertimbangannya sendiri dan menjelaskan proses mencapai kesimpulan.

2.1.2 Keuntungan dan Kelemahan Sistem Pakar

Keuntungan dan kelemahan sistem pakar adalah sebagai berikut, (Kusumadewi, 2003):

2.1.2.1 Keuntungan Sistem Pakar

Banyak manfaat yang dapat diambil dengan adanya sistem pakar, adalah:

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Meningkatkan kualitas.
6. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).

7. Mampu beroperasi dalam lingkungan yang berbahaya.
8. Memiliki kemampuan untuk mengakses pengetahuan.
9. Memiliki reliabilitas.
10. Meningkatkan kapabilitas sistem komputer.
11. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
12. Sebagai media pelengkap dalam pelatihan.
13. Meningkatkan kapabilitas dalam penyelesaian masalah.
14. Menghemat waktu dalam pengambilan keputusan.

2.1.2.2 Kelemahan Sistem Pakar

Di samping memiliki keuntungan, sistem pakar juga memiliki beberapa kelemahan, yaitu:

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan oleh sistem.
3. Sulit dikembangkan, hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
4. Sistem pakar tidak 100% bernilai benar.

2.1.3 Alasan Pengembangan Sistem Pakar

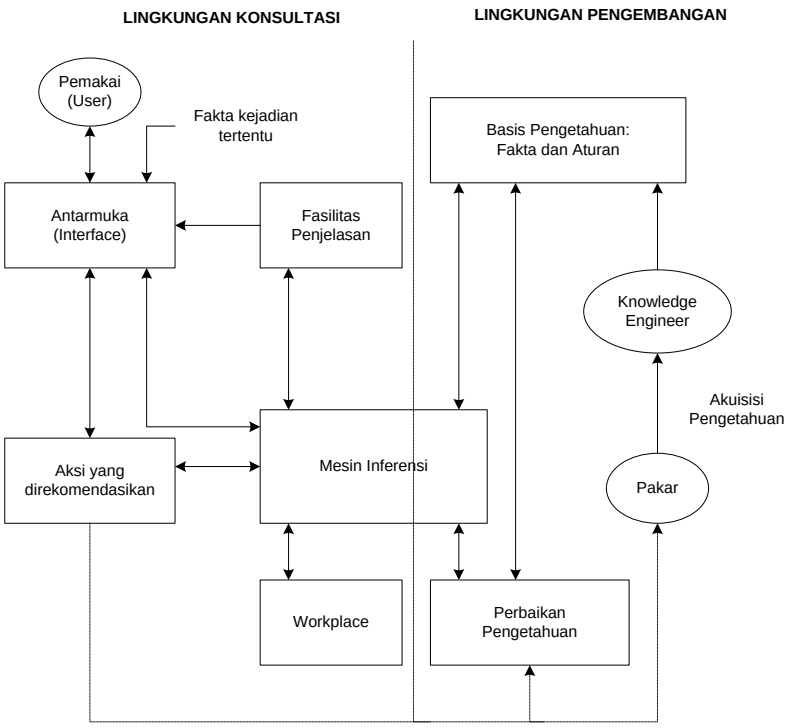
Sistem pakar dikembangkan lebih lanjut dengan alasan:

1 Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar mempunyai masa produktif atau pergi (meninggal dunia).
4. Pengetahuan seorang pakar mahal nilainya.
5. Kepakaran dibutuhkan juga pada situasi yang kurang mendukung.

2.1.4 Struktur Sistem Pakar

Sistem pakar dapat ditampilkan dengan dua lingkungan, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh sistem pakar (ES) *builder* untuk membangun komponen dan memasukkan pengetahuan ke dalam basis pengetahuan. Lingkungan konsultasi digunakan oleh nonpakar untuk memperoleh pengetahuan dan nasihat pakar. Lingkungan ini dapat dipisahkan setelah sistem lengkap. Bentuk Struktur Sistem Pakar pada penelitian ini dapat dilihat pada Gambar 2.1 berikut ini

.



Gambar 2.1 Arsitektur Sistem Pakar *(Sumber : Turban, 2005)*

Keterangan Gambar 2.1 Arsitektur Sistem Pakar:

1. **Pakar**

   Merupakan orang yang memiliki pengetahuan, penilaian, pengalaman dan metode khusus, serta kemampuan untuk menerapkan bakat ini dalam memberikan nasihat dan memecahkan persoalan.

2. **Akuisisi Pengetahuan**

   Merupakan akumulasi, transfer dan transformasi keahlian pemecahan masalah dari pakar atau sumber pengetahuan terdokumentasi ke program komputer, untuk membangun atau memperluas basis pengetahuan. Sumber pengetahuan potensial antara lain pakar manusia, buku teks dan dokumen multimedia.

3. *Knowledge Engineer*

Yaitu seorang spesialis sistem yang menterjemahkan pengetahuan yang dimiliki seorang pakar menjadi pengetahuan yang akan tersimpan dalam basis pengetahuan pada sebuah sistem pakar.

4. Basis Pengetahuan

Berisi pengetahuan relevan yang diperlukan untuk memahami, merumuskan, dan memecahkan persoalan. Basis pengetahuan mencakup dua elemen dasar, yaitu :

1) Fakta, misalnya situasi persoalan dan teori area persoalan (apa yang diketahui tentang area domain).
2) *Rule* atau aturan khusus yang mengarahkan penggunaan pengetahuan untuk memecahkan persoalan khusus dalam domain tertentu (referensi logika, misalnya, antara gejala dan penyebab).

5. Perbaikan Pengetahuan

Pakar manusia memiliki sistem perbaikan pengetahuan, yakni mereka dapat menganalisis pengetahuannya sendiri, belajar darinya, dan meningkatkannya untuk konsultasi mendatang. Serupa pula, evaluasi tersebut diperlukan dalam pembelajaran komputer sehingga program dapat menganalisis alasan keberhasilan atau kegagalannya. Hal ini dapat mengarah kepada peningkatan sehingga menghasilkan basis pengetahuan yang lebih akurat dan pertimbangan yang lebih efektif. Dengan komponen

ini, pakar mampu menganalisis kinerja dari Sistem pakar, belajar daripadanya, dan meningkatkannya pada konsultasi selanjutnya.

6. Mesin Inferensi

Merupakan otak dari Sistem pakar. Komponen ini sebenarnya adalah program komputer yang menyediakan metodologi untuk *reasoning* (pertimbangan) mengenai informasi dalam basis pengetahuan dan dalam ”*workplace*”, dan digunakan untuk merumuskan kesimpulan.

Mesin Inferensi mempunyai 3 elemen utama, yaitu :

1) *Interpreter* adalah elemen yang mengeksekusi item agenda yang dipilih dengan mengaplikasikannya pada basis pengetahuan *rule* yang berhubungan.
2) *Scheduler* adalah elemen yang menjaga kontrol di sepanjang agenda.Memperkirakan akibat dari pengaplikasian *rule inferensia* yang menampakkan prioritas *item* atau kriteria lain pada agenda.
3) *Consistency enforcer* adalah elemen yang mencoba menjaga konsistensi representasi solusi yang muncul.

7. *Workplace*

Merupakan area kerja memori yang disimpan sebagai *database* untuk deskripsi persoalan terbaru yang ditetapkan oleh data *input*, digunakan juga untuk perekaman hipotesis dan keputusan sementara.

Tiga tipe keputusan dapat direkam dalam *workplace*: rencana (cara mengatasi persoalan), agenda (tindakan potensial sebelum eksekusi), dan

solusi (hipotesis kandidat dan arah tindakan alternatif yang telah dihasilkan sistem sampai dengan saat ini).

8. Fasilitas Penjelasan

Ini adalah kemampuan penelusuran kebenaran dari konklusi yang didapat dari sumber-sumbernya. Hal ini krusial untuk transformasi kepakaran dan penyelesaian masalah. Komponen ini mampu menelusuri kebenaran dan untuk menerangkan perilaku Sisem Pakar secara interaktif, menjawab pertanyaan seperti: Mengapa pertanyaan tertentu ditanyakan oleh Sistem pakar? Bagaimana konklusi tertentu dicapai? Mengapa alternatif tertentu ditolak? Rencana apakah yang ada untuk mencapai solusi? Dan apa-apa saja selanjutnya yang harus dilakukan sebelum diagnosis final dapat ditentukan?

9. Antar muka Pemakai (*User Interface*)

Sistem pakar berisi *prosessor* bahasa untuk komunikasi berorientasi persoalan yang mudah antara pengguna dan komputer. Komunikasi ini paling baik dilakukan dalam bahasa alami. Dikarenakan batasan teknologi, maka kebanyakan sistem yang ada menggunakan pendekatan pertanyaan dan jawaban untuk berinteraksi dengan pengguna.

10. Aksi yang direkomendasikan

Merupakan saran atau solusi yang direkomendasikan untuk permasalahan yang sedang dihadapi oleh *user*.

11. *User*

Umumnya *user* yang dimaksud ini adalah:

1) *Klien* (yaitu bukan pakar) yang menginginkan *advis*/nasehat. Di sini Sistem pakar bertindak seperti seorang konsultan atau penasehat.
2) *Learner* (pelajar) untuk mempelajari bagaimana Sistem pakar menyelesaikan permasalahan. Pada *Learner* Sistem pakar bertindak sebagai seorang instruktur.
3) Pembangun Sistem pakar yang ingin meningkatkan basis pengetahuannya. Di sini Sistem pakar bertindak sebagai seorang rekan.

2.1.5 Representasi Pengetahuan

Pengetahuan merupakan kemampuan untuk membentuk model mental yang menggambarkan obyek dengan tepat dan merepresentasikannya dalam aksi yang dilakukan terhadap suatu obyek.

Representasi pengetahuan merupakan metode yang digunakan untuk pengkodean pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Perepresentasian dimaksudkan untuk menangkap sifat-sifat penting problema dan membuat informasi itu dapat diakses oleh prosedur pemecahan problema.

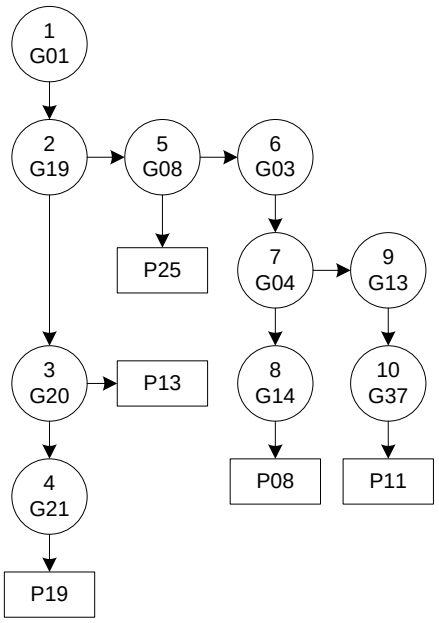
2.1.5.1 Tabel Keputusan

Pengetahuan relasi dapat pula direpresentasikan dalam tabel keputusan. Dalam tabel keputusan, pengetahuan disusun dalam format *spreadsheet* menggunakan kolom dan baris. Tabel dibagi menjadi dua bagian. Pertama, dikembangkan suatu daftar atribut, dan untuk tiap aribut dirinci semua kemugkinan nilai. Kemudian daftar kesimpulan dikembangkan. Akhirnya, kombinasi atribut yang berbeda disesuaikan terhadap kesimpulan.

Pengetahuan untuk tabel dikumpulkan dalam sesi akuisisi pengetahuan. Setelah terbentuk, pengetahuan dalam tabel dapat digunakan sebagai *input* untuk metode representasi pengetahuan yang lain. Tidak mungkin melakukan inferensi dengan hanya tabel domain kecuali pada saat digunakan induksi aturan.

2.1.5.2 Pohon Keputusan

Pohon keputusan dihubungkan ke tabel dan populer di banyak tempat. Pohon ini terdiri dari *node* yang menyatakan tujuan dan *link* yang menyatakan keputusan. Manfaat utama pohon keputusan adalah dapat menyederhanakan proses akuisisi pengetahuan. Pohon keputusan dapat dengan mudah diubah ke aturan. Contoh dari pohon keputusan dapat dilihat pada Gambar 2.2.



Gambar 2.2 Contoh Pohon Keputusan

Pengetahuan dapat direpresentasikan dalam bentuk yang sederhana atau kompleks tergantung dari masalahnya, beberapa model representasi pengetahuan yang penting, adalah:

1. **Logika (*logis*)**

   Logika adalah bentuk representasi pengetahuan yang paling tua.

   Proses logika adalah proses membentuk kesimpulan atau menarik suatu inferensi berdasarkan fakta yang telah ada. *Input* dari proses logika berupa premis atau fakta-fakta yang diakui kebenarannya sehingga dengan melakukan penalaran pada proses logika dapat dibentuk suatu inferensi atau kesimpulan yang benar pula.

Ada 2 penalaran yang dapat dilakukan untuk mendapat konklusi:

1) Penalaran Deduktif

Penalaran dimulai dari prinsip umum untuk mendapatkan konklusi yang lebih khusus.

Contoh:

Premis Mayor : jika warna daun dari hijau menjadi kuning berarti terserang penyakit layu bakteri.

Premis Minor : Warna daun menjadi kunig.

Konklusi : Oleh karena itu jahe terserang penyakit layu bakteri.

2) Penalaran Induktif

Penalaran dimulai dari fakta-fakta khusus untuk mendapatkan kesimpulan umum.

Contoh :

Premis-1 : Penyakit layu bakteri adalah penyakit berbahaya.

Premis-2 : Penyakit busuk rimpang adalah penyakit berbahaya.

Premis-3 : Penyakit bercak daun adalah penyakit berbahaya.

Konklusi : Penyakit jahe adalah penyakit berbahaya.

2. Jaringan Semantik (*Semantic Nets*)

Jaringan semantik merupakan jaringan data dan informasi, yang menunjukkan hubungan antar berbagai objek dimana informasi yang terhubung tersebut adalah informasi yang proporsional (suatu pernyataan

yang dapat bernilai benar atau salah). Dalam matematika, istilah jaringan semantik merupakan suatu label atau graph berarah.

Struktur jaringan semantik terdiri dari *node* atau simpul dan busur atau arc yang menghubungkannya. Simpul menyatakan objek sedangkan busur menyatakan *links*. *Links* dari jaringan semantik digunakan untuk menunjukkan hubungan (*relationship*) antar simpul-simpul tersebut.

1. Bingkai (*Frame*)

Bingkai adalah struktur data yang mengandung semua informasi/pengetahuan yang relevan dari suatu obyek. Pengetahuan ini diorganisasi dalam struktur hirarkis khusus yang memungkinkan pemrosesan pengetahuan. Bingkai merupakan aplikasi dari pemrograman berorientasi obyek dalam AI dan Sistem Pakar. Pengetahuan dalam bingkai dibagi-bagi ke dalam slot atau atribut yang dapat mendeskripsikan pengetahuan secara deklaratif ataupun prosedural.

Contohnya adalah bingkai untuk merepresentasikan pengetahuan mengenai gajah:

Mamalia

Subkelas : Binatang

berdarah_panas : Ya

Gajah

Subkelas : Mamalia

Warna : Abu-abu

Ukuran : Besar

Clyde

Instance : Gajah

Warna : Merah_muda

Pemilik : Fred

Nellie:

Instance : Gajah

Ukuran : Kecil

2. Kaidah Produksi (*Production Rule*)

Kaidah produksi adalah bentuk representasi pengetahuan yang paling populer untuk sistem pakar. Pengetahuan direpresentasikan dalam bentuk pasangan kondisi-aksi : **IF** kondisi ini (atau premis atau *anteseden*) terjadi, **THEN** beberapa tindakan (atau hasil atau kesimpulan atau *konsekuensi*) akan (atau sebaiknya) terjadi. Pada dasarnya penyajian pengetahuan secara kaidah produksi terdiri dari dua hal, yaitu :

1. *Anteseden,* bagian yang mengekspresikan situasi atau premis (pernyataan berawalan If).
2. *Konsekuen,* bagian yang menyatakan suatu tindakan, hasil atau kesimpulan yang diperoleh jika premis bernilai benar (pernyataan berawalan THEN).

Hal ini dapat dituliskan sebagai IF *anteseden* THEN *konsekuen-konsekuen* atau kesimpulan yang dinyatakan pada bagian THEN baru dinyatakan benar. Jika bagian IF pada kaidah tersebut juga bernilai benar atau sesuai

ketentuan tertentu. Kaidah bisa dalam bentuk berbeda seperti contoh di bawah ini :

a. IF *premise*, THEN *conclusion*. IF pendapatan tinggi THEN kesempatan pemeriksaan keuangan tinggi.
b. *Conclusion*, IF *premise*. Kesempatan pemeriksaan keuangan tinggi pendapatan tinggi.

Ada kemungkinan beberapa *premise* dan beberapa *conclusion* di dalam satu kaidah, sehingga beberapa *premise* diberi istilah dengan *premise clauses* dan beberapa *conclusion* diberi istilah *conclusion clauses*.

Kaidah produksi merupakan keterkaitan antara fakta-fakta yang didapat dari pakar, ilmu pengetahuan serta pengalaman. Skema representasi pengetahuan kaidah produksi biasanya dituliskan dalam bentuk jika – maka (IF – THEN). Kaidah yang terdapat pada pangkalan pengetahuan tentang diagnosis hama dan penyakit pada tanaman jahe adalah sebagai berikut :

Kaidah 1 :	IF	Daun bagian bawah melipat & menggulung
	AND	Daun berlubang-lubang
	AND	Batang dan daun tanaman kering
	THEN	**Hama Kepik**

Kaidah 2 :	IF	Akar tanaman luka
	AND	Akar tanaman menjadi kering & mati
	AND	Akar tanaman Busuk dan keropos
	AND	Batang dan daun tanaman kering
	THEN	**Hama Ulat**

Kaidah 3 : IF Daun Bagian Bawah Melipat dan Menggulung
AND Warna Daun Dari Hijau Menjadi Kuning
AND Tunas Batang Menjadi Busuk
AND Pinggir daun busuk dan basah
AND Batang tanaman Busuk dan keropos
AND Tunas batang baik menjadi busuk
AND Akar tanaman Busuk dan keropos
THEN Penyakit Layu Bakteri

Kaidah 4 : IF Daun bagian bawah menjadi kuning
AND Daun layu
AND Batang tanaman layu
THEN Penyakit Busuk Rimpang

Kaidah 5 : IF Pada daun bercak-bercak
AND Daun berwarna abu-abu
AND Daun bintik-bintik berwarna hitam
AND Pinggir daun busuk dan basah
AND Batang tanaman Busuk dan keropos
AND Tunas batang baik menjadi busuk
AND Akar tanaman Busuk dan keropos
THEN Penyakit Bercak Daun

Kaidah 6 : IF Akar tanaman luka
AND Patogen tanah mudah masuk
AND Akar tanaman menjadi kering & mati
AND Akar tanaman Busuk dan keropos
AND Batang tanaman luka
THEN Penyakit Buncak Akar

Kaidah 7 : IF Rimpang keriput
AND Batang tanaman Busuk dan keropos
AND Akar tanaman Busuk dan keropos
THEN Penyakit Lalat Rimpang

Kaidah 8 : IF Kulit rimpang kusam
AND Rimpang dihisap dan kering
THEN Penyakit Kutu Perisai

3. *Object-Atribut-Value* (OAV)

Object dapat berupa bentuk fisik atau konsep. Attribute adalah karakteristik atau sifat dari object tersebut. *Values* (nilai) – besaran/nilai/takaran spesifik dari attribute tersebut pada situasi tertentu, dapat berupa *numeric, string* atau *bolean*. Sebuah object bisa memiliki beberapa attribute, biasa disebut OAV Multi attribute. Contoh representasi pengetahuan dengan OAV ditunjukkan pada Tabel 2.1

Tabel 2.1 Representasi dengan OAV

Objek	**Atribut**	**Nilai**
Mangga	Warna	Hijau, Orange
Mangga	Berbiji	Tunggal
Mangga	Rasa	Asam, Manis
Mangga	Bentuk	Oval
Pisang	Warna	Hijau, Kuning

2.1.6 Metode Inferensi Dalam Sistem Pakar

Metode Inferensi dalam sistem pakar adalah bagian yang menyediakan mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar.

a) Metode ini menganalisa masalah tertentu dan selanjutnya mencari jawaban atau kesimpulan yang terbaik.
b) Metode ini memulai pelacakannya dengan mencocokan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.

Dua metode inferensi yaitu:

1. ***Backward Chaining***

   Backward Chaining adalah pendekatan *goal-driven* yang dimulai dari harapan apa yang terjadi (hipotesis) dan kemudian mencari bukti yang mendukung (atau berlawanan) dengan harapan. Sering hal ini memerlukan perumusan dan pengujian hipotesis sementara (subhipotesis). Cara kerja mesin inferensi *Backward Chaining* ditunjukkan pada Gambar 2.3

SUBTUJUAN	← **ATURAN** ←	**TUJUAN**
A = 1	IF A = 1 AND B = 2	
B = 2	THEN C = 3	C = 3

Gambar 2.3 Cara Kerja Mesin Inferensi *Backward Chaining*

Misal: A dan B adalah gejala dan C adalah hama/penyakit.

1 = Warna daun menjadi kuning.

2 = Batang menjadi busuk.

3 = Penyakit layu bakteri

Contoh: Jahe terserang penyakit layu bakteri.

IF Warna daun menjadi kuning AND Batang menjadi busuk.

2. *Forward Chaining*

Forward Chaining adalah pendekatan *data-driven yang* dimulai dari informasi yang tersedia atau dari ide dasar, kemudian mencoba menarik kesimpulan. Cara kerja mesin inferensi *Forward Chaining* ditunjukkan pada Gambar 2.4

DATA	→	**ATURAN**	→	**KESIMPULAN**
A = 1		IF A = 1 AND B = 2		
B = 2		THEN C = 3		C = 3

Gambar 2.4 Cara Kerja Mesin Inferensi *Forward Chaining*

Contoh : IF Warna daun menjadi kuning.

AND Batang menjadi busuk.

THEN Terserang penyakit layu bakteri.

2.2 Rekayasa Perangkat Lunak

Beberapa pengertian menurut para ahli di bidang perangkat lunak tentang rekayasa perangkat lunak, (Andi Kristanto, 2004) yaitu:

1. Menurut Stephen R. Schach

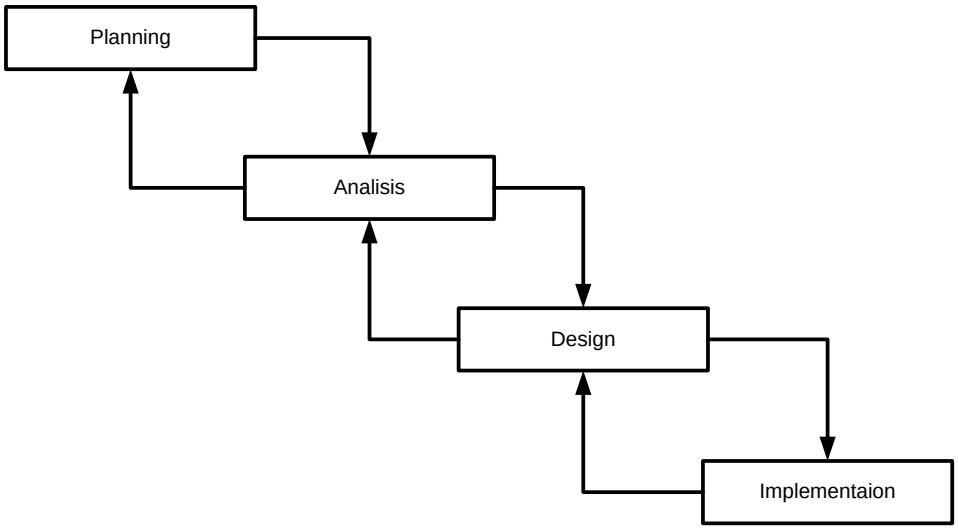
   Rekayasa perangkat lunak adalah sebuah disiplin ilmu dimana dalam menghasilkan perangkat lunak bebas dari kesalahan dan dalam pengiriman anggaran tapat waktu serta memuaskan keinginan pemakai.

2. Menurut Fritz Baver

   Rekayasa perangkat lunak adalah sebuah studi pendekatan dua aplikasi secara sistematis, disiplin pengembangan operasi dan pemeliharaan perangkat lunak yang kesemuanya itu merupakan aplikasi rekayasa yang berkaitan dengan perangkat lunak.

Metode pengembangan Perangkat Lunak menggunakan model *waterfall*, (Pressman, Roger S,2002) :

Model ini adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Metode pengembangan Perangkat Lunak Model *Waterfall* ditunjukkan pada Gambar 2.5



Gambar 2.5 Metode Pengembangan Perangkat Lunak Model *Waterfall*

Keterangan:

1. ***Planning*** **(Perencanaan)**

   Tahap perencanaan merupakan proses penting untuk mengetahui mengapa perangkat lunak dibuat dan menentukan bagaimana cara membangun perangkat lunak tersebut. Langkah pertama dari proses tersebut adalah dengan mengidentifikasi peluang apakah dapat memberikan kemungkinan biaya rendah tetapi menghasilkan keuntungan.

2. ***Analysis*** **(Analisis)**

   Analisis perangkat lunak dilakukan untuk memberikan jawaban pertanyaan siapa yang menggunakan perangkat lunak, apa yang dilakukan oleh perangkat lunak, dimana dan kapan perangkat lunak tersebut digunakan. Pada tahap ini pembuat perangkat lunak melakukan observasi dan pengamatan terhadap perangkat lunak yang lama. Kemudian mengidentifikasi, memanfaatkan, mengembangkan peluang dan membangun konsep untuk sebuah perangkat lunak baru.

3. *Design* (Perancangan)

Tahap perancangan dilakukan untuk menetapkan bagaimana perangkat lunak dioperasikan. Hal ini berkaitan dan menentukan perangkat keras, perangkat lunak, tampilan program, *form* dan laporan yang dipakai.

4. *Implementation* (Penerapan)

Merupakan tahapan penerapan sistem setelah dilakukan uji coba, apakah sistem telah dapat layak digunakan sebagai sebuah sistem yang utuh dengan melakukan serangkaian uji coba terhadap data real serta implementasi terhadap sistem secara keseluruhan.

Tahap ini dari 2 sub bagian yaitu:

a. Coding

Tahapan ini merupakan tahapan untuk menerjemahkan data atau pemecahan masalah yang telah dirancang kedalam bahasa pemprograman komputer yang telah ditentukan.

b. Testing

Pada tahapan ini dilakukan pengujian perangkat lunak oleh peneliti untuk mengetahui apakah perangkat lunak yang dibangun masih ada kesalahan atau tidak, Jika pada saat pengujian ini masih terdapat kesalahan, maka langsung dilakukan perbaikan dan apabila sudah tidak adalagi kesalahan maka perangkat lunak tersebut dapat langsung diterapkan.

5. *Maintenance* (Perawatan)

Tahap ini meliputi pemeliharaan sistem yang telah dibuat yang selalu disesuaikan dengan perkembangan *software* yang ada.

2.2.1 Pengertian Perancangan Sistem

Perancangan sistem adalah merancang suatu sistem yang baru sehingga dapat digunakan untuk menggantikan sistem yang lama secara keseluruhan atau yang digunakan sebagai acuan dalam pembuatan suatu program. Perancangan sistem dapat digunakan untuk memperbaiki sistem yang telah ada sebelumnya.

Perancangan sistem merupakan langkah awal dalam suatu penyelesaian. Untuk mencapai tujuan yang diinginkan agar diperoleh hasil yang maksimal sesuai dengan informasi yang dibutuhkan, maka perancangan sistem sangat dibutuhkan untuk memberikan gambaran dari suatu proses untuk menghasilkan suatu informasi yang dibutuhkan.

2.2.2 Alat Bantu Perancangan Sistem

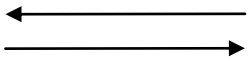
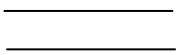
Alat bantu yang digunakan untuk merancang sistem yaitu:

1. DFD (*Data Flow Diagram*)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan asal data dan tujuan data yang keluar dari sistem, tempat data disimpan, proses yang menghasilkan proses tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut,

(Jogiyanto,2005:700). Simbol yang digunakan dalam membuat DFD ditunjukkan pada Tabel 2.2

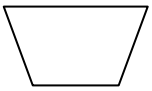
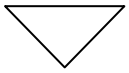
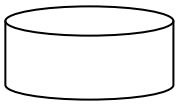
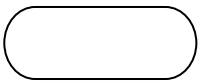
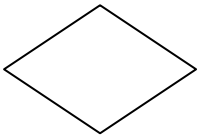
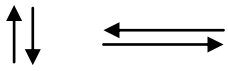
Tabel 2.2 Simbol untuk *DFD*

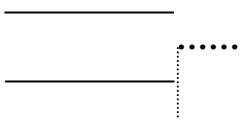
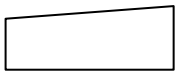
Simbol	Keterangan
(*external entitity*)	Merupakan sumber atau tujuan dari aliran data dari atau ke sistem.
Arus data (*data flow*) 	Menggambarkan aliran data dari satu proses ke proses lainnya.
Proses (*process*)	Proses atau fungsi yang menstransformasikan data.
Simpanan data *(data store*) 	Komponen yang berfungsi untuk menyimpan data atai file.

2. Diagram Alir Dokumen (*Document Flowchart*)

Diagram Alir Dokumen berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu, (Jogiyanto, 2005:712). Simbol yang digunakan untuk menggambar Diagram Alir Dokumen ditunjukkan pada Tabel 2.3

Tabel 2.3 Simbol Diagram Alir Dokumen

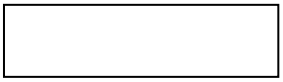
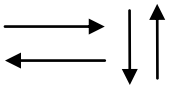
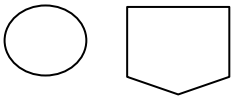
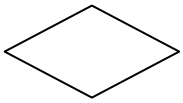
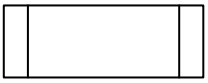
SIMBOL	KETERANGAN
Simbol dokumen	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
kegiatan manual 	Proses manual.
Simbol simpanan 	Menunjukkan dokumen yang diarsipkan (arsip manual)
Simbol proses	Proses yang dilakukan oleh komputer
Simbol Data Storage 	Data penyimpanan (*Data Storage*)
Simbol terminasi 	Terminasi yang menandakan awal dan akhir dari suatu aliran
Simbol keputusan 	Pengambilan keputusan (*Decision*)
Simbol garis alir 	Menunjukkan arus dari data

Simbol penjelasan 	Menunjukkan penjelasan dari suatu proses.
Simbol penghubung	Menunjukkan penghubung kehalaman yang masih sama atau ke halaman lain.
Simbol pemasukan 	Pemasukan data secara manual

3. Bagan Alir Program *(Program Flowchart)*

Menurut Jogiyanto Hartono, MBA dalam buku "*Analisis dan Disain Sistem Informasi*", Bagan alir program adalah suatu bagan yang menggambarkan arus logika dan data yang akan diproses dalam suatu program dari awal sampai akhir. Bagan alir terdiri dari simbol-simbol yang mewakili fungsi-fungsi langkah program dan garis alir (*flowlines*) menunjukkan urutan dari simbol-simbol yang dikerjakan. Bagan Alir Program ditunjukkan pada Tabel 2.4

Tabel 2.4 Simbol Bagan Alir Program

SIMBOL	KETERANGAN
Input/Output	Digunakan untuk mewakili data *input* atau *output*
Proccess 	Digunakan untuk mewakili suatu proses
Line 	Digunakan untuk menunjukkan arus dari proses
On-page/Off-page reference 	Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang masih sama atau dihalaman berikutnya
Decision 	Digunakan untuk menyelesaikan kondisi didalam program
Prosedur 	Digunakan untuk menunjukkan suatu oprasi yang rinciannya ditujukan ketempat lain

2.3 Tanaman Jahe

Jahe merupakan tanaman obat berupa tumbuhan rumpun berbatang semu. Jahe berasal dari Asia Pasifik yang tersebar dari India sampai Cina. Oleh karena itu kedua bangsa ini disebut-sebut sebagai bangsa yang pertama kali memanfaatkan jahe terutama sebagai bahan minuman, bumbu masak dan obat-obatan tradisional.

Jahe termasuk dalam suku temu-temuan (*Zingiberaceae*), se-famili dengan temu-temuan lainnya seperti temu lawak (*Cucuma xanthorrizha*), temu hitam (*Curcuma*

aeruginosa), kunyit (*Curcuma domestica*), kencur (*Kaempferia galanga*), ataupun lengkuas (*Languas galanga*).

2.4 Hama dan Penyakit Jahe

a. Hama

Hama yang dijumpai pada tanaman jahe adalah:

Kepik, menyerang daun hingga berlubang-lubang dan tanaman menjadi kering.

Ulat penggesek akar, menyerang akar tanaman jahe hingga menyebabkan tanaman jahe menjadi kering dan mati.

b. Penyakit

1) Penyakit layu bakeri

Gejala:

Mula-mula helaian daun bagian bawah melipat dan menggulung kemudian terjadi perubahan warna dari hijau menjadi kuning dan mengering. Kemudian akar dan batang serta tunas menjadi busuk dan akhirnya tanaman mati rebah. Bila diperhatikan, rimpang yang sakit itu berwarna gelap dan sedikit membusuk, kalau rimpang dipotong akan keluar lendir berwarna putih susu sampai kecoklatan. Penyakit ini menyerang tanaman jahe pada umur 3-4 bulan dan yang paling berpengaruh adalah faktor suhu udara yang dingin, genangan air dan kondisi tanah yang terlalu lembab.

Pengendalian:

Jaminan kesehatan bibit jahe

Karantina tanaman jahe yang terkena penyakit

Pengendalian dengan pengolahan tanah yang baik

Pengendalian fungisida dithane M-45 (0,25%), Bavistin (0,25%). Tanaman yang terserang penyakit layu bakteri ditunjukkan pada Gambar 2.6

Gambar 2.6 Penyakit layu bakeri

2) Penyakit busuk rimpang

Penyakit ini dapat masuk ke bibit rimpang jahe melalui lukanya. Ia akan tumbuh dengan baik pada suhu udara 20-25 derajat C dan terus berkembang akhirnya menyebabkan rimpang menjadi busuk.

Gejala:

Daun bagian bawah yang berubah menjadi kuning lalu layu dan akhirnya tanaman mati.

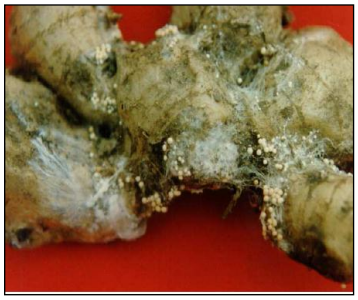
Pengendalian:

penggunaan bibit yang sehat

penerapan pola tanam yang baik

penggunaan fungisida.

Tanaman yang terserang penyakit busuk rimpang ditunjukkan pada Gambar 2.7



Gambar 2.7 Tanaman Terserang Penyakit Busuk Rimpang

3) Penyakit bercak daun

Penyakit ini dapat menular dengan bantuan angin, akan masuk melalui luka maupun tanpa luka.

Gejala:

Pada daun yang bercak-bercak berukuran 3-5 mm, selanjutnya bercak-bercak itu berwarna abu-abu dan di tengahnya terdapat bintik-bintik berwarna hitam, sedangkan pinggirnya busuk basah. Tanaman yang terserang bisa mati.

Pengendalian:

baik tindakan pencegahan maupun penyemprotan penyakit bercak daun sama halnya dengan cara-cara yang dijelaskan di atas.

Tanaman yang terserang penyakit bercak daun ditunjukkan pada gambar 2.8

Gambar 2.8 Tanaman Terserang Penyakit Bercak Daun

4) Penyakit Buncak Akar

Gejala :

Akar luka sehingga penyerapan hara terganggu, patogen tanah mudah masuk, akar dan batang tanaman menjadi kering, akar dan batang tanaman menjadi busuk dan keropos.

Pengendalian :

Bibit diambil dari tanaman induk sehat

Pasteuria penetrans (2-5 kapsul/tanaman/6 bulan)

Tepung biji mimba (25-50 g/tanaman/3 bulan)

Mulsa (10-20 ton/ha) dan karbofuran pada saat tanam 20 – 30 kg/ha

5) Penyakit Lalat Rimpang

Gejala :

Rimpang keriput, busuk dan keropos

Pengendalian :

Perlakuan benih dengan air panas 50ºC selama 10 menit, 40 ºC selama 20 menit, atau dengan insektisida botani seperti ekstrak mimba 2.5 dan ekstrak bungkil jarak 2.5%.

Penyemprotan dengan diklorfos, interval 3 minggu.

Karbofuran 20 – 30 kg/ha pada saat tanam. Tanaman yang terserang penyakit lalat rimpang ditunjukkan pada Gambar 2.9

Gambar 2.9 Tanaman Terserang Penyakit Lalat Rimpang

6) Penyakit Kutu Perisai

Gejala :

Kulit rimpang kusam, karena rimpang dihisap dan kering.

Pengendalian :

Perlakuan fumigasi benih dengan metil bromida atau aluminium fosfida

Perlakuan benih dengan air panas 50oC selama 10 menit, insektisida karbosulfan (2ml/l), insekti sida botani (seperti ekstrak mimba 2.5% dan atau ekstrak bungkil jarak (2.5%).

Tanaman yang terserang penyakit kutu perisai ditunjukkan pada Gambar 2.10

Gambar 2.10 Tanaman Terserang Penyakit Kutu Perisai

2.5 Perangkat Lunak yang Digunakan Dalam Pembuatan Sistem

Dalam perancangan sistem yang baru ini, digunakan perangkat lunak *Macromedia Dreamweaver, Microsoft MySQL, XAMPP dan PHP*

2.5.1 *Macromedia Dreamweaver*

Dreamweaver adalah sebuah *editor* program yang komplit yang dapat digunakan untuk membuat animasi sederhana yang berbentuk *layer*. Dengan adanya program ini kita tidak akan susah-susah mengetik *script-script* format *HTML*, *PHP*, *ASP* maupun program lainnya (Bunafit Nugroho, 2004).

2.5.2 *Microsoft MySQL*

MySQL merupakan software sistem manajemen database (*Database Management System* - *DBMS*) yang sangat populer di kalangan pemrograman *web*, terutama di lingkungan linux dengan menggunakan *script PHP* dan *Perl*. *Software* database ini kini telah tersedia juga pada *platform* sistem operasi *windows* (98/ME atupun NT/2000/XP) (Betha Sidik, 2005).

MySQL merupakan *database* yang paling populer digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya. Kepopuleran *MySQL* dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah kecil. *MySQL* merupakan *database* yang digunakan oleh situs-situs terkemuka di *internet* untuk menyimpan datanya (Betha Sidik, 2005).

2.5.3 *PHP*

Menurut Sunarfrihanto (2002) , *PHP* adalah bahasa *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* yang dinamis. Pembuatan *web* dengan menggunakan *PHP* sebagai bahasa pemograman dan *HTML* sebagai pembangun halaman *web*.

Menurut Kadir (2002), *PHP* merupakan bahasa berbentuk *script* yang ditempatkan pada *server* dan diproses *server* lalu hasilnya dikirim ke *client*, dimana *client* menerima atau melihat hasil yang dikirimkan oleh *server* dengan menggunakan *web browser*.

2.5.4 XAMPP

XAMPP merupakan salah satu *software* yang dapat digunakan untuk mengatasi masalah *PHP* dan *MySQL* jika dirasa sulit. *PHP*, *MySQL*, *Apache* tergabung dalam satu *software* yaitu *XAMPP*. Dalam peng-*install*-an *XAMPP* tidak perlu melakukan perubahan pada *register Windows* dan tidak perlu meng-*edit file* konfigurasinya. Jika telah meng-*install XAMPP* tidak perlu meng-*install PHP*, *MySQL* ataupun *Apache* (XAMPP, 2008).

XAMPP merupakan kependekan dari X yaitu *multi platform*, bisa *Windows*, *Linux*, *Mac OS*, maupun *Solaris*, A yaitu *Apache HTTP Server*, M yaitu *MySQL Database Server*, P yaitu *PHP Scripting Language*, dan P selanjutnya yaitu *Perl Scripting Language*, dengan adanya *XAMPP* mempermudah *user* untuk membuat *server* lokal sendiri (XAMPP, 2008).