

BAB II TINJAUAN PUSTAKA

2.1 Teori Dasar

Menurut (Danny, 2014) (*expert system*) adalah sistem/program yang bertindak laku seperti ahlinya atau pakarnya. Sistem pakar merupakan sistem yang berbasis pengetahuan digunakan untuk membantu dalam menyelesaikan masalah-masalah yang ada dalam dunia nyata.

2.1.1 Kecerdasan Buatan atau *Artificial Intelligence (AI)*

Menurut (Budiharto & Suhartono, 2014, p. 2) kecerdasan buatan atau *Artificial intelligence (AI)* merupakan bidang ilmu komputer yang mempunyai peran penting di era kini dan masa akan datang, di dalam bidang ini telah berkembang sangat pesat di 200 tahun sering adanya pertumbuhan kebutuhan akan perangkat cerdas dalam rumah tangga maupun industri.

2.1.1.1 Sejarah Kecerdasan Buatan (*artificial intelligence*)

Menurut (Budiharto & Suhartono, 2014, p. 3) McMulloh dan Pitts pada tahun 1943 mengajukan model matematis bernama *perceptron* dari neuron di dalam otak. Mereka juga menunjukkan bagaimana neuron menjadi aktif seperti sakelar *on-off*. Neuron tersebut mampu belajar dan memberikan aksi berbeda terhadap waktu dari

input yang diberikan. “Dapatkah komputer berfikir?” dengan menciptakan *Turing Machine*.

Paper Alan Turing pada tahun 1950 berjudul *Computing Machinery and Intelligence* mendiskusikan syarat sebuah mesin dianggap cerdas. Turing beranggapan bahwa jika mesin dapat dengan sukses berperilaku seperti manusia, maka mesin itu dapat dianggap cerdas. Pada akhir 1955, Newell dan Simon mengembangkan *The Logic Theorist*, program AI pertama. Program ini mempresentasikan masalah sebagai model pohon, lalu penyelesaiannya dengan memilih cabang yang akan menghasilkan kesimpulan terbenar. Program ini berdampak besar dan menjadi batu loncatan penting dalam mengembangkan bidang AI.

Pada tahun 1956, John McCarthy dari *Massachusetts Institute of Technology* yang dianggap sebagai bapak AI, menyelenggarakan konferensi bertajuk *The Dartmouth Summer Research Project on Artificial Intelligence*. Konferensi ini bertujuan menarik bakat, keahlian orang banyak untuk masuk dalam dunia kecerdasan buatan.

McCarthy mengartikan sebagai, “ AI cabang dari ilmu komputer yang hanya fokus pada perkembangan komputer untuk dapat memiliki kemampuan dan berperilaku seperti manusia”. Masa-masa dikategorikan sebagai *Classical AI*. Pada tahun 1980, komputer semakin mudah diperoleh dengan harga yang lebih murah. Akibatnya, berbagai riset di bidang kecerdasan buatan berkembang sangat tinggi (Budiharto & Suhartono, 2014, p. 5)

2.1.1.2 Logika Fuzzy (*Fuzzy Logic*)

Menurut (Budiharto & Suhartono, 2014, p. 150) *fuzzy logic* mempunyai derajat keanggotaan dalam rentang 0 (nol) hingga 1 (satu), hal ini berbeda dengan logika digital atau diskrit yang hanya memiliki dua nilai yaitu 1 (satu) atau 0 (nol). Logika fuzzy digunakan untuk menerjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*). Komputer tidak mengerti nilai asli dari kata “sebentar”. Komputer dapat mengelola ketidak pastian tersebut sehingga digunakan untuk memutuskan sesuatu yang membutuhkan kepintaran manusia dalam penalaran.

2.1.1.3 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

Jaringan syaraf tiruan salah satu upaya manusia untuk merangkap cara kerja atau fungsi sistem syaraf manusia dalam melakukan tugas. Hal ini didasari oleh kemampuan berfikir manusia dalam mengorganisasikan sel penyusunan yang disebut *neuron*, sehingga mampu melaksanakan tugas tertentu, khususnya pengenalan pola dengan efektivitas yang sangat tinggi. Jaringan syaraf tiruan prosesor tersebar paralel yang sangat besar mempunyai kecenderungan untuk menyimpan pengetahuan yang bersifat pengalaman dan membuatnya siap digunakan. Jaringan syaraf tiruan menyerupai otak manusia dalam dua hal, yaitu: Pengetahuan diperoleh jaringan melalui proses belajar; Keuatan hubungan antar sel

syaraf (*neuron*) yang dikenal sebagai bobot sinaptik digunakan untuk menyimpan pengetahuan (Suyanto, 2014).

Jaringan Saraf Tiruan paradigma dalam pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dan paradigma ini struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu. Cara kerja JST seperti cara kerja manusia, yaitu belajar melalui contoh. Sebuah JST dikonfigurasi untuk aplikasi tertentu. Belajar dalam sistem biologis melibatkan penyesuaian terhadap koneksi *synaptic* yang ada antara *neuron*. Hal ini berlaku juga untuk JSTI (Sutojo et al., 2011)

Sel syaraf (*neuron*) unit memproseskan suatu informasi yang merupakan dasar dari operasi jaringan syaraf tiruan. Terdapat tiga elemen dasar dari model *neuron*, yaitu (Suyanto, 2014):

1. Sekumpulan sinapsis atau jalur hubungan, dimana sinapsis memiliki bobot atau kekuatan hubungan.
2. Suatu *adder* untuk menjumlahkan sinyal *input* yang diberi bobot oleh sinapsis *neuron* yang sesuai. Operasi yang digambarkan di sini mengikuti aturan *linear combiner*.
3. Suatu fungsi aktivasi untuk membatasi amplitudo *output* dari setiap *neuron*.

Kelebihan-kelebihan yang diberikan jaringan saraf tiruan antara lain (Sutojo et al., 2011)

1. *Belajar Adaptive*: Kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengamatan awal.
2. *Self-Organisation*: Sebuah JST dapat membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real Time Operation*: Perhitungan JST dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Selain mempunyai kelebihan tersebut, JST mempunyai kelemahannya tersendiri (Sutojo et al., 2011):

1. Tidak efektif jika digunakan untuk melakukan operasi numerik dengan presisi tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatik, operasi logika, dan simbolis.
3. Untuk beroperasi butuh pelatihan sehingga bila jumlah datanya besar, waktu yang dipergunakan membutuhkan waktu yang sangat lama.

Beberapa arsitektur jaringan yang sering digunakan dalam jaringan saraf tiruan antara lain (Sutojo et al., 2011):

- a. Jaringan Lapisan Tunggal.

Jaringan dengan lapisan tunggal terdiri dari 1 lapisan *input* dan 1 lapisan *output*. Setiap unit dalam lapisan input selalu terhubung dengan setiap unit yang terdapat pada lapisan *output*.

- b. Jaringan Lapisan Banyak

Jaringan lapisan banyak mempunyai 3 jenis lapisan, yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*.

Hebb Rule. Model *neuron* McCulloch-Pitts akan mengalami kesulitan bila berhadapan dengan fungsi yang sama kompleks. Hal ini karena dalam menentukan bobot w dan nilai ambang Θ harus diuji secara analitik atau menggunakan cara coba-coba. Pada 1949, D.O. Hebb memperkenalkan cara menghitung bobot w dan bias secara iteratif dengan memanfaatkan model pembelajaran dengan supervisi sehingga bobot w dan bias dapat dihitung secara otomatis tanpa harus melakukan cara coba-coba. Model *Hebb* merupakan model jaringan tertua yang menggunakan pembelajaran dengan supervisi. Arsitektur jaringan Hebb sama seperti arsitektur jaringan McCulloch-Pitts, yaitu beberapa input terhubung langsung dengan sebuah unit output, ditambahkan dengan sebuah bias (Sutojo et al., 2011).

1. *Perceptron*. Model jaringan perceptron merupakan model yang paling baik pada saat itu. Model ditemukan oleh Rosenblatt (1962) dan Minsky – Papert.
2. *Delta Rule*. Selama pelatihan pola, *delta rule* akan mengubah bobot dengan cara meminimal error antara jaringan y dengan target t .
3. *Backpropagation*. *Backpropagation* metode gradien untuk meminimalkan kuadrat *error* keluaran. Ada tiga tahap yang harus dilakukan pelatihan jaringan, yaitu tahap perambatan maju (*forward propagation*), tahap perambatan balik, dan tahap perubahan bobot dan bias. Arsitektur jaringan ini terdiri dari input layer, hidden layer dan output layer.
4. *Heteroassociative Memory*. Jaringan saraf *heteroassociative memory* jaringan yang dapat menyimpan kumpulan pengelompokan pola, menentukan bobot-

bobotnya sedemikian rupa. Setiap kelompok pasangan vektor $(s(n), t(n))$ dengan $n=1, 2, \dots, N$. Algoritma pelatihan yang biasa digunakan adalah *Hebb Rule* (Sutojo et al., 2011).

5. *Learning Vector Quantization (LVQ)*. *Learning Vector Quantization (LVQ)* metode pelatihan pada lapisan yang kompetitif terawasi yang akan belajar secara otomatis untuk mengklasifikasikan vektor-vektor input ke dalam kelas-kelas tertentu. Kelas-kelas yang dihasilkan tergantung pada jarak antara vektor input. Jika ada 2 vektor input yang hampir sama maka lapisan kompetitif akan mengklasifikasikan kedua vektor input tersebut dalam kelas yang sama (Sutojo et al., 2011).

2.1.1.4 Sistem Pakar (*Expert System*)

Sistem pakar cabang dari *Artificial Intelligence (AI)* yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose problem solver (GPS)* yang dikembangkan oleh Newel dan Simon. Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti MYCIN untuk diagnosis penyakit, DENDRAL untuk mengidentifikasi struktur campuran yang tak dikenal, XCON & XSEL untuk membantu konfigurasi sistem komputer besar, SOPHIE untuk analisis sirkuit elektronik, *Prospector* digunakan di bidang geologi untuk membantu mencari dan menemukan deposit, FOLIO digunakan untuk membantu memberi keputusan bagi

seorang manager dalam stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik disel, dan sebagainya (Sutojo et al., 2011).

Sistem pakar merupakan program yang berfungsi menirukan pakar manusia harus bisa melakukan hal-hal yang dapat dikerjakan oleh seorang pakar. Untuk membangun sistem yang seperti itu maka komponen yang harus dimiliki sebagai berikut (Giarratano dan Riley, 2005 *dalam* (Hartati & Iswanti, 2008):

a. Antar Muka Pengguna (*User Interface*)

Sistem pakar menggantikan seorang pakar dalam suatu situasi tertentu, maka sistem harus menyediakan pendukung yang diperlukan oleh pemakai yang tidak memahami masalah teknis. Sistem pakar juga menyediakan komunikasi antara sistem dan pemakainya, yang disebut antar muka. Antar muka yang efektif dan ramah pengguna (*user friendly*) penting sekali terutama bagi pemakai yang tidak ahli dalam bidang yang diterapkan pada sistem pakar.

b. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan merupakan kumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu. Pengetahuan ini diperoleh dari akumulasi pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti yang telah disebutkan sebelumnya. Basis pengetahuan bersifat dinamis, bisa berkembang dari waktu ke waktu. Perkembangan ini disebabkan karena pengetahuan selalu bertambah, *terupdate*. Pada sistem pakar basis pengetahuan terpisah dari mesin inferensi, pemisahan ini bermanfaat untuk pengembangan sistem pakar secara leluasa disesuaikan dengan perkembangan pengetahuan pada suatu domain.

Penambah dan pengurangan dapat dilakukan pada basis pengetahuan ini tanpa mengganggu mesin inferensi.

c. Mesin Inferensi (*Inference Machine*)

Mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, biasa dikatakan sebagai mesin pemikir (*thinking machine*). Pada prinsipnya mesin inferensi inilah yang akan mencari solusi dari suatu permasalahan.

Mesin inferensi sesungguhnya adalah program komputer yang menyediakan metodologi untuk melakukan penalaran tentang informasi pada basis pengetahuan dan pada memori kerja, serta untuk merumuskan kesimpulan-kesimpulan.

d. Fasilitas Penjelasan (*Explanation Facility*)

Proses menentukan keputusan yang dilakukan oleh mesin inferensi selama sesi konsultasi mencerminkan proses penalaran seorang pakar. Karena pemakai kadangkala bukanlah ahli dalam bidang tersebut, maka dibuatlah fasilitas penjelasan. Fasilitas penjelasan inilah yang dapat memberikan informasi kepada pemakai mengenai jalannya penalaran sehingga dihasilkan suatu keputusan. Bentuk penjelasannya dapat berupa keterangan yang diberikan setelah suatu pertanyaan diajukan, yaitu penjelasan atas pertanyaan mengapa, atau penjelasan atas pertanyaan bagaimana sistem mencapai konklusi.

Tujuannya adanya fasilitas penjelasan dalam sistem pakar antara lain membuat sistem menjadi lebih cerdas, menunjukkan adanya proses analisa dan yang tidak kalah pentingnya memuaskan psikologis pemakai.

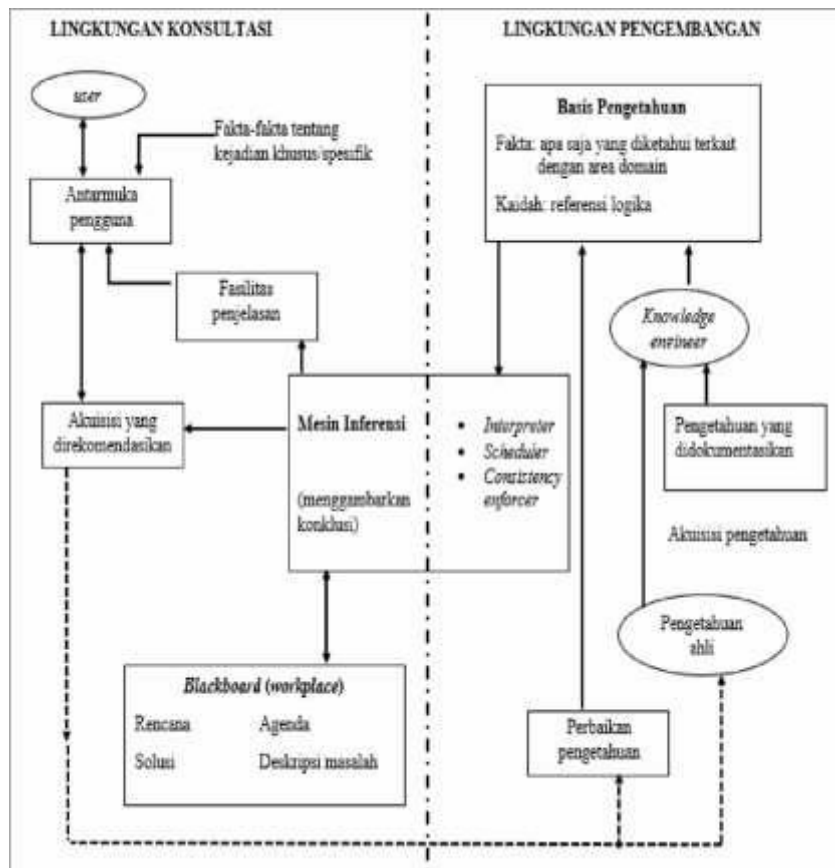
e. Fasilitas Akuisisi Pengetahuan (*Knowledge Acquisition Facility*)

Dengan fasilitas pada sistem, seorang pakar akan dengan mudah menambahkan pengetahuan ataupun kaidah baru pada sistem pakar. Untuk menjamin bahwa hanya bisa diakses oleh pakar. Pengguna awam tidak berhak memakai fasilitas pengetahuan.

f. Memori Kerja (*Working Memory*)

Bagian dari sistem pakar yang menyimpan fakta yang diperoleh saat dilakukan proses konsultasi. Fakta inilah yang nantinya akan diolah oleh mesin inferensi berdasarkan pengetahuan yang disimpan dalam basis pengetahuan untuk menentukan suatu keputusan pemecahan masalah. Konklusinya bisa berupa hasil diagnosa, tindakan, akibat.

Sistem pakar juga dapat dilihat dari sudut pandang lingkungan (*environment*) dalam sistem. Terdapat dua lingkungan konsultasi pengembangan. Lingkungan konsultasi bagi pengguna non pakar untuk melakukan konsultasi dengan sistem yang tujuannya adalah mendapatkan nasehat pakar. Sedangkan, lingkungan membangun komponen dan memasukkan pengetahuan hasil akuisisi pengetahuan ke dalam basis pengetahuan. Secara lengkap struktur sistem pakar yang menekankan pada lingkungan yang ada dalam sistem terlihat pada gambar di bawah ini (Hartati & Iswanti, 2008).



Gambar 2.1 Struktur sistem pakar
(Sumber: Hartati dan Iswanti, 2008:9)

Menurut (Hartati & Iswanti, 2008) pengetahuan dimaksudkan untuk mengorganisasikan pengetahuan dalam bentuk dan format tertentu agar dapat dipahami oleh komputer. Dalam pemilihan representasi pengetahuan yang tepat akan menghasilkan sebuah sistem pakar yang efektif. Salah satu model representasi pengetahuan yang penting yaitu kaidah produksi (*production rule*).

Pada sistem pakar berbasis *rule*, domain pengetahuan direpresentasikan dalam sebuah kumpulan *rule* berbentuk *IF-THEN*, sedangkan data dalam sebuah kumpulan adalah fakta-fakta tentang kejadian saat ini. Mesin inferensi membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan

fakta-fakta yang terdapat dalam database. Jika bagian IF (kondosi) dari *rule* cocok dengan fakta maka *rule* dieksekusi dan bagian *THEN* (aksi) diletakkan dalam database sebagai fakta baru yang ditambahkan. (Sutojo et al., 2011)

Teknik inferensi dalam sistem pakar tersebut ada dua, (Sutojo et al., 2011):

1. *Forward chaining.*

Forward chaining teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta tersebut dengan bagian IF dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian IF, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam database. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja.

2. *Backward chaining.*

Backward chaining metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada dibagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis dibagian *THEN* ditempatkan di basis data sebagai fakta baru. Jika tidak cocok, simpan premis dibagian *IF* ke dalam *stack* sebagai *subGoal*. Proses berakhir jika *Goal* ditemukan atau tidak ada *rule* yang bisa membuktikan kebenaran dari *subGoal* atau *Goal*. (Sutojo et al., 2011)

Secara umum, konsep ini diaplikasikan ketika ditentukan sebagai kondisi atau keadaan awal. Konsep ini disebut juga *goal-driven search*. Arah penalaran atau peruntutan dalam konsep ini berlawanan dengan *forward chaining*. Konsep ini dapat dimodelkan sebagai berikut:

Proses penalaran pada *backward chaining* dimulai dari tujuan kemudian merunut balik ke jalur yang mengarah ke tujuan tersebut, untuk membuktikan bahwa bagian kondisi pada kaidah atau aturan benar-benar terpenuhi. Proses *internal* selalu memeriksa konklusi terlebih dahulu sebagai praduga awal, kemudian memeriksa dan memastikan gejala-gejala (kondisi) telah terpenuhi dan selanjutnya mengeluarkan konklusi sebagai *output*. Jika sistem ditemukan ada bagian kondisi yang terpenuhi maka sistem akan memeriksa konklusi (tujuan) pada aturan atau kaidah berikutnya. (Hartati & Iswanti, 2008)

Kaidah menyediakan cara formal yang dituliskan dalam bentuk jika-maka (*IF-THEN*) untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah *IF-THEN* menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Berikut ini adalah contoh struktur kaidah *IF-THEN* yang menghubungkan obyek: (Hartati & Iswanti, 2008)

1. *IF* premis *THEN* konklus
2. *IF* masukan *THEN* keluaran
3. *IF* kondisi *THEN* tindakan
4. *IF* antesende *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 konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi ini mengacu kepada keadaan yang harus berlaku sebelum tindakan dapat diambil. Antesende mengacu situasi yang terjadi sebelum konse

kuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Aksi mengacu pada kegiatan yang menyebabkan bermunculnya efek dari tindakan tersebut. Gejala mengacu pada keadaan yang ditimbulkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan (diagnosa). (Hartati & Iswanti, 2008)

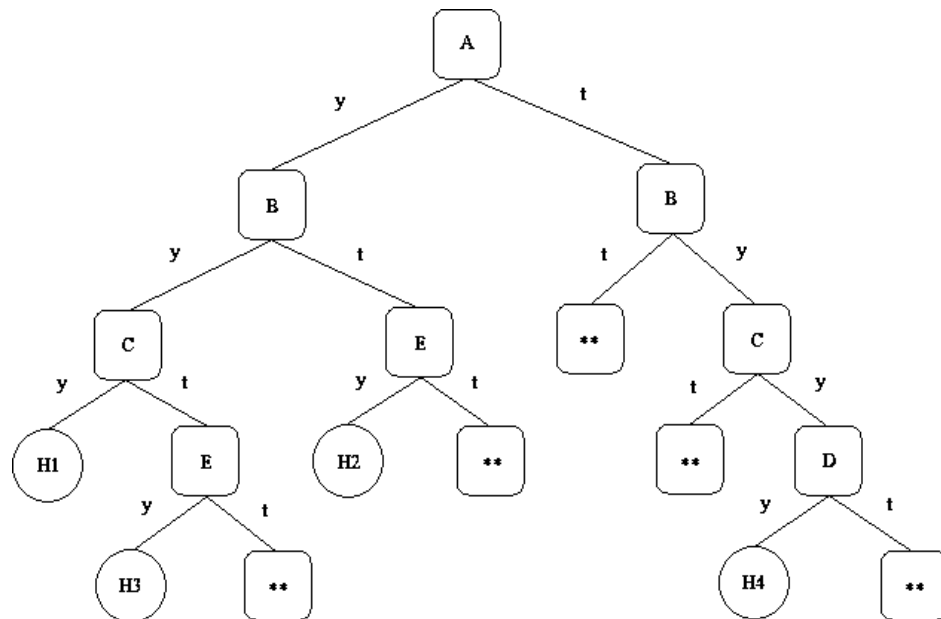
Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk tabel keputusan kemudian dibuat pohon keputusannya. Contoh tabel keputusan dan pohon keputusan (Hartati & Iswanti, 2008)

Tabel 2. 1 Tabel Keputusan

Hipotesas	Hipotesa	Hipotesa	Hipotesa	Hipotesa
<i>Evidence A</i>	Ya	ya	ya	tidak
<i>Evidence B</i>	Ya	tidak	ya	ya
<i>Evidence C</i>	Ya	Tidak	tidak	ya
<i>Evidence D</i>	tidak	Tidak	tidak	ya
<i>Evidence E</i>	tidak	Ya	ya	tidak

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 a4

Gambar 2. 1 Pohon Keputusan

Sumber: (Hartati & Iswanti, 2008)

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 dapat dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi *node (evidence)* di atasnya, notasi “t” artinya tidak terpenuhi.

Dalam sesi konsultasi pada sistem pakar, *node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat

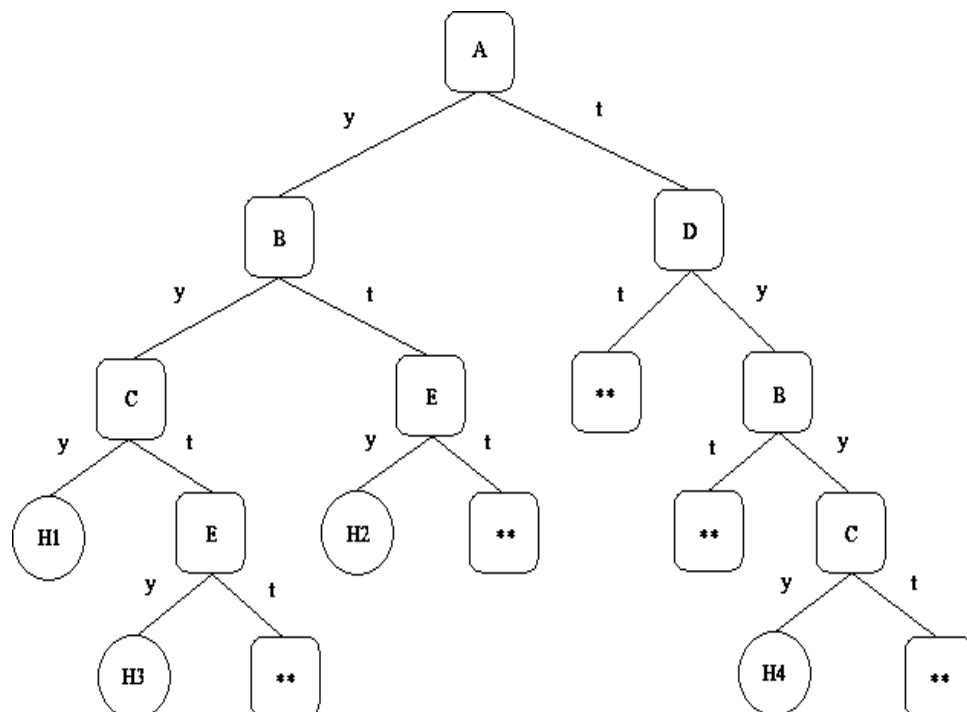
pohon keputusan pada gambar 2.2 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* A?”. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.2.

Tabel 2. 2 Alternatif Tabel Keputusan

Hipotesa	Hipotesa	Hipotesa	Hipotesa	Hipotesa
<i>Evidence A</i>	Ya	Ya	Ya	Tidak
<i>Evidence D</i>	Tidak	Tidak	Tidak	Ya
<i>Evidence B</i>	Ya	Tidak	Ya	Ya
<i>Evidence C</i>	Ya	Tidak	Tidak	Ya
<i>Evidence E</i>	Tidak	Ya	Ya	Tidak

Sumberd: (Hartati & Iswanti, 2008)

Berdasarkan tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



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

Gambar 2. 2 Alternatif Pohon Keputusan

Sumber: (Hartati & Iswanti, 2008)

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.

Kaidah yang dapat dihasilkan berdasarkan pohon keputusan pada gambar 2.3 adalah sebagai berikut:

1. Kaidah 1: IF A AND B AND C THEN H1
2. Kaidah 2: IF A AND B AND E THEN H3
3. Kaidah 3: IF A AND E THEN H2
4. Kaidah 4: IF D AND B AND C THEN H4

Model representasi pengetahuan kaidah banyak pada aplikasi sistem pakar karena model representasi ini sangat mudah dipahami.

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikan, diantaranya (Sutojo et al., 2011):

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat.
2. Membuat orang awam bekerja seperti layaknya seorang pakar.
3. Mampu menangkap pengetahuan dan kepakaran seseorang.
4. Dapat beroperasi dialingkungan yang berbahaya.

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya (Sutojo et al., 2011):

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% benar.

Ciri-ciri dari sistem pakar adalah sebagai berikut: (Sutojo et al., 2011)

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data tidak pasti.
3. Dapat menjelaskan alasan dengan cara yang mudah dipahami.
4. Bekerja berdasarkan kaedah/*rule* tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

2.1.2 Web

World Wide Web (WWW) suatu program yang ditemukan oleh Tim Berneres-Lee pada tahun 1991. Awalnya Barners-Lee hanya ingin menemukan cara untuk menyusun arsip-arsip risetnya. Untuk itu, beliau merancang suatu sistem untuk keperluan pribadi. Sistem itu adalah program piranti lunak yang diberi nama Enquire. Dengan program itu, Berners-Lee berhasil menciptakan jaringan yang

memudahkan pencarian informasi yang dibutuhkan. Inilah yang kelak menjadi dasar dari sebuah perkembangan yang dikenal sebagai WWW. (Hidayatullah & Kawistara, 2014)

Web suatu layanan di dalam jaringan sebuah internet yang berupa ruang informasi dengan adanya *web*, *user* dapat memperoleh atau menemukan informasi yang diinginkan dengan cara mengikuti *link* (*hyperlink*) yang disediakan di dalam dokumen yang ditampilkan oleh aplikasi *webdbrowser*. (Raharjo, 2011)

Banyak keuntungan yang dapat diperoleh dari layanan *web*, diantaranya adalah sebagai berikut: (Raharjo, 2011)

1. Informasi mudah dikembangkan dan dapat diakses oleh semua pengguna dan penikmat internet diseluruh dunia.
2. Konfigurasi *server* dapatd proses secara lebih mudah
3. Instalasi (*upload*) aplikasi hanya dilakukan sekali, tanpa harus melakukan instalasi aplikasi disetiap komputer *user* yang ingin mengakses aplikasi tersebut.
4. Tidak tergantung pada *platform*, artinya informasi maupun aplikasi dapat diakses dari komputer yang memiliki sistem operasi berbeda.

2.1.3 Basis Data (*Database*)

Database atau basis data dapat diartikan sebagai sekelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar suatu saat dapat dimanfaatkan dengan cepat dan mudah. (Hidayatullah & Kawistara, 2014)

Menurut (S & M. Shalahuddin, 2013) sistem basis data sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. *Database* yaitu media untuk penyimpanan data agar dapat diakses dengan mudah dan cepat. 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 persyaratan minimal sebagai berikut (S & M. Shalahuddin, 2013)

1. Menyediakan fasilitas untuk mengelol akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dicoba secara bersamaan
4. Mampu menangani *backup* data

Ada beberapa versis *BMS* yang paling sering digunakan saat ini yaitu (S & M. Shalahuddin, 2013):

1. *DBMS* versi komersial yaitu, *Oracle*, *Microsoft SQL Server*, *IBM DB2*, dan *Microsoft Access*.
2. *DBMS* versi *open source* yaitu, *MySQL*, *PostgreSQL*, *Firebird*, dan *SQLite*.

Dalam alur hidup basis data (*Database Life Cycle*), terdapat tahapan yang dinamakan *physical database design*. Biasanya pada tahap ini dibuat rancangan fisik *database* yaitu *Physical Data Model (PDM)*. *PDM* merupakan konsep yang digunakan untuk menerangkan secara detail bagaimana data disimpan dalam *database*. *PDM* sudah dalam bentuk fisik perancangan *database* yang siap

diimplementasikan ke dalam *DBMS* sehingga nama tabel pada *PDM* merupakan nama asli tabel yang akan di implementasikan ke dalam *DBMS*.(S & M. Shalahuddin, 2013)

2.1.4 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 (S & M. Shalahuddin, 2013):

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.

Pengujian dengan membuat kasus penguji 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 mampu menghasilkan fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan.

2.2 Variabel

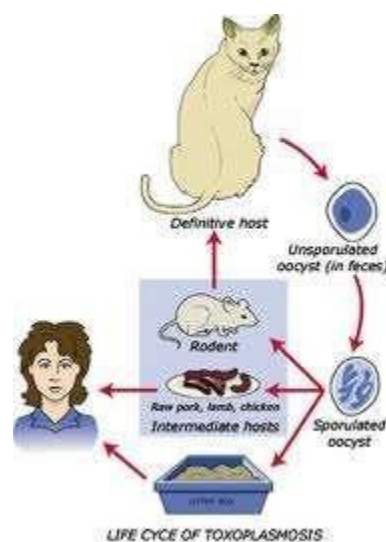
Variabel pada penelitian ini adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti memahami informasi tentang hal tersebut, kemudian ditarik kesimpulannya. Dinamakan variabel karena variasinya. Misalnya berat badan dapat disebut variabel, karena berat badan sekelompok orang itu bervariasi antara satu orang dengan yang lain.(Sugiyono, 2014). Variabel yang digunakan penelitian ini penyakit kucing.

2.2.1 Kucing

Dalam bahasa Mesir, kucing dipanggil “mau”. Kata ini berasal dari kata *meow* yang berarti dapat melihat. Penduduk Mesir mengakui kucing mempunyai kekuatan untuk melihat kebenaran dan dalam kehidupan setelah mati (Efendi & Budiana, 2014)

2.2.2 *Toxoplasma*

Toxoplasma ini jenis penyakit yang disebabkan oleh *protozoa parasit* atau *Toxoplasma gondii*. *Toxoplasma* biasanya menyerang Penyakit jenis mamalia yang berdarah panas, seperti kucing, anjing, kambing, kelinci dan lain-lain. Parasit *Toxoplasma gondii* binatang mamalia yang berdarah panas. Penularan *Toxoplasma* ini biasanya melalui kotoran kucing yang mengandung *okista* (telur *toxoplasma*), yang kemudian ikut tertelan kedalam tubuh kita (manusia). Maka dari itu kita dianjurkan selalu mencuci tangan menggunakan sabun setelah berintraksi dengan kucing atau binatang peliharaan lainnya. Untuk terhindar dari penyakit *toxoplasma* yang mungkin bisa terjadi.



Gambar 2.4 *Toxoplasma*
Sumber: Data Penelitian 2018

Gejala yang dialami oleh kucing yang terkena *toxoplasma* diantaranya flu, merasa lelah, nyeri pada kepala, demam. Pencegahan yang dapat dilakukan antara lain :

1. Selalu memandikan kucing secara teratur minimal 1 minggu sekali.

2. Bersihkan kandang kucing setiap hari serta membersihkan atau mencuci kandang kucing setiap 1 minggu sekali secara teratur.
3. Berikan tempat untuk membuang kotoran yang khusus, agar kucing tidak membuang kotoran disembarang tempat yang dapat mendatangkan parasit *toxoplasma gondii* yang sangat berbahaya pada kucing pemeliharaan.
4. Berikan kucing peliharaan pakan yang matang serta sehat, hindari pemberian makan daging-dagingan yang mentah, itu juga bisa membuat kucing mencret.
5. Memberikan vaksin kucing secara teratur agar kucing tidak terjangkit penyakit lainnya seperti rabies, mencret, dan lain-lain.

2.2.3 *Pyometra*

Pyometra adalah kondisi medis dimana pus (nanah) terakumulasi di badan. Pada saat kondisi ini terjadi, cairan akan mulai mengisi dinding *uterus* yang mengalami penebalan, terjadi infeksi dan *uterus* akan terisi cairan berupa nanah.



Gambar 2.5 Gejala *Pyometra*
Sumber: Data Penelitian 2018

karena pertahanan tubuh terhadap infeksi. Pada kondisi ini, *cervix* akan terbuka sehingga nanah keluar melalui vagina dan dapat mengeluarkan bau yang tidak sedap, Kondisi yang dapat mengancam jiwa hewan. Namun, jika kondisi *cervix* tertutup, maka *pus* (nanah) akan terjebak didalam uterus dan tidak dapat dikeluarkan dari dalam tubuh hewan. Pada kondisi ini, infeksi akan mengancam kematian pada hewan (*ovariohysterectomy*). Kondisi *pyometra* dapat terjadi pada setiap kucing betina yang tidak atau belum *disterilisasi*. Tetapi usia 3 tahun keatas dan usia tua (lebih dari 7 tahun) merupakan usia yang beresiko tinggi. *Pyometra* biasanya terjadi pada saat 1-2 bulan setelah kucing mengalami *estrus (heat)* yang terakhir.

Gejala klinis yang dapat dilihat tergantung dari terbuka atau tidaknya *cervix*. Jika dalam keadaan *cervix* terbuka, maka nanah (*pus*) akan mengalir keluar dari *uterus* melalui vagina dari luar. Nanah tersebut akan berwarna kekuningan, kecoklatan atau kemerahan. Biasanya leleran dapat terlihat di bulu dekat ekor, alas kandang (*bedding*) dan tempat dimana kucing biasanya tidur atau bermain. Pemilik hewan tidak melihat leleran nanah dikarenakan leleran terlebih dahulu dibersihkan (dijilat) oleh hewan. Demam, lemah, kehilangan nafsu makan dan depresi dapat juga terlihat sebagai gejala klinis. Jika *cervix* dalam keadaan tertutup rapat, leleran nanah tidak dapat mengalir keluar tubuh dan tidak dapat terlihat. Nanah akan terakumulasi di *uterus* dan menyebabkan *uterus* menjadi membesar (bengkak). Dari kasus-kasus yang lain pembengkakan di *abdomen* dapat terlihat. Bakteri akan mengeluarkan racun (*toksin*) yang akan diserap sistem sirkulasi darah. Hewan yang mengalami kejadian *pyometra* biasanya akan terlihat sakit. Bakteri mempengaruhi

kemampuan ginjal untuk mempertahankan cairan. Hal tersebut dapat terjadi baik di *pyometra* dengan *cervix* terbuka ataupun tertutup.

Diagnosa terhadap *pyometra* terkadang sangat susah, terlebih pada kasus dimana *cervix* dalam keadaan tertutup dimana tidak ada leleran nanah terlihat. Karena. Uji darah dan urin dapat dilakukan untuk membantu peneguhan diagnosa. Hewan yang mengalami *pyometra* biasanya akan menunjukkan peningkatan pada penghitungan sel darah putih (*White Blood Cell Count*) dan urin biasanya encer (*dilute*). Namun, abnormalitas seperti ini dapat terlihat juga dikebanyakan kasus dengan infeksi bakteri. Pemeriksaan radiografi (x-ray) atau *USG* dibagian perut (*abdomen*) dapat membantu untuk mengidentifikasi pembesaran *uterus* dan membedakan nya dengan kebuntingan. Sekaan (swab) dapat diambil dari vagina untuk membantu mengidentifikasi jenis bakteri yang dapat menyebabkan infeksi. Penanganan terhadap *pyometra* dengan melakukan pembedahan (*Surgery*). Yaitu penanganan yang dianjurkan untuk mengangkat (*remove*) *uterus* dan ovarium.

Penanganan ini dinamakan *ovariohysterectomy* (*spay*) dan penanganan ini adalah penanganan yang paling aman dan efektif untuk menangani kasus *pyometra*. Pada penanganan kasus *pyometra* ini sangatlah berbeda dibandingkan dengan pembedahan *spay* yang rutin dilakukan pada hewan yang sehat. Pada saat mengangkat *uterus* harus dilakukan secara hati-hati dikarenakan besar dan lemahnya *uterus*, serta jangan ada leleran nanah yang keluar selama pembedahan karena akan mencemari jaringan organ yang lain. Cairan *intravena* diperlukan sebelum dan setelah pembedahan. *Antibiotik* dapat diberikan selama 1 sampai 2 minggu. *Pyometra* dapat dicegah dengan mensteril sasikan hewan (*desexin* atau

spay) pada saat mereka masih muda. Pada saat usia yang dianjurkan untuk melakukan sterilisasi hewan kucing yang tidak dibiakan (*breeding*) pada umur 5-6 bulan.

2.2.4 Mastitis

Mastitis penyakit radang ambing yang merupakan radang infeksi. Biasanya penyakit ini berlangsung secara akut, sub akut maupun kronis. *Mastitis* ditandai dengan peningkatan jumlah sel di dalam air susu, perubahan fisik maupun susunan air susu dan disertai atau tanpa disertai perubahan *patologis* atau kelenjarnya sendiri. Di Indonesia, kasus *mastitis* masih banyak terjadi, terutama pada peternakan kecil yang kurang memperhatikan kondisi kandang maupun tingkat kebersihannya.



Gambar 2.6 Gejala *Mastitis*
Sumber: Data Penelitian 2018

Mastitis ditandai dengan peningkatan jumlah sel di dalam air susu, perubahan fisik maupun susunan air susu dan disertai atau tanpa disertai perubahan *patologis*

atau kelenjarnya sendiri. Menurut faktor penyebabnya, mastitis dapat disebabkan oleh bakteri *Streptococcus agalactiae*, *Str. dysgalactiae*, *Str. uberis*, *Str. zooepidemicus*, dan *Staphylococcus aureus*, serta berbagai spesies lain yang juga bisa menyebabkan terjadinya *mastitis* walaupun dalam persentase kecil. Sindrom biasanya terjadi dalam waktu 12 jam setelah kelahiran. Biasanya tanda pertama diikuti oleh depresi, gelisah ketika sedang menyusui dan melemahnya kondisi anak kucing. Terjadi demam pada induk kucing 39,5-41 °C jika *mastitis* hadir. Dalam banyak kasus, hanya satu kelenjar *mastitic*. Penyakit ini berlangsung selama minimal 3 hari dan kemudian sembuh secara spontan. Kondisi dapat didahului oleh penundaan dalam proses kelahiran (diatas 5 jam) dan dapat bervariasi dalam intensitasnya. Dalam kasus susu tanpa disertai *hypogalactia mastitis* atau unsur-unsur lain yang kompleks, akan berkurang berat badan pada anak-anak kucing (105 gr per hari, normal 125gr per hari) mungkin satu-satunya indikasi dari masalah. Penyebab terjadinya antara lain:

1. Lantai yang terlampau kasar dan kotor terjadi gesekan yang mengakibatkan luka-dalam pada kantung susu.
2. Lantai yang kotor, tidak berarti lantai yang bersih tidak terlihat kotoran berarti bersih, kadang kala berkumpul kuman-kuman pada pori-pori lantai semen atau lantai yang beralaskan kayu.
3. Kebiasaan membersihkan kucing pada bagian tersebut dengan cara mengelap dengan handuk ataupun sejenisnya akibat tekanan dan gesekan menimbulkan luka didalam kantung susu kucing tersebut tanpa kita sadari.

4. Kebiasaan keingin tahuan kita, apakah kucing kita hamil atau tidak dengan menekan/memencet puting.
5. Radang kulit seputar kantong susu, sehingga kucing tersebut mengaruk-garuk dengan kakinya

Untuk pengobatannya sendiri penggunaan antibiotik sangatlah tepat untuk pengobatan penyakit ini, terutama *penicillin* (*Benzyl penicillin G*, *procain penicillin-G*, *ampicilin*), *cephalosporin*, *erythromycin*, *neomycin*, *novobiosin*, *oksitetrasiklin*, dan *streptomycin*. Dewasa ini, seiring dengan banyaknya penelitian penelitian yang telah dilakukan beberapa sejawat dokter hewan maupun mahasiswa di kalangan medis *veteriner*, semakin banyak pula penemuan-penemuan baru yang menarik untuk dicoba dan dibuktikan kebenarannya.

2.2.5 Distemper



Gambar 2.7 Gejala *Distemper*
Sumber: Data Penelitian 2018

Distemper merupakan salah satu penyakit yang sangat menular. Penyakit ini merupakan virus yang bersifat sistemik. *Distemper* sendiri merupakan salah satu

penyakit dengan kematian yang sangat tinggi. Apalagi jika penyakit ini menyerang anak kucing, anjing atau *puppy*. Penyakit yang cukup berbahaya ini disebabkan infeksi virus *Distemper* atau *Canine Distemper Virus*. Virus ini juga lebih suka menyerang kucing yang masih muda daripada yang sudah dewasa. Semua jenis serta segala umur kucing bisa saja terserang penyakit *distemper* ini. Apalagi anak kucing yang berumur 3-6 bulan jauh lebih rentan terinfeksi serta mengalami komplikasi yang serius. Seperti munculnya peradangan paru-paru atau *pneumonia*, serta peradangan otak atau *encephalitis*. Gejala ini berbeda dibandingkan dengan gejala yang akan timbul pada kucing dewasa.

Virus *distemper* dapat menyerang beberapa sistem pada tubuh kucing, yaitu sistem pernafasan, sistem pencernaan, sistem kekebalan tubuh, sistem syaraf dan sistem urogenital. Karena banyaknya sistem tubuh yang diserang (bisa satu atau ketiganya sekaligus), gejalanya pun bisa beragam. Gejala pada tahap awal bisa berupa :

1. Demam Tinggi

Salah satu gejala *distemper* yang biasanya terjadi adalah demam tinggi. Sama dengan manusia pada umumnya, kucing memiliki suhu normal di sekitaran 38-39.5 derajat Celcius. Terkadang, Demam pada kucing ini ditandai dengan menggigil dan gemetaran.

2. Radang Pada Mata

Kucing tidak mengalami demam tinggi saja, Kucing yang terjangkit *distemper* awalnya akan terkena radang pada mata. Biasanya peradangan ini akan

disertai kotoran mata atau belek yang cukup banyak, tebal serta berwarna kekuningan. Jika sudah terjadi harus rutin membersikannya.

3. Ingus Pada Hidung

Kucing yang terkena *distemper* adalah mengeluarkan cairan dalam hidung dengan warna sedikit kekuningan.

4. Depresi

Kucing terkena *distemper* yaitu kucing yang terlihat stres. Kucing yang terlihat murung dan sulit untuk diajak bermain atau berintraksi. Saat kucing yang depresi akan terlihat tidak mempunyai energi saat mengibaskan ekornya. Depresi yang kucing alami bisa membuat perubahan atau gangguan dalam tidurnya.

5. Nafsu Makan Hilang

Pada saat kucing yang terkena *distemper* nafsu makan hilang, Jika nafsu makan kucing hilang pemelika dapat memaksa kucing untuk makan, atau memberikan makanan favorit kucing. Tidak diperbolehkan memberikan makanan yang mentah agar terhindar dari memperparah nafsu makan.

6. Batuk Kering

Selain nafsu makan hilang kucing juga sering mengalami batuk kering. Jadi saat mendengar kucing terbatuk-batuk kecil. Jangan khawatir, karena biasa terjadi pada kucing yang terkena *distemper*. Bahkan terkadang disertai dengan ingus yang berwarna kekuningan.

7. Benjolan Bernanah di Bagian Perut

Saat mereba berut kucing bisa merasakan benjolan pada area perut kucing. Harus berhati-hati pada saat merabanya, karena benjolan ini merupakan benjolan

yang berisi nanah. Usahakan untuk tidak menekan terlalu kuat agar benjolannya tidak pecah.

8. Muntah Diare

Jika saat *distemper* mulai parah, kucing biasanya akan mengalami muntah disertai diare. Untuk mengurangi diare sementara menghentikan pemberian makanan menghentikannya selama 12 sampai 24 jam. Menghentikan memberi makan bertujuan untuk menenangkan saluran pencernaan. Sediakan air bersih agar kucing bisa minum disaat merasa lapar.

9. Jalan Serampangan

Selain muntah yang disertai diare, kucing akan terlihat jalan serampangan. Kucing berjalan tidak lurus dan terasa lemas. Hal biasa terjadi kepada kucing yang terserang *distemper* yang sudah terlanjur parah.

10. Kepala gemetar

Selain jalan yang serempangan, kucing juga terlihat gemetar. Bahkan terkadang terlihat seperti menggerakkan rahang yang terlihat seperti mengunyah terkadang terlihat tidak terkendali.

11. Kejang-kejang

Gejala lain yang dialami oleh kucing *distemper* adalah kejang-kejang serta kebingungan. Pada saat kucing sedang kejang-kejang. Jangan mencoba memegang

mulut kucing untuk mencegah menggigit lidahnya, Karena jika melakukannya pada saat kucing kejang-kejang kan membuat kucing tersebut bertambah stres..

12. Gerakan Kaki Tidak Terkoordinasi

Salah satu gejala yang akan timbul pada saat kucing terkena *distemper* yang parah adalah gerakan kaki yang tidak terkoordinasi. Kontraksi otot atau kedutan biasanya akan tidak terkontrol, ritmis serta terjadi kekakuan. Biasanya ini terjadi pada otot disekitar leher serta kepala.

13. Kelumpuhan

Jika kucing sudah parah kucing bisa mengalami kelumpuhan. Bahkan kucing bias mengalami kebutaan. Selian ini akan sulit untuk bernapas. Pada umumnya, belum ditemukan obat yang bisa mengobati *distemper* pada kucing. Hal ini dikarenakan *distemper* memiliki sifat suportif dan *simptomatis*. Namun, berikut beberapa cara mencegah *distemper* pada kucing, diantaranya:

1. Memberikan *antibiotik spektrum* untuk pencegahan *invasi* bakteri sekunder.
2. Memberikan cairan elektrolit untuk menjaga keseimbangan cairan.
3. Memberikan *antipiretik*.
4. Memberikan *analgesik* pereda sakit
5. Memberikan nutrisi dengan *parenteral*.
6. Memberikan *discharge* pada bagian mata dan hidung secara rutin.
7. Memberikan *glukokortikoid* untuk mencegah kebutaan.
8. Memberikan *antikonvulsan* untuk mengendalikan kejang.

2.2.6 *Pthiriasis* (Kutu)

Kutu kucing (tungau) dapat menjadi masalah yang cukup serius apabila tidak segera mendapatkan penanganan yang benar. Hewan parasit seperti pinjal pada kucing yang dibiarkan berkembang biak akan membuatnya terasa gatal yang disebabkan oleh pinjal. Dengan begitu kucing akan sering menggaruk, dan dalam beberapa kasus bahkan kucing akan menggigit tubuhnya sendiri. Jika masalah ini dibiarkan dalam waktu yang lama, maka kulit si kucing akan menimbulkan alergi, lecet bahkan infeksi sekunder.



Gambar 2.8 Serangan Parasit
Sumber: Data Penelitian 2018

Gejala serangan kutu yang terjadi pada kucing sangatlah bervariasi, contohnya seperti berikut ini:

1. Masalah kulit

Pada tahap awal serangan hewan parasit tersebut pada kucing biasanya akan terjadi iritasi kulit. *Pustula*, gatal serta benjolan akan muncul akibat sensitivitas kucing peliharaan kita terhadap air liur dari dampak seekor kutu.

2. Sering menggaruk

Kucing akan terlihat sering menggaruk tubuhnya dikarenakan respon alergi dan gatal terhadap gigitan pinjal.

3. Bulu rontok

Kerontokan bulu akan terjadi apabila sering menggaruk dikarenakan infeksi. Jika terjadi iritasi kulit, maka periksalah apakah ada area botak yang menandakan terjadinya serangan kutu.

4. Lemah dan lesu

Meski mendapatkan makanan normal biasanya kucing yang terkena serangan hewan parasit akan terlihat lemah dan lesu. Jika serangan parasit tersebut cukup parah maka akan mengakibatkan dehidrasi.

5. Menghindari suatu tempat

Kucing akan terlihat menghindari suatu tempat yang yang berbahan kayu, terutama jika tempat tersebut biasanya menjadi tempat favorit. Mungkin hal tersebut dikarenakan tempat tersebut merupakan sarang kutu.

6. Kucing terlihat gelisah

Karna rasa gatal yang membuatnya bisa gila, kucing akan tiba-tiba berperilaku liar. Dari sering menggeram, lari melesat hingga menggosok-gosok kepala dan tubuh pada lantai.

7. Anemia

Gejala terakhir yang didapat kucing saat terserang pinjal adalah anemia. Kucing akan semakin banyak kehilangan darah apabila hewan parasit seperti pinjal

yang hidup pada tubunya semakin banyak. Kondisi seperti ini harus segera ditangani oleh dokter hewan.

2.3 Software Pendukung

Software pendukung merupakan beberapa perangkat lunak yang digunakan untuk mendukung pembuatan sistem pakar pada penelitian ini. Perangkat lunak tersebut antara lain adalah *Star UML*, *HTML5*, *CSS3*, *PHP*, *MySQL*, *XAMPP*, *PHPMyAdmin*, *Notepad++*, dan *Bootstrap*.

2.3.1 XAMPP (X Apache MySQL PHP Perl)

Menurut (Dudung, 2016) XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.



Gambar 2.9 Logo XAMPP
Sumber: Aplikasi 2018

2.3.2 PHPMYAdmin dan MySQL

PHP singkatan dari “*Hypertext Preprocessor*” yang digunakan sebagai bahasa *scrip server side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Berbeda dengan HTML yang hanya dapat menampilkan konten statis, PHP bisa berinteraksi dengan *database*, *file* dan *folder*, sehingga membuat PHP bisa menampilkan konten yang dinami udari sebuah *website*. Karena penggunaan PHP memungkinkan *web* dapat dibuat dinamis, *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. PHP *software oper source* yang disebar dan dilisensikan secara Cuma-Cuma serta dapat di-*download* secara bebas dari situs reminya <http://www.php.net>. PHP ditulis menggunakanyahasa C. (KUNCIKOM, 2012, p. 5)



Gambar 2.10 Logo phpMyAdmin
Sumber: Aplikasi 2018

Menuruti(Saputra, 2012, p. 77) MySQL salah satu *database* kelas dunia yang cocok jika dikaitkan dengan bahasa pemograman PHP. MySQL bekerja dengan menggunakan bahasa SQL (*Structure Query Language*) yang merupakan bahasa standar yang digunakan untuk memanipulasi *database*. Pada umumnya perintah yang paling sering digunakan dalam MySQL adalah *SELECT* (mengambil),

INSERT (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, atau pun *index* untuk menambah atau *query* language tersebut pada sub-sub selanjutnya.

Menurut (Saputra, 2012, p. 78) alasan yang menjadikan *database* MySQL sangat diminati oleh para programmer, diantaranya:

1. Bersifat *openisource*.
2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dalam pengolahan data.
3. *Performance* dan *reliable*, pemrosesan *database* nya sangat cepat dan stabil.
4. Sangat mudah dimengerti.
5. Memiliki dukungan *(group)* pengguna MySQL.
6. Lintas *Platform*, dapat digunakan pada berbagai sistem operasi berbeda.
7. *Multiuser*, dimana MySQL dapat dipergunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.



Gambar 2.11 Logo MySQL
Sumber: Aplikasi 2018

Menurut Madcoms (2016: 2) UlfaMicheal Widenius penemu awal versi pertama *MySQL* yang kemudian pengembangan selanjutnya dilakukan oleh perusahaan *MySQL AB*. *MySQL AB* yang merupakan sebuah perusahaan komersial

yang didirikan oleh pengembang *MySQL*. *MySQL* seluruh tipe data numerik standar ANSI (*American National Standards Institute*). Berikut ini tipe-tipe yang digunakan dalam *MySQL*:

1. Di bawah ini adalah tipe data numerik yang biasanya digunakan beserta penjelasannya dari tipe data numerik:

Tabel 2. 3 Tipe Data Numerik

Tipe Data	Deskripsi
<i>INT</i>	Nilai integer yang bisa bertanda atau tidak. Jika bertanda, maka rentang yang diperbolehkan.
<i>TINYINT</i>	Nilai integer yang sangat kecil. Rentang nya 128 – 127 untuk yang bertanda dan 0 – 255 untuk yang tidak bertanda.
<i>SMALLINT</i>	Nilai integer yang sangat kecil dengan rentang 31768 sampai 32767 dihasilkan untuk yang tidak bertanda dari 0 – 65535
<i>MEDIUMINT</i>	Integer yang berukuran sedang dan rentang -83886083 sampai 8388607 atau 0 sampai 16777215
<i>BIGINT</i>	Integer dengan ukuran besar dengan rentang -9223372036854775808 sampai 922337203685477580 atau 0 sampai 18446744073709551615
<i>FLOAT(M,D)</i>	Bilangan pecahan dengan panjang (termasuk jumlah desimal) M dan jumlah desimal D. Presisi desimalnya bisa sampai 24 digit. <i>Default Float</i> (10,2). Bilangan <i>float</i> selalu bisa bertanda
<i>DOUBLE(M,D)</i>	Adalah bilangan pecahan dengan presisi dua kali lipat hasil. Panjang (termasuk jumlah desimal) dan dengan jumlah desimal D. Presisi desimalnya bisa sampai 53 digit. <i>Default Double</i> (16,4). Bilangan <i>float</i> selalu bertanda. Sinonim dari <i>DOUBLE</i> adalah <i>REAL</i> .
<i>DECIMAL(M,D)</i>	Adalah bilangan pecahan dan harus didefinisikan M dan D nya Setiap desimal membutuhkan tempat 1 byte. Sinonim dari <i>DECIMAL</i> adalah <i>NUMERIC</i> .

Sumber: Rekayasa Perangkat Lunak 2013

2. Di bawah ini adalah tipe data tanggal dan waktu yang digunakan di dalam *MySQL* :

Tabel 2. 4 Tipe Data Tanggal dan Waktu

Tipe Data	Deskripsi
<i>DATE</i>	Adalah tipe data tanggal dengan format YYYY-MM-DD, diantara 1000-01-01 dan 9999-12-31.
<i>DATETIME</i>	Adalah kombinasi tanggal dan waktu dengan format dYYYY-MM-DD HH:MM:SS dan rentang data antara 1000-01-01 00:00:00s sampai dengan 9999-12-31 23:59:59.
<i>TIMESTAMP</i>	Penanda waktu antara 1 Januari 1970 tengah malam sampai dengan tahun 2037. Formatnya mirip dengan <i>DATETIME</i> tetapi tanpa pembatas di antara angkanya.
<i>TIME</i>	Penyimpanan waktu dalam format HH:MM:SSdx
<i>YEAR(M)</i>	Menyimpan data tahun dalam format 2 atau 4 digit. Jika M diisi dengan nilai 2, maka rentang tahunnya dari 19702069 sedangkan jika M diisi dengan nilai 4 maka <i>YEAR</i> bisa bernilai 1901 sampai dengan 2155. Default nilai M adalah 4.

Sumber: Rekayasa Prangkat Lunak 2013

3. Berikut ini adalah tipe data string yang paling umum didalam *MySQL*:

Tabel 2. 5 Tipe Data *String*

Tipe Data	Deskripsi
<i>CHAR(M)</i>	String dengan ukuran yang sama. Ukurannya diantara 1 sampai 255 karakter. Ukuran ditentukan dengan nilai M. Contoh: <i>CHAR(6)</i>
<i>VARCHAR(M)</i>	String dengan ukuran bervariasi diantara 1 sampai dengan 255 karakter. Contoh: <i>VARCHAR(25)</i>

Tabel 2. 5 Lanjutan

<i>TEXT</i>	String dengan ukuran maksimum 65535 karakter. String yang tersimpan di dalam TEXT dianggap tidak case sensitive. Untuk kapasitas yang lebih kecil bisa menggunakan TINYTEXT dengan kapasitas maksimal 255 karakter, sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMTEXT (maksimal 16777215 karakter) dan LONGTEXT (maksimal 4294967295 karakter).
<i>BLOB</i>	<i>Binary Large Object</i> (BLOB) adalah data yang dipergunakan untuk menyimpan data binary dalam jumlah besar. Biasa digunakan untuk menyimpan citra. Untuk penyimpanan data yang lebih kecil bisa menggunakan TINYBLOB (maksimal 255 karakter) sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMBLOB (maksimal: 16777215 karakter) dan LONGBLOB (maksimal: 4294967295 karakter).
<i>ENUM</i>	Enumerasi atau sebuah list (daftar). Contoh: ada sebuah nilai yang dibatasi hanya boleh dengan nilai tertentu saja. Misalnya saja nilai itu hanya bisa terdiri dari A-E, maka bisa dibuat menjadi ENUM('A', 'B', 'C', 'D', 'E')

Sumber: Rekayasa Perangkat Lunak 2013

2.3.3 PHP (Hypertext Preprocessor)

PHP singkatan dari “*Hypertext Preprocessor*” yang digunakan sebagai bahasa *scrip server side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan *database*, *file* dan *folder*, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah *website*. Karena penggunaan PHP memungkinkan *web* dapat dibuat dinamis, *maintenance* situs *web* tersebut menjadi

lebih mudah dan efisien. PHP merupakan *software oper source* yang disebarakan dan dilisensikan secara gratis. PHP ditulis menggunakan bahasa C. (KUNCIKOM, 2012, p. 5)



Gambar 2.12 Logo *PHP (Hypertext Preprocessor)*
Sumber: Aplikasi 2018

PHP sudah menjadi bahasa *scripting* pada umumnya yang banyak dipergunakan di kalangan *developer web*. *PHP* mempunyai banyak kelebihan menjadi alasan utama kenapa *PHP* lebih terpilih sebagai basis umum dalam membuat sebuah *web*. (Hidayatullah & Kawistara, 2014)

PHP bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. *PHP* disebut sebagai pemrograman *server-side* karena *PHP* diproses pada komputer server. dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser (client)*. (Madcoms, 2016)

PHP dapat dipergunakan secara gratis (*free*) dan bersifat *Open Source*. *PHP* dirilis dalam lisensi *PHP License*, berbeda dengan lisensi *GNU General Public License (GPL)* yang biasa digunakan untuk proyek *Open Source*. (Madcoms, 2016)

PHP dapat membuat tugas-tugas yang dilakukan dengan mekanisme *CGI (Common Gateway Interface)* seperti mengambil, mengumpulkan data dari

database, meng-*generate* halaman dinamis, atau bahkan menerima dan mengirim *cookie*. (Hidayatullah & Kawistara, 2014)

Berikut ini adalah beberapa kelebihan lain dari *PHP* (Madcoms, 2016):

1. *Web* menjadi dinamis.
2. *PHP* bersifat *Open Source* yang berarti dapat oleh siapa saja secara gratis.
3. Program yang dibuat dengan *PHP* bisa dijalankan oleh semua Sistem Operasi (OS) karena *PHP* berjalan secara *web base* yang diartikan semua Sistem Operasi bahkan *handphone web browser* dapat menggunakan program *PHP*.
4. Aplikasi *PHP* lebih cepat dibandingkan dengan *ASP* maupun *Java*.
5. Mendukung banyak paket *database* seperti *MySQL*, *Oracle*, *PostgreSQL*.
6. Pemrograman *PHP* tidak memerlukan kompilasi (*compile*) dalam penggunaannya.
7. Banyak *web server* yang mendukung *PHP* seperti *Apache*, *Lighttpd*, *IIS*.
8. Pengembangan aplikasi *PHP* tidak banyak dokumentasi, referensi, dan *developer* yang membantu dalam pengembangannya.
9. Banyaknya aplikasi dan program *PHP* yang tersebar secara gratis dan siap pakai seperti *WordPress*, *PrestaShop*.

2.3.4 Bootstrap

Bootstrap (Otto & Thornton, 2016) *Bootstrap* telah menjadi salah satu *front end framework* dan proyek *open source* terkenal di dunia. *Bootstrap* diciptakan oleh Twitter di pertengahan 2010 oleh Mark Otto dan Jacob Thornton. Sebelumnya,

Bootstrap dikenal sebagai Twitter Blue Print. Twitter pertama kalinya menyelenggarakan *Hack Week* dan proyek ini meledak sebagai proyek pengembangan dari semua tingkat keahlian tanpa ada bimbingan. Dan menyediakan panduan gaya untuk alat pengembangan internal di perusahaan selama lebih dari satu tahun sebelum dirilis untuk publik dan terus melakukannya hingga hari ini.



Gambar 2. 13 Logo Bootstrap
Sumber: Aplikasi 2018

Bootstrap dirilis pada Jumat, 19 Agustus 2011, setelah kami merilis 20 lebih, termasuk dua penulisan utama dengan versi 2 dan versi 3. Dengan *Bootstrap 2*, kami menambahkan fungsi responsif ke seluruh *framework* sebagai *stylesheet* opsional. *Bootstrap 3*, membantu para developer untuk membuat tampilan responsif secara default dengan ponsel.

2.3.5 HTML (*Hypertext Markup Language*)

Menurut (Saputra, 2012, p. 1) HTML singkatan dari *Hyper Text Markup Language*. HTML bisa disebut bahasa paling dasar dan sangat penting yang dipergunakan untuk menampilkan dan mengolah tampilan pada halaman *website*.

HTML yang paling marak dibicarakan didunia maya. HTML 5 layaknya sebuah HTML yang sering dipergunakan aplikasi *web*, hanya saja HTML 5 ini memiliki keunggulan tersendiri bandingkan versi lamanya menjadi lebih ringkas. Yang paling mencolok adalah tersediannya fitur baru seperti elemen multimedia, yang tak lain adalah fungsi untuk memutar audio dan juga video. Pada versi HTML sebelumnya, Dengan adanya penambahan fitur dan tag khusus yang ada pada HTML 5, membuat semuanya lebih mudah (Saputra, 2012).



Gambar 2.14 Logo HTML
Sumber: Aplikasi 2018

Dokumen *HTML* memiliki sebuah struktur yang harus diikuti aturan pembuatannya. Berikut ini adalah elemen yang wajib ada di dalam file *HTML* yang apabila ingin membangun pondasi kerangka *website*: (A. Saputra, 2012)

1. Elemen *HTML*

Elemen dasar dalam memulai dokumen *HTML*. Tag ini dipergunakan bahwa dokumen *HTML*. Contoh penulisannya adalah: **<html>d** dan diakhiri dengan **</html>**.

2. Elemen *Head*

Head merupakan *tag* yang berfungsi untuk menuliskan keterangan tentang dokumen *web* yang akan ditampilkan. Elemen ini nantinya akan diakhiri dengan tanda penutup **</head>**. Secara runtun dapat diktik dengan format berikut:

```
<html>
```

```
<head>
```

```
</head>
```

```
</html>
```

3. Elemen *Title*

Elemen *title* berfungsi untuk menambah pada *caption browser web* tentang topik atau tema atau judul dari suatu dokumen *web* yang ditampilkan pada *browser*. Berikut struktur penggunaannya:

```
<html>
```

```
<head>
```

```
    <title> ..judul dokumen HTML </title>
```

```
</head>
```

```
</html>
```

4. Elemen *Body*

Elemen *body* bagian paling pertama dalam dokumen *web*. Berfungsi sebagai *tag* yang menghasilkan suatu teks atau informasi atau yang dikenal dengan sebutan konten, maka teks tersebut diletakkan pada elemen *body*. Berikut ini struktur elemennya adalah:

```
<html>  
  
<head>  
  
    <title> ..judul dokumen HTML </title>  
  
</head>  
  
<body>  
  
    ..isi dokumen HTML  
  
</body>  
  
</html>
```

2.3.6 CSS (*Cascading Style Sheet*)

Cascading Style Sheet (CSS) adalah suatu bahasa yang bekerja sama dengan dokumen *HTML* untuk mendefinisikan cara bagaimana suatu isi halaman *web* yang ditampilkan atau dipresentasikan. Presentasi ini meliputi *style* atau gaya teks, *link*, maupun letak (*layout*) pada halaman. (Raharjo, 2011)



Gambar 2.15 Logo CSS3
Sumber: Aplikasi 2018

2.3.5 Notepad++

Notepad++ sebuah *text* editor yang sangat dibutuhkan oleh semua orang dan khususnya bagi para *developer* dalam membuat program. *Notepad++* menggunakan komponen *Scintilla* untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan di atas sistem operasi *Microsoft Windows*. (Madcoms, 2016).



Gambar 2.16 Logo Notepad ++
Sumber: Aplikasi 2018

Beberapa urutan bahasa pemrograman yang didukung oleh *Notepad++* adalah C, C++, Java, C#, dXML, dHTML, hPHP, Javascript dan masih banyak lagi

bahasa program yang didukung. *Notepad++* mempunyai fitur yang memudahkan untuk membedakan alur sintaks yaitu fitur *highlighting*. Fitur ini akan menandai sintaks dan variabel yang digunakan dalam *source code*. Tersedianya fitur *tab* yang dapat membantu mengelola beberapa kode dalam waktu bersamaan. Juga ada fitur penomoran baris turut membantu dalam mencari kode yang *error* ketika dijalankan. *Function List* tersedia untuk mengelola. *Notepad++* ini juga disertai fitur dalampencarian yang sangat mudah dan praktis. Suatu kata kunci yang dimasukkan, maka *Notepad++* bisa menandai semuanya dalam *source code*. Serta, bisa mengubah semuanya hanya dengan sekalin erintah karena ada fitur “*Replace All*”. (Madcoms, 2016)

2.3.7 Star UML

Star UML salah satu *CASE (Computer-Aided Software Engineering) tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upper CASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak (S & Shalahuddin, 2013)



Gambar 2.17 Logo StarUML
Sumber: Aplikasi 2018

Salah satu pemodelan yang saat ini yang banyak dipergunakan adalah *UML* (*Unified Modeling Language*). *UML* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, memuat analisis dan desain, untuk menggambarkan arsitektu bdalam pemrograman berorientasi objek. (S & Shalahuddin, 2013)

UML ada karena ada nya pemodelan visual untuk menspesifikasikan, menggambar, membangun dan dokumentasi dari sistem perangkat lunak. *UML* tidak membatasi pada program, Meskipun sebenarnya *UML* banyak digunakan metodologi berorientasi objek. (S & Shalahuddin, 2013).

Ada terdapat tiga macam diagram dalam *UML* 2.3 yang dibagi menjadi 3 kategori yaitu (S & Shalahuddin, 2013).

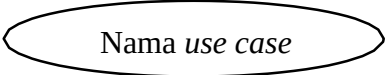
1. *Structure diagrams* yaitu sekumpulan diagram yang berguna untuk menggambarkan *UML*.
2. *Behavior diagram* tergolong diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram* yaitu sekumpulan diagram yang menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Dalam penelitian ini, berikut diagram yang penulis gunakan dalam membuat desain sistem yaitu:

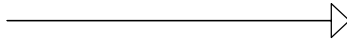
1. *Use Case Diagram*

Use case atau diagram *use case* pemodelan Buntut kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih faktor dengan sistem informasi yang akan dibuat. *Use case* sangat berfungsi mengetahui apa yang ada didalam sebuah sitem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu *use case* dan aktor (S & Shalahuddin, 2013).

Tabel 2. 6 Simbol *Use Case Diagram*

Simbol	Deskripsi
<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>
Aktor/<i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi/<i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi/<i>extend</i> <<extend>> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)

Tabel 2. 6 Lanjutan

Menggunakan/ <i>include/uses</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana
<<include>>	<i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini
	Untuk menjalankan fungsinya atau sebagai syarat di
<<uses>>	Jalankannya <i>use case</i> ini. Arah panah mengarah
	pada <i>use case</i> yang ditambahkan.

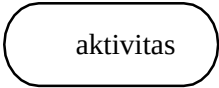
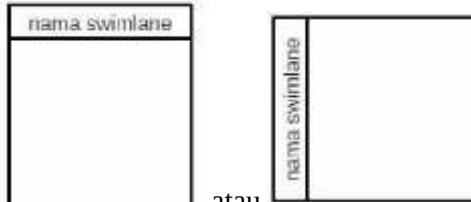
Sumber: Rekayasa Perangkat Lunak 2013

2. Activity Diagram

Menurut (A.S & Shalahuddin, 2013a, p. 161) diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujiannya.
4. Rancangan menu yang tampil pada perangkat lunak.

Tabel 2. 7 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, disebut diagram aktifitas memiliki sebuah status awal.
Aktivitas 	Aktifitas yang dilakukan sistem, aktifitas Biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu.
Status akhir 	Dalam hasil akhir yang dilakukan oleh sistem, adalah sebuah diagram aktifitas yang memiliki sebuah hasil akhir.
Penggabungan/ <i>join</i> 	Asosiasi dalam penggabungan dimana lebih sering aktifitas tergabung menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitasnya yang terjadi.

Sumber: Rekayasa Perangkat Lunak 2013

2.4 Penelitian Terdahulu

Penelitian terdahulu digunakan dalam mendukung teori yang berkaitan dengan penelitian, Berikut ini peneliti mencantumkan beberapa penelitian terdahulu untuk mendukung dan membantu penelitian ini:

1. **Tutur Larasati, M. Rudyanto Arief (2016)**, dengan ISSN 2338-137 mengatakan, **Sistem Pakar Diagnosa Awal Penyakit Kucing Berbasis Web Menggunakan Metode *Certain Factor*** menjelaskan bahwa media konsultasi merupakan aplikasi dari sistem pakar berbasis web dengan menggunakan fakta dan teknik penalaran dalam memecahkan suatu masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tertentu. Sistem pakar telah memberikan nilai tambahan pada teknologi untuk membantu dalam menangani era informasi yang semakin canggih. Aplikasi sistem pakar ini menghasilkan *output* berupa program aplikasi yang dapat digunakan untuk mendiagnosa kemungkinan penyakit kulit pada hewan kucing berdasarkan gejala yang diinputkan oleh *user*. Sistem ini juga menampilkan besarnya kepercayaan gejala tersebut terhadap penyakit kulit yang diinputkan oleh *user*. Pengujian sistem menunjukkan bahwa sistem mampu melakukan diagnosa penyakit kucing berdasarkan gejala-gejala yang diderita pasien meskipun gejala-gejala tersebut mengandung ketidakpastian. Hasil diagnosa disertai nilai *Certainty Factor* yang menunjukkan tingkat kebenaran, keakuratan dari kemungkinan penyakit kulit pada kucing.
2. **Alexius Endy Budianto, S.Kom, MM (2015)**, dengan ISSN : 2476-9754 **Aplikasi Sistem Pakar Menggunakan Metode *Backward Chaining* Untuk**

Analisis Penyakit Hewan Ternak. Berisi sistem pakar umumnya sebagai sistem yang menyerap pengetahuan manusia ke komputer. Dimana sebuah komputer dapat menyelesaikan suatu permasalahan yang sebagaimana para ahli melakukannya. Penyakit adalah virus ataupun serangga yang mampu menyerang suatu makhluk hidup, salah satunya adalah hewan ternak sapi. Sehingga hewan sapi mengalami gangguan dan juga dapat mengalami kematian. Teknik pengumpulan data dalam penelitian adalah mengumpulkan data dalam penelitian adalah mengumpulkan data-data dan informasi melalui wawancara pada para peternak hewan sapi yang terdapat pada desa Jabung Malang, dan dari buku referensi. Hasil penelitian pengumpulan data ini disebut sistem pakar yang sebagai sarana media informasi yang mampu mendiagnosa penyakit yang terjadi pada hewan sapi dan dapat mempermudah para peternak atau *user* dalam mengenali suatu penyakit hewan sapi berdasarkan diagnosa.

3. **Made Bela Pramesthi Putri, Edy Santoso dan Marji (2017)** dengan ISSN 2548-964 dalam penelitian **Diagnosis Penyakit Kulit Pada Kucing Menggunakan Metode *Modified KNearest Neighbor*** peneliti mengatakan Kucing yang sering dijadikan hewan peliharaan oleh manusia ini tak luput dari banyaknya penyakit yang menyerang. Negara Indonesia merupakan negara yang memiliki kelembapan udara yang tinggi, maka jamur dan parasit dapat dengan mudah menyebar dan berkembangbiak hingga dapat menyebabkan penyakit kulit. Penyakit kulit merupakan penyakit yang umum diderita oleh hewan mamalia ini, jika tidak ditangani dengan cepat dan tepat maka penyakitnya dapat dengan cepat meluas hingga dapat mengganggu aktifitas kucing atau bahkan dapat menyebabkan kematian. Gejala awal dari penyakit kulit terkadang tidak

begitu terlihat dan tidak begitu mengganggu, oleh karena itu terkadang kucing pun terlihat baik-baik saja sehingga pemilik tidak begitu peduli. Sangat terbatasnya pengetahuan pemilik tentang penyakit kulit yang dialami oleh kucing, serta banyaknya kemiripan gejala-gejala dari berbagai penyakit kulit yang sulit diidentifikasi oleh orang awam menjadi alasan utama penulis untuk melakukan penelitian tentang diagnosis penyakit kulit pada kucing menggunakan metode *Modified K-Nearest Neighbor*. Metode *Modified K-Nearest Neighbor* digunakan untuk pengklasifikasian data baru yang kelasnya belum diketahui berdasarkan nilai k terdekat. Dataset yang digunakan dalam penelitian ini terdiri dari 240 data penyakit kucing dengan 14 parameter dan 5 jenis penyakit kulit yang berbeda, keluaran dari sistem ini berupa hasil diagnosis penyakit. Hasil akurasi tertinggi yang didapatkan berdasarkan pengujian yang telah dilakukan sebesar 100% pada nilai $k=1$ dan akurasi terendah sebesar 89.668%. Dari hasil akurasi tersebut, maka dapat disimpulkan bahwa metode *Modified K-Nearest Neighbor* dapat diimplementasi ke dalam sistem diagnosis penyakit kulit pada kucing.

4. **SAURKAR A.V. dan WATANE H.N. (2012)**, dengan ISSN : 2249-9466 dan E-ISSN : 2249-9474 **AN EXPERT SYSTEM FOR DISEASES DIAGNOSIS IN PET**. Mengatakan *This paper presents an architectural framework of an Expert System in the area of ANIMAL HUSBANDARY and describes the design and development of the rule based expert system, The designed system is intended for the diagnosis of common diseases occurring in the animals. An*

Expert System is a computer program normally composed of a Knowledge base, inference engine and user-interface. The proposed expert system facilitates different components including decision support module with interactive user interfaces for diagnosis on the basis of response(s) of the user made against the queries related to particular disease symptoms. The system integrates a structured knowledge base that contains knowledge about symptoms and remedies of diseases in the animals appearing during their life span. An image database is also integrated with the system for making the decision support more interactive. The pictures related to disease symptoms are stored in the picture database and the intelligent system module prompts these with the interface based on rule based decision making algorithms. The system has been tested with domain dataset, and results given by the system have been validated with domain experts.

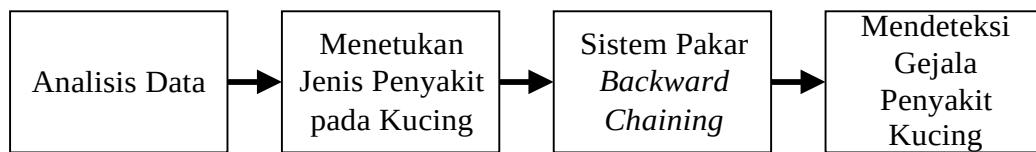
5. **Siska Iriani (2015)**, dengan ISSN 2580-5495 Penerapan Metode *Backward Chaining* pada Sistem Pakar Diagnosa Penyakit Tulang Manusia. Sistem Pakar Untuk Mendiagnosa Penyakit Tulang pada Manusia merupakan aplikasi yang berguna untuk mengetahui jenis penyakit pada tulang manusia, beserta gejala yang dialami pemakai. Pembahasan utama dalam sistem ini adalah perancangan dan pembuatan sistem pakar untuk melakukan diagnosa dan memberikan informasi – informasi mengenai penyakit tulang, gejala-gejala pada penyakit tersebut serta cara pencegahan, pengobatan dan penyebabnya. Model inferensi yang digunakan dalam pembuatan sistem pakar ini adalah penalaran mundur (*Backward Chaining*) sedangkan teknik pencarian menggunakan *Depth First*

Search. Penentuan diagnosa dalam sistem pakar ini dilakukan melalui proses konsultasi antara sistem dan pemakai. Jawaban disesuaikan dengan aturan yang berada di dalam sistem, jika jawaban yang dimasukkan sesuai dengan aturan yang berlaku, maka sistem ini akan memberikan hasil diagnosa berupa informasi penyakit. Diharapkan dengan dibuatnya Sistem Pakar Untuk Mendiagnosa Penyakit Tulang Pada Manusia ini dapat memberikan hasil diagnosa, penyebab, pengobatan, serta pencegahan terhadap suatu penyakit. Sistem ini disebut dengan Sistem Pakar (*Expert System*).

6. Indri Mansyur, Wawan Kurniawan (2017). **Sistem Pakar Mendiagnosa Penyakit Paru-Paru Pada Manusia Berbasis Web**. Berdasarkan penelitian yang dilakukan menjelaskan, sistem pakar adalah salah satu bagian dari kecerdasan buatan yang mengandung pengetahuan tertentu sehingga setiap orang dapat menggunakannya untuk memecahkan berbagai masalah yang bersifat spesifik dalam hal ini adalah permasalahan kesehatan paru-paru pada manusia. Metode penelusuran adalah *backward chaining* dengan teknik pencarian *breadth first search*. Berdasarkan gejala dan memberikan solusi jenis penyakit paru-paru.

2.5 Kerangka Pemikiran

Kerangka pemikiran memuat pemikiran terhadap alur yang dipahami sebagai acuan dalam memecahkan suatu masalah yang diteliti secara logis dan sistematis. Kerangka berfikir yang baik menjelaskan secara teoritis pertautan antara variabel yang diteliti (Sugiyono, 2014). Berikut ini kerangka pemikiran yang mendasari penelitian ini.



Gambar 2.17 Kerangka Pemikiran
Sumber: Data Penelitian 2018

Data-data yang dibutuhkan tentang diagnosa penyakit pada kucing dianalisis terlebih dahulu agar lebih mudah dalam melakukan pengolahan datanya. Dari data yang sudah diperoleh dan dianalisis kemudian diolah dengan metode sistem pakar *backward chaining*. Sistem pakara engan metode *backward chaining* dibuat menggunakan bahasa pemrograman *PHP* dan *database MySQL* sehingga menghasilkan sistem pakar untuk mendiagnosa penyakit kucing menggunakan metode *backward chaining* berbasis *web*.