

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Teori adalah sekumpulan dari beberapa konsep, definisi dan proporsi yang berfungsi untuk melihat fenomena secara sistematis, melalui spesifikasi hubungan antar variabel, sehingga dapat berguna untuk menjelaskan dan meramalkan fenomena (Sugiyono, 2014:52). Teori dasar yang dicantumkan pada bab ini adalah sebagai berikut:

2.1.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat AI, yaitu *intelligence* yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud disini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. 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 di masa-masa yang akan datang (Sutojo, Mulyanto, & Suhartono, 2011: 1-3).

Persoalan-persoalan yang ditangani oleh kecerdasan buatan semakin lama semakin berkembang sehingga memungkinkan bagi kecerdasan untuk merambah ke bidang ilmu yang lain. Hal ini disebabkan karakteristik cerdas sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. Klasifikasi subdisiplin ilmu dalam kecerdasan buatan didasarkan pada hasil perkawinan antara kecerdasan buatan dengan bidang ilmu lainnya (Sutojo et al., 2011: 12).

1. Logika *Fuzzy* (*Fuzzy Logic*)

Konsep tentang logika *fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962. Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”, dan lain-lain. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi, dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1. Artinya, bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara

bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya (Sutojo et al., 2011: 211-212).

Beberapa metode yang digunakan dalam sistem inferensi fuzzy adalah (Sutojo et al., 2011: 233-237):

a) Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan sebagai berikut:

- 1) *Fuzzyfikasi*
- 2) Pembentukan basis pengetahuan *Fuzzy* (*rule* dalam bentuk *IF...THEN*)
- 3) Mesin inferensi menggunakan fungsi implikasi MIN untuk mendapatkan nilai α -predikat tiap-tiap *rule* ($\alpha_1, \alpha_2, \alpha_3, \dots \alpha_n$). Kemudian masing-masing nilai α -predikat ini digunakan untuk menghitung keluaran hasil inferensi secara tegas (*crisp*) masing-masing *rule* ($z_1, z_2, z_3, \dots z_n$)
- 4) *Defuzzyfikasi* menggunakan metode rata-rata (*Average*)

b) Metode Mamdani

Metode Mamdani paling sering digunakan dalam aplikasi-aplikasi karena strukturnya yang sederhana, yaitu menggunakan operasi *MIN-MAX* atau *MAX-PRODUCT*. Untuk mendapatkan *output*, diperlukan 4 tahapan berikut.

- 1) *Fuzzyfikasi*
- 2) Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- 3) Aplikasi fungsi implikasi menggunakan fungsi MIN dan komposisi antar-*rule* menggunakan fungsi MAX (menghasilkan himpunan *fuzzy* baru)
- 4) *Defuzzyfikasi* menggunakan metode *Centroid*

c) Metode Sugeno

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

- 1) *Fuzzyfikasi*
- 2) Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- 3) Mesin inferensi menggunakan fungsi implikasi MIN untuk mendapatkan nilai α -predikat tiap-tiap *rule* ($\alpha_1, \alpha_2, \alpha_3, \dots \alpha_n$). Kemudian masing-masing nilai α -predikat ini digunakan untuk menghitung keluaran hasil inferensi secara tegas (*crisp*) masing-masing *rule* ($z_1, z_2, z_3, \dots z_n$)
- 4) *Defuzzyfikasi* menggunakan metode rata-rata (*Average*)

2. Jaringan Saraf Tiruan (*Artificial Neural Network*)

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu (Sutojo et al., 2011: 283).

Kelebihan-kelebihan yang diberikan oleh jaringan saraf tiruan antara lain (Sutojo et al., 2011: 284):

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

Selain mempunyai kelebihan-kelebihan tersebut, jaringan saraf tiruan juga mempunyai kelemahan-kelemahan berikut (Sutojo et al., 2011: 284-285):

- a. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi tinggi.
- b. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatika, operasi logika, dan simbolis.
- c. Untuk beroperasi, jaringan saraf tiruan butuh pelatihan sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.

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

1. Pelatihan dengan Supervisi (pembimbing)

Pada pelatihan jenis ini, jaringan akan dipandu oleh beberapa pasang data (masukan dan target) yang berfungsi sebagai pembimbing untuk melatih jaringan memperoleh hasil bobot yang terbaik. Algoritma yang termasuk dalam pelatihan dengan supervisi antara lain: *Hebb-Rule*, *Perceptron*, *Delta-Rule*, *Backpropagation*, *Heteroassociative Memory*, *Bidirectional Associative Memory (BAM)*, dan *Learning Vector Quantization (LVQ)*.

2. Pelatihan tanpa Supervisi

Dalam pelatihan jenis 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.

3. Sistem Pakar (*Expert System*)

Sistem pakar merupakan sebuah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti seorang pakar. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli di bidangnya. Sistem pakar dapat membantu aktivitas para pakar sebagai asisten yang berpengalaman. 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 (Mujilawati, 2014).

2.1.2 Sistem Pakar

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

Suatu sistem dapat dikatakan sebagai sistem pakar jika memiliki ciri-ciri sebagai berikut (Sutojo et al., 2011: 162):

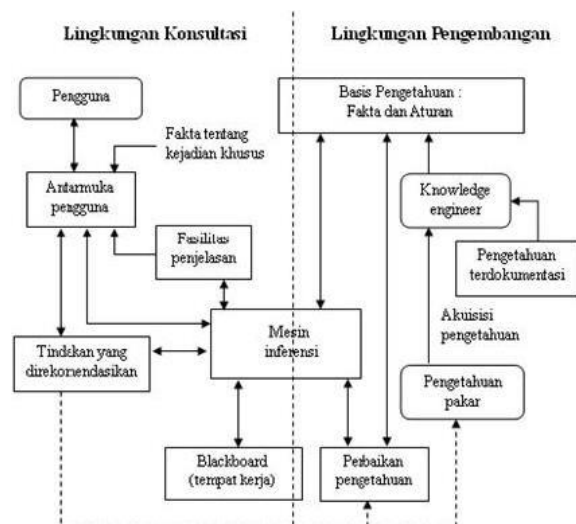
- a. Terbatas pada domain keahlian tertentu.
- b. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
- c. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
- d. Bekerja berdasarkan kaidah/*rule* tertentu.
- e. Mudah dimodifikasi.
- f. Basis pengetahuan dan mekanisme inferensi terpisah.
- g. Keluarannya bersifat anjuran.
- h. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar kedalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu akuisisi pengetahuan (dari pakar atau

sumber lain), representasi pengetahuan (pada komputer), inferensi pengetahuan, dan pemindahan pengetahuan ke pengguna (Sutojo et al., 2011: 164).

1. Struktur Sistem Pakar

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. Gambar 2.1 menunjukkan komponen penting dalam sebuah sistem pakar (Sutojo et al., 2011: 166-167).



Gambar 2.1 Komponen Penting Sistem Pakar
(Sumber: Sutojo et al., 2011: 167)

Berikut adalah penjelasan dari masing-masing komponen struktur sistem pakar diatas (Sutojo et al., 2011: 167-169):

a) Akuisisi Pengetahuan

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

b) Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan berisi pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari 2 elemen dasar, yaitu:

- 1) Fakta, misalnya situasi, kondisi, atau permasalahan yang ada.
- 2) *Rule* (aturan), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

c) Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan proses penalaran. Ada tiga

teknik pengendalian yang digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik tersebut.

d) Daerah Kerja (*Blackboard*)

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

- 1) rencana : bagaimana menghadapi masalah
- 2) agenda : aksi-aksi potensial yang sedang menunggu untuk dieksekusi
- 3) solusi : calon aksi yang akan dibangkitkan

e) Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi akan lebih baik jika disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu, dan formulir elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

f) Subsistem Penjelasan (*Explanation Subsystem / Justifier*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat

penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

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

Kemampuan memperbaiki pengetahuan (*knowledge refining system*) dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang. Kemampuan evaluasi diri diperlukan oleh program agar dapat menganalisis alasan-alasan kesuksesan dan kegagalannya dalam mengambil kesimpulan.

h) Pengguna (*User*)

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

2. Teknik Inferensi

Pada sistem pakar berbasis *rule*, domain pengetahuan direpresentasikan dalam sebuah kumpulan *rule* berbentuk IF-THEN, sedangkan data direpresentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian saat itu. Mesin inferensi memabandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam *database*. Jika bagian IF (kondisi) dari *rule* cocok dengan fakta, maka *rule* dieksekusi dan bagian THEN (aksi) diletakkan dalam

database sebagai fakta baru yang ditambahkan. Beberapa teknik inferensi yang digunakan dalam Sistem Pakar adalah (Sutojo et al., 2011: 171-178):

a. Penalaran Maju (*Forward Chaining*)

Forward chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari *rules* IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka *rule* tersebut dieksekusi. 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. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi

b. Penalaran Mundur (*Backward Chaining*)

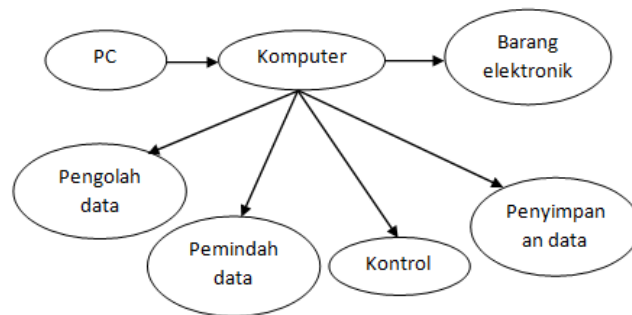
Backward chaining adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada dibagian THEN dari *rule* IF-THEN), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis dibagian IF. Jika cocok, *rule*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

3. Representasi Pengetahuan

Representasi pengetahuan dimaksudkan untuk mengorganisasikan pengetahuan dalam bentuk dan format tertentu untuk bisa dimengerti oleh komputer. Oleh karena itu, untuk membuat sistem pakar yang efektif harus dipilih representasi pengetahuan yang tepat sehingga membuat sistem pakar dapat mengakses basis pengetahuan untuk keperluan pembuatan keputusan. Berikut adalah beberapa model representasi pengetahuan (Hartati & Iswanti, 2008: 22-40).

a) Jaringan Semantik (*Semantic Nets*)

Jaringan semantik adalah teknik representasi pengetahuan yang digunakan untuk informasi proporsional, sedangkan yang dimaksud dengan informasi proporsional adalah pernyataan yang mempunyai nilai benar atau salah. Representasi jaringan semantik merupakan penggambaran grafis dari pengetahuan yang memperlihatkan hubungan hirarkis dari obyek-obyek. Komponen dasar untuk merepresentasikan pengetahuan dalam bentuk jaringan semantik adalah simpul (*node*) dan penghubung (*link*). Obyek direpresentasikan oleh simpul. Hubungan antara obyek-obyek dinyatakan oleh penghubung yang diberi label untuk menyatakan hubungan yang direpresentasikan. Gambar 2.2 berikut ini menunjukkan contoh jaringan semantik.



Gambar 2.2 Contoh Jaringan Semantik
(Sumber: Hartati & Iswanti, 2008: 24)

b) Bingkai (*Frame*)

Bingkai berupa kumpulan slot-slot yang berisi atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berupa kejadian, lokasi, situasi ataupun elemen-elemen lainnya. Bingkai digunakan untuk representasi pengetahuan deklaratif. Bingkai memuat deskripsi sebuah obyek dengan menggunakan tabulasi informasi yang berhubungan dengan obyek. Representasi pengetahuan menggunakan bingkai sesuai untuk jenis pengetahuan yang memiliki subyek sempit, lebih bersifat pasti dan jarang berubah-ubah isinya kecuali terdapat kondisi khusus. Contoh bingkai terlihat pada gambar 2.3.

<i>Slots</i>	<i>Fillers</i>
nama	sepeda motor
spesialisasi	jenis kendaraan beroda dua
produk	Honda, Yamaha, Kawasaki, Daiheiyo
bahan bakar	Bensin

Gambar 2.3 Contoh Bingkai
(Sumber: Hartati & Iswanti, 2008: 24)

c) Kaidah Produksi (*Production Rule*)

Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dan bentuk jika-maka (*if-then*). Kaidah *if-then* menghubungkan anteseden (*antecedent*) dengan konsekuensi yang diakibatkannya. Berbagai struktur kaidah *if-then* yang menghubungkan obyek atau atribut sebagai berikut (Adedeji, 1992 dalam Hartati & Iswanti, 2008: 25-26):

IF premis THEN konklusi	IF masukan THEN keluaran
IF kondisi THEN tindakan	IF antesenden THEN konsekuen
IF data THEN hasil	IF tindakan THEN tujuan
IF aksi THEN reaksi	IF sebab THEN akibat
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. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat tertentu. Gejala mengacu pada keadaan yang

menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan.

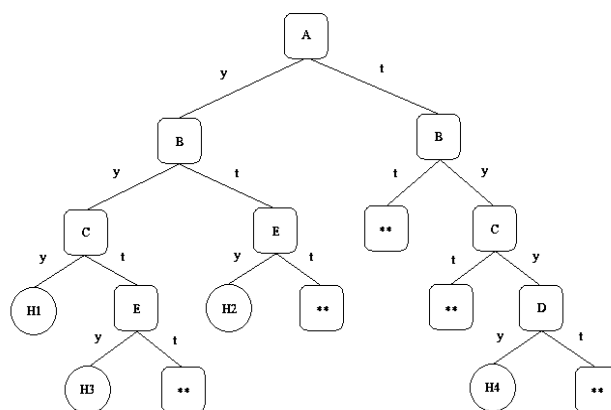
Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk tabel keputusan (*decision table*) kemudian dari tabel keputusan dibuat pohon keputusannya (*decision tree*). Berikut ini adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan (Hartati & Iswanti, 2008: 26-39).

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: Hartati & Iswanti, 2008: 34)

Mengacu tabel keputusan pada tabel 2.1 dapat dihasilkan pohon keputusan sebagai berikut:



Gambar 2.4 Pohon Keputusan
(Sumber: Hartati & Iswanti, 2008: 34)

2.2 Variabel Penelitian

Variabel penelitian merupakan suatu atribut atau sifat atau nilai dari orang, obyek atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya (Sugiyono, 2014: 38).

Berdasarkan pengertian diatas, berikut akan dijelaskan variabel yang digunakan dalam penelitian ini yaitu penyakit fisik akibat kerja serta akan diuraikan indikator-indikator yang dijadikan fokus pada penelitian ini.

2.2.1 Penyakit Fisik Akibat Kerja

Berdasarkan hasil wawancara bersama pakar, penyakit fisik akibat kerja adalah penyakit akibat kerja yang disebabkan oleh faktor fisika yang ada di lingkungan kerja yaitu kebisingan, pencahayaan yang kurang atau berlebihan, radiasi, tekanan panas dan juga getaran. Penyakit akibat kerja itu sendiri adalah setiap penyakit yang disebabkan oleh pekerjaan atau lingkungan kerja. Berikut dijelaskan penyakit akibat kerja yang disebabkan oleh faktor fisik.

1. Kebisingan

Kebisingan adalah semua suara yang tidak dikehendaki yang bersumber dari alat-alat proses produksi atau alat-alat kerja yang pada tingkat tertentu dapat menimbulkan gangguan pada pendengaran. Suara yang keras, berlebihan atau berkepanjangan dapat merusak jaringan saraf sensitif di telinga, menyebabkan

kehilangan pendengaran sementara atau permanen (International Labour Organization [ILO], 2013: 10).

Dampak negatif dari terpaparnya kebisingan di tempat kerja adalah gangguan pendengaran akibat bising. Gangguan pendengaran akibat bising (Noise Induced Hearing Loss) adalah penurunan pendengaran atau tuli akibat pajanan bising yang melebihi nilai ambang batas (NAB) di lingkungan kerja. Gejala-gejala yang dapat terjadi antara lain tinitus (telinga berdenging), sukar menangkap percakapan, dan penurunan pendegaran (Kementerian Kesehatan RI, 2011: 4).

Gangguan pendengaran akibat bising bersifat permanen atau *irreversible* dan tidak dapat diobati dengan obat maupun pembedahan. Sehingga tindakan penanganan yang dapat dilakukan adalah mencegah perburukan penurunan pendengaran dengan memberikan rekomendasi (Kementerian Kesehatan RI, 2011: 5-6):

- a. Hindarkan penderita dari tempat kerja / lingkungan bising dengan cara melakukan rotasi atau penjadwalan kerja
- b. Penggunaan Alat Pelindung Pendengaran (*Ear Plug / Earmuff*)
- c. Bila gangguan pendengaran sudah mengakibatkan kesulitan berkomunikasi dengan volume percakapan biasa, dapat dicoba pemasangan alat bantu dengar/ABD (*hearing aid*).

Pencegahan dapat dilakukan dengan melaksanakan Program Konservasi Pendengaran di tempat kerja dengan baik. Program Konservasi Pendengaran (PKP) sebagai berikut.

- a. Identifikasi sumber bising (*walk through survey*)
- b. Pengukuran dan analisis kebisingan (*Octave Band Analyzer*)
- c. *Pengendalian* bising dalam bentuk kontrol *engineering* maupun kontrol administrasi. Kontrol *engineering* dapat dilakukan dengan cara meredam sumber bunyi yang berasal dari generator diesel, mesin dan peralatan kerja atau bising yang ditimbulkan oleh aktivitas pekerja seperti di tempat penempaan logam. Kontrol administrasi dilakukan dengan menghindarkan pekerja dari tempat kerja/lingkungan bising dengan melakukan rotasi atau pembatasan jam kerja
- d. Tes Audiometri (*pemeriksaan* untuk mengevaluasi fungsi pendengaran dengan menggunakan alat yang bernama audiometer) secara berkala
- e. Memberikan informasi dan edukasi kepada tenaga kerja
- f. Menggunakan APD (Alat Pelindung Diri / *Personal Protective Equipment*); sumbat telinga (*ear plug*), tutup telinga (*earmuff*) dan pelindung kepala (*helmet*)
- g. Pencatatan dan pelaporan data pemeriksaan kesehatan tenaga kerja (Kementerian Kesehatan RI, 2011: 6-7).

2. Pencahayaan

Pencahayaan merupakan sejumlah penyinaran pada suatu bidang kerja yang diperlukan untuk melaksanakan kegiatan secara efektif. Fungsi dari pencahayaan di area kerja antara lain memberikan pencahayaan kepada benda-benda yang menjadi objek kerja operator tersebut, seperti mesin atau peralatan, proses produksi, dan lingkungan kerja. Pencahayaan di setiap tempat kerja harus memenuhi syarat untuk melakukan pekerjaan. Pencahayaan yang sesuai sangat penting untuk peningkatan kualitas dan produktivitas (Rahmayanti & Artha, 2015).

Dampak negatif dari intensitas cahaya yang kurang atau berlebih adalah kelelahan mata. Kelelahan mata merupakan akibat dari ketegangan pada mata dan disebabkan oleh penggunaan indera penglihatan dalam bekerja yang memerlukan kemampuan untuk melihat dalam jangka waktu yang lama dan biasanya disertai dengan kondisi pandangan yang tidak nyaman. Kelelahan mata tersebut tentunya memiliki tanda-tanda serta karakteristik antara lain mata berair, kelopak mata berwarna merah, penglihatan ganda, sakit kepala, dan ketajaman penglihatan menurun (penglihatan kabur). Penanganan kelelahan mata dapat dilakukan dengan mengistirahatkan mata sejenak secara berkala dari kegiatan kerja, melakukan peregangan otot, serta melakukan perawatan mata seperti mengompres mata dengan air hangat atau meneteskan obat tetes mata (Rahmayanti & Artha, 2015).

Kelelahan pada mata dapat dicegah dengan melakukan beberapa cara sebagai berikut:

- a. Pengaturan tata letak ruangan mulai dari penataan posisi sumber cahaya, pengorganisasian peralatan kerja yang ergonomis, serta memperhatikan area gerak bebas dari ruang kerja
- b. Mengadakan perawatan dan pemeliharaan sumber cahaya secara rutin seperti melakukan penggantian lampu yang rusak, berkedip, redup, atau mati
- c. Pengadaan benda-benda yang memiliki fungsi sebagai penyegar indera penglihatan seperti tanaman, lukisan atau berbagai bentuk lainnya agar tenaga kerja dapat mengalihkan perhatian sejenak dari pekerjaan terhadap objek-objek tersebut
- d. Menggunakan alat pelindung diri (kaca mata keselamatan) selama bekerja di area wajib penggunaan alat pelindung diri
- e. Perlu adanya sosialisasi edukasi lebih lanjut mengenai program-program yang dapat meningkatkan kemampuan mata, seperti waktu istirahat mata, peregangan otot sejenak dan kegiatan lainnya yang dapat menghilangkan kelelahan mata (Rahmayanti & Artha, 2015).

3. Getaran

Getaran adalah gerakan yang teratur dari benda atau media dengan arah bolak-balik dari kedudukan keseimbangan. Getaran terjadi saat mesin atau alat dijalankan dengan motor, sehingga pengaruhnya bersifat mekanis. Alat untuk mengukur getaran dinamakan vibrasi meter (Suhardi, 2008).

Getaran mekanis dibedakan atas getaran seluruh tubuh (*whole body vibration*) dan getaran alat-lengan (*tool-hand vibration*). Masing-masing menimbulkan penyakit akibat kerja yang berbeda (Anies, 2014: 138).

- a. Getaran seluruh tubuh (*whole body vibration*). Getaran ini dihasilkan karena seluruh masa tubuh berhadapan dengan getaran mekanis. Contoh: getaran permukaan penyangga pada mesin kontraktor.
- b. Getaran alat lengan (*tool-hand vibration*) atau getaran setempat (*hand arm vibration*), merupakan getaran yang terjadi pada alat tubuh yang bersentuhan langsung dengan media getaran dan bagian tubuh yang lain berada pada posisi diam. Biasanya bagian tubuh yang terkena getaran yaitu lengan tangan sehingga sering disebut *hand arm vibration*. *Hand arm vibration* disebabkan oleh pengoperasian peralatan tangan bertenaga (*Hand-held Power Tools*). Getaran jenis ini biasanya dialami oleh tenaga kerja yang dipekerjakan pada operator gergaji rantai, tukang semprot, potong rumput, gerinda, dan penempa palu.

Pemaparan pada getaran dapat menyebabkan akibat negatif yang permanen bila dibiarkan tidak diperiksa dan tidak ditangani. Berikut ini adalah efek pemaparan getaran setempat (*hand arm vibration*) dan getaran seluruh tubuh (*whole body vibration*).

- 1) Efek pemaparan *hand arm vibration*

Getaran dapat menyebabkan perubahan dalam tendon, otot, tulang dan sendi, dan dapat mempengaruhi sistem saraf. Secara kolektif, efek *hand arm*

vibration dikenal dengan *hand-arm vibration syndromes* (HAVS). Perkembangan dari HAVS bersifat bertahan dan keparahan semakin lama semakin meningkat. HAVS mungkin menjadi dapat diamati secara klinis setelah beberapa bulan atau beberapa tahun. Tenaga kerja yang mengalami HAVS akan mengalami gejala:

- a. Serangan pemutihan (*blancing*) satu jari atau lebih bila terpapar dingin
- b. Rangsangan nyeri seperti disengat (*tingling*) dan kehilangan rasa di jari
- c. Kehilangan rasa rabaan lembut
- d. Sensasi nyeri dan dingin di antara serangan jari menjadi putih
- e. Kehilangan kekuatan menggenggam
- f. Struktur tulang membentuk kista (berbentuk benjolan) di jari dan pergelangan tangan (Soedirman & Prawirakusumah, 2014: 122-123).

Perawatan yang dapat dilakukan untuk pemulihan gejala agar peredaran darah normal kembali yaitu pemanasan tangan dalam air hangat, pemijatan, meniupkan udara panas ke arah tangan, serta menggerakkan tangan secara berputar (Untari, 2017).

2) Efek getaran seluruh tubuh (*whole-body vibration*)

Pemaparan pada getaran seluruh tubuh mengakibatkan suatu penyakit menyeluruh, yang dikatakan sebagai *vibration sickness*. Gejala yang timbul yaitu pegal-pegal (kelelahan), insomnia, sakit kepala, mengalami masalah

pada bagian perut, mual, kaki kesemutan, dan mata berkunang-kunang. Pada kondisi ini tenaga kerja perlu untuk beristirahat sejenak kemudian melakukan pemeriksaan lebih lanjut dengan pihak medis (Soedirman & Prawirakusumah, 2014: 127).

Pencegahan yang dapat dilakukan terhadap akibat negatif pemaparan pada getaran adalah sebagai berikut (Soedirman & Prawirakusumah, 2014: 127-128).

- a) Merancang ulang untuk meminimalisasi getaran pada penggunaan alat-alat vibrasi yang dipegang tangan
- b) Jika perancangan ulang tidak memungkinkan, reduksi atau mengurangi penggunaan alat-alat vibrasi harus dilakukan
- c) Substitusi alat vibrasi manual dengan mesin vibrasi
- d) Bila mungkin, alat-alat dengan vibrasi tinggi harus diganti dengan alat-alat yang lebih baik
- e) Membatasi waktu pemaparan vibrasi dan menggunakan waktu istirahat untuk menghindari pemaparan vibrasi konstan dan kontinu
- f) Istirahat selama 10 menit setiap 1 jam dilakukan untuk pekerja yang menggunakan alat vibrasi secara terus menerus.

4. Tekanan Panas

Tenaga kerja yang bekerja dengan beban kerja tertentu di lingkungan kerja dengan panas yang tinggi dapat menderita gangguan dan penyakit yang dikenal dengan penyakit yang berhubungan dengan suhu udara panas (*heat-related*

disease). Ada beberapa jenis *heat-related disease*, berkisar dari yang menyebabkan ketidaknyamanan sementara (*temporary discomfort*), sampai yang biasanya berupa kondisi fatal, antara lain *heat stroke* (Soedirman & Prawirakusumah, 2014: 108-110).

- a. *Heat rash*. *Heat rash* adalah iritasi kulit yang disebabkan oleh keringat yang terlalu banyak karena panas dan lembap. *Heat rash* memiliki gejala bintik-bintik kemerahan pada kulit yang berisi cairan (keadaan seperti biang keringat) dan blister kecil (lecet atau melepuh) pada kulit. Pada kondisi ini tenaga kerja perlu beristirahat pada tempat yang lebih sejuk, menjaga kulit agar tetap terlindung dan kering dan menggunakan bedak penghilang keringat.
- b. *Heat cramp* ialah kekejangan otot yang disebabkan bekerja di lingkungan kerja yang panas, sehingga banyak keluar keringat dan mengakibatkan hilangnya garam Na dari tubuh. Gejala yang dirasakan adalah otot lengan, kaki, atau perut menjadi nyeri akibat kontraksi mendadak atau kejang, pingsan, mual, muntah, suhu badan normal, kulit lembab dan dingin, serta berkeringat. Cara penanggulangan adalah dengan beristirahat di tempat yang dingin dan minum cairan elektrolit (garam).
- c. *Heat syncope* merupakan kondisi dimana tiba-tiba terserang pusing atau *fainting* yaitu keadaan tidak sadar secara sementara atau lemah sesudah bekerja atau mengeluarkan tenaga dalam lingkungan yang panas atau terpapar suhu tinggi dengan tanda-tanda kulit pucat dan berkeringat, tetapi

tetap dingin, denyut nadi cepat tetapi lemah, dan suhu tubuh normal. Penanganan penderita *heat syncope* adalah dengan mengeluarkan penderita dari pemaparan dan dipindahkan ke tempat yang dingin, melonggarkan pakaian, memposisikan kaki lebih tinggi dari jantung, dan segera menghubungi pihak medis.

- d. Kelelahan akibat panas (*Heat exhaustion*), terjadi karena cuaca kerja yang sangat panas sehingga menyebabkan turunnya volume air darah karena dehidrasi (terlalu banyak berkeringat dan tidak cukup minum). Penderita *heat exhaustion* akan mengalami haus, pusing, mengalami *malaie* (lemah lesu), mual, berkeringat sangat banyak, suhu tubuh normal, kulit dingin dan lembap, pucat, dan mungkin pingsan. Cara mengatasi, jika pekerja sadar, istirahatkan di tempat yang sejuk, minum cairan yang mengandung elektrolit, melonggarkan pakaian, dan bila kedinginan perlu memakai selimut. Jika pekerja pingsan, segera mencari bantuan medis dan jangan memberi minum jika pekerja pingsan.
- e. *Heat stroke* adalah kerusakan serius yang berkaitan dengan kesalahan pada pusat pengatur suhu tubuh di otak. *Heat stroke* adalah kondisi serius yang mengancam nyawa yang terjadi bila tubuh kehilangan kemampuan mengontrol suhu. Penderita *heat stroke* hampir selalu meninggal dunia, sehingga tindakan medis segera adalah hal yang sangat penting bila gejala-gejala awal muncul. Gejala *heat stroke* yaitu sakit kepala, malaise (lemah lesu), suhu tubuh yang cepat naik hingga melebihi 40°C, kulit panas,

kemerahan dan kering, tidak berkeringat, penderita merasa kebingungan ditandai dengan percakapan membingungkan, halusinasi, dan sebagainya, pernafasan cepat, nadi cepat, secara tiba-tiba terjadi penurunan kesadaran (*delirium*), serta kejang jika suhu terus naik. Pada kondisi ini, penanggulangan yang harus dilakukan adalah mencari bantuan medis segera, memindahkan penderita ke tempat yang dingin, melonggarkan pakaian, menggunakan handuk basah atau air dan kipas untuk mendinginkan penderita, dan segera mencari bantuan medis.

Pencegahan terhadap *heat-related disease* dilakukan dengan cara berikut ini (Soedirman & Prawirakusumah, 2014: 111-113):

- 1) Menurunkan kondisi panas lingkungan kerja dengan penerapan teknologi pengendalian. Teknologi pengendalian dilakukan dengan cara:
 - a) Pendinginan setempat (*spot cooling*), yaitu pendinginan yang dilakukan dengan mengalirkan udara segar berkecepatan tinggi ke arah tubuh menggunakan kipas angin
 - b) Ventilasi, dilakukan dengan cara *cross ventilation* dan *natural draft*. *Cross ventilation* dilakukan dengan memasukkan udara segar ke dalam lingkungan kerja melalui bukaan pada dinding di satu sisi, yang mendinginkan ruangan panas, sekaligus mendorong udara panas keluar melalui bukaan di seberang yang lain. Sedangkan *natural draft* dilakukan dengan mengeluarkan udara panas ke atas melalui cerobong atau bangunan terbuka di atap.

- c) Perisai panas (*metal shielding*) yaitu memasang pelat logam yang cekung ke dalam antara sumber panas dan pekerja. Antara sumber panas dan perisai logam dialirkan udara.
 - d) Memasang pendingin udara (*air conditioning*).
 - e) Menggunakan Alat Pelindung Diri (APD)
- 2) Pengaturan waktu kerja
 - 3) Suplementasi air dan garam. Tenaga kerja perlu untuk sering minum dalam jumlah sedikit namun sering, dan air harus dijaga cukup dingin dengan suhu diantara 10 hingga 15°C. Tenaga kerja juga harus menjaga ketersediaan garam dalam tubuh. Air minum dengan garam harus tersedia dengan konsentrasi 0,1% (1 g NaCl dalam 1 liter air).

2.3 Software Pendukung

Software pendukung merupakan perangkat lunak komputer yang digunakan untuk merancang suatu sistem. Adapun perangkat lunak yang digunakan dalam penelitian ini adalah:

2.3.1 Unified Modeling Language (UML)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu

Unified Modeling Language (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa & Shalahuddin, 2014: 137).

2.3.1.1 Use Case Diagram

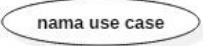
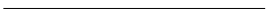
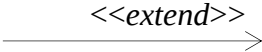
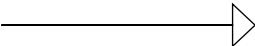
Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa & Shalahuddin, 2014: 155).

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada 2 hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* (Rosa & Shalahuddin, 2014: 155).

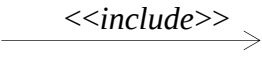
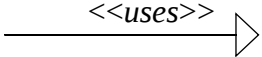
1. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case* (Rosa & Shalahuddin, 2014: 156-158):

Tabel 2.2 Simbol Diagram *Use Case*

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 di awal 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. 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> 	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; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan
Generalisasi / <i>generalization</i> 	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

Tabel 2.2 Lanjutan

Menggunakan / <i>include</i> / <i>uses</i>  	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. ada dua sudut pandang mengenai <i>include</i> di <i>use case</i> : 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan 2. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan
---	--

Sumber: Rosa & Shalahuddin (2014: 156-158)

2.3.1.2 Activity Diagram

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 oleh aktor, jadi aktivitas yang dapat lakukan oleh sistem (Rosa & Shalahuddin, 2014: 161).

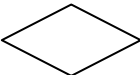
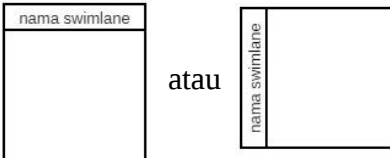
Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa & Shalahuddin, 2014: 161-162).

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokan tampilan dari sistem / *use interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang diperlukan didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak

Berikut adalah simbol-simbol yang ada pada diagram aktivitas (Rosa & Shalahuddin, 2014: 162-163):

Tabel 2.3 Simbol Diagram Aktivitas

Simbol	Deskripsi
Status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Rosa & Shalahuddin (2014: 162-163)

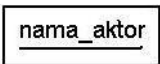
2.3.1.3 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa & Shalahuddin, 2014: 165).

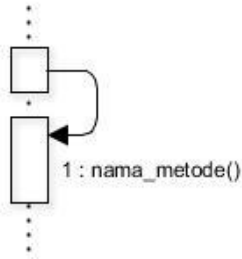
Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (Rosa & Shalahuddin, 2014: 165).

Berikut adalah simbol-simbol yang ada pada diagram sekuen (Rosa & Shalahuddin, 2014: 165-167):

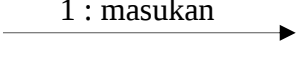
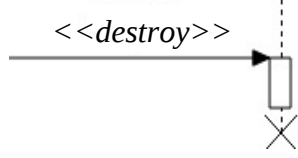
Tabel 2.4 Simbol Sequence Diagram

Simbol	Deskripsi
Aktor  nama aktor atau  nama_aktor tanpa waktu aktif	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, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor

Tabel 2.4 Lanjutan

Garis hidup / <i>lifeline</i> 	menyatakan kehidupan suatu objek
Objek nama objek: nama kelas	menyatakan objek yang berinteraksi pesan
Waktu aktif 	menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
Pesan tipe <i>creator</i> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe <i>call</i>	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas sesuai dengan kelas objek yang berinteraksi

Tabel 2.4 Lanjutan

Pesan tipe <i>send</i> 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe <i>return</i> 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe <i>destroy</i> 	menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber: Rosa & Shalahuddin (2014: 165-167)

2.3.1.4 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, dan metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa & Shalahuddin, 2014: 141-142).

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak

sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas sebagai berikut (Rosa & Shalahuddin, 2014: 142):

1. Kelas *main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

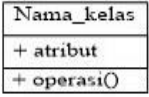
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

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

Berikut adalah simbol-simbol yang ada pada diagram kelas (Rosa & Shalahuddin, 2014: 146-147):

Tabel 2.5 Simbol Diagram Kelas

Simbol	Deskripsi
Kelas 	kelas pada struktur sistem
Antarmuka / <i>interface</i>  nama_interface	sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek

Tabel 2.5 Lanjutan

Asosiasi / <i>association</i> _____	relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> _____>	relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi _____▷	relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> _____>	relasi antarkelas dengan makna sebuah kebergantungan antarkelas
Agregasi / <i>aggregation</i> _____◊	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber: Rosa & Shalahuddin (2014: 146-147)

2.3.2 Notepad++



Gambar 2.5 Logo Notepad++
(Sumber: Supono & Putratama, 2016: 13)

Notepad++ merupakan aplikasi teks editor gratis dan powerful yang dapat digunakan oleh seorang pengembang aplikasi (*programmer*) untuk menuliskan kode-kode program. *Notepad++* mendukung banyak bahasa pemrograman, diantaranya *Assembly*, *C*, *C++*, *C#*, *CSS*, *HTML*, *Java*, *Javascript*, *Pascal*, *Perl*, *PHP*, *Python*,

Ruby, Shell, SQL, VB, XML, dan lain sebagainya. *Notepad++* memiliki banyak kelebihan bila dibandingkan dengan *Notepad* bawaan *Windows* yang pertama, seperti memiliki *GUI* yang baik dan menarik. Selain itu *Notepad++* juga dapat ditambahkan sebagai *plugin* yang bisa semakin mempermudah pekerjaan *programmer*. Kelebihan lainnya adalah *Notepad++* memiliki versi portabel (Supono & Putratama, 2016: 13).

2.3.3 MySQL

MySQL merupakan salah satu *database* kelas dunia yang bekerja dengan menggunakan bahasa *SQL (Structure Query Language)*. *SQL* merupakan bahasa standar yang digunakan untuk memanipulasi *database*. Perintah dasar (*query commands*) dalam *MySQL* adalah *select* (mengambil), *insert* (menambah), *update* (mengubah), dan *delete* (menghapus). *SQL* juga menyediakan perintah untuk membuat *database, field*, ataupun *index* untuk menambah atau menghapus data.



Gambar 2.6 Logo MySQL
(Sumber: Saputra, 2012: 77)

2.3.4 XAMPP (X Apache MySQL PHP Perl)

XAMPP adalah sebuah perangkat lunak bebas yang mendukung banyak sistem operasi dan merupakan kompilasi dari beberapa program. Fungsi *XAMPP*

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*. Program ini tersedia dalam *GNU General Public License* (Aditya, 2011: 16).



Gambar 2.7 **Logo XAMPP**
(Sumber: Aditya, 2011)

2.3.5 *PHP (Hypertext Preprocessor)*

PHP atau *Hypertext Processor* adalah bahasa yang berbentuk *script* yang ditempatkan dalam server dan dieksekusi di dalam server untuk selanjutnya ditransfer dan dibaca oleh *client*. *PHP* tidak dapat lepas dari *database MySQL*. Oleh karena itu, dalam membuat suatu *website* dengan bahasa pemrograman *PHP*, *web server* sangat dibutuhkan (Adelheid & Nasution, 2012: 2).



Gambar 2.8 **Logo PHP**
(Sumber: Adelheid & Nasution, 2012: 2)

2.3.6 *HTML*

HTML (Hyper Text Markup Language) merupakan bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website*. *HTML* berguna untuk menampilkan berbagai informasi didalam sebuah penjajah *web* internet dan *formatting hypertext* sederhana yang ditulis ke dalam berkat format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. *HTML* memiliki 2 jenis ekstensi file yaitu .htm dan .html (Saputra, 2012: 1).



Gambar 2.9 Logo *HTML*
(Sumber: Agus Saputra, 2012:1)

2.3.7 *CSS (Cascading Style Sheet)*

CSS (Cascading Style Sheet) adalah bahasa pemrograman web yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam *web* menjadi lebih rapi, terstruktur dan seragam. *CSS* merupakan salah satu pemrograman wajib selain *html* yang harus dikuasai oleh pemrogram *web*. Tujuan utama dari *CSS* adalah untuk memisahkan konten utama dengan tampilan

dokumen lainnya. *Web* dengan CSS akan lebih ringan dan mudah untuk dibuka dibandingkan dengan *web* yang tidak menggunakan CSS. Keuntungan yang dapat diperoleh dengan menggunakan CSS diantaranya (Saputra, 2012: 27-29):

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



Gambar 2.10 Logo CSS
(Sumber: Saputra, 2012: 27-29)

2.3.8 *jQuery*

jQuery merupakan salah satu dari *JavaScript library* yaitu kumpulan fungsi *JavaScript* siap pakai sehingga mempermudah dan mempercepat *programmer*

dalam membuat kode *JavaScript*. *jQuery* memiliki dua jenis skrip, yaitu skrip untuk *development* yang dikhususkan untuk para pengembang *website* dan skrip yang telah dikompres untuk para pengguna. Berikut adalah beberapa kelebihan dari *jQuery* (Rohingun, 2015: 1-3):

- 1) *jQuery* telah banyak dipakai oleh *website-website* terkemuka di dunia
- 2) Kompatibel dengan dengan semua *browser*
- 3) Kompatibel dengan semua versi *CSS*
- 4) Dokumentasi, tutorial, dan contoh-contoh lengkap
- 5) Didukung oleh komunitas yang besar dan aktif
- 6) Ketersediaan *plugin* yang sangat banyak
- 7) File-nya hanya ada satu dan berukuran kecil, sehingga akses cepat
- 8) *Open source* gratis dengan lisensi dari *GNU General Public License* dan *MIT License*

2.3.9 Framework CodeIgniter

CodeIgniter merupakan aplikasi *open source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan *PHP*. *CodeIgniter* memudahkan *developer website* untuk membuat aplikasi *website* dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. *CodeIgniter* dirilis pertama kali pada 28 Februari 2006 (Wahana Komputer, 2011: 2).



Gambar 2.11 Logo *CodeIgniter*
(Sumber: Wahana Komputer, 2011: 2)

Beberapa kelebihan *framework CodeIgniter (CI)* dibandingkan dengan *framework PHP* lain adalah sebagai berikut (Wahana Komputer, 2011: 4):

- a) Performa sangat cepat
- b) Konfigurasi yang sangat minim (*Nearly Zero Configuration*)
- c) Banyak komunitas. Dengan adanya komunitas, akan memudahkan *developer website* untuk melakukan interaksi dengan *developer website* lainnya
- d) Dokumentasi yang sangat lengkap. Paket instalasi *CodeIgniter* sudah disertai *user guide* yang sangat lengkap dengan bahasa yang mudah dipahami.

2.4 Penelitian Terdahulu

Penelitian terdahulu merupakan penelitian-penelitian yang dilakukan oleh para ahli terdahulu sebelum penelitian ini. Hasil penelitian-penelitian tersebut dijadikan referensi dalam penelitian ini, baik variabel-variabel terkait maupun

asumsi-asumsi yang relevan dari hasil penelitian tersebut. Penjelasan lebih lanjut tentang hasil penelitian yang telah dilakukan dan berkaitan dengan penelitian ini adalah sebagai berikut:

1. (Prabowo, Widyananda, & Santoso, 2008) dalam penelitiannya tentang “Sistem Pakar Berbasis Web Untuk Diagnosa Awal Penyakit THT” diperoleh kesimpulan: Penyakit THT (Telinga, Hidung, dan Tenggorokan) merupakan penyakit yang sering terjadi di masyarakat. Banyaknya keluhan dan gejala yang ada dan berbagai macam jenis penyakit THT, menyebabkan identifikasi penyakit THT menjadi sulit. Maka dari itu dibutuhkan sebuah aplikasi untuk mendiagnosa gejala-gejala dan keluhan yang dirasakan pasien untuk mengidentifikasi apakah merupakan gejala dari penyakit THT. Sistem pakar merupakan salah satu solusi untuk mendiagnosis penyakit berdasarkan gejala yang dirasakan oleh penderita. Pada penelitian ini dibuat sebuah sistem pakar menggunakan konsep *Forward Chaining*, dengan menggunakan metode *Certainty Factor* untuk mendiagnosa penyakit THT pada manusia. Sistem ini dapat memberikan diagnosa awal penyakit THT yang diderita oleh penderita dari gejala-gejala yang dirasakan oleh penderita, tanpa harus bertanya langsung ke pakar. Hasil penelitian menunjukkan bahwa CF dapat digunakan sebagai cara untuk mengatasi ketidakpastian untuk kasus diagnosa awal THT. ISSN: 1907-5022.

2. (Minarni & Ariani, 2013) dalam penelitiannya tentang “Perancangan Perangkat Lunak Diagnosa Penyakit Mata Khusus Gangguan Konjungtiva Dengan Metode Forward Chaining Berbasis Web” diperoleh kesimpulan: Konjungtivitis adalah peradangan pada konjungtiva dan penyakit ini merupakan penyakit mata yang paling umum di dunia. Penyakit ini bervariasi mulai dari hiperemia ringan dengan mata berair sampai konjungtivitis berat dengan banyak sekret purulen kental menyebabkan infeksi pada mata semakin banyak karena meningkatnya penggunaan obat-obatan topical dan agen immunosupresif sistemik. Penelitian ini bertujuan untuk merancang sebuah aplikasi untuk membantu pengguna untuk mendeteksi dini penyakit gangguan konjungtiva Aplikasi ini dibangun dalam rekayasa perangkat lunak berbasis web, dengan menggunakan bahasa pemrograman PHP dan database yang digunakan adalah MySQL. Dengan rekayasa perangkat lunak akan memberikan informasi kepada user bagaimana mengenali dan mengetahui jenis penyakit mata khusus gangguan konjungtiva beserta solusinya. ISSN: 2086-4981.
3. (Rimantho & Cahyadi, 2015) dalam penelitiannya yang berjudul “Analisis Kebisingan Terhadap Karyawan Di Lingkungan Kerja Pada Beberapa Jenis Perusahaan” diperoleh kesimpulan: Gangguan pendengaran yang disebabkan oleh kebisingan di lingkungan pekerjaan telah menjadi perhatian para peneliti. Pemerintah memberikan aturan secara jelas mengenai ambang batas mengenai kebisingan di lingkungan kerja dalam kaitannya dengan

pencegahan penyakit akibat kerja. Penelitian ini menganalisa paparan kebisingan kerja dan penggunaan alat pelindung diri kebisingan pada beberapa industri yang berbeda di Jakarta. Kuesioner digunakan untuk menggali informasi pada responden yang dianggap berpotensi terpapar oleh kebisingan di lingkungan kerjanya. Responden dipilih secara acak yaitu 400 orang pekerja pada 3 lingkungan industri yang berbeda seperti permesinan, industri daur ulang biji plastik, dan industri konveksi. Studi menunjukkan bahwa industri permesinan memiliki tingkat kebisingan yang lebih tinggi, yaitu sekitar 97 dB, sedangkan industri pengolahan biji plastik sekitar 92 dB dan industri konveksi sekitar 65 dB. Proporsi terbesar penggunaan APD adalah wanita yaitu sekitar 75% sementara laki-laki hanya sekitar 65%. Sedangkan berdasarkan usia, diperoleh informasi bahwa usia responden 21-35 tahun merupakan pengguna APD terbesar yaitu sekitar 67.8% dan usia di atas 46 tahun menggunakan APD sekitar 37.2%. Para stakeholder mempunyai peranan yang cukup penting dalam upaya mereduksi potensi risiko yang dapat muncul dari paparan tingkat kebisingan pada lingkungan pekerjaan serta senantiasa memperhatikan faktor-faktor kesehatan dan keselamatan kerja (K3) karyawan. ISSN: 2085-1669.

4. (Zubair et al., 2011) dalam penelitiannya yang berjudul “*Diagnosis of Skin Diseases using Online Expert System*” diperoleh kesimpulan: *This paper describes Expert System (ES) for diagnosis and management of skin diseases. More than 13 types of skin diseases can be diagnosed and treated*

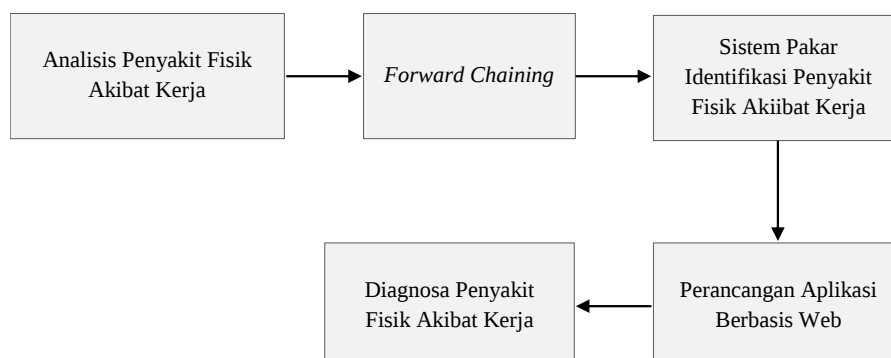
by our system. It is rule based web-supported expert system, assisting skin specialists, medical students doing specialization in dermatology, researchers as well as skin patients having computer know-how. System was developed with Java Technology. The expert rules were developed on the symptoms of each type of skin disease, and they were presented using tree-graph and inferred using forward-chaining with depth-first search method. User interaction with system is enhanced with efficient user interfaces. The web based expert system described in this paper can detect and give early diagnosis of thirteen plus skin diseases. This ES can be extended to diagnose all types of skin-diseases. ISSN: 1947-5500.

5. (Amarathunga, Ellawala, Abeysekara, & Amalraj, 2015) dalam penelitiannya tentang “Expert System For Diagnosis of Skin Diseases” diperoleh kesimpulan: *Dermatology is a one of major session of medicine that concerned with the diagnosis and treatment of skin diseases. Skin diseases are the most common form of disease in humans. Recently, many of researchers have advocated and developed the imaging of human vision or in the loop approach to visual object recognition. This research paper presents a development of a skin diseases diagnosis system which allows user to identify diseases of the human skin and to provide advises or medical treatments in a very short time period. For this purpose, user will have to upload an image of skin disease to our system and answer questions based on their skin condition or symptoms. It will be used to detect diseases*

of the skin and offer a treatment recommendation. This system uses technologies such as image processing and data mining for the diagnosis of the disease of the skin. The image of skin disease is taken and it must be subjected to various preprocessing for noise eliminating and enhancement of the image. This image is immediately segmentation of images using threshold values. Finally data mining techniques are used to identify the skin disease and to suggest medical treatments or advice for users. This expert system exhibits disease identification accuracy of 85% for Eczema, 95% for Impetigo and 85% for Melanoma. ISSN: 2277-8616.

2.5 Kerangka Pemikiran

Kerangka pemikiran yang baik menjelaskan secara teoritis hubungan variable yang diteliti. Berdasarkan identifikasi masalah dan rumusan masalah pada bab sebelumnya, maka kerangka pemikiran dari penelitian ini adalah sebagai berikut:



Gambar 2.12 Kerangka Pemikiran
(Sumber: Data Penelitian, 2017)

Berikut adalah keterangan gambar kerangka pemikiran.

1. Analisis data penyakit fisik akibat kerja ini menggunakan sumber data dari buku dan jurnal ilmiah (penelitian terdahulu) serta wawancara langsung dengan pakar yang berkaitan (dokter perusahaan).
2. Metode *Forward Chaining* sebagai pencarian yang dimotori data (data driven search). Runut maju melakukan proses peruntutan (penalaran) dimulai dari premis-premis atau informasi yang berupa diagnosa penyakit sebagai masukan (IF) terlebih dahulu kemudian menuju konklusi atau derived information (THEN).
3. Pembuatan sistem pakar identifikasi penyakit fisik akibat kerja berdasarkan hasil analisis data yang telah diruntutkan melalui metode *forward chaining*.
4. Sistem pakar direalisasikan menjadi sebuah aplikasi berbasis web dengan menggunakan bahasa pemrograman HTML dan manajemen *database* berupa MySQL.

Keluaran (*output*) menghasilkan aplikasi untuk mendiagnosa penyakit fisik akibat kerja dengan menggunakan metode *forward chanining* dalam tampilan *web*.