

BAB II TINJAUAN PUSTAKA

2.1. Teori Dasar

Deskripsi teori dalam suatu penelitian merupakan uraian sistematis tentang teori (dan bukan sekedar pendapat pakar atau penulis buku) dan hasil-hasil penelitian yang relevan dengan variabel yang diteliti. Beberapa jumlah kelompok teori yang perlu dikemukakan/dideskripsikan, akan tergantung pada luasnya permasalahan dan secara teknis tergantung pada jumlah variabel yang diteliti (Sugiyono, 2012: 52) .

Pada bab ini akan dijelaskan tentang beberapa teori dasar antara lain kecerdasan buatan atau *Artificial Intelligence (AI)* dan beberapa subdisiplin ilmunya seperti logika *fuzzy (fuzzy logic)*, jaringan saraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*); *web*, basis data, dan validitas sistem.

2.1.1. Kecerdasan Buatan atau *Artificial Intelligence (AI)*

Kata *intelligence* berasal dari bahasa Latin *intelligo* yang berarti ‘saya paham’. Jadi, dasar dari *intelligence* adalah kemampuan memahami dan melakukan aksi. Sebenarnya, area Kecerdasan Buatan (*Artificial Intelligence*) atau

disingkat dengan AI, bermula dari kemunculan komputer sekitar tahun 1940-an, meskipun sejarah perkembangannya dapat dilacak hingga zaman Mesir kuno. Pada masa sekarang, perhatian difokuskan pada kemampuan komputer untuk mengerjakan sesuatu yang dapat dilakukan oleh manusia. Dalam hal ini, komputer tersebut dapat meniru kemampuan kecerdasan dan perilaku manusia (Budiharto.W, Derwin, S, 2014: 3).

Menurut Sutojo. T, Edy. M, (2011: 3) Program konvensional hanya dapat menyelesaikan persoalan yang diprogram secara spesifik. Jika ada informasi baru, sebuah program konvensional harus diubah untuk menyesuaikan diri dengan informasi baru tersebut. Hal ini tidak hanya menyebabkan boros waktu, namun juga dapat menyebabkan terjadinya eror. Sebaiknya, kecerdasan buatan memungkinkan komputer untuk berpikir atau menalar dan menirukan proses belajar manusia sehingga informasi baru dapat diserap sebagai pengetahuan, pengalaman, dan proses pembelajaran serta dapat digunakan sebagai acuan dimasa-masa yang akan datang.

2.1.1.1. Logika *Fuzzy* (*Fuzzy Logic*)

Logika *fuzzy* adalah sebagai suatu jenis logic yang bernilai ganda dan berhubungan dengan ketidakpastian dan kebenaran parsial, seperti *propositional logic*, objek dasar dari suatu logic adalah proposition atau pernyataan yang menyatakan suatu fakta (Suyanto, 2014:103).

Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang sesuai untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan komputer, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol. Dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1, artinya suatu keadaan memungkinkan mempunyai dua nilai “Ya” dan “Tidak” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan di berbagai bidang seperti pada sistem diagnosis penyakit (dalam bidang kedokteran); pemodelan sistem pemasaran, sistem operasi (dalam bidang ekonomi); kendali kualitas air, prediksi adanya gempa bumi, klasifikasi dan pencocokan pola (Sutojo.T, Edy. M, dkk, 2011: 211-212).

Ada beberapa keuntungan yang dapat diambil ketika menggunakan logika *fuzzy* untuk memecahkan suatu masalah yaitu :

1. Perancangannya tidak memerlukan persamaan matematik yang rumit.
2. Mudah dimengerti.
3. Memiliki toleransi terhadap data-data yang tidak tepat.
4. Mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Dapat bekerja sama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

Sistem inferensi *fuzzy* adalah cara memetakan ruang *input* menuju ruang *output* menggunakan logika *fuzzy*. Empat elemen dasar sistem inferensi *fuzzy* antara lain (Sutojo.T, Edy. M, dkk,2011: 232):

1. Basis pengetahuan *fuzzy*, yaitu kumpulan aturan (*rule*) *fuzzy* dalam bentuk pernyataan *IF...THEN*.
2. Fuzzifikasi, yaitu proses untuk mengubah *input* sistem yang mempunyai nilai tegas menjadi variabel linguistik menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.
3. Mesin inferensi, yaitu proses untuk mengubah *inputfuzzy* menjadi *outputfuzzy* dengan cara mengikuti aturan-aturan yang telah ditetapkan pada basis pengetahuan *fuzzy*.
4. Defuzzifikasi, yaitu mengubah *outputfuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzifikasi.

Beberapa metode yang digunakan dalam sistem inferensi *fuzzy* adalah (Sutojo.T, Edy. M, dkk,2011: 233-237):

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan sebagai berikut:

- a) Fuzzifikasi
- b) Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c) Mesin inferensi menggunakan fungsi implikasi *MIN* (*Minimum*)
- d) Defuzzifikasi menggunakan metode Rata-rata (*Average*)

2. Metode Mamdani

Metode ini sering digunakan karena strukturnya yang sederhana. Pada metode ini, untuk mendapatkan *output* diperlukan 4 tahapan sebagai berikut:

- a) Fuzzifikasi
- b) Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c) Aplikasi fungsi implikasi menggunakan fungsi *MIN* (*Minimum*) dan komposisi antar-*rule* menggunakan fungsi *MAX*(*Maximum*) dengan menghasilkan himpunan *fuzzy* baru
- d) Defuzzifikasi menggunakan metode *Centroid* (Titik Tengah)

3. Metode Sugeno

Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Dalam metode ini, *output* sistem berupa konstanta atau persamaan linier. Dalam inferensinya, metode Sugeno menggunakan tahapan sebagai berikut:

- a) Fuzzifikasi
- b) Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c) Mesin inferensi menggunakan fungsi implikasi *MIN* (*Minimum*)
- d) Defuzzifikasi menggunakan metode Rata-rata (*Average*)

2.1.1.2 Jaringan Saraf Tiruan (*Artificial Neural Network*)

Jaringan saraf tiruan merupakan merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak

manusia dalam mengorganisasikan sel-sel penyusunnya yang disebut neuron, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektifitas yang sangat tinggi (Suyanto, 2014:169). Beberapa kelebihan yang dimiliki jaringan saraf tiruan antara lain:

1. Belajar adaptif, yaitu kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. *Self-Organization*, yaitu kemampuan membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real Time Operation*, yaitu perhitungan jaringan saraf tiruan yang dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Selain mempunyai beberapa kelebihan, jaringan saraf tiruan juga mempunyai kelemahan-kelemahan:

1. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatika, operasi logika, dan simbolis.
3. Membutuhkan pelatihan untuk dapat beroperasi sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.

Salah satu elemen yang menentukan baik tidaknya suatu mode jaringan saraf tiruan adalah hubungan antar-*neuron* atau arsitektur jaringan. *Neuron-neuron*

tersebut terkumpul dalam lapisan-lapisan yang disebut *neuron layers*. Terdapat 3 bagian lapisan penyusun jaringan saraf tiruan, yaitu (Kusumadewi Sri, 2009: 292):

1. Lapisan *Input (Input Layer)*

Unit-unit dalam lapisan ini disebut unit-unit *input* yang bertugas menerima pola *input*-an dari luar yang menggambarkan suatu permasalahan.

2. Lapisan Tersembunyi (*Hidden Layer*)

Unit-unit dalam lapisan ini disebut unit-unit tersembunyi, yang mana nilai *output*-nya tidak dapat diamati secara langsung.

3. Lapisan *Output (Output Layer)*

Unit-unit dalam lapisan ini disebut unit-unit *output*, yang merupakan solusi jaringan saraf tiruan terhadap suatu permasalahan.

Beberapa arsitektur jaringan yang sering digunakan dalam jaringan saraf tiruan antara lain (Suyanto, 2014: 171):

1. Jaringan Lapisan Tunggal

Jaringan ini terdiri dari 1 lapisan *input* dan 1 lapisan *output*, yang mana setiap unit dalam lapisan *input* selalu terhubung dengan setiap unit yang terdapat pada lapisan *output*. Jaringan ini menerima *input* kemudian mengolahnya menjadi *output* tanpa melewati lapisan tersembunyi. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *ADALINE*, *Hopfield*, dan *Perceptron*.

2. Jaringan Lapisan Banyak

Jaringan ini mempunyai 3 jenis lapisan, yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Jaringan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan jaringan lapisan tunggal. Contoh

jaringan saraf tiruan yang menggunakan jaringan ini adalah *MADALINE*, *backpropagation*, dan *Neocognitron*.

3. Jaringan dengan Lapisan Kompetitif

Jaringan ini memiliki bobot yang telah ditentukan dan tidak memiliki proses pelatihan. Jaringan ini digunakan untuk mengetahui *neuron* pemenang dari sejumlah *neuron* yang ada sehingga sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *Learning Vector Quantization (LVQ)*.

Berdasarkan cara memodifikasi bobotnya, pelatihan jaringan saraf tiruan dibagi menjadi dua, yaitu (Sutojo, 2011: 301- 392):

1. Pelatihan dengan Supervisi (pembimbing)

Dalam pelatihan ini, jaringan dipandu oleh sejumlah pasangan data (masukan dan target) yang berfungsi sebagai pembimbing untuk melatih jaringan hingga diperoleh bobot yang terbaik. Algoritma yang termasuk dalam pelatihan dengan supervisi antara lain:

a. *Hebb-Rule*

Model ini diperkenalkan oleh D.O. Hebb yang menggunakan cara menghitung bobot dan bias secara iteratif dengan memanfaatkan model pembelajaran dengan supervisi sehingga bobot dan bias dapat dihitung secara otomatis tanpa harus melakukan cara coba-coba. Arsitektur jaringan ini terdiri dari beberapa unit *input* dihubungkan langsung dengan sebuah unit *output*, ditambah dengan sebuah bias.

b. *Perceptron*

Model ini ditemukan oleh Rosenblatt (1962) dan Minsky – Papert (1969). Model jaringan ini merupakan model yang terbaik pada saat itu. Algoritma pelatihan *perceptron* digunakan baik untuk *input* biner maupun bipolar, dengan tertentu.

c. *Delta-Rule*

Selama pelatihan pola, *Delta-Rule* akan mengubah bobot dengan cara meminimalkan *error* antara *output* jaringan dengan target.

d. *Backpropagation*

Backpropagation adalah metode penurunan gradien untuk meminimalkan kuadrat *error* keluaran. Pelatihan jaringan ini terdiri dari 3 tahap, 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*.

e. *Heteroassociative Memory*

Jaringan saraf *heteroassociative memory* adalah jaringan yang dapat menyimpan kumpulan pengelompokan pola dengan cara menentukan bobot-bobotnya sedemikian rupa. Algoritma pelatihan yang biasa digunakan adalah *Hebb-Rule*.

f. *Bidirectional Associative Memory (BAM)*

Bidirectional Associative Memory (BAM) adalah model jaringan saraf yang memiliki 2 lapisan, yaitu lapisan *input* dan lapisan *output* yang mempunyai hubungan timbal balik antara keduanya (bersifat *bidirectional*). Arsitektur jaringan ini terdiri dari 3 *neuron* pada lapisan *input* dan 2 *neuron* pada lapisan

output. Model jaringan ini terbagi menjadi 2 jenis yaitu *BAM* Diskrit dan *BAM* Kontinu.

g. *Learning Vector Quantization (LVQ)*

Learning Vector Quantization (LVQ) adalah suatu model pelatihan pada lapisan kompetitif terawasi yang akan belajar secara otomatis untuk mengklasifikasikan vektor-vektor *input* ke dalam kelas-kelas tertentu.

2. Pelatihan tanpa Supervisi

Dalam pelatihan ini, tidak ada pembimbing yang digunakan untuk memandu proses pelatihan. Jaringan hanya diberi *input* tetapi tidak mendapatkan target yang diinginkan sehingga modifikasi bobot pada jaringan dilakukan menurut parameter tertentu. Model jaringan yang termasuk dalam pelatihan tanpa supervisi adalah jaringan kohonen yang diperkenalkan oleh Prof. Teuvo Kohonen pada tahun 1982.

Pada jaringan kohonen, *neuron-neuron* pada suatu lapisan data akan menyusun dirinya sendiri berdasarkan *input* nilai tertentu dalam suatu *cluster*. *Cluster* yang dipilih sebagai pemenang adalah *cluster* yang mempunyai vektor bobot paling cocok dengan pola *input*, yaitu *cluster* yang memiliki jarak yang paling dekat.

2.1.1.3 Sistem Pakar (*Expert System*)

Sistem pakar mulai dikembangkan pada pertengahan 1960, ditandai dengan lahirnya sistem pakar pertama bernama *General-purpose Problem Solver (GPS)*

yang dikembangkan oleh Newel dan Simon. Kemudian bermunculan sistem pakar lain di berbagai bidang seperti *MYCIN* untuk diagnosis penyakit, *DENDRAL* untuk mengidentifikasi struktur molekul 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 memberikan keputusan bagi seorang manajer dalam masalah stok dan investasi, *DELTA* dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya (Suyanto, 2014: 1).

Menurut Kusumadewi Sri (2009: 109) sistem pakar merupakan salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia. Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Dengan bantuan sistem pakar, seseorang yang bukan pakar dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar.

Menurut Sutojo (2011:14-15) sistem pakar dapat digunakan oleh orang awam yang bukan pakar untuk meningkatkan kemampuan dalam memecahkan masalah. Sistem pakar juga dapat digunakan oleh pakar sebagai *assistant* yang berpengetahuan, serta digunakan untuk memperbanyak atau menyebarkan sumber pengetahuan yang semakin langka.

Suatu sistem dikatakan sebagai sistem pakar jika memiliki ciri-ciri sebagai berikut (Kusumadewi Sri, 2009: 122):

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi diletakkan terpisah.
7. Keluarannya (*output*) bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara terpisah secara searah, sesuai dengan dialog dengan pengguna.

Representasi pengetahuan merupakan metode yang digunakan untuk mengodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Hal ini dimaksudkan untuk menangkap sifat-sifat penting dari suatu masalah sehingga informasi itu dapat diakses oleh prosedur pemecahan masalah. Bahasa representasi harus dirancang agar fakta-fakta dan pengetahuan lain yang terkandung di dalamnya dapat digunakan untuk penalaran.

Menurut Suyanto (2014: 95) representasi pengetahuan dimaksudkan untuk mengorganisasikan pengetahuan dalam bentuk dan format tertentu agar dapat dimengerti oleh komputer. Pemilihan representasi pengetahuan yang tepat akan menghasilkan sebuah sistem pakar yang efektif. Salah satu model representasi pengetahuan yang penting yaitu kaidah produksi (*production rule*).

Sistem pakar pada penelitian ini menggunakan model representasi pengetahuan berbasis kaidah produksi. Menurut Firebaugh (1988) dalam

(Kusumadewi Sri, 2009: 113-114) struktur sistem pakar yang berbasis kaidah produksi terdiri dari 4 komponen, yaitu:

1. Antarmuka pemakai

Antarmuka merupakan penghubung antara pemakai dengan sistem pakar. Komponen ini berfungsi sebagai alat komunikasi antara sistem dan pengguna (*user*) yang penting sekali bagi pengguna. Komponen ini harus didesain sedemikian rupa sehingga efektif dan mudah digunakan terutama bagi pengguna yang tidak ahli dalam bidang yang diterapkan pada sistem pakar.

2. Basis pengetahuan

Basis pengetahuan adalah komponen yang berisi sekumpulan kaidah yang berasal dari pengetahuan dalam domain tertentu dan secara umum disajikan dalam bentuk kaidah produksi (*IF...THEN...*). Pengetahuan pakar yang disajikan dalam format tertentu didapat dari sekumpulan pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti buku-buku, jurnal ilmiah, majalah, maupun dokumentasi tercetak lainnya. Basis pengetahuan diletakkan terpisah dari mesin inferensi agar pengembangan pengetahuan sistem pakar dapat dilakukan secara leluasa tanpa mengganggu mesin inferensi.

3. Struktur kontrol (Mesin Inferensi)

Struktur kontrol merupakan *interpreter* kaidah atau mesin inferensi yang menggunakan pengetahuan-pengetahuan yang tersimpan dalam basis pengetahuan untuk memecahkan atau menyelesaikan permasalahan yang ada. Inferensi merupakan proses untuk menghasilkan informasi berupa konklusi logis berdasarkan informasi yang tersedia atau fakta yang diketahui.

Dalam melakukan proses inferensi, sistem pakar memerlukan adanya proses pengujian kaidah-kaidah dalam urutan tertentu untuk mencari suatu kondisi yang sesuai dengan kondisi awal atau untuk memastikan kondisi yang sedang berjalan sudah dimasukkan ke dalam *database*. Proses pengujian itu disebut dengan perunutan atau penalaran, yaitu proses pencocokan fakta atau kondisi tertentu yang tersimpan dalam basis pengetahuan maupun pada memori kerja dengan kondisi yang dinyatakan dalam premis atau bagian kondisi pada suatu kaidah atau aturan (Kusumadewi Sri, 2009: 14).

Ada beberapa konsep penalaran yang dapat digunakan oleh mesin inferensi yaitu:

a. Penalaran maju (*forward chaining*)

Konsep ini dapat juga disebut sebagai pencarian yang dimotori data (*data driven search*). Runut maju melakukan proses perunutan (penalaran) dimulai dari premis-premis atau informasi masukan (*IF*) terlebih dahulu kemudian menuju konklusi atau *derived information (THEN)*. Konsep ini dapat dimodelkan sebagai berikut:

IF (informasi masukan) *THEN* (konklusi)

Informasi masukan dapat berupa suatu pengamatan sedangkan konklusi dapat berupa diagnosa sehingga dapat dikatakan jalannya penalaran runut maju dimulai dari pengamatan menuju diagnosa. Pada metode ini, sistem tidak melakukan praduga apapun terhadap konklusi, namun sistem akan menerima semua gejala yang diberikan pengguna lalu sistem akan memeriksa gejala-gejala

tersebut dan selanjutnya mencocokkan dengan konklusi yang sesuai (Kusumadewi Sri, 2009: 19).

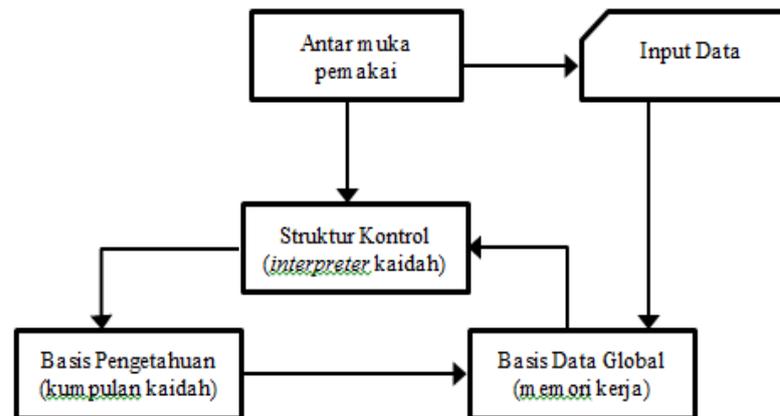
b. Penalaran mundur (*backward chaining*)

Secara umum konsep ini diaplikasikan ketika tujuan ditentukan sebagai kondisi atau keadaan awal. Konsep ini disebut juga *goal-driven search*. Arah penalaran atau peruntukan 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 (tujuan) terlebih dahulu sebagai praduga awal, kemudian memeriksa dan memastikan gejala-gejala (kondisi) telah terpenuhi dan selanjutnya mengeluarkan konklusi sebagai *output*. Jika sistem menemukan ada bagian kondisi yang tidak terpenuhi maka sistem akan memeriksa konklusi (tujuan) pada aturan atau kaidah berikutnya (Kusumadewi Sri, 2009: 21-22).

2. *Working memory* (memori kerja) atau basis data global

Berfungsi untuk mencatat status masalah yang terjadi dan *history* solusi. Memori kerja merupakan bagian yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi saat proses konsultasi terjadi (Kusumadewi. Sri, 2009: 19).



Gambar 2.1 Struktur Sistem Pakar Kaidah Produksi

Kusrini (2008: 33) menjelaskan bahwa 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 (Adedeji, 1992 dalam Kusumadewi. Sri, 2009):

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 konklusi 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 konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan diagnosa (Kusumadewi. Sri, 2008: 25-26).

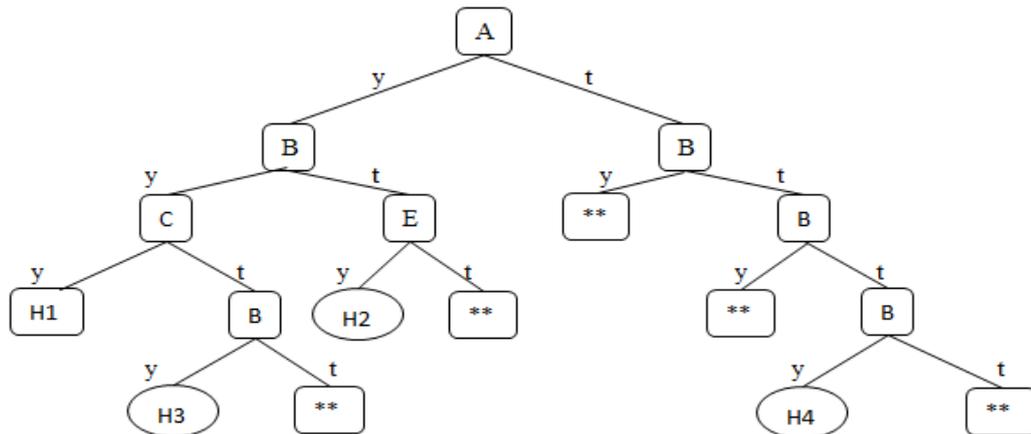
Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk tabel keputusan kemudian dibuat pohon keputusannya. Berikut ini adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan (Kusumadewi. Sri, 2009: 18).

Tabel 2. 1 Tabel Keputusan

Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<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: Kusumadewi. Sri, 2009: 19)



Gambar 2.2 Pohon Keputusan

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.

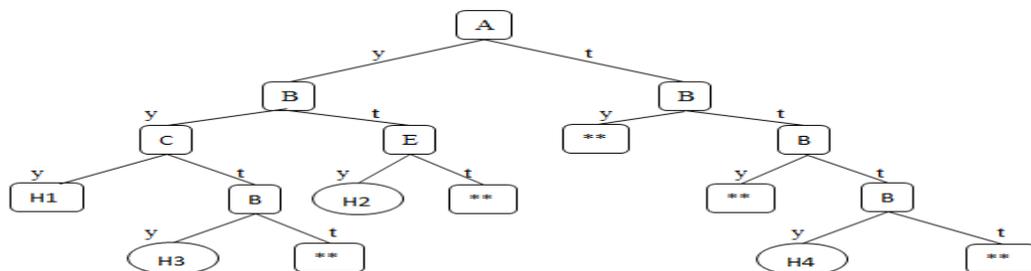
Dalam sesi konsultasi pada sistem pakar, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.2 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* 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 <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence</i> A	Ya	Ya	ya	Tidak
<i>Evidence</i> D	Tidak	Tidak	tidak	Ya
<i>Evidence</i> B	Ya	Tidak	ya	Ya
<i>Evidence</i> C	Ya	Tidak	tidak	Ya
<i>Evidence</i> E	Tidak	Ya	ya	tidak

(Sumber: Kusumadewi. Sri, 2009:22)

Berdasarkan tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



Gambar 2.3 Alternatif Pohon Keputusan

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.

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 produksi banyak digunakan pada aplikasi sistem pakar karena model representasi ini mudah dipahami dan bersifat deklaratif sesuai dengan jalan pikiran manusia dalam menyelesaikan suatu masalah, dan mudah diinterpretasikan.

Menurut Kusumadewi. Sri, (2009: 109) terdapat beberapa alasan mengapa sistem pakar dikembangkan untuk menggantikan seorang pakar:

1. Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.

4. Menghadirkan/menggunakan jasa seorang pakar memerlukan biaya yang mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat (*hostile environment*).

Adapun kelebihan yang dimiliki sistem pakar antara lain (Kusrini, 2006: 15):

1. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
3. Meningkatkan *output* dan produktifitas, bekerja lebih cepat dari manusia sehingga mengurangi jumlah pekerja yang dibutuhkan dan akan mereduksi biaya.
4. Meningkatkan kualitas.
5. Sistem pakar menyediakan nasihat yang konsisten dan dapat mengurangi tingkat kesalahan.
6. Membuat peralatan yang kompleks lebih mudah dioperasikan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
7. Handal (*realibility*).
8. Sistem pakar tidak dapat lelah atau bosan serta konsisten dalam memberikan jawaban dan selalu memberikan perhatian penuh.
9. Memiliki kemampuan untuk memecahkan masalah yang kompleks.
10. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dipakai dimana saja. Sistem pakar merupakan arsip yang terpercaya dari sebuah keahlian

sehingga *user* seolah-olah berkonsultasi langsung dengan sang pakar meskipun sang pakar sudah pensiun.

Selain memiliki beberapa kelebihan yang dapat dimanfaatkan, sistem pakar juga memiliki beberapa kekurangan, yaitu (Sutojo, 2011: 161):

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

2.1.2 Web

Menurut Raharjo (2011: 67) *web* adalah suatu layanan di dalam jaringan 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 *web browser*. Web telah menjadi (*interface*) standar untuk layanan-layanan lain yang ada di internet, misalnya *e-mail*. Dengan menggunakan teknologi *web*, *user* akan lebih mudah dalam berinteraksi dengan data yang tersimpan di dalam suatu *web server*, tanpa harus menuliskan perintah apa pun. Permintaan dokumen atau data dari sumber *web server* dapat dilakukan hanya dengan mengikuti (klik) *link* yang disediakan di dalam dokumen.

2.1.3 Database (basis data)

Menurut Rosa A & Shalahuddin (2015: 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 berikut (A.S & Shalahuddin, 2011: 45):

1. Menyediakan fasilitas untuk mengelola akses data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data yang dilakukan.

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 antara lain (Kusumadewi Sri, 2009: 275-276):

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. Variabel Penelitian

Variabel penelitian merupakan segala sesuatu yang berbentuk apa saja yang dimiliki orang, obyek atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya (Sugiyono, 2014: 38). Pada penelitian ini yang menjadi variabel adalah penyakit udang vannamei, penyakit udang vannamei ini juga ada beberapa jenis yaitu: *Infectious myo necrotic virus* (IMNV), dan *white spot syndrome virus* (WSSV).



Gambar 2.4. Udang Vannamei

2.2.1 *Infectious myo necrotic virus (IMNV)*

Penyakit IMNV sering disebut oleh para petambak dengan nama myo. Penyakit ini disebabkan oleh invaksi virus myo dengan material genetik berupa RNA yang berukuran kecil, kurang dari 40 nm. Penyakit ini ditandai dengan timbulnya otot putih, ekor berwarna kemerahan, dan terjadinya pebesaran pada limfoid organ udang.

Penyakit IMNV bersipat akut dengan tingkat kematian berkisar antara 40-70% dan akan meningkat secara bertahap dan terus menerus. Kematian terjadi ketika terjadi ketika udang menampilkan gejala klinis yang parah, yaitu ekor kemerahan atau tubuh udang telah memutih.

Proses infeksi virus myo sampai timbulnya penyakit myo pada udang vannamei dimulai dari terinduksinya virus tersebut ke dalam perairan tambak. laju perkembangan penyakit pada udang yang terinfeksi dipengaruhi oleh memburuknya kualitas air tambak. Makin buruk kualitas air tanbak, makin

meningkat laju perkembangan penyakit. Kualitas air yang buruk akan memicu penurunan sistem kekebalan tubuh sehingga mengakibatkan terjadinya propagasi dan penyebaran virus di dalam lingkungan tambak (Erick Erlangga, 2012:108).



Gambar 2.5: IMNV (*Infectious myo necrotic virus*)

2.2.2 *White spot syndrome virus* (WSSV)

Sejak ditemukan pada tahun 1992 di Taiwan penyakit ini telah menjadi ancaman yang serius terhadap budi daya udang vannamei. WSSV merupakan penyakit yang di sebabkan oleh virus dari golongan *nimaviridae* dan memiliki bentuk basis. Virus ini memiliki tingkat virulensi yang tinggi dan menyebabkan kematian massal dalam jangka 2-3 hari setelah udang vannamei terindetivikasi terserang penyakit. Udang yang terkena penyakit ini biasanya menunjukkan tanda adanya bercak putih pada bagian karapas. Biasanya tanda tersebut akan terlihat jelas ketika udang telah mencapai bobot 3-5 gram, sedangkan dibawah itu akan sulit untuk mentukan apakah udang terserang penyakit atau tidak.

Beberapa kasus penyakit bercak putih yang terjadi pada udang vannamei umumnya akan menyerang beberapa organ tubuh yang sangat vital. Organ-organ target yang sering diserang oleh virus ini di antaranya, sel-sel insang, hepatopankreas, dan usus. Adanya bercak putih pada kerapas dan sefalotoraks disebabkan oleh terjadinya abnormalitas pada penyimpanan garam kalsium dalam tubuh udang (Erick Erlangga, 2012:102)



Gambar 2.6 WSSV (white spot syndrome virus)

2.3 Software Pendukung

2.3.1 Xampp



Gambar 2.7 Logo *Xampp*

Menurut Aditya, (2011: 16) XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MYSQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

2.3.2 php Myadmin



Gambar 2.8 Logo *phpMyAdmin*

phpmyadmin adalah perangkat lunak gratis yang ditulis dalam bahasa pemrograman *PHP* bertujuan untuk menangani administrasi *MySQL* melalui *web*. *phpMyAdmin* mendukung berbagai operasi pada *MySQL* dan *MariaDB*. Operasi-operasi yang sering digunakan seperti mengelola *database*, tabel, kolom, relasi, indeks, *users*, *permissions*, dan lain-lain, dapat dilakukan melalui antarmuka pengguna dengan tetap dapat mengeksekusi pernyataan *SQL* secara langsung (Abdulloh, Rohi, 2015: 4).

2.3.3 Hypertext Preprocessor (PHP)



Gambar 2.9 Logo PHP

Menurut Aditya (2011: 1–2) *Hypertext Preprocessor* (PHP) adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memprogram situs web dinamis. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreter* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Pada juni 2004, Zend merilis PHP 5.0 dalam versi ini, inti dari *interpreter* PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi obyek kedalam PHP untuk menjawab perkembangan

bahasa pemrograman ke arah paradigama berorientasi obyek. Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari *apace*, *IIS*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relative mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dn *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.3.4 HTML (*Hyper Text Markup Language*)



Gambar 2.10 Logo HTML

Menurut Saputra (2012: 11–18) HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML biasa disebut bahasa paling dasar dan penting yang digunakan uuntuk menampilkan dan mengelola tampilan pada halaman website. Menurut sumber yang penulis kutip dari Wikipedia, HTML digunakan

untuk menampilkan berbagai informasi didalam sebuah penjelajah web internet dan formatting hypertext sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintergritas.

HTML berawal pada tahun 1980 ketika IBM berniat untuk membuat suatu bahasa kode untuk menggabungkan teks dengan memformat agar mengenali elemen dokumen. Bahasa yang menggunakan tanda-tanda ini dinamakan *Markup Language*. Namun pihak IBM memberi nama *Generalize Markup Language*. Pada tahun 1986, ISO mengeluarkan standarisasi bahasa *markup* berdasarkan GML dengan nama *standard Generalize Markup Language* (SGML). Pada tahun 1989, Caillau Tim bekerja sama dengan Bannes Lee Robert, ketika bekerja di CERN mencoba untuk mengembangkan SGML. Dari tangan merekalah lahir HTML (*Hyper Text Markup Language*) yang kini digunakan untuk membuat webside. HTML ini pertama kali dipopulerkan oleh Browser Mosaic. Sejak tahun 1990, bahasa ini mengalami perkembangan yang cukup pesat. Apalagi sejak tahun 1996, *World Wide Web Consortium* (W3C) turut mengembangkan html dan mengeluarkan versi 3.2 sejak itu html digunakan sebagai bahasa standar internet yang kini dikendalikan oleh W3C.

HTML versi 5 yang paling marak dibincangkan didunia maya, HTML 5 layaknya sebuah html biasa yang sering kita gunakan 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 pembuatannya. 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 ditampilkan, 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.

2.3.5 CSS (*Cascading Style sheet*)



Gambar 2.11 Logo CSS

Menurut Saputra (2012: 27–29) *Cascading Style Sheet (CSS)* merupakan bahasa pemrograman web yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web lebih rapih, terstruktur, dan seragam. CSS merupakan salah satu program wajib disamping html yang harus dikuasai oleh para setiap program web, terlebih lagi itu adalah Web Designer. Tujuan utama CSS adalah untuk memisahkan konten utama dengan tampilan dokumen lainnya (html dan sejenisnya). Dengan adanya pemisah ini, akses konten pada web meningkat. Web yang menggunakan CSS akan lebih ringan dan mudah untuk dibuka dibandingkan dengan web yang tidak menggunakan CSS. Perbedaan ini akan semakin terasa ketika web yang anda buka mempunyai data yang banyak.

Tujuan lainnya adalah untuk mempercepat pembuatan halaman web. Hanya perlu membuat satu property dan property tersebut dapat digunakan pada halaman lainnya, tidak perlu menulis ulang kode program yang digunakan berulang kali. CSS ini dikembangkan oleh *World Wide Web Consortium* atau yang biasa lebih dikenal dengan istilah W3C. Sehingga CSS menjadi bahasa standard dalam pembuatan web. CSS bukan menggantikan kode html, tetapi hanya

difungsikan sebagai penopang atau pendukung (pelengkap) dari file html yang berperan dalam penataan kerangka dan layout.

Dengan menggunakan CSS, akan banyak keuntungan yang dapat kita peroleh, diantaranya:

- a) Memisahkan pembuatan dokumen (CSS dan HTML).
- b) Mempermudah dan mempersingkat pembuatan dan pemeliharaan dokumen web.
- c) Akses web lebih pat saat di-loading (mempercepat pembacaan HTML).
- d) Flaksibel, interaktif, tampilan lebih menarik dan nyaman dipandang.
- e) Lebih kecil ukuran file sehingga bandwidth yang digunakan juga otomatis lebih kecil.
- f) Dapat digunakan pada semua browser.

2.3.6 JavaScript



Gambar 2.12 Logo *JavaScript*

Menurut Raharjo (2011: 221) *JavaScript* adalah bahasa yang berfungsi untuk membuat skrip-skrip program yang dapat dikenal dan dieksekusi oleh *web browser* dengan tujuan untuk menjadikan halaman *web* lebih bersifat interaktif. Meskipun banyak fitur dari bahasa dan java yang diadopsi oleh JavaScript, namun

javascript dikembangkan secara terpisah dan independen. Javascript dan java merupakan dua hal yang konsepnya sangat berbeda, meskipun ada kemiripan dalam penulisan sintaksnya. Javascript dikembangkan oleh Netscape dan merupakan bahasa yang bersifat terbuka (*open*) sehingga setiap orang dapat menggunakannya tanpa harus membeli lisensi.

2.3.7 MySQL dan SQL



Gambar 2.13 Logo *MySQL*

Menurut A.S & Shalahuddin, (2011: 47–49) SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. SQL berkembang pada tahun 1970-an. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standards for Institute*) dan disebut sebagai SQL-86. Pada perkembangannya, SQL beberapa kali dilakukan revisi. Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

Pada sebuah tabel terdapat field yang berisi nilai dari data. Nilai data dalam field mempunyai tipe sendiri. Ada beberapa tipe data yang dapat digunakan MySQL antara lain (Utomo, Eko Priyo 2014: 61–63) :

1. Tipe Data Numerik dibedakan menjadi dua kelompok, yaitu *integer* dan *floating point*. *Integer* digunakan untuk data bilangan bulat, sedangkan *floating point* digunakan untuk bilangan desimal.
2. Tipe Data String merupakan rangkain karakter.
3. Tipe Data Char dan Varchar untuk tipe data char dan varchar sebenarnya sama, perbedaannya terletak pada memori terletak pada jumlah memori yang dibutuhkan untuk penyimpanan.
4. Tipe Data Tanggal ada beberapa tipe dapat anda gunakan, antara lain DATETIME, DATA < TIME, TIMESTRAMP, atau YEAR.

2.3.8 Adobe Dreamweaver CS6



Gambar 2.14 Logo *Adobe Dreamweaver CS6*

Menurut Madcoms, (2015: 23), *Adobe Dreamweaver CS6* adalah perangkat lunak terkemuka untuk desain *web* yang menyediakan kemampuan visual yang intuitif termasuk dalam tingkat kode, yang dapat digunakan untuk

dapat membuat dan mengedit *website HTML*, serta aplikasi *mobile* seperti *smartphone*, *tablet*, dan perangkat lunak lainnya.

Dengan adanya fitur *layout Fluid Grid* yang dirancang khusus untuk memungkinkan lintas platform, maka akan membuat *layout* adaptif atau dapat menyesuaikan dengan *browser* yang di pakai. *Dreamweaver* menjadi *web* desain standard dan alat pengembangan untuk banyak organisasi dan *Dreamweaver CS6* dibutuhkan untuk merespon dengan cara *web* Fitur-fitur dalam *Dreamweaver CS6* antara lain: *Layout fluid grid*, peningkatan kinerja *FTP* karena didukung oleh *FTPS* dan *FTPeS*, integrasi dengan *Adobe Business Catalyst*, peningkatan *support jQuery Mobile*, *support PhoneGap* diperbarui, transisi *CSS3* dan *HTML5*, fitur *Live View* diperbarui, panel *Multiscreen Preview* diperbarui, integrasi *Adobe BrowserLab*, mendukung integrasi *CMS*, isyarat kode yang lebih spesifik, integrasi *Adobe Creative Suite*, komunitas *Dreamweaver* yang semakin diperluas, didukung oleh teknologi terkemuka termasuk *HTML*, *XHTML*, *CSS*, *JavaScript*, *Ajax*, *PHP*, perangkat lunak *Adobe ColdFusion*, dan *ASP*; selalu menjadi yang terdepan dengan *World Wide Web Consortium (W3C)* validasi, mendukung *Subversion*, inspeksi atau pengecekan *CSS* dengan dukungan *CSS* yang komprehensif, bantuan pengkodean yang cerdas dengan isyarat kode *custom class PHP*, *setup* situs yang sederhana, *CSS stater pages*, dan terintegrasi dengan konten *FLV*.

2.3.9 StarUML



Gambar 2.15 Logo *StarUML*

StarUML (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 obyek. *StarUML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa A & Shalahuddin, 2015: 117–136).

Terdapat 13 macam diagram yang di kelompokkan dalam 3 kategori, berikut ini penjelasan singkat dari kategori tersebut:

1. *Structure diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Diagram *UML* yang termasuk dalam kategori ini antara lain *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.

2. *Behaviour diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Diagram *UML* yang termasuk dalam kategori ini antara lain *use case diagram*, *activity diagram*, dan *state machine diagram*.

3. Interaction diagrams

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Diagram *UML* yang termasuk dalam kategori ini antara lain *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction overview diagram*.

Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem yaitu:

1. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterikatan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Berikut ini adalah simbol-simbol yang digunakan dalam *class diagram*:

Tabel 2. 3 Simbol *class diagram*

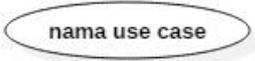
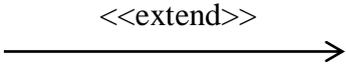
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/association 	Relasi antar kelas dengan makna umum
Asosiasi berarah/directed association 	Relasi antar kelas dengan makna yang satu digunakan oleh kelas yang lain
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan/dependency 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/aggregation 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

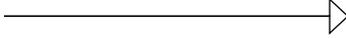
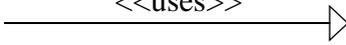
Sumber : (A.S & Shalahuddin, 2011)

2. Use case diagram

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut ini adalah simbol-simbol yang digunakan dalam *use case diagram*

Tabel 2. 4 Simbol Use Case Diagram

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>
<p>asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>
<p>Ekstensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu. Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>

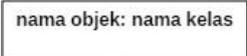
<p>generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya</p>
<p>Menggunakan/<i>include/uses</i></p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya <i>use case</i> ini. Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>

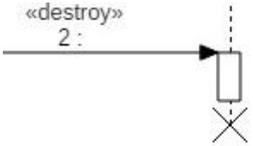
Sumber : (A.S & Shalahuddin, 2011)

3. Sequence Diagram

diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Dibawah ini akan menjelaskan simbol-simbol *Sequence Diagram* yaitu:

Tabel 2. 5 Simbol Sequence Diagram

Simbol	Deskripsi
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>

<p>pesan tipe <i>call</i></p> <p>1 : nama_metode() →</p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.</p>
<p>Pesan tipe <i>send</i></p> <p>1 : masukan →</p>	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju</p>
<p>pesan tipe <i>return</i></p> <p>----- 1 : keluaran →</p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu. Arah panah mengarah pada objek penerima</p>
<p>Pesan tipe <i>destroy</i></p> <p>«destroy» 2 : →</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri</p>

Sumber :(A.S & Shalahuddin, 2011)

2.4 Penelitian Terdahulu

Untuk mendukung teori yang berkaitan dengan penelitian, peneliti mencantumkan beberapa penelitian terdahulu di bidang sistem pakar.

Windah Supartini, Hindarto. kinetik, vol.1, no.3, 2016, ISSN : 2503-2259. Sistem pakar berbasis web dengan metode forward chaining dalam mendiagnosis dini penyakit tuberkulosis di jawatimur. Dalam penelitian ini, metode yang dilakukan, adalah memeriksa berkas pasien Tuberkulosis dan melakukan wawancara di sub recipient TB Care Aisyiyah Jawa Timur yang beralamatkan di Jl. Kertomenanggal IV No.1 Gayungan, Kota Surabaya, Jawa Timur, 60234 dengan dokter yang direkomendasikan untuk penelitian dari Rumah

Sakit/Puskesmas Program Penanggulangan TB Care Aisyiyah, yaitu dr. Daniek Suryaningdiah sebagai dokter yang menangani program penyakit Tuberkulosis. Dalam pengembangan sistem pakar, dilakukan tanya jawab secara langsung kepada dokter spesialis penyakit Tuberkulosis untuk mendapatkan informasi yang dibutuhkan dalam proses perancangan sistem pakar berbasis web dengan metode Forward Chaining dalam upaya mendiagnosis dini penyakit Tuberkulosis di Jawa Timur. Ada dua pendapat mengenai pelaksanaan metode ini. Pertama dengan cara membawa seluruh data yang didapat ke sistem pakar. Kedua dengan membawa bagian-bagian penting saja dari data yang didapat ke sistem pakar. Cara pertama lebih baik digunakan jika sistem pakar terhubung dengan proses otomatis dan penerima seluruh data dari database. Cara kedua menghemat waktu serta biaya dengan mengurangi data dan mengambil data yang dianggap perlu. Sebagai contoh, seperti kasus pada kedua metode di atas, maka berdasarkan metode ini langkah-langkah yang diambil

Prista Amanda Putri, dan Hindayati Mustafidah, ISSN: 2086-9398 Vol. I Nomor 4, 2011. Sistem Pakar untuk Mendiagnosa Penyakit Hati Menggunakan Metode Forward Chaining. Sistem pakar merupakan program komputer untuk dapat meniru proses pemikiran dan pengetahuan pakar untuk menyelesaikan suatu masalah yang spesifik. Implementasi sistem pakar banyak digunakan untuk kepentingan masyarakat karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu ke dalam suatu program, sehingga dapat memberikan keputusan dan melakukan penalaran secara cerdas. Sistem pakar merupakan cabang dari kecerdasan buatan dan juga merupakan

bidang ilmu yang muncul seiring perkembangan ilmu komputer saat ini. Sistem ini bekerja untuk mengadopsi pengetahuan manusia ke komputer yang menghubungkan dasar pengetahuan dengan sistem inferensi untuk menggantikan fungsi seorang pakar dalam menyelesaikan suatu masalah. Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (inference rules) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu. Sistem pakar dibuat dengan mendapatkan pengetahuan dari seorang pakar, kemudian dilakukan pengkodean ke bentuk yang dapat diproses oleh komputer untuk menyelesaikan persoalan yang sejenis. Sistem pakar sangat tergantung pada suatu bidang dalam menyusun penyelesaian persoalan yang dihadapi oleh sistem.

Samsilul Azhar, Herlina Latipa Sari, Leni Natalia Zulita, Jurnal Media Infotama Vol. 10 No. 1, ISSN 1858 – 2680, 2014. Sistem pakar penyakit ginjal pada manusia menggunakan metode forward chaining. Kemajuan teknologi komputer yang pesat dapat membantu kehidupan manusia bahkan di dalam bidang-bidang di luar disiplin ilmu komputer. Sistem pakar adalah merupakan salah satu cabang kecerdasan buatan yang mempelajari bagaimana meniru cara berpikir seorang pakar dalam menyelesaikan suatu permasalahan, membuat keputusan maupun mengambil kesimpulan sejumlah fakta. Dimana sistem pakar (expert system) merupakan suatu perangkat lunak komputer yang memiliki basis pengetahuan untuk domain tertentu dan menggunakan penalaran inferensi

menyerupai seorang pakar dalam menyelesaikan masalah. Mengamati kehidupan sehari-hari di masyarakat, rupanya bukan hanya faktor pendidikan, ekonomi, dan budaya saja yang menjadi masalah besar bagi masyarakat saat ini. Ternyata faktor social yang menyangkut taraf kesejahteraan, dan kesehatan masyarakat merupakan masalah yang jauh lebih penting untuk di perhatikan. Karena seperti kita ketahui bahwa taraf kesejahteraan hidup sangat berdampak pada tingkat kesehatan dari masyarakat itu sendiri. Dengan kata lain, bagi mereka yang hidup dengan taraf kesejahteraan baik, pola hidup serta kesehatan mereka cenderung lebih terjaga, sedangkan bagi mereka yang hidup dengan taraf kesejahteraan kurang, mereka biasanya kurang peduli atau bahkan tidak menjaga pola hidup dan kesehatan mereka.

(Nusai, Cheechang, Chaiphech, & Thanimkan, 2015) *Procedia Computer Science*, Nomor 1, Volume 63, ISSN 1877-0509, dengan judul *Swine-Vet : a Web-based Expert System of Swine Disease Diagnosis. A web-based expert system of swine disease diagnosis was developed for swine farmers and animal husbandmen. Our expert system was divided into three steps. First step was disease screening. We established the novel model of knowledgerepresentation for inference using swine's gender and age range which defined by the veterinarian. Second step was disease diagnosis using the symptoms. To make a diagnosis using symptoms which are accurate and efficient, we established the novel model of uncertain knowledge representation for inference using determination of significant weight of each symptom which defined by the veterinarian and using the certainty factor of occurred symptom, the value was specified by user. Third*

step was the disease diagnosis using swine necropsy lesion. We established the novel model of knowledge representation for inference using major lesion group which defined by the veterinarian for confirmation of morbidity. From the results of diagnosis by our expert system compared with veterinarian, we found that it could disease screening accurately for 97.50 %, could diagnose by symptom accurately for 92.48% and could diagnose by lesion accurately for 95.62%. And the results of evaluation of satisfaction with Likert-scale by the swine farmers and animal husbandmen were 4.7 and 4.5 respectively.

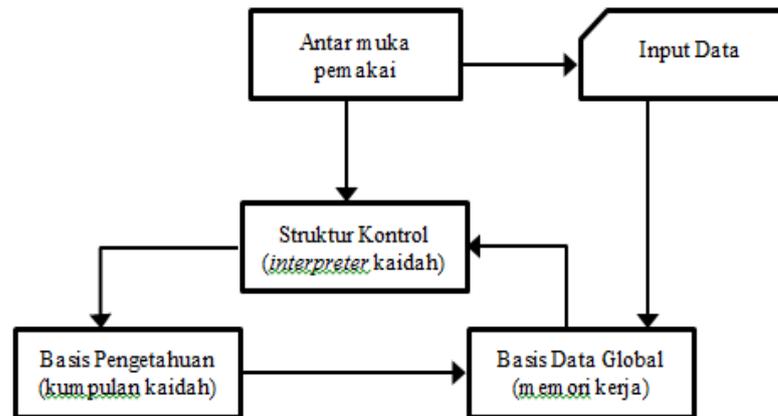
Sistem pakar diagnosis penyakit babi berbasis web dikembangkan untuk petani babi dan penggarap hewan. Sistem pakar kami terbagi dalam tiga tahap. Langkah pertama adalah skrining penyakit. Kami membentuk model baru representasi pengetahuan untuk kesimpulan menggunakan jenis kelamin dan rentang usia babi yang ditentukan oleh dokter hewan. Langkah kedua adalah diagnosa penyakit dengan menggunakan gejalanya. Untuk membuat diagnosis dengan menggunakan gejala yang akurat dan efisien, kami menetapkan model baru representasi pengetahuan yang tidak pasti untuk kesimpulan dengan menggunakan bobot signifikan setiap gejala yang ditentukan oleh dokter hewan dan dengan menggunakan faktor kepastian gejala yang terjadi, nilai tersebut ditentukan oleh Pengguna. Langkah ketiga adalah diagnosis penyakit dengan menggunakan lesi nekropsis babi. Kami menetapkan model baru representasi pengetahuan untuk kesimpulan menggunakan kelompok lesi utama yang ditetapkan oleh dokter hewan untuk konfirmasi ketidakcukupan. Dari hasil diagnosa dengan sistem pakar kami dibandingkan dengan dokter hewan, kami menemukan bahwa dapat disaring

penyakit secara akurat sebesar 97,50%, dapat didiagnosis dengan gejala secara akurat sebesar 92,48% dan dapat didiagnosis dengan lesi secara akurat sebesar 95,62%. Dan hasil evaluasi kepuasan dengan skala likert oleh petani babi dan peternak masing-masing adalah 4,7 dan 4,5.

(C. Fiarni, A. Samuel & H. Maharani, 2015) *Procedia Computer Science*, Nomor 1, Volume 72, dengan judul *Automated Scheduling System for Thesis and Project Presentation Using Forward Chaining Method With Dynamic Allocation Resources Cut. This paper presents a practical method for modeling and solving a dynamic resource allocation of automatic scheduling problem using forward chaining heuristic approach, in the case of undergraduate Student's Thesis and Project presentations timetable. Poor scheduling practices would cause double-assignments of lecturers, prolonged postponement and cancellations of presentations as well as inefficient use of time and resources. This method will follow a pre-assigned logic rules and algorithm to fit the optimization criteria's. The output of this research will be an automatic set of presentation schedule alternatives that will take into account all the constraints. The proposed algorithm for this automatic scheduling system could generate optimal presentation timetable and enables direct interaction with lecturers in order to gather data of their availability time among other its functionalities. The proposed system performs satisfactorily in term of accuracy, data handling and adaptability on helping the faculty to arrange presentations more easily, yield a reliable record and increase efficient use of resources.*

Makalah ini menyajikan metode praktis untuk pemodelan dan pemecahan alokasi sumber daya dinamis dari masalah penjadwalan otomatis dengan menggunakan pendekatan heuristik forward chaining, dalam hal Skripsi Tesis dan sarjana mahasiswa. Jadwal proyek presentasi. Praktik penjadwalan yang buruk akan menyebabkan dua kali penempatan dosen, penundaan dan pembatalan presentasi yang berkepanjangan serta penggunaan waktu dan sumber daya yang tidak efisien. Metode ini akan mengikuti aturan logika dan algoritma yang telah ditetapkan agar sesuai dengan kriteria optimasi. Output dari penelitian ini akan menjadi alternatif pilihan presentasi otomatis yang akan memperhitungkan semua kendala. Algoritma yang diusulkan untuk sistem penjadwalan otomatis ini dapat menghasilkan jadwal penyajian yang optimal dan memungkinkan interaksi langsung dengan dosen untuk mengumpulkan data tentang ketersediaan waktu mereka di antara fungsionalitasnya. Sistem yang diusulkan berjalan dengan memuaskan dalam hal akurasi, penanganan data dan kemampuan beradaptasi dalam membantu fakultas untuk mengatur presentasi dengan lebih mudah, menghasilkan catatan yang andal dan meningkatkan penggunaan sumber daya secara efisien.

2.5 Kerangka Pemikiran



Gambar 2.5 Kerangka Pemikiran

Mempersiapkan data yang dibutuhkan untuk menganalisis penyakit Udag vanamei agar lebih mudah untuk memproses pengolahan datanya. Kemudian data-datanya tersebut di olah menggunakan sistem pakar menggunakan metode *forward chaing*. Bahasa prnograman yang dipakai dalam sistem pakar ini *PHP* dan *database MySQL* yang digunakan untuk proses mendiagnosa penyakit yang diderita udang vanamei dan menghasilkan *output* (hasil diagnosa).