

BAB II

TINJAUAN PUSTAKA

Pada bab ini dijelaskan tentang penelitian-penelitian terdahulu yang menerapkan sistem pakar dengan metode yang berbeda untuk mendiagnosa sebuah penyakit. Pada bab ini juga dijelaskan tentang teori-teori yang digunakan sebagai landasan pemikiran untuk membangun sebuah aplikasi sistem pakar berbasis *web*.

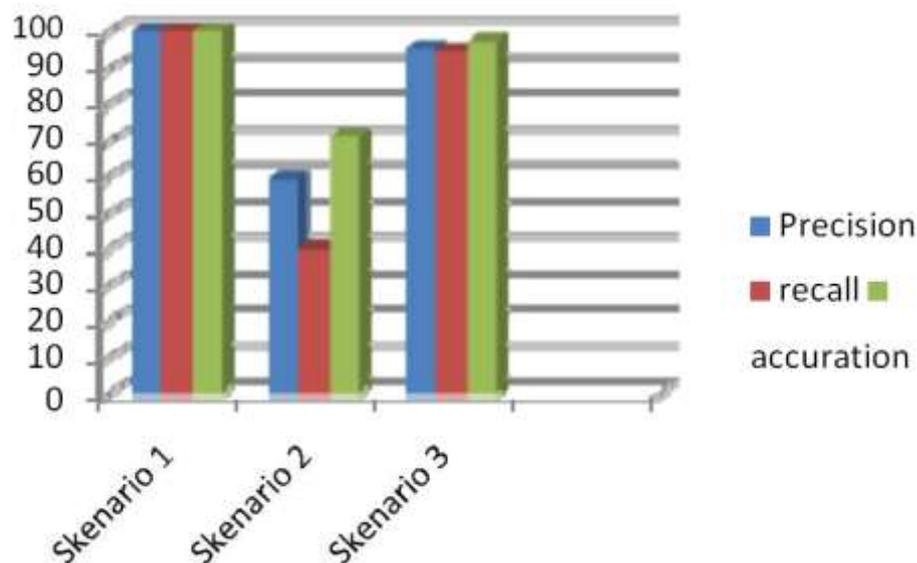
2.1 Penelitian Sebelumnya

Sistem pakar sebelumnya pernah digunakan untuk melakukan beberapa penelitian dengan berbagai metode-metode yang berbeda, salah satunya adalah Penelitian tentang sistem pakar identifikasi penyakit pada tanaman padi, yang dilakukan oleh Naufal Irfan pada tahun 2016, dengan judul “Aplikasi sistem pakar identifikasi penyakit pada tanaman padi menggunakan metode *certainly factor*”. Pada penelitian ini, peneliti menggunakan metode *certainly factor* untuk menentukan hasil identifikasi. Beberapa *tools* pendukung juga digunakan seperti PHP sebagai bahasa pemrograman, MySQL sebagai penyimpanan *database*. Dan XAMPP sebagai server lokal. Hasil pengujian fungsional pada aplikasi ini sudah berjalan 100% serta perhitungan tingkat kevalidan sesuai hasil dari para pakar. Pengujian *user* didapat 82,67% yang menyatakan baik, 14,67% menyatakan cukup, dan 2,67% yang menyatakan kurang. Berdasarkan hasil tersebut sistem pakar identifikasi penyakit padi ini dapat diterima dengan baik.

Penelitian yang kedua dilakukan oleh Prakoso dkk. pada tahun 2012. dengan judul “Penerapan *case-based reasoning* pada sistem cerdas untuk pendeteksian dan penanganan dini penyakit sapi”. Penelitian ini bertujuan untuk mendeteksi dan mencari cara penanganan dini untuk mencegah tingginya kerugian yang terjadi akibat penyakit sapi. *Case-based reasoning* (CBR) memberikan hasil diagnosa berdasarkan permasalahan terdahulu yang dapat direvisi untuk memecahkan permasalahan terbaru. Dari ketiga uji coba dengan *case* didalam *case memory*(skenario 1), diluar *case memory*(skenario 2), dan gejala parsial dari *case memory*(skenario 3) mendapatkan hasil yang baik dengan nilai *precision*

100% dan 95.83% untuk skenario 1 dan 3. Serta nilai precision yang memang kurang baik untuk skenario 2 sebesar 59.31%. Dalam studi kasus ini, saat menerapkan metode CBR, ketersediaan *case* yang lengkap dan variatif di dalam *case memory* dapat memberikan pengaruh sangat besar dalam peningkatan ketepatan hasil diagnosa karena CBR mampu mengidentifikasi *case* berdasarkan kemiripan dengan sangat baik, baik *case* yang diinputkan memiliki kesamaan identik maupun parsial dengan *case* yang ada di dalam *case memory*. Namun kelemahan menggunakan metode CBR adalah tidak menjamin solusi terbaik atau optimum karena penalaran ini berdasarkan kasus lampau, jika solusi lampau salah maka tahapan revisi sangat diperlukan untuk mengurangi tingkat kesalahannya. Pengujian sistem cerdas berbasis CBR ini dilakukan dengan menggunakan tiga skenario pengujian yang dijabarkan sebagai berikut:

1. Pengujian menggunakan *case* yang terdapat di dalam *case memory*.
2. Pengujian menggunakan *case* diluar *case memory*.
3. Pengujian menggunakan *case* dengan gejala parsial dari *case memory*.



Gambar 2.1. Grafik Perbandingan *Precision*, *Recall*, dan *Accuration* Dari Ketiga Skenario (Prakoso dkk., 2012)

Pada gambar 2.1 dapat dilihat bahwa pada skenario 1, dengan melakukan uji coba dengan mengujikan kembali *case* yang sudah terdapat di dalam *case memory* memiliki hasil precision, recall, dan *accuration* yang paling tinggi.

Sedangkan skenario 3 memiliki hasil sedikit lebih rendah dibandingkan dengan skenario 1 namun dengan nilai precision, recall, dan accuracy yang masih sangat baik. Sedangkan skenario 2 dengan melakukan uji coba menggunakan case yang tidak terdapat dalam case memory memiliki hasil precision, recall, dan accuracy yang paling rendah dan kurang baik dibandingkan dengan skenario 1 dan 3. Hal ini menunjukkan bahwa sistem cerdas berbasis CBR memiliki kinerja yang baik untuk memberikan solusi yang akurat baik ketika menyelesaikan permasalahan dengan ciri-ciri yang identik dengan case yang sudah terdapat di dalam case memory, maupun menyelesaikan permasalahan dengan ciri-ciri yang sebagian sama dengan case yang sudah ada di dalam case memory, Sedangkan untuk memecahkan masalah dengan ciri-ciri yang berbeda dengan case yang ada di dalam case memory sistem cerdas kurang dapat memberikan hasil diagnosa dan solusi yang akurat. Oleh karena itu dari hasil yang didapatkan pada ketiga skenario diatas dapat disimpulkan bahwa proses pembelajaran pada CBR merupakan hal yang penting untuk menjaga agar sistem cerdas mampu memberikan hasil diagnosa dan solusi yang baik pada permasalahan baru.

Tampilan desain antarmuka (*interface*) hasil diagnosa sistem cerdas pendeteksian dan penanganan dini penyakit pada sapi menggunakan metode CBR dapat dilihat pada gambar 2.2. Dimana *output* yang dihasilkan berupa nama penyakit, solusi untuk penanganannya beserta konfirmasi *user* tentang hasil yang diberikan oleh sistem. Apakah hasil diagnosa benar atau salah.



Gambar 2.2 Tampilan antarmuka hasil diagnosa dan solusi (Prakoso dkk., 2012)

Sedangkan penelitian yang dilakukan oleh Windah Supartini pada tahun 2016 adalah memanfaatkan sistem pakar dengan menggunakan metode *forward*

chaining untuk mendiagnosis dini penyakit *tuberculosis*. Pada penelitian ini diagnosis sistem pakar, memiliki nilai keakuratan 93,333 % dan nilai eror 6,667 % untuk uji coba pada 15 pasien. Sehingga dapat disimpulkan bahwa sistem pakar cukup layak untuk digunakan oleh pasien dalam mendiagnosis dini pada penyakit *tuberculosis*. Sistem pakar yang menggunakan metode Forward Chaining berbasis *web* ini cukup membantu untuk mendiagnosis penyakit *tuberculosis* berdasarkan gejala-gejala yang dikeluhkan oleh pasien. Proses diagnosa *user* menjawab pertanyaan-pertanyaan berupa gejala yang dikeluhkan kemudian sistem akan melakukan diagnosa berdasarkan relasi antara data penyakit dan data gejala.

Gusviantoko pada tahun 2011 juga melakukan penelitian tentang sistem pakar dengan judul “Membangun sistem cerdas menggunakan *Backpropagation Artificial Neural Network (ANN)*”. Namun metode ANN memiliki kelemahan karena memerlukan jumlah data yang besar yang diperlukan dalam fase *training* dan dalam memberikan hasil diagnosa bergantung sepenuhnya terhadap data yang diolah dalam fase *training*. ANN juga memiliki masalah dengan *overfitting* karena ANN menggunakan pola dari hasil perhitungan bobot data *training* untuk menghasilkan hasil diagnosa. ANN juga memiliki keterbatasan yang disebabkan adanya *hidden layer* sehingga ANN disebut model *black-box*, sehingga representasi *knowledge* tidak terlihat. Padahal dalam proses diagnosa penyakit diperlukan penjabaran ciri-ciri gejala yang menyertai hasil diagnosa dan dapat melakukan evaluasi apakah hasil diagnosa tersebut perlu dilakukan modifikasi atau tidak. *Case-based Reasoning (CBR)* memiliki kemampuan diagnosa berbasis kasus dan memberikan informasi secara otomatis berdasarkan pengetahuan terdahulu yang dapat direvisi untuk menyesuaikan dengan permasalahan terbaru. Sehingga pengetahuan CBR akan terus berkembang. Pemecahan masalah baru pada CBR dilakukan dengan cara mencari permasalahan sejenis di masa lampau dan memberikan solusi berdasarkan permasalahan yang paling mirip yang ada di dalam *case memory*. permasalahan yang dapat digunakan untuk memecahkan masalah tersimpan di dalam *case memory* yang dapat direvisi untuk memecahkan permasalahan di masa datang.

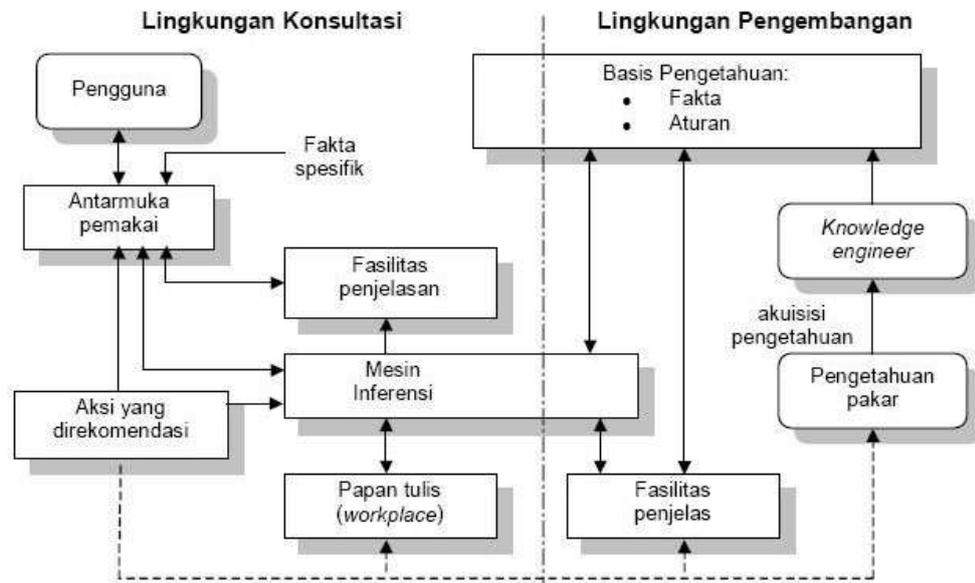
2.2 Pengertian Sistem Pakar

Sistem Pakar adalah sebuah program komputer yang mencoba meniru atau

mensimulasikan pengetahuan (*knowledge*) dan keterampilan (*skill*) dari seorang pakar pada area tertentu. Selanjutnya sistem ini akan mencoba memecahkan suatu permasalahan sesuai dengan kepekarannya (Subakti dan Irvan, 2006).

Sistem pakar adalah sistem perangkat lunak komputer yang menggunakan ilmu, fakta, dan teknik berpikir dalam pengambilan keputusan untuk menyelesaikan masalah-masalah yang biasanya hanya dapat diselesaikan oleh tenaga ahli dalam bidang yang bersangkutan (Marimin, 2002).

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 (Turban,1995). Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Sistem pakar memiliki banyak definisi, tetapi pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah (Turban,1995).



Gambar 2.3 Struktur Sistem Pakar Menurut Turban (Turban., 1995)

Sistem pakar dapat ditampilkan dengan dua lingkungan, yaitu lingkungan pengembangan dan lingkungan konsultasi, lihat pada gambar 2.3. 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 (Turban, 1995).

2.2.1 Konsep Dasar dalam Sistem Pakar

Menurut Sutojo dan Mulyanto (2010) konsep dasar sistem pakar meliputi enam hal, antara lain adalah sebagai berikut :

1. Kepakaran (*Expertise*)

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan tentang:

- a. Fakta-fakta tentang bidang permasalahan tertentu.
- b. Teori-teori tentang bidang permasalahan tertentu.
- c. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya.
- d. Aturan *heuristic* yang harus dikerjakan dalam suatu situasi tertentu.
- e. Strategi global untuk memecahkan permasalahan.
- f. Pengetahuan tentang pengetahuan (*meta knowledge*).

2. Pakar (*Expert*)

Pakar adalah seorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal yang berkaitan dengan topic permasalahan. Kemampuan seorang pakar minimal dapat mengenali (*recognizing*) dan merumuskan masalah, menjelaskan solusi, belajar dari pengalaman dan memahami batas kemauan. Sumber pengetahuan potensial antara lain pakar manusia, buku teks, dokumen multimedia, *database* (*public* dan *privat*), laporan riset khusus, dan informasi yang terdapat dalam *web*.

3. Pemindahan Keahlian (*Transferring Expertise*)

Tujuan dari sistem pakar adalah memindahkan keahlian dari seseorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan yaitu :

- a. Akuisisi pengetahuan (dari pakar atau sumber lain).
- b. Representasi pengetahuan (pada komputer).
- c. Inferensi pengetahuan.
- d. Pemindahan pengetahuan ke pengguna.

4. Inferensi (*Inferencing*)

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan.

5. Aturan (*Rule*)

Kebanyakan *software* sistem pakar komersional adalah sistem yang berbasis rule (*rule-based sistem*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.

6. Kemampuan Menjelaskan (*Explanation Capability*)

Fasilitas lain dari sistem pakar adalah kemampuan untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukannya dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya. Karakteristik dan kemampuan yang dimiliki oleh sistem pakar berbeda dengan sistem konvensional. Menurut Sotojo (2010) Perbedaan sistem konvensional dan sistem pakar ditunjukkan pada tabel 2.1:

Tabel 2.1 Perbandingan Antara Sistem Konvensional Dengan Sistem Pakar (Sotojo., 2010)

Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesan biasanya digabungkan dalam satu program.	Basis pengetahuan dipisahkan secara jelas dengan mekanisme inferensi.
Program tidak membuat kesalahan (yang membuat kesalahan: pemrograman atau pengguna)	Program dapat berbuat kesalahan.
Biasanya tidak menjelaskan mengapa data masukan diperlukan atau bagaimana output dihasilkan.	Penjelasan merupakan bagian terpenting dari sistem pakar.
Perubahan program sangat menyulitkan.	Perubahan dalam aturan-aturan mudah untuk dilakukan.
Eksekusi dilakukan langkah demi langkah (algoritmik).	Eksekusi dilakukan dengan menggunakan <i>heuristic</i> dan logika pada seluruh basis pengetahuan.
Perlu informasi lengkap agar bisa beroperasi.	Dapat beroperasi dengan informasi yang mengandung ketidakpastian.
Manipulasi efektif dari basis data yang besar.	Manipulasi efektif dari basis pengetahuan 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 numerik atau informasi	Menangkap, menambah, dan mendistribusikan akses ke pertimbangan dan pengetahuan.

2.2.1 Komponen-Komponen Sistem Pakar

Menurut Aziz (2004) komponen-komponen sistem pakar terdiri dari:

1. Basis Pengetahuan

Merupakan inti dan sistem pakar dimana basis pengetahuan merupakan representasi pengetahuan dan dapat juga untuk menyimpan, mengorganisasikan pengetahuan dari seorang pakar. Basis Pengetahuan ini tersusun atas fakta yang berupa informasi, tentang obyek dan kaidah (rule) yang merupakan informasi tentang cara bagaimana membangkitkan fakta baru dari fakta yang sudah diketahui. Menurut Gondran (1986) Basis pengetahuan merupakan representasi pengetahuan dari seorang pakar. Yang kemudian dapat dimasukkan kedalam bahasa pemrograman khusus untuk kecerdasan buatan (misalnya prolog atau lips) atau cangkang (*shell*) sistem pakar

2. Basis Data

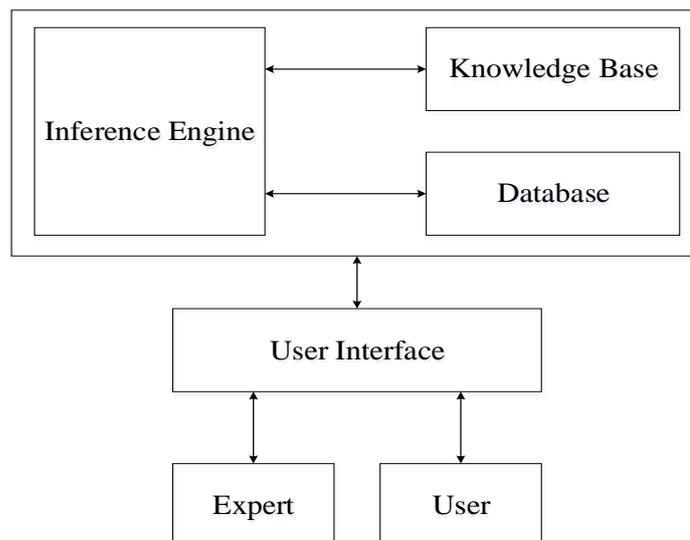
Merupakan bagian yang mengandung semua fakta-fakta baik fakta awal pada saat sistem mulai beroperasi maupun fakta yang didapatkan pada saat pengambilan kesimpulan yang sedang dilaksanakan. Dalam prakteknya, Basis data berada di dalam memori komputer. Kebanyakan sistem pakar mengandung basis data untuk menyimpan data hasil observasi dan data lainnya yang dibutuhkan untuk pengolahan.

3. Mesin Inferensi

“Brain” pada sistem pakar adalah mesin inferensi. Mesin inferensi dikenal sebagai struktur kontrol atau interpreter dan rule (dalam rule-base sistem pakar). Komponen ini secara esensial merupakan program komputer yang menyediakan metodologi untuk reasoning tentang informasi dalam basis pengetahuan dan untuk kesimpulan. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model dan fakta yang disimpan pada basis pengetahuan dalam rangka mencapai solusi atau kesimpulan. Dalam sistem pakar terdapat dua strategi dalam mesin inferensi, yaitu strategi penalaran dan strategi pengendalian.

4. Antarmuka Pemakai (*user interface*)

Merupakan tatap muka pengguna berupa program komputer yang telah diprogramkan dengan algoritma tertentu seperti *Visual Basic*, *Delphi*, *PHP*, *Codeigniter*, *Bootstrap* dan sebagainya.



Gambar 2.4 Struktur Dari Sistem Pakar (Aziz., 2004)

Keterangan:

1. *Knowledge Base* : Basis Pengetahuan yang dimiliki oleh seorang pakar yang merupakan bagian terpenting dalam Sistem Pakar.
2. *Database* : Basis data mencatat semua fakta – fakta baik dari awal pada saat sistem mulai beroperasi atau fakta yang didapat dari hasil kesimpulan.
3. *Inference Engine* : Pembangkit inferensi merupakan mekanisme analisa dari sebuah masalah tertentu yang selanjutnya mencari jawaban dari kesimpulan terbaik.
4. *User Interface* : Bagian ini merupakan sarana komunikasi antar pemakai dan sistem.

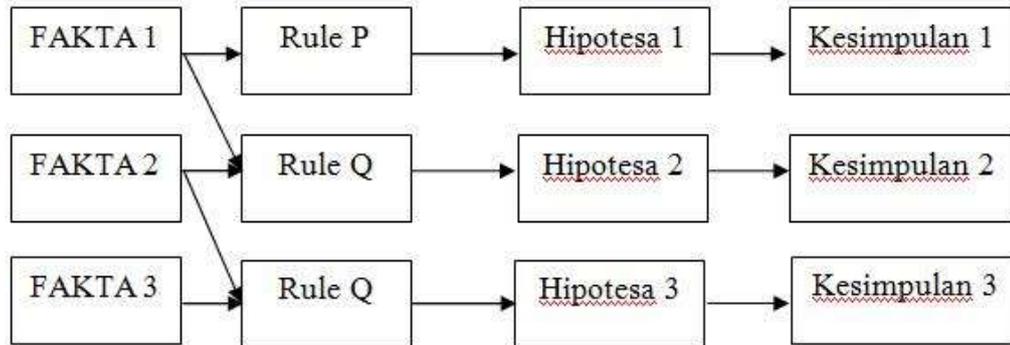
2.3 Metode *Forward Chaining*

Metode *Forward Chaining* adalah metode pencarian atau pelacakan ke depan yang dimulai dengan informasi yang ada dengan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. *Forward chaining* menggunakan pendekatan berorientasi data. Dalam pendekatan ini dimulai dari informasi yang tersedia, atau dari ide dasar, kemudian mencoba menggambarkan kesimpulan. Komputer akan mencoba menganalisa permasalahan dengan mencari fakta yang cocok dengan bagian IF dari aturan IF-THEN. Metode *forward chaining* merupakan suatu strategi pengambilan keputusan yang dimulai dari bagian kiri (IF terlebih dahulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis (Tutik A,2009). Berikut aturan dasar *forward chaining*:

Rule Base	Workspace
R1: IF A AND B THEN D	A,B
R2: IF B THEN C	
R3: IF C AND D THEN E	

Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri. Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu, lalu dicari *rule* yang sesuai

dengan fakta – fakta yang diberikan untuk menguji kebenaran hipotesa. Metode *forward chaining* akan ditunjukkan pada gambar 2.5.



Gambar 2.5 Konsep Penggunaan Metode *Forward Chaining* (Tutik A, 2009)

Metode inferensi ini yang akan digunakan dalam system pakar yang akan dibangun dengan contoh penalaran sebagai berikut:

IF Demam Menggigil AND Nyeri Badan AND Lelah Berlebih THEN

Anthrax

Penjelasan:

Jika sapi mengalami demam menggigil disertai nyeri badan dan lelah berlebih, maka sapi sedang menderita penyakit ANTHRAX. Jika salah satu gejala tidak diderita oleh sapi maka, kemungkinan sapi tidak menderita ANTHRAX.

2.4 Penyakit pada Hewan Ternak Sapi

Menurut Dodi M (2015) sapi yang menderita penyakit akan menimbulkan kerugian ekonomi yang cukup besar bagi peternak khususnya dan masyarakat luas pada umumnya. Kesehatan ternak adalah suatu keadaan atau kondisi dimana tubuh hewan dengan seluruh sel yang menyusun dan cairan tubuh yang dikandungnya secara fisiologis berfungsi normal. Salah satu bagian yang paling penting dalam penanganan kesehatan ternak adalah melakukan pengamatan terhadap ternak yang sakit melalui pemeriksaan ternak yang diduga sakit. Menurut Dodi M (2015) beragam faktor dapat mempengaruhi kesehatan sapi.

Namun diantara beragam faktor tersebut, lingkungan dan penularan merupakan faktor yang paling banyak membuat ternak sapi terserang penyakit.

2.4.1 Pemeriksaan Penyakit Pada Sapi

Pemeriksaan ternak yang diduga sakit adalah suatu proses untuk menentukan dan mengamati perubahan yang terjadi pada ternak melalui tanda-tanda atau gejala-gejala yang nampak sehingga dapat diambil suatu kesimpulan dan suatu penyakit dapat diketahui penyebabnya. Jenis-jenis pemeriksaan penyakit sapi seperti dapat dilihat pada tabel 2.2:

Tabel 2.2 Ciri-ciri visual ternak sehat dibandingkan dengan ternak sakit
(Dodi M., 2015)

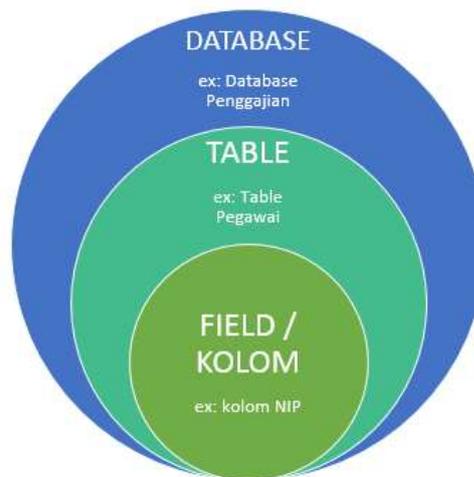
No	Kategori	Sehat	Sakit
1	Pergerakan	Aktif dan Lincah	Kurang Aktif dan Lincah
2	Mata	Jernih	Pucat dan Sayu
3	Bulu	Halus dan Bersih	Kasar, Berdiri dan Kusam
4	Nafsu Makan	Normal	Berkurang
5	Lendir Lubang Alami	Tidak Ada	Ada
6	Suara Nafas	Halus, dan Teratur	Ngorok, Tidak Teratur

Pemeriksaan tersebut merupakan pemeriksaan klinis terhadap ciri-ciri fisik gejala yang sedang dialami oleh sapi. Pemeriksaan klinis ini sangat jelas terlihat langsung mengenai gejala yang sedang diderita.

2.5 Pengertian Database

Database atau basis data merupakan sebuah koleksi atau kumpulan dari data yang bersifat mekanis, terbagi, terdefinisi secara formal serta terkontrol. Pengontrolan dari sistem database tersebut adalah terpusat, yang biasanya dimiliki dan juga dipegang oleh suatu organisasi (Gordon C. Everest).

Sistem basis data pada dasarnya dapat dianggap sebagai tempat atau lokasi untuk sekumpulan berkas data yang sudah terkomputerisasi dengan tujuan untuk memelihara informasi, dan juga memuat informasi tersebut, terutama apabila informasi tersebut sedang dibutuhkan. Database pada MySQL terdiri dari table table yang didalamnya berisi field-field atau kolom.



Gambar 2.6 Struktur Database (Gordon C. Everest., 2008)

Struktur Database / Basis Data

1. Data

Sekumpulan fakta mengenai objek tertentu, orang dan lain-lain yang dinyatakan dengan angka, huruf, gambar, film, suara dan sebagainya yang relevan dan belum mempunyai arti.

2. Informasi

Hasil pengolahan data yang konkrit dan sudah mempunyai arti untuk mencapai suatu tujuan tertentu.

3. Tabel

Merupakan hal yang paling mendasar dalam hal penyimpanan data yang terdiri dari field dan record.

4. Field (kolom)

Merupakan elemen dari tabel yang berisikan informasi tertentu yang spesifik tentang sub judul tabel pada sebuah item data. Syarat-syarat pembentukan Field Name pada tabel:

- a. Harus Unik atau Spesifik
- b. Boleh disingkat
- c. Pemisah sebagai pengganti spasi dalam pembentuk field adalah tanda lambang (_) .

Contoh: Kode Barang menjadi KdBarang, KodeBrg, Kd_Brg, Kd_Barang.

Dalam sistem manajemen basisdata, terdapat tiga macam field:

1. Harus diisi (required)
2. Dapat diabaikan (optional)
3. Penghitungan dari field lainnya (calculated).

2.6 Pengertian Web

Website atau yang biasa disebut WWW (*World Wide Web*) adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser* (Yuhfizar,1998).

Web merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup local maupun jarak jauh. Dokumen pada website disebut web page dan link dalam website memungkinkan pengguna bias berpindah dari satu page ke page lain(hyper text), baik diantara page yang disimpan di server yang sama maupun server di seluruh dunia.(Hakim, 2004).

Adapun cara kerja dari *web* adalah:

1. Informasi web disimpan dalam dokumen dalam bentuk halaman-halaman *web* atau *web page*.
2. Halaman *web* tersebut disimpan dalam komputer *server web*.
3. Sementara dipihak pemakai ada komputer yang bertindak sebagai *computer client* dimana ditempatkan program untuk membaca halaman *web* yang ada di server web (*browser*).
4. *Browser* membaca halaman *web* yang ada di *server web*.

2.6.1 Unsur-Unsur Website

Menurut Bahar (2013) untuk menyediakan keberadaan sebuah website, maka harus tersedia unsur-unsur penunjangnya, adalah sebagai berikut:

1. Nama domain (Domain name/URL – Uniform Resource Locator)

Pengertian Nama domain atau biasa disebut dengan *Domain Name* atau URL adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah *website*, atau dengan kata lain *domain name* adalah alamat yang digunakan untuk menemukan sebuah *website* pada dunia internet. Nama domain

diperjualbelikan secara bebas di internet dengan status sewa tahunan. Nama domain sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan *website* tersebut. Contoh nama domain ber-ekstensi internasional adalah com, net, org, info, biz, name, ws. Contoh nama domain ber-ekstensi lokasi Negara Indonesia adalah co.id (untuk nama domain website perusahaan), ac.id (nama domain website pendidikan), go.id (nama domain website instansi pemerintah), or.id (nama domain website organisasi).

2. Rumah tempat *website* (*Web hosting*)

Pengertian *Web Hosting* dapat diartikan sebagai ruangan yang terdapat dalam *harddisk* tempat menyimpan berbagai data, file-file, gambar dan lain sebagainya yang akan ditampilkan di *website*. Besarnya data yang bisa dimasukkan tergantung dari besarnya *web hosting* yang disewa/dipunyai, semakin besar *web hosting* semakin besar pula data yang dapat dimasukkan dan ditampilkan dalam *website*. *Web Hosting* juga diperoleh dengan menyewa. Besarnya hosting ditentukan ruangan *harddisk* dengan ukuran MB(*Mega Byte*) atau GB(*Giga Byte*). Lama penyewaan *web hosting* rata-rata dihitung per tahun. Penyewaan *hosting* dilakukan dari perusahaan-perusahaan penyewa *web hosting* yang banyak dijumpai baik di Indonesia maupun Luar Negri.

3. Bahasa Program (*Scripts Program*)

Adalah bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website* yang pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis atau interaktifnya sebuah *website*. Semakin banyak ragam bahasa program yang digunakan maka akan terlihat *website* semakin dinamis, dan interaktif serta terlihat bagus.

Beragam bahasa program saat ini telah hadir untuk mendukung kualitas *website*. Jenis jenis bahasa program yang banyak dipakai para desainer *website* antara lain HTML, ASP, PHP, JSP, Java Scripts, Java applets dsb. Bahasa dasar yang dipakai setiap situs adalah HTML sedangkan PHP, ASP, JSP dan lainnya merupakan bahasa pendukung yang bertindak sebagai pengatur dinamis, dan interaktifnya situs. Bahasa program ASP, PHP, JSP atau lainnya bisa dibuat sendiri. Bahasa program ini biasanya digunakan untuk membangun portal berita,

artikel, forum diskusi, buku tamu, anggota organisasi, email, mailing list dan lain sebagainya yang memerlukan update setiap saat.

4. Desain *website*

Setelah melakukan penyewaan *domain name* dan *web hosting* serta penguasaan bahasa program (*scripts program*), unsur *website* yang penting dan utama adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*. Desain sangat berpengaruh kepada penilaian pengunjung akan bagus tidaknya sebuah *website*. Untuk membuat *website* biasanya dapat dilakukan sendiri atau menyewa jasa *website designer*. Saat ini sangat banyak jasa *web designer*, terutama di kota-kota besar. Perlu diketahui bahwa kualitas situs sangat ditentukan oleh kualitas *designer*. Semakin banyak penguasaan *web designer* tentang beragam program/*software* pendukung pembuatan situs maka akan dihasilkan situs yang semakin berkualitas, demikian pula sebaliknya. Jasa *web designer* ini yang umumnya memerlukan biaya yang tertinggi dari seluruh biaya pembangunan situs dan semuanya itu tergantung kualitas *designer*.