BAB II
KAJIAN PUSTAKA

2.1 Teori Dasar

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

2.1.1 Kecerdasan buatan

(Sutojo, S Mulyanto, Edy Suhartono, 2011) Dimulai dari sejarah berbagai litelatur mengenai kecerdasan buatan atau *artificial intelligence* menyebutkan bahwa ide ini di awali pada awal abad 17 ketika *Rene descartes* mengemukakan bahwa tubuh hewan bukanlah apa – apa melainkan hanya mesin – mesin yang rumit. Kemudian *Blaise Pascal* menciptakan mesin penghitung digital mekanis pertama pada 1642. Selanjutnya pada abad ke 19, *Charles Babbage* dan ada Lovelace bekerja pada mesin penghitung mekanis yang dapat di program. Perkembangan terus berlanjut, *Bertrand Russell* dan *Alfred North Whitehead* menerbitkan Principia Mathematica yang merombak logika formal. *Warren McCulloch dan Walter Pitts* menerbitkan "*Kalkulus Logis Gagasan* yang tetap

ada dalam aktivitas" pada 1943 yang meletakkan pondasi awal untuk jaringan syaraf.

Tahun 1950 – an adalah periode usaha aktif dalam AI. Program AI yang pertama ditulis pada 1951 untuk menjalankan mesin *Farranti Mark I di University Of Manchester (UK).* Sebuah program permainan naskah yang ditulis oleh Christopher Strachey dan program permainan catur yang dtulis oleh *Diet Rich Prinz*. *John McCarthy* membuat istilah " Kecerdasan Buatan" pada konferensi pertama pada tahun 1956 , selain itu dia juga menemukan bahasa pemrograman Lisp. *Alan Turing* memperkenalkan "Turing Test" sebagai sebuah cara untuk mengoprasionalkan test prilaku cerdas. *Joseph Weizenbaum* membangun ELIZA, sebuah Chatterbot yang menerapkan psikoterapi Rogerian.

Selama tahun 1960-an sampai 1970-an , *Joel Moses* mendemonstarasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah didalam program Macsyma, program berbasis pengetahuan yang sukses pertama kali dibidang matematika. *Marvin Minsky* dan *Seymour Papert* menerbitkan Perceptrons, yang mendemonstrasikan batas jaringan syaraf sederhana dan *Alain Colmerauer* mengembangkan bahasa komputer prolog. *Ted Shortliffe* mendemonstrasikan kekuatan sistem berbasis atauran untuk refresentasi pengetahuan dan inferensi dalam diagnosa dan terapi medis yang di yakni sebagai sistem pakar pertama. *Hans Moravec* mengembangkan kendaraan terkendali komputer pertama untuk mengatasi jalan yang mempunyai rintangan secara mandiri. Adapun bagian dari Artificial Intelligance ini adalah sebagai berikut :

1. **Sistem Pakar (*Expert System*)**

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo, S Mulyanto, Edy Suhartono, 2011, p. 13).

2. **Fuzzy Logic**

Fuzzy adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk di implementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, embended system, jaringan PC, multi-channel atau workstasion berbasis akuisisi data, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”. Oleh karena itu, semua in dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi dalam logika fuzzy memungkinkan nilai keanggotaan berada di antara 0 dan 1. Artinya bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimiliknya. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

3. Jaringan saraf tiruan

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

3.1.2. Sistem Pakar

Sistem pakar berasal dari istilah *knowledge-based expert system*. Yaitu suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo, S Mulyanto, Edy Suhartono, 2011a, p. 13). Sistem pakar didefenisikan sebagai sistem komputer yang mempunyai kemampuan meniru pengetahuan dan keterampilan pada seorang pakar tertentu. Secara umum sistem pakar dibangun atas tiga komponen utama, yaitu :

1. *Knowledge base* (basis pengetahuan) merupakan bagian yang ada didalam sebuah sistem pakar yang berfungsi untuk menyimpan pengetahuan. *Knowledge base* yang di simpan oleh sistem pakar

berbeda antara satu dengan yang lain tergantung dari bidang kepakaran yang dibuat.

2. *Working memory* berfungsi untuk menyimpan fakta – fakta yang di dapat dalam proses konsultasi yang dilakukan dengan seorang pakar. *Working memory* akan menyimpan segala informasi yang user masukan berdasarkan hasil konsultasi user dengan seorang pakar.
3. *Inference engine* mempunyai tugas dalam hal mencari fakta – fakta yang terdapat di dalam *working memory* dengan fakta – fakta tertentu yang ada didalam *knowdledge base.*

2.1.1.1. Ciri-Ciri Sistem Pakar

Adapun ciri – ciri yang dimiliki sistem pakar, yaitu:

1. Memiliki informasi yang handal.
2. Mudah di modifikasi.
3. Dapat digunakan dalam berbagai jenis komputer.
4. Memiliki kemampuan untuk belajar beradaptasi.

2.1.1.2. Keuntungan Sistem Pakar

Secara garis besar, banyak manfaat yang diambil dengan adanya sistem pakar, antara lain:

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

2.1.1.3. Kelemahan Sistem Pakar

Disamping mempunyai kelebihan sistem pakar juga mempunyai kelemahan, kelemahan yang dimiliki oleh sistem pakar, sebagai berikut:

1. Biaya yang di perlukan untuk membuat dan memiliharanya sangat mahal.
2. Sulit dikembangkan, hal ini tentu saja erat kaitan nya dengan ketersedian pakar di bidang nya.
3. Sistem pakar tidak 100% bernilai benar.

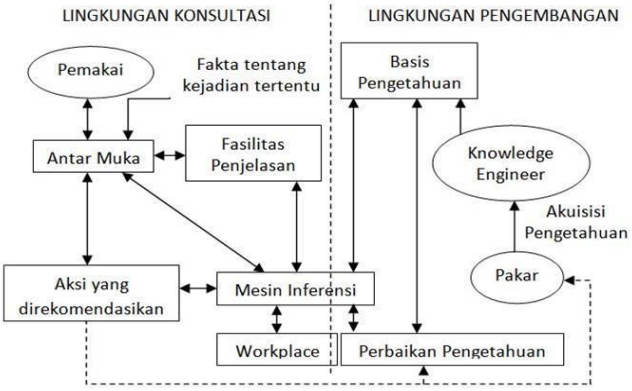
2.1.1.4. Bentuk Sistem Pakar

Ada 4 bentuk sistem pakar yaitu:

1. Berdiri sendiri: sistem pakar jenis ini merupakan software yang berdiri sendiri tidak tergabung dengan software yang lainnya.
2. Tergabung: sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu algoritma (konvensional), atau merupakan program dimana didalam nya memanggil algoritma subrutin lain (konvensional).
3. Menghubung ke software lain: bentuk ini merupakan sistem pakar yang menghubungkan kesuatu paket program tertentu, misalnya dengan DBMS.
4. Sistem mengabdi: sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu, misalnya digunakan untuk membantu menganalisis data radar.

2.1.1.5. Struktur Sistem Pakar

Sistem pakar terdiri dari 2 bagian pokok, yaitu lingkungan pengembangan dan lingkungan konsultasi



Gambar 2.1 Struktur Sistem Pakar
Sumber: (Sutojo, S Mulyanto, Edy Suhartono, 2011)

Berdasarkan gambar diatas, berikut ini adalah penjelasan komponen komponen yang ada pada sistem pakar.(Sutojo, S Mulyanto, Edy Suhartono, 2011 p. 167)

a. Akuisisi Pengetahuan

   Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan).

b. Basis pengetahuan (*Knowledge Base*)

   Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah.

c. Mesin Inferensi (*Inference Engine*)

   Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, an fakta yan disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan.

d. Daerah Kerja (*Blackboard*)

   Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang akan terjadi, sistem pakar membutuhkan *Blackboard*, yaitu area pada memori yang berfungsi sebagai basis data.

e. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling bagus bila disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu, dan formulir elektronik.

f. Subsistem Penjelasan (*Explanation Subsystem / Justifer*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

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

Kemampuan memperbaiki pengetahuan (*Knowledge Refining System*) dari seorang pakar diperlakukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudia memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang.

h. Pengguna (*User*)

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

Untuk membuat sistem pakar yang efektif harus dipilih representasi pengetahuan yang tepat. Pemilihan representasi pengetahuan yang tepat akan membuat sistem pakar dapat mengakses basis pengetahuan tersebut untuk keperluan pembuatan keputusan. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

1. Jaringan Semantik (*Semantic Nets*)

Jaringan semantik merupakan refresentasi pengetahuan yang digunakan untuk menggambarkandata dan informasi, yang menunjukkan hubungan antara berbagai objek. Objek bisa berupa benda fisik seperti mobil, rumah, orang, atau konsep berupa pikiran, kejadian, atau tindakan. Jaringan semantik merupakan grafik, yang terdiri dari simpul-simpul (*nodes*), yang mempresentasikan objek, dan busur-busur (*arcs*) yang menunjukan relasi antara objek-objek tersebut. Jaringan semantik merupakan alat efektif untuk mempresentasikan pemetaan data agar tidak terjadi duplikasi data.

2. Bingkai (*Frame*)

Frame pertama kali ditemukan oleh Marvin Minsky pada 1970. Menurut Minsky, *frame* dapat dipandang sebagai struktur data yang digunakan untuk merepresentasikan pengetahuan yang didasarkan pada karakteristikyang sudah dikenal, yang merupakan pengalaman-pengalaman. *Frame* merupakan kumpulan pengetahuan tentang suatu objek tertentu, peristiwa, lokasi, situasi, dan lain-lain. *Frame* terdiri dari dua elemen dasar, yaitu slot dan subslot. Slot merupakan kumpulan atribut atau properti yang menjelaskan objek yang direpresentasikan oleh *frame.* Sedangkan subslot menjelaskan pengetahuan atau prosedur dari atribut pada slot. Slot dalam *frame* berisi beberapa informasi sebagai berikut:

a. Informasi identifikasi *frame.*
b. Hubungan *frame* dengan *frame* yang lain.
c. Menggambarkan persyaratan yang dibutuhkan *frame.*
d. Informasi prosedural untuk menggunakan struktur yang digambarkan.

e. Informasi *default frame.*

f. Informasi baru.

Sedangkan, subslot terdiri dari beberapa bentuk antara lain:

a. *Value:* nilai dari suatu atribut.

b. *Default:* nilai yang digunakan jika slot kosong atau tidak dideskripsikan pada instansiasi *frame.*

c. *Range:* jenis informasi yang muncul pada slot.

d. *If added:* berisi informasi tindakan yang akan dikerjakan jika nilai slot diisi.

e. *If needed:* subslot ini digunakan pada kasus dimana tidak ada *value* pada slot.

f. *Other:* slot dapat berisi *frame, rule,* jaringan semantik ataupun tipe lain dari informasi.

3.1.3. Representasi Pengetahuan

Pemrosesan yang dilakukan oleh sistem pakar merupakan pemrosesan pengetahuan, bukan pemrosesan data seperti yang dikerjakan dengan pemrograman secara konvensional yang kebanyakan dilakukan oleh sistem informasi. Pengetahuan (*knowledge*) adalah pemahaman secara praktis maupun teoritis terhadap suatu obyek atau domain tertentu(Hartati & Iswanti, 2008 p 18). Berikut ini adalah contoh struktur kaidah *IF-THEN* yang menghubungkan objek. (Hartati & Iswanti, 2008:25)

1. *IF* premis *THEN* konklusi
2. *IF* masukan *THEN* keluaran
3. *IF* kondisi *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan
7. *IF* aksi *THEN* reaksi
8. *IF* gejala *THEN* diagnosa

Premis mengacu pada fakta yang harus benar sebelum konkluksi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Antesenden mengacu situasi yang terjadi sebelum kosekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan menace pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat tertentu. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan.

Sebelum sampai pada bentuk kaidah produksi, terdapat langkah-langkah yang harus ditempuh dari pengetahuan yang didapatkan dalam domain tertentu. Langkah-langkah tersebut adalah menyajikan pengetahuan yang berhasil

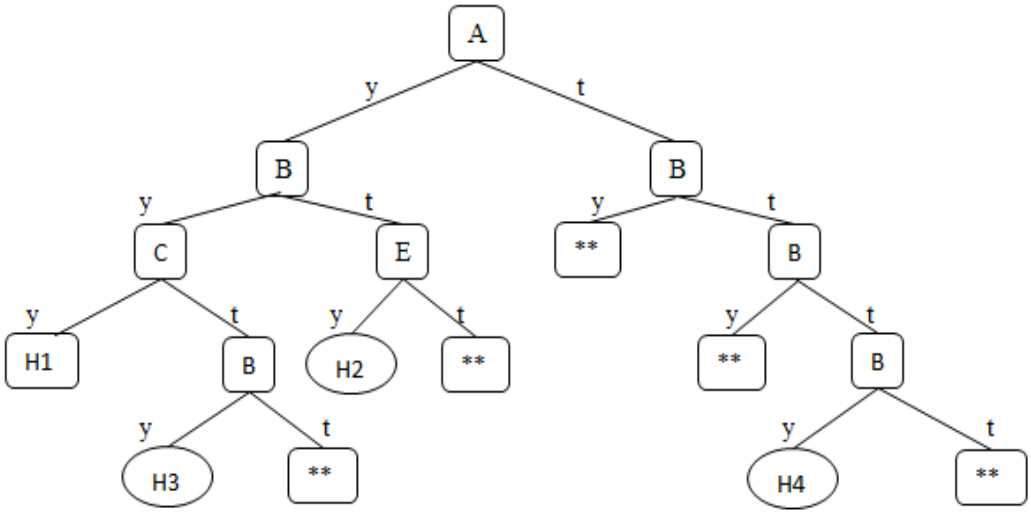
didapatkan dalam bentuk tabel keputusan (*decision table*) kemudian dari tabel keputusan dibuat pohon keputusan (*decision tree*).

Tabel keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Tabel keputusan merupakan matrik kondisi yang dipertimbangkan dalam pendeskripsian kaidah. Meskipun kaidah secara langsung dapat dihasilkan dari table keputusan tetapi untuk menghasilkan kaidah yang efesien terdapat suatu langkah yang harus ditempuh yaitu membuat pohon keputusan terlebih dahulu. Dari pohon keputusan dapat diketahui atribut (kondisi) yang dapat direduksi sehingga menghasilkan kaidah yang efisien dan optimal. Berikut ini variasi contoh terkait dengan penyajian dalam bentuk tabel keputusan dan pohon keputusan.

Tabel 2.1 keputusan

Hipotesa / ***Evidence***	**Hipotesa 1**	**Hipotesa 2**	**Hipotesa 3**	**Hipotesa 4**
Evidence A	Ya	Ya	Ya	Tidak
Evidence B	Ya	Tidak	Ya	Ya
Evidence C	Ya	Tidak	Tidak	Ya
Evidence D	Tidak	Tidak	Tidak	Ya
Evidence E	Tidak	Ya	Ya	Tidak

Sumber: (Hartati & Iswanti, 2008)



Gambar 2.2 Pohon Keputusan
Sumber: (Hartati & Iswanti, 2008)

Keterangan:

A = *evidence* A, H1 = hipotesa 1, y = ya

B = *evidence* B, H2 = hipotesa 2, t = tidak

C = *evidence* C, H3 = hipotesa 3, ** = tidak menghasilkan hipotesa tertentu

D = *evidence* D, H4 = hipotesa 4

Dari gambar 2.2 dapat diketahui bahwa hipotesa H1 terpenuhi jika memenuhi *evidence* A, B, dan C. Hipotesa H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. Hipotesa H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. Hipotesa H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi "y" mengandung arti memenuhi *node* (*evidence*) di atasnya, notasi "t" artinya tidak memenuhi.

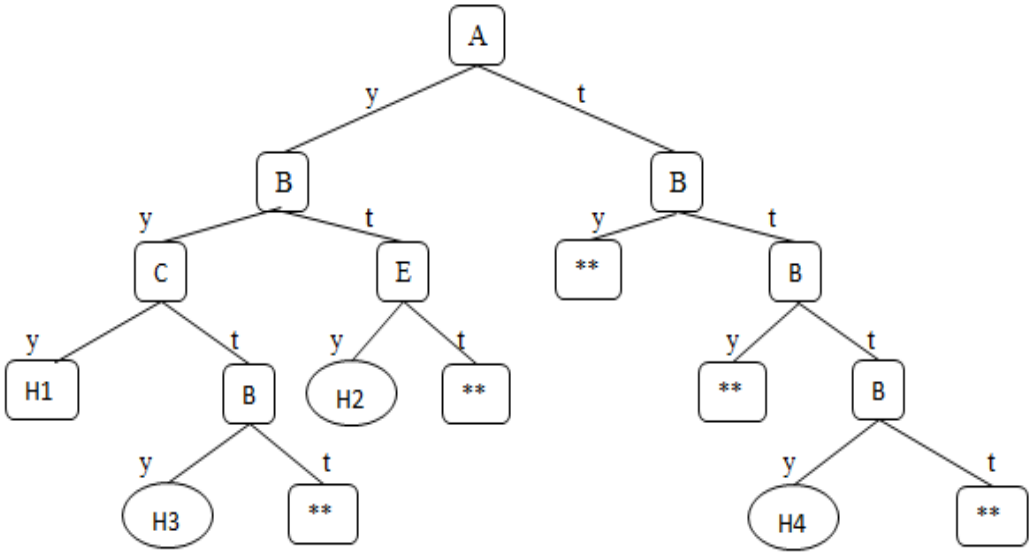
Dalam implementasi sistem pakar terutama dalam sesi konsultasi pada, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.2 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan "apakah memiliki *evidence* A?". Permasalahannya adalah apapun jawaban pengguna baik "ya" atau "tidak" maka sistem akan menanyakan *evidence* B. Ini berarti jawaban pengguna tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.2.

Tabel 2.2 Alternatif Tabel Keputusan

Hipotesa / ***Evidence***	**Hipotesa 1**	**Hipotesa 2**	**Hipotesa 3**	**Hipotesa 4**
Evidence A	Ya	Ya	ya	Tidak
Evidence D	Tidak	Tidak	tidak	ya
Evidence B	Ya	Tidak	ya	Ya
Evidence C	Ya	Tidak	tidak	Ya
Evidence E	Tidak	Ya	ya	Tidak

Sumber: (Hartati & Iswanti, 2008)

Berdasarkan tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



Gambar 2.3 Alternatif Pohon Keputusan
Sumber: (Hartati & Iswanti, 2008a)

Keterangan:

A = *evidence* A, H1 = hipotesa 1, y = ya

B = *evidence* B, H2 = hipotesa 2, t = tidak

C = *evidence* C, H3 = hipotesa 3, ** = tidak menghasilkan hipotesa tertentu

D = *evidence* D, H4 = hipotesa 4

Dilihat dari gambar 2.3, masing-masing *node* yang mewakili *evidence* tertentu untuk kondisi "y" dan "t" sudah tidak mengarah pada *evidence* yang sama. Hal ini berarti jawaban pengguna yang berbeda akan mengarah pada pertanyaan yang berbeda pula.

Proses akuisisi pengetahuan dalam sistem pakar bukanlah hal yang mudah dan sederhana. Pengetahuan yang berhasil didapatkan dan disajikan dalam salah satu bentuk representasi pengetahuan misalnya disajikan dalam bentuk tabel keputusan dan dapat diubah susunannya, secara teori sudah benar tetapi pada domain pengetahuan tertentu, perubahan seperti itu terkadang tidak diperkenankan. Dengan kata lain, pengetahuan yang akan disajikan dalam bentuk representasi tertentu harus mengikuti kaidah-kaidah kepakaran dengan domainnya dan tidak boleh bertentangan.

4. Logika Predikat

Logika pertama kali dikembangkan oleh Aridtoteles merupakan logika formal. Logika formal adalah logika yang berhubungan dengan bentuk (*syntax*) pernyataan bukan pada arti (*semantic*) dari pernyataan. Logika formal berkosentrasi pada penalarannya itu sendri tanpa terganggu oleh obyek yang sedang dinalar. Pada alogika simbolis dikenal aksioma yaitu simbol-simbol yang merepresentasikan obyek dan kelas serta operasi aljabar untuk memanipulasi simbol-simbol tersebut. Logika personal, bisa juga disebut kulkulus proposional, adalah logika simbolis yang memanipulasi proposisi.

Logika propositional hanya mampu menangani pernyataan yang komplit dan tidak biasa menganalisa struktur internal sebuah pernyataan. Sehingga untuk menganalisa kasus yang lebih umum dikembangkan logika predikat yang dapat menganalisis struktur internal kalimat (*sentence*). Bentuk paling sederhana dari logika predikat adalah logika derajat pertama (*first order logic*). Logika derajat pertama tersebut terbentuk dengan menambahkan fungsi atau analisis lain pada

kulkulus predikat. Fungsi merupakan logika yang menghasilkan nilai (*value*). Logika predikat selain digunakan untuk menentukan kebenaran (*truthfulness*) tau kesalahan (*falsity*) sebuah pernyataan, juga dapat digunakan untuk merepresentasikan pernyataan tentang obyek tertentu.

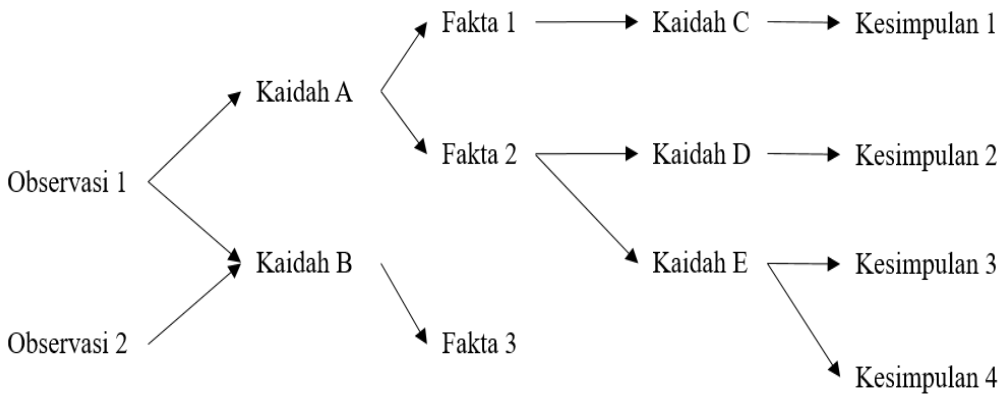
2.1.4. *Database* (Basis Data)

Menurut (A.S. & Shalahuddin, 2011, p. 44) Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data meliputi memasukkan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan. Salah satu bentuk basis data yang dibutuhkan dalam sebuah sistem yaitu *Database Management System* (*DBMS*). *DBMS* adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi syarat minimal sebagai beriku:

1. Menyediakan fasilitas untuk mengelala akses data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data yang dilakukan.
4. Mampu menangani backup data.

2.1.5. *Forward Chaining*

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam database. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. (Sutojo, S Mulyanto,Edy Suhartono, 2011, p. 171)



Gambar 2.4 Diagram *Forward Chaining*
Sumber: (Sutojo, S Mulyanto,Edy Suhartono, 2011)

Kelebihan:

1. Kelebihan utama dari *Forward Chaining* yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan dan menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.

2. Metode ini mampu menyediakan banyak sekali informasi dari hanya jumlah kecil data.

Kelemahan:

1. Kelemahan utama metode ini yaitu kemungkinan tidak adanya cara untuk mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
2. Sistem bisa saja menanyakan pertanyaan yang tidak berhubungan. walaupun jawaban dari pertanyaan tersebut penting. Namun hal ini akan membingungkan user untuk menjawab pada subjek yang tidak berhubungan.

2.1.6. Validasi Sistem

Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa sistem atau perangkat lunak yang dibangun telah sesuai dengan yang diharapkan. Beberapa pendekatan dalam melakukan pengujian untuk validasi sistem antara lain. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

1. *Black-Box Testing* (pengujian kotak hitam)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Tujuannya untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari sistem atau perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem atau perangkat lunak apakah sesuai dengan spesifikasi yang

dibutuhkan.Kasus uji yang dibuat untuk melakukan *black-box testing* harus dibuat dengan kasus benar dan kasus salah.

2. *White-Box Testing* (pengujian kotak putih)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. *White-box testing* dilakukan dengan memeriksa logika dari kode program.Pembuatan kasus uji dapat mengikuti standar pengujian dari standar pemrograman yang ada.

2.2 Penyakit Sapi

Dalam penelitian ini, ada dua variabel yang akan mendukung jalanya penelitian. Variabel yang pertama adalah penyakit *Milk fever (parturient-hypocalcemia)*, dan variabel kedua penyakit *Hypomagnesemia (grass tetany)*, ketiga Penyakit *Asetonemia (Ketosis)*. Dibawah ini adalah penjelasan dari variabel yang akan dibahas untuk memenuhi kebutuhan sistem pakar. Berikut jenis-jenis penyakit yang terdapat pada ternak sapi.

2.2.1. Penyakit *Milk fever (parturient-hypocalcemia)*

Milk fever secara teknis disebut sebagai *parturient hypocalcemia* atau *parturient paresis* yang berarti penurunan kadar kalsium darah pada saat melahirkan. Penyakit ini biasanya disertai dengan penurunan suhu tubuh menjadi subnormal. Kejadian penyakit berlangsung secara akut yang di ikuti dengan penurunan kadar kalsium darah (*hipokalsemia*).

Gambar 2.*5* penyakit *Milk Fever (parturient hypocalcemia)*
Sumber : Data Penelitian (2018)

Gejala paresis muncul seiring dengan penurunan kadar kalsium darah dan diikuti dengan *comatose* (pingsan) (Neves, Leno, Stokol, Overton, & McArt, 2017). Umumnya penyakit ini muncul dalam 3 hari setelah melahirkan.

1. Penyebab penyakit

Banyak faktor yang dapat mempengaruhi timbulnya gejala klinis *Milk fever* pada sapi perah, seperti:

a. Tingkat produksi susu. Tingkat produksi susu yang sangat tinggi sering mengalami *parturient hypocalcemia.*

b. Umur. Bertambahnya umur seekor hewan akan menurunkan tinggat metabolisme umum. Sapi yang lebih tua mengalami penurunan pergantian mineral tulang dan begitu pula kapasitas penyerapan kalsium oleh lambung. Sapi yang lebih tua akan mengalami resiko tinggi terhadap *parturient hypocalcemia.*

c. Asupan (*intake*) diet kalsium sebelum melahirkan. Sapi yang mendapat diet kalsium yang berlebihan akan lebih peka di banding yang menerima diet kalsium yang rendah.

d. Statis saluran pencernaan. Proses laktasi pada sapi perah bergantung pada kondisi fungsional saluran pencernaan dan bila terjadi gangguan dapat menimbulkan *hypocalcemia* klinis.

e. Keseimbangan diet. Keseimbangan diet merupakan faktor predisposisi *parturient hypocalcemia* sebagai contoh status asam atau basa (pH) dari total diet.

2. Gejala yang ditimbulkan

Rendahnya kadar kalsium darah dapat menimbulkan *hipersensitivitas* pada membran syaraf serta otot dan kemudian terjadi *hipereksibilitas* dan *grass tetany*. Namun pada stadium akhir *milk fever* akan terjadi paralisis otot. Sapi perah yang menderita *milk fever* umumnya melalui dua stadium yaitu :

a. Stadium pertama ditandai dengan berbaring pada sternal *(sternal recumbency)*. Dan kepala yang mengarah kebelakang.

b. Stadium kedua melibatkan kolaps dan koma.

3. Pencegahan dan pengobatannya

a. Mengembalikan kadar kalsium darah pada kondisi normal.

b. Jangan memberikan rumput mengandung kalsium yang rendah.

c. Jangan memberikan rumput mengandung magnesium yang rendah.

d. Selama melahirkan biasanya terjadi periode statis lambung dan akan menurunkan kemampun sapi mengabsorbsi kalsium.

2.2.2. Penyakit *Grass Tetany (Hypomagnesemia)*

Grass tetany disebut juga *hipomagnesemia* akut. Penyakit ini bersifat fatal pada ruminansia yang berlangsung secara cepat tanpa gejala klinis sebelum kematian hewan pada lokasi dimana ditemukan terjatuh.

Gambar 2.6 Penyakit *Grass Tetany (Hypomagnesemia)*
Sumber : Data Penelitian (2018)

Grass tetany sering di temukan pada hewan yang di kembalakan di lapangan sapi yang sedang laktasi paling peka terhadap penyakit. Penyakit ini adalah *hypomagnesemia* dimana terjadi penurunan kadar magnesium didalam darah. (Neves et al., 2017)

1. Penyebab penyakit

Grass tetany disebabkan karena menurunnya *(defisiensi)* kadar magnesium dalam darah sapi yang dikembalakan di lapangan. Sapi yang di kandangkan dan diberi *ransum konsentrat* serta rumput kering jarang mengalami grass tetany. Masa laktasi dapat meningkatkan kebutuhan magnesium sehingga sapi laktasi menjadi beresiko untuk mengalami *grass tetany.* Masalah utama adalah tidak cukupnya asupan magnesium dari pakan yang disertai dengan beberapa faktor lainnya dimana terbatas nya penyerapan magnesium dari usus. Secara umum, beberapa faktor yang perlu diperhatikan pada penyakit *grass tetany* adalah :

a. Rumput muda mengandung magnesium yang lebih rendah dibandingkan dengan rumput tua atau kering.
b. Rumput dengan daun tunggal dan ramping mengandung megnesium lebih rendah dari pada rumput dengan daun jamak (*multiple*) dan lebar.
c. Pemupukan rumput dengan nitrogen dan kalium dapat menghambat penyerapan magnesium oleh tanaman dan mencegah penyerapan magnesium oleh mukosa usus.
d. Diet kalsium yang berlebihan dapat menghambat absorbsi magnesium, dan oleh sebab itu pengapuran lahan rumput dapat meningkatkan potensi tetany pada sapi yang di kembalakan.
e. Pakan yang mengantung protein tinggi yang meningkatkan kandungan anomia dalam rumen dapat menghambat penyerapan magnesium disebabkan terjadinya presipitasi *magnesium amonium forsfat*.
f. Beberapa jenis pengikat mineral *(magnesium)* terdapat di dalam saluran pencernaan hewan seperti asam *ketobutirat* atau asam *transakonitat.*

4. Gejala yang ditimbulkan

a. Penyakit ini berlangsung secara akut (sangat cepat) sapi terlihat menderita sangat parah atau mati.
b. Sapi terlihat berlari – lari secara liar dan jatuh dengan gejala konvulsi.
c. *Hyperthermia* kontraksi otot yang parah berkaitan dengan tetany serta detak jantung yang tidak teratur dan berbunyi.
d. Sapi akan mengalami hipersensitif.

e. Sapi akan terlihat galak serta sulit untuk dipegang gejala ini berlangsung beberapa hari sampai memasuki stadium konvulsi.

5. Pencegahan dan pengobatannya

a. Memberikan teraoi magnesium.

b. Segera berikan larutan magnesium sulfat 25% sebanyak 400 ml secara oral.

c. Hindari pemberian secara intravenus karena dapat mempercepat timbulnya serangan jantung yang bersifat fatal.

d. Bila terlihat gejala spasmus, maka berikan sedativa untuk menenangkan hewan dan mengurangi resiko serangan jantung.

e. Berikan kombinasi preparat magnesium dan garam kalsium secara perlahan melalui intravena.

f. Memberikan suplemen magnesium pada masa – masa kritis seperti pada musim hujan dan pengembalaan pada rumput basah.

2.2.3. Penyakit *Asetonemia (Ketosis)*

Asetonemia juga di sebut *ketosis* atau *hypoglisemia ketosis* yang sering ditemukan pada sapi perah pada masa laktasi. *Ketosis* disebabkan karena ketidak seimbangan antara input dan output energi metabolisme.

Gambar 2.7 Penyakit *Asetonemia (Ketosis)*
Sumber : Data Penelitian (2018)

Penyakit ini merupakan gangguan metabolisme yang terjadi pada awal laktasi dan berkaitan erat dengan *hypoglisemia, ketonaemia, ketonuria*, hilang nafsu makan, kehilangan berat badan dan inkoordinasi. (Neves et al., 2017)

1. Penyebab penyakit

Mekanisme asetonemia diawal dengan gangguan metabolisme karbohidrat untuk menghasilkan energi. Pakan yang mengandung karbohidrat tinggi dipecah oleh mikro organisme rumen menjadi asam propionat dan kemudian dibawa kedalam hati untuk digunakan menghasilkan glukosa. Sapi pada masa awal laktasi umumnya tidak mampu mengkonsumsi cukup energi dari diet pakan nya untuk mencukupi kebutuhan energi dalam menghasilkan susu, oleh karena itu, tubuh mendapatkan energi dari cadangan lemak. Lemak dipecahkan menjadi asam lemak kemudian melalui aliran darah dibawa ke jaringan hati.

2. Gejala Yang ditimbulkan

Gejala awal dari *asetonemia* yaitu:

a. penurunan produksi susu dan bau aseton dari pernapasan, susu dan urin sapi penderita.
b. Sapi tidak mau mengkonsumsi konsentrat meskipun tetap mengkonsumsi rumput kering atau silase.
c. Bulu terlihat kusam dan sapi mengalami *lethargy*(lemah).
d. Berhentinya ruminasi dan penurunan produksi susu secara drastis.
e. Aseton dapat mempengaruhi otak dimana sapi menjadi liar.
f. Keluar air liur dari mulut.

g. Menjilati benda sekitar dan berdiri dengan menjulurkan kepala ke atas.

3. Pengobatan dan pencegahan

Pengobatan dan pencegahan penyakit dapat dilakukan melalui tiga tahapan pengobatan yaitu:

a. Penyuntikan obat untuk meningkatkan kadar glukosa dan mempercepat metabolisme hati. Obat yang digunakan umum nya *anabolik steroid* atau kelompok *glukokortikoid.*

b. Obat yang diberikan secara oral untuk meningkatkan kadar gula darah dan memperbaiki metabolisme seperti *sodium propionat* dan *gliserol*.

c. Perawatan individual secara terpisah pada hewan yang sakit dengan memberikan pakan khusus seperti *molases.*

Tabel 2.3 Variabel, Penyebab Dan Indikator

Variabel	Penyebab	Indikator
PENYAKIT SAPI	1. tingkat produksi susu 2. umur 3. asupan 4. saluran pencernaan 5. keseimbangan diet.	1. Penyakit *Milk Fever (Parturient hypocalcemia)*
	1. Menurunnya kadar magnesium.	2. Penyakit *Grass Tetany (Hypomagnesemia)*
	2. Gangguan metabolisme karbohidrat. 3. energi tidak seimbang	3. Penyakit *Asetonemia (Ketosis)*

Sumber: Data Penelitian (2018)

2.3 *Software* Pendukung

Software merupakan istilah umum untuk data yang diformat dan disimpan secara digital, termasuk program komputer. Dokumentasinya dan berbagai informasi yang bisa di baca dan ditulis oleh komputer. Definisi *software* lain ialah intruksi dalam bahasa (formal) pemrograman , disusun oleh programmer untuk dikerjakan komputer.

2.3.1. UML

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

UML (*Unified Modeling Language*) adalah notasi yang lengkap untuk membuat visualisai model suatu sistem sistem berisi informasi dan fungsi, tetapi secara normal digunakan untuk memodelkan sistem komputer didalam pemodelan obyek guna menyajikan sitem yang berorientasi objek pada orang lain. (A.S & Shalahuddin, 2011, p. 113)

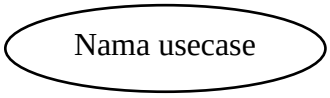
Gambar 2.8 *Logo UML*
Sumber : Data Penelitian (2017)

2.3.1.1. *Use case* Diagram

Menurut (A.S & Shalahuddin, 2011, p.130) Diagram *Use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara garis besar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Ada 2 hal utama pada *use case* antaranya:

1. Aktor merupakan orang yang berinteraksi pada sistem informasi baik pengguna maupun pembuat sistem infromasi tersebut.
2. *Use case* adalah fungsional yang disediakan sistem sebagai tempat bertukar pesan.

Tabel 2.4 Simbol- Simbol Pada *Use case* Diagram

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal *frase* nama *use case*
 Nama aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal *frase* nama aktor.
Asosiasi / *association* 	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
Extensi / *Extend*	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang

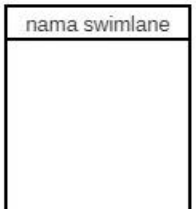
<<*Extend*>> ---------->	ditambah dapat berdiri sendiri walau tanpa *use case* tambahan. Itu biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan.
Generalisasi/generalization ——————▶	Hubungan generalisasi dan spesialisasi (umum-khusus) antara 2 buah *use case* dimana fungsi satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / *include* / *uses* <<*include*>> ---------- > <<*uses*>> ——————▶	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsi atau sebagai syarat dijalankan *use case* ini.

2.3.1.2. *Activity Diagram*

Menurut (A.S & Shalahuddin, 2011, p. 134) Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diketahui bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.5 *Activity* diagram

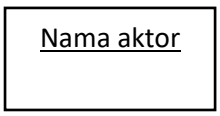
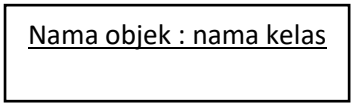
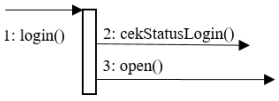
Simbol	**Deskripsi**
Status awal ●	Status awal aktivitas sistem, sebuah diagram aktifitas memiliki sebuah status awal
Aktifitas aktifitas	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
Percabangan/*decision*	Asosiasi percabangan dimana jika

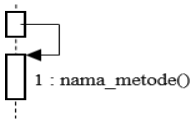
	ada pilihan aktifitas lebih dari satu
Penggabungan/*join* 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi

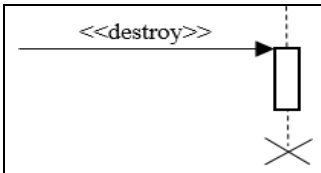
2.3.1.3. *Sequence Diagram*

Menurut (A.S&Shalahuddin, 2011, p.137) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterim antar objek.

Tabel 2.6 *Sequence* Diagram

Simbol	Deskripsi
Aktor Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, walaupun simbol aktor itu berbentuk gambar orang, tetapi belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Garis hidup/ *lifeline* 	Menyatakan kehidupan suatu objek
Objek  	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan  aktif dan berinteraksi pesan.
Pesan tipe *create* <<*create*>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Pesan tipe *call* 1: nama_metode()	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau diri sendiri.  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe *send* 1 : masukkan	Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 1 : keluaran ---------->	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe *destroy*	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah

	mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*.

Sumber: (A.S. & Shalahuddin, 2011)

2.3.1.4. *Class Diagram*

Menurut (A.S.&Shalahuddin, 2011, p. 122) Diagram kelas atau *class diagram* menggambarkan struktur dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dari metode atau operasi. Atribut merupakan variabel- vareiabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

   Kelas yang memiliki fungsi awal dieksekusiketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

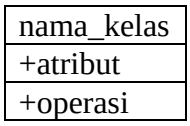
3. Kelas yang diambil dari pendefinisian *use case*

   Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case.*

4. Kelas yang diambil dari pendefinisian data

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan kebasis data.

Tabel 2.7 *Class* Diagram

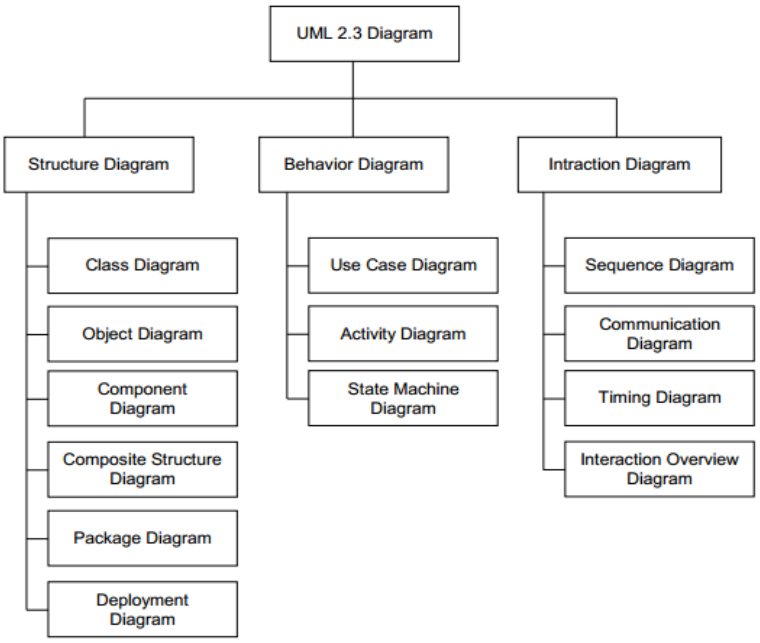
Simbol	Deskripsi
Kelas  	Kelas pada struktur sistem
Antarmuka / *interface* nama_interface	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi / *association* 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah / *directed association* 	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / *defedency* ..... >	Relasi anata kelas dengan makna kebergantungan antar kelas
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua-bagian (*whole-part*)

Sumber:(A.S. & Shalahuddin, 2011)

2.3.1.4. Diagram UML

Menurut (A.S. & Shalahuddin, 2011, p. 120) UML 2.3 memiliki 13 macam diagram yang dikelompokkan dalam 3 kategori. Dibawah ini adalah pembagian dari kategori diagram UML:

1. *Strukture diagram*: kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagram*: kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram*: kumpilan diagram – diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



Gambar 2.9 Diagram UML
Sumber : Data Penelitian (2018)

2.3.2. *star*UML

starUML adalah *platform* pemodelan yang mendukung UML (*Unified Modeling Language*). *Star*UML mempunyai sebelas macam diagram yang berbeda yang juga mendukung pendekatan MDA (*Model Driven Architecture*) dengan mendukung konsep UML *profile. Star*UML dapat memaksimalkan produktifitas dan kualitas dari suatu *software project*.

Fitur – fitur yang ada didalam *star*UML adalah :

1. dukungan terhadap UML:
   a. *usecase diagram.*
   b. *class diagram.*
   c. *activity diagram.*
   d. *sequence diagram.*
2. dukungan terhadap beberapa bahasa pemrograman:
   a. *Java profile, code generator, dan reverse engineering.*
   b. *C++ profile, code generator, dan reserve engineering.*
   c. *C# profile, code generator, dan reserve engineering.*
   d. *Microsoft office documen generator.*
   e. *Microsoft word document template and generation.*
   f. *Microsoft excel document template and generation.*
   g. *Microsoft powerpoint document template and generation.*
3. *Customizable code generation.*
4. Mendukung teknologi MDA.
5. Diagram yang dapat diperluas (menentukan sendiri jenis diagram).

6. *Extensibility*.
7. Kompatible yang tinggi.
8. *Editing*.
9. *User interface*

2.2.4. PHP

PHP merupakan singkatan dari "*Hypertext Preprocessor"*. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (situs personal) dan pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995 dan pada saat itu PHP masih bernama FI (*Form Interpreter*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengelola data form dari *Web*. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum. PHP adalah sebuah bahasa scripting yang terpasang pada HTML. (Saputra, 2012)

PHP digunakan untuk membuat tampilan *Web* menjadi lebih dinamis, dengan PHP Anda bisa menampilkan atau menjalankan beberapa file dalam satu file dengan cara di-*include* atau require. PHP itu sendiri sudah dapat berinteraksi dengan database walaupun dengan kelengkapan yang berbeda. (Saputra, 2012)

Gambar 2.10 Logo
Sumber : http://php.net

2.2.5. *HTML* (*Hyper Text Markup Language*)

Gambar 2.11 Logo HTML
Sumber: Data Penelitian (2018)

HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML biasa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *Website*.

HTML versi 5 yang paling marak dibincangkan didunia maya, HTML 5 layaknya sebuah html biasa yang sering kita guanakan dalam membangun aplikasi *Web*, hanya saja html 5 ini memiliki keunggulan disbanding versi terdahulunya. HTML 5 juga mampu menyederhanakan kode-kode html terdahulu menjadi lebih ringkas. Hal yang paling mencolok adalah tersedianya fitur baru seperti elemen multimedia, misalnya <audio> dan <video>, yang tak lain adalah fungsi untuk memutar audio dan juga video. Pada versi HTML sebelumnya, jika anda ingin memutar perangkat multimedia, haruslah menggunakan perintah <*embed*>. Dengan adanya penambahan fitur dan tag khusus yang ada di HTML 5, akan membuat semuanya menjadi sangat mudah. Dokumen html memiliki sebuah struktur yang harus kita ikuti aturan pembuatanya. Kita akan mengenal beberapa elemen-elemen wajib yang ada pada file html apabila kita ingin membangun sesuatu pondasi kerangka wenside, elemen tersebut diantaranya:

1. Elemen HTML <html> merupakan tag dasar apabila kita ingin melalui suatu dokumen html. Tag ini merupakan perintah wajib bagi pemogram *Web* untuk menuliskan tag pertama dalam dokumen html.
2. Elemen *Head* </*head*> merupakan tag berikutnya setelah html (<html>), yang berfungsi untuk menuliskan keterangan tentang dokumen *Web* yang akan ditampilkanm, elemen ini nantinya akan diakhiri dengan tanda penutup </*head*>.
3. Elemen Title <*title*> merupakan suatu elemen yang harus dituliskan didala elemen *head* yang digunakan untuk memberikan judul/ informasi pada caption browser *Web* tentang topic/ tema atau judul dari suatu dokumen *Web* yang ditampilkan pada browser
4. Elemen Body <*body*> merupakan bagian utama dalam dokumen *Web*. Jika kita ingin menampilkan suatu teks atau informasi atau yang dikenal dengan sebutan konten, maka kita harus meletakkan teks tersebut pada elemen *body*.

2.2.6. XAMPP

XAMPP adalah sebagai server yang berdiri sendiri (*localhost*) yang terdiri beberapa program seperti *Apache* HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang dituliskan dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, MySQL, PHP dan *Perl*. Program ini tersedia dalam GNU *General Publik license* dan bebas, merupakan *Web* server yang mudah untuk digunakan yang dapat menampilkan halaman *Web* yang dinamis secara gratis dan mendukung instalasi di *Linux* dan *Windows*. (Saputra, 2012)

Gambar 2.12 Logo XAMPP
Sumber : Aplikasi

2.2.7. Notepad++

Adalah program aplikasi yang berguna untuk mengedit teks dan script kode pemrograman. Software Notepad++ dibuat dan dikembangkan oleh tim Notepad++. Perangkat lunak komputer ini memiliki kelebihan pada peningkatan kemampuan sebuah program teks editor, lebih dari sekedar program Notepad bawaan windows. Notepad++ bisa mengenal tag dan kode dalam berbagai bahasa pemrograman dan bersifat *opensource* (gratis). (Saputra, 2012)

Gambar 2.13 Logo Notepad++
Sumber : Aplikasi

2.2.8. MySQL

Sejarah diciptakan pada tahun 1979, oleh Michael ”Monty” widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem *database* sederhana dinamakan UNIREG yang menggunakan koneksi Low-level ISAM database engine dengan *indexing.* MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management*

system) atau DBMS yang Multithread, multi-user, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License*(GPL). (Saputra, 2012)

Gambar 2.14 *Logo MySQL*
Sumber : Data Penelitian (2018)

1. Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain:

a. Probabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, Freedsb, Mac Os X Server, Solaris, Amiga.
b. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
c. Multi-user. MySQL dapat diguanakan beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
d. *Performance tuning*. MySQL memiliki kecepatan yang menakjubkan menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

e. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsigned integer, float, double, chart, text, date, timestamp*, dan lain-lain.
f. Perintah dan fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
g. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan perizinan yang mendetail secara sandi terenskripsi.
h. Skalabilitas dan pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indek dapat ditampung mencapai 32 indeks pada tiap tabelnya.
i. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix soket* (UNIX), atau named pipes (NT).
j. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa indonesia belum termasuk didalamnya.
k. Antar muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
l. Klien dan peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

m. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menagani *ALTER TABLE*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.

2. Kelebihan MySQL

a. Berlisensi GPL dan multi platform.
b. dapat diintegrasikan dengan bebrapa bahasa pemrograman seperti Net, java, python, perl yang merupakan bahasa pemrograman yang paling domain dikalangan programmer.
c. Mendukung ODBC untuk sistem operasi windows sehingga bisa digunakan aplikasi yang berjalan di windows.
d. Bisa dijalankan pada spesifikasi hardware yang rendah karena lebih hemat *resource memory* (dibandingkan database lain) sehingga mudah digunakan untuk bahan pembelajaran.
e. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa meskipun bahasa indonesia belum termasuk di dalamnya.

3. Kekurangan MySQL

1. Banyak mengklaim kurang support terhadap pemrograman Visual/Desktop, sehingga sedikit yang menggunkan untuk aplikasi visual.
2. Karena berlisensi GPL sehingga sulit mendapatkan *update* untuk problem yang urgent, sehingga perusahaan skala menengah keatas lebih memilih RDBMS berlisensi dan support seperti *Oracle* dan MS SQL server.

3. Sangat diragukan dalam menagani data skala besar, karena ada beberapa opini yang pro dan kontrak terhadap kemampuan MySQL terhadap pengolahan data yang besar.

2.4 Penelitian Terdahulu

Penelitian yang dilakukan oleh peneliti merupakan penelitian terapan yang dilakukan dengan mengambil beberapa penelitian terdahulu yang relevan dengan penelitian ini, yang dijadikan peneliti sebagai referensi. Dari penelitian terdahulu dapat disimpulkan.

Restia Dwi Oktavianing Tyas, Arief Andy Soebroto, M. Tamzil Furqon (2015). **Pengembangan sistem pakar diagnosa penyakit sapi potong dengan metode fuzzy k-nearest neighbour**. Hasil dan pembahasan merupakan tahap hasil perancangan dan implementasi sistem pakar. . Pengujian sistem pakar diagnosa penyakit sapi potong terdiri dari tiga bagian yaitup pengujian black box, pengujian variasi nilai variabel k dan variasi data latih dan pengujian nilai variabel m.(Technology, Dwi, Tyas, Soebroto, & Furqon, 2015).

Reski Mai Candra, Weni Rahim (2014). **Sistem pakar diagnosa bibit unggul sapi dan kambing dengan metode certainty factor.** Berdasarkan dari hasil analisa, perancangan dan implementasi pada sistem ini, dapat diambil beberapa kesimpulan diantaranya adalah. Metode Certainty Factor dapat di terapkan untuk menentukan jenis induk yang dapat menghasilkan bibit unggul. Berdasarkan hasil pengujian user acceptance test menyatakan bahwa sistem dapat memudahkan para peternak menentukan calon induk sapi dan kambing yang dapat

menghasilkan bibit unggul. Berdasarkan pengujian oleh pakar, menyatakan bahwa sistem Dianosa Bibit Unggul Sapi Dan Kambing Dengan Metode Certainty Factor sesuai dengan yang diterapkan oleh pakar. (Ilmiah et al., 2014).

I Kadek Dwi Gandika Supartha, Ida Nirmala Sari(2014). **Sistem pakar diagnosa awal penyakit kulit pada sapi bali dengan metode forward chaining dan certainty factor. Setelah dilakukan analisis dan pengujian terhadap Sistem.** Pakar Diagnosa Awal Penyakit Kulit pada Sapi Bali dengan Menggunakan Metode forward chaining dan Certainty Factor, dapat disimpulkan bahwa. Sistem ini dapat digunakan untuk mengetahui jenis penyakit yang di dialami oleh sapi Bali di BPTU Sapi Bali berdasarkan inputan gejala-gejala yang dimasukan oleh user. Sistem ini mampu menyimpan representasi pengetahuan pakar dengan metode Foward Chaining dan Certainty Factor berdasarkan nilai Measure of Believe (MB) dan Measure of Disbelieve (MD). Selain menghasilkan penyakit sistem pakar ini dapat menjelaskan definisi dan cara pengobatan penyakit kulit pada sapi Bali. (Dwi, Supartha, Sari, & Informatika, 2014)

Indra Fauz i Rohman, Prihastuti Harsani, Arie Qur'ania. (2016) **Aplikasi Diagnosis Penyakit Sapi Menggunakan Metode Certainty Factors Berbasis Android.** Berdasarkan penelitian yang dilakukan pada aplikasi diagnosispenyakit sapi, maka dapat disimpulkan bahwa metode *Certainty Factors* dapat diimplementasikan dalam aplikasi Diagnosis Penyakit sapi dan penerapan metode *Certainty Factors* sangat membantu dalam pembuatan aplikasi ini, dengan mengandalkan bobot dari masing- masing gejala. Setelah melalui pengujian didapat bahwa Implementasi aplikasi dengan perhitungan manual hasilnya sama

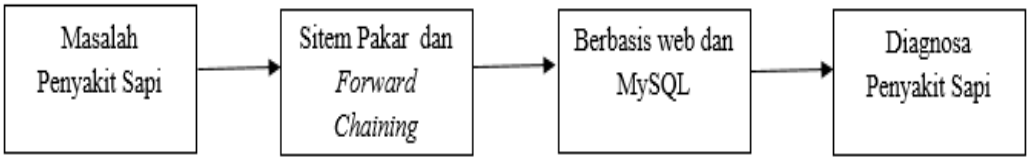
atau valid. Dengan tingkat Akurasi dari aplikasi dengan metode certainty factors ini mencapai 86%. Berdasarkan hasil pengujian yang telah dilakukan pada perangkat android versi 4.0 atau Ice Cream Sandwich sampai android versi 5.0 atau Lollipop, semua fitur yang ada pada aplikasi dapat berjalan dengan sangat optimal. Aplikasi ini akan sangat membantu kedua belah pihak, antara peternak dan dokter hewan. Peternak sangat terbantu karena aplikasi ini sudah dapat menentukan penyakit yang diderita sapi sehingga lebih cepat dalam penanganannya dan dokter hewan pun terbantu dengan aplikasi ini karena dokter dapat dengan cepat memfokuskan penanganan sesuai saran aplikasi. (Harsani, Komputer, & Bogor, 2016)

Alexius Endy Budianto.(2015) **Aplikasi Sistem Pakar Menggunakan Metode Backward Chaining Untuk Analisis Penyakit Hewan Ternak.** Berdasarkan permasalahan yang telah dibahas dan diselesaikan melalui laporan ini maka terdapat beberapa kesimpulan yaitu untuk membangun sistem pakar ini dibutuhkan inputan gejala-gejala yang menggunakan metode backward chaining atau runut balik. Berdasarkan hasil pengujian aplikasi sistem pakar ini berguna untuk membantu para peternak sapi untuk mengenali gejala-gejala yang ditimbulkan dan memberikan tindakan yang tepat untuk mengatasi serangan penyakit pada hewan ternak sapi. (Budianto & Kom, 2015)

2.5 Kerangka Berpikir

Sistem pakar untuk mendiagnosa penyakit sapi merupakan suatu sistem pakar yang dirancang sebagai alternatif untuk membantu masyarakat khusus nya

peternak sapi dalam memberikan kemudahan dan solusi terhadap penyakit sapi. Dengan menggunakan metode Forward Chaining berbasis *Web* sangat membantu para peternak sapi dimanapun dan kapanpun dalam mengetahui dan menangani gejala apa yang ada pada sapi tersebut sehingga bisa meminimalisirkan ternak sapi yang terjangkit penyakit sampai berakibat kematian. Dengan berbasiskan sistem *Web* diharapkan tingkat kemudahan pengaksesan sistem pakar ini lebih mudah dan efisien.



Gambar 2.15 Kerangka Berfikir
Sumber : Data Penelitian (2018)

Uraian Kerangka Berfikir:

1. Masalah Penyakit sapi: penelitian ini mengangkat tiga variabel penyakit metabolik pada ternak sapi yang pertama penyakit *Milk Fever (Parturient-Hypocalcemia,* Kedua penyakit *Grass Tetany (Hypomagnesemia),* ketiga Penyakit *Asetonemia (Ketosis).*
2. Sistem pakar dan metode *Forward Chaining*: penelitian ini membangun sistem pakar untuk mendiagnosa penyakit pada hewan ternak dengan menggunakan metode *Forward Chaining.*
3. Berbasis *Web* dan MySQL: penelitian ini menggunakan tools berbasis *Web* dan MySQL
4. Diagnosa Penyakit Sapi: berdasarkan hasil analisis penelitian maka didapatkan diagnosa jenis penyakit sapi serta cara penanganannya.