BAB II
TINJAUAN PUSTAKA

2.1 Teori Dasar

Teori adalah seperangkat konstruk (konsep), definisi dan proposal yang berfungsi untuk melihat fenomena secara sistematik melalui spesifikasi hubungan antar variabel (Sugiyono, 2014:52). Teori dasar yang dicantumkan pada bab ini adalah sebagai berikut:

2.1.1 Kecerdasan Buatan

Beberapa macam cabang ilmu kecerdasan buatan adalah sebagai berikut (Sutojo, dkk., 2011: 283):

1. **Jaringan Saraf Tiruan**

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

Berdasarkan cara memodifikasi bobotnya, pelatihan jaringan saraf tiruan dibagi menjadi dua, yaitu (Sutojo, dkk., 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:*Hebb-Rule*, *Perceptron*, *Delta-Rule*, *Backpropagation*, *Heteroassociative Memory*, *Bidirectional Associative Memory* (*BAM*), dan *Learning Vector Quantization* (*LVQ*).

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. Logika *Fuzzy* (*Fuzzy Logic*)

Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya 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, dkk., 2011: 211).

Beberapa metode yang digunakan dalam sistem inferensi fuzzy adalah (Sutojo, 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 liniear. 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*)

3. Sistem Pakar (*Expert System)*

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

Beberapa metode yang digunkaan dalam Sistem Pakar adalah(Sutojo, dkk., 2011:171-178):

1. *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.*

2. *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 di bagian *IF*.

2.1.2 Sistem Pakar

Salah satu bagian dari "kecerdasan buatan (AI)" yang akhir-akhir ini mengalami perkembangan pesat adalah Sistem Pakar (*expert system*), yaitu suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah, Sistem pakar (*expert system*) adalah kerja komputer yang berkemampuan untuk menyimpan pengetahuan dan aturan dari domain pakar yang khusus. Fungsi yang utama kerja ini adalah untuk memindahkan secara efektif kumpulan pengetahuan kepada mereka yang bukan pakar. Sistem Pakar akan memberikan pemecahan suatu masalah ketika pemakai melakukan konsultasi atau dialog kepada sistem atau program komputer. Dengan bantuan sistem pakar, seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah, serta mengambil

keputusan yang biasanya dilakukan oleh seorang pakar (Sasongko, Jati., 2007: 37-50).

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 (Sutojo, dkk., 2011: 159-160).

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 (Hartati dan Iswanti, 2008: 45-47).

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 perunutan dalam konsep ini berlawanan dengan *forward chaining*. Konsep ini dapat dimodelkan sebagai berikut:

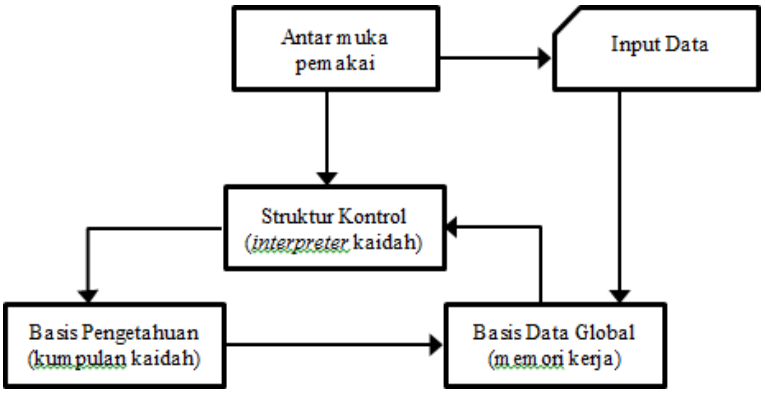
Tujuan,

IF (kondisi)

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 (Hartati dan Iswanti, 46-47).

1. *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 (Kusrini, 2006: 19).



Gambar 2.1 Struktur Sistem Pakar Kaidah Produksi
(Sumber: Firebaugh, 1988 *dalam* Hartati dan Iswanti, 2008: 10)

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* Hartati dan Iswanti, 2008: 25):

1. *IF* premis *THEN* konklusi
2. *IF* masukan *THEN* keluaran
3. *IF* kondisi *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan

7. *IF* aksi *THEN* reaksi
8. *IF* gejala *THEN* diagnose

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 sehingg 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) (Hartati dan Iswanti, 2008: 25-26).

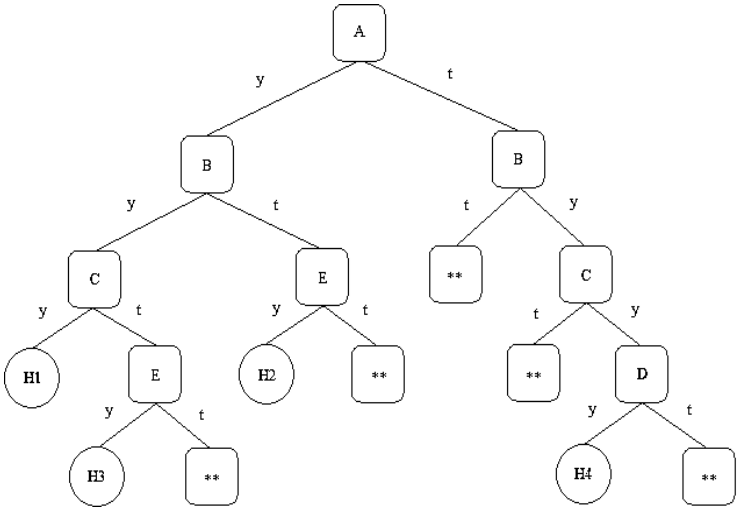
Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk tabel keputusan kemudian dibuat pohon keputusannya. Berikut ini adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan (Hartati dan Iswanti, 2008: 26-39).

Tabel 2.1 Tabel Keputusan

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

Sumber: Hartati dan Iswanti (2008: 32)

Keterangan:



Gambar 2.2 Pohon Keputusan
(Sumber: Hartati dan Iswanti, 2008: 33)

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"

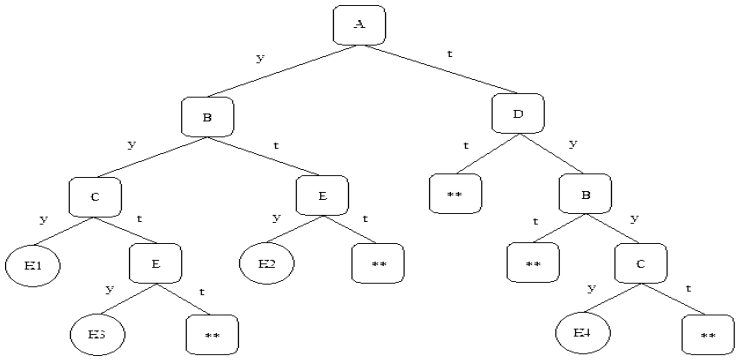
maka sistem akan menanyakan *evidence* B. Ini berarti jawaban pengguna tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.2.

Tabel 2.2 Alternatif Tabel Keputusan

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

Sumber: Hartati dan Iswanti (2008: 34)

Berdasarkan tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



Gambar 2.3 Alternatif Pohon Keputusan
(Sumber: Hartati dan Iswanti, 2008: 35)

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 Kusrini (2006: 14) 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, dkk., 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.

2.2 Variabel

Variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya (Sudaryono,2015). Variabel pada penelitian ini adalah sebagai berikut:

2.2.1 Trouble Shooting Hardware Komputer

Sebuah komputer terdiri dari komponen keras dan komponen lunak. Komponen keras atau *hardware* adalah komponen yang bisa dilihat oleh mata dan dipegang oleh tangan. Perangkat keras komputer adalah komponen penyusun komputer. Pada umunya, perangkat keras ditancapkan ke *casing* komputer atau ditancapkan ke kabel melalui port. Jika ditancapkan ke kabel, perangkat keras ini juga sering disebut sebagai periferal (Penerbit Andi, 2014: 2).

Program ini akan membantu dalam menyelesaikan masalah-masalah yang berkaitan dengan kerusakan atau penyelesaian masalah perangkat keras (hardware) komputer secara cepat. Adapun Indikator yang terdapat dalam penelitian ini adalah sebagai berikut:

1. Power Supply

Catu daya atau *power supply unit* (PSU) adalah kompenen yang sangat penting pada komputer. Jika tidak ada PSU, komputer tidak dapat menyala. *Power Supply* akan mengkonversi daya *AC* dari PLN ke DC untuk kemudian dialirkan ke *motherbard* komponen komputer lainya. PC modern umumya memiliki PSU yang memiliki fitur *switched-mode.* Beberapa *power supply* memiliki selektor manual untuk voltase input, sementara lainnya bisa beradaptasi dengan input yang diberikan.

Mayoritas PC modern memiliki *power supply* yang kompatibel dengan spesifikasi ATX. Termasuk *form factor* dan toleransi voltasenya. Ketika *power supply* ATX dikoneksikan ke *supply* utama, arus listrik yang dihasilkan selalu memiliki voltase *standby 5 volt* digunakan untuk membuat fungsi standby pada komputer tetap berjalan. Voltase ini menyediakan sinyal ke *motherboard* untuk mengindikasikan bahwa voltase DC tersedia. Sehingga komputer bisa dinyalakan dan dibooting dengan aman (Penerbit Andi, 2014: 15).

1. Mengenal Jenis-Jenis *Power Supply*

Sampai saat ini, ada beberapa jenis *power supply* yang biasa digunakann di sebuah komputer. Berikut beberapa jenis *power supply* dan sedikit penjelasannya :

1. Power Supply AT

Power Supply dengan jenis ini adalah *power supply* yang mepunyai kabel power terpisah menjadi dua untuk dipasangkan ke motherboard. Konektor yang pertama adalah P8 dan yang ke dua P9. Jika anda menggunakan *power supply* jenis ini, anda harus berhati-hati agar tidak terjadi kesalahan pemasangan. Cara yang benar adalah kabel power berwarna hitam dari kabel power P8 dan P9 harus berremu di tengah-tengah jika kedua konektor tersebut disatukan. Pada *power supply* jenis ini, tombol on-off langsung dihubungkan pada casing komputer. Jadi, untuk menyalakan atau mematikan, anda harus menekan tombol on-off tersebut. Pada *power supply* jenis ini banyak digunakan di komputer Pentium II dan Pentium III, dan sekarang jarang sekali orang yang menggunakan *power supply* jenis ini *Power Supply AT* (Penerbit Andi, 2014: 40).

Gambar 2.4 *Power Supply AT*
(Sumber: Penerbit Andi, 2014: 40)

2. Power Supply ATX

Power Supply jenis ini adalah penyempurnaan dari *power supply* jenis AT. Di jenis ATX, kabel power yang dihubungkan ke motherboard sudah menjadi satu bagian yang berjumlah 20 PIN dan disebut dengan ATX 20 PIN. Jadi, dalam pemasangannya sangat jarang terjadi kesalahan. *Power Supply* jenis ini juga sudah dilengkapi dengan *power switch* atau *auto-shutdown*. Jadi, untuk mematikan komputer, anda hanya perlu menggunakan perintah *shutdown* pada komputer saja tanpa harus menekan tombol on-off. (Penerbit Andi, 2014: 42).

Gambar 2.5 *Power Supply ATX*
(Sumber: Penerbit Andi, 2014: 41)

3. Power Supply modular

Power Supply jenis ini merupakan inovasi baru dalam pemasangan kabel. *Power Supply modular* memungkinkan pengguna untuk menghilangkan kabel yang tidak diperlukan dalam pemasangan CPU. Kabel-kabel yang digunakan di *power supply* jenis ini bersifat tidak permanen sehingga bisa dilepas pasang sesuai kebutuhan pengguna (Penerbit Andi, 2014: 42).

Gambar 2.6 *Power Supply Modular*
(Sumber: Penerbit Andi, 2014: 42)

4. Power Supply non modular

Power Supply non-modular adalah yang banyak digunakan dari zaman Pentium II sampai sekarang. *Power Supply* jenis ini mempunyai kabel yang bersifat permanen (Penerbit Andi, 2014: 43).

Gambar 2.7 *Power Supply Non Modular*
(Sumber: Penerbit Andi, 2014: 43)

2. Kerusakan *Power Supply*

Power Supply adalah satu-satunya komponen komputer yang membagikan daya listrik ke semua komponen lain. Jika *power supply* ini membagikan mengalami kerusakan pastinya aliran listrik tidak akan

terpenuhi. Anda bisa mengecek kabel input atau output yang ada di *power supply* untuk memastikan PSU (*power supply unit*) bekerja dengan baik (Penerbit Andi, 2014: 45).

2. Motherboard

Motherboard adalah sebuah papan sirkuit yang digunakan di perangkat komputer yang menampung berbagai komponen elektronik saling berhubungan, seperti chip BIOS, soket, konektor, dan berbagai jalur untuk koneksi ke masing-masing perangkat. *Motherboard* merupakan komponen utama dari perangkat komputer disatukan dan membuat komponen-komponen tersebut bekerja sama sehingga komputer bekerja dengan baik. Banyak sekali gejala kerusakan yang terjadi jika *motherboard* tidak bekerja dengan baik. Berikut ini beberapa gejala *motherboard* tidak bekerja dengan baik :

1. Komputer menjadi lambat.
2. Transfer file (copy-paste) sering gagal.
3. Hang saat menjalankan lebih dari 3 aplikasi pada waktu bersamaan.
4. Terdengar suara aneh saat memainkan musik.
5. Terjadi *Blue Screen On Death (BSOD).*
6. Kegagalan CMOS.
7. Tidsk bisa booting.

8. Kapasitas *harddisk* menjadi tidak normal.
9. Tampilan huruf di layar berkedip dan muncul garis.
10. Komponen *hardware* lain menjadi tidak berfungsi.
11. Arus listrik tidak stabil (Penerbit Andi, 2014: 45).

Gambar 2.8 *Motherboard*
(Sumber: Penerbit Andi, 2014: 23)

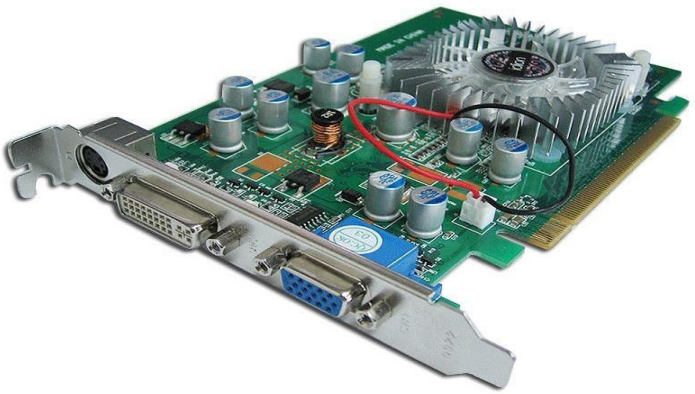
3. Harddisk

Harddisk drive (HDD) atau sering disebut harddisk, adalah sebuah perangkat keras (*hardware)* untuk menyimpan informasi digital dan data-data yang di komputer. *Harddisk* digunakan sebagai penyimpanan utama pada sebuah komputer , data yang tersimpan di *harddisk* bisa bersifat *long-term*(jangka panjang). Sampai saat ini, *harddisk* sudah berkembang sangat pesat dari segi ukuran fisik, daya tampung penyimpanan, kecepatan, dan harga (Penerbit Andi, 2014: 56).

Gambar 2.9 *Harddisk*
(Sumber: Penerbit Andi, 2014: 51)

4. VGA

Video Graphics Adapter (VGA) *card* adalah salah satu komponen keras komputer (*hardware*) yang berguna untuk menampilkan output gambar dan teks yang dihasilkan oleh komputer ke dalam layar monitor. Tanpa adanya VGA card ini, layar monitor anda tidak akan ada gambarnya atau "*blank*". VGA *card* berbentuk seperti kartu yang tertanam di *motherboard PC* (Penerbit Andi, 2014: 81).



Gambar 2.10 VGA Card
(Sumber: Penerbit Andi, 2014: 77)

5. DVD ROM

Optical Drive adalah perangkat keras yang berfungsi untuk membaca dan menulis data pada kepingan DVD/CD. *Optical Drive* menggunakan sinar laser atau gelombang elektromagnetik untuk melakukan tugasnya (Penerbit Andi, 2014: 85).

Gambar 2.11 DVD-ROM
(Sumber: Penerbit Andi, 2014: 85)

6. RAM

Memori akses acak atau yang biasa disebut *Random access memory (RAM)* adalah sebuah tipe penyimpanan komputer yang isinya dapat diakses dalam waktu yang tetap tidak memperdulikan letak data tersebut dalam memori. Ini berlawanan dengan alat memori urut, seperti tape magnetik, disk dan drum, di mana gerakan mekanikal dari media penyimpanan memaksa komputer untuk mengakses data secara berurutan. RAM biasanya digunakan untuk penyimpanan primer (memori utama) dalam komputer untuk digunakan dan mengubah informasi secara aktif, meskipun beberapa alat menggunakan beberapa jenis RAM untuk menyediakan penyimpanan sekunder jangka-panjang. (Penerbit Andi, 2014: 85).

Gambar 2.12 RAM
(Sumber: Penerbit Andi, 2014: 85)

2.3 *Software* Pendukung

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

2.3.1 Unified Modeling Language (UML)

Pada perkembangan teknik pengumpulan pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unifed 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 A.S dan M. Shalahuddin, 2011:118).

2.3.1.1. Use Case Diagram

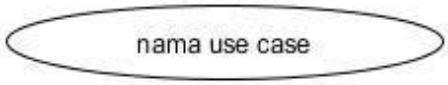
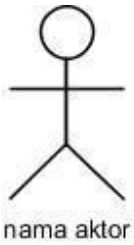
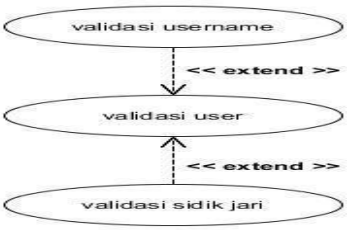
Use case atau diagram *use* case merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. *Use* case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa A.S dan M. Shalahuddin, 2011:130).

Menurut Rosa A.S dan M. Shalahuddin (2011:131) Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case*yaitu pendefinisian apa yang disebut aktor dan *use case*.

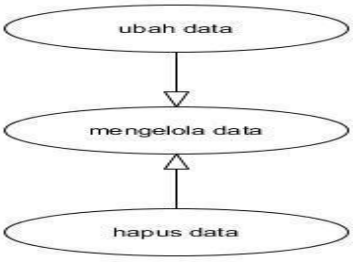
1. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use Case* merupakan fungsionalitas yang disediakan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

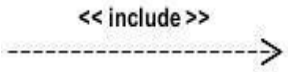
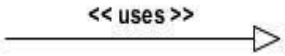
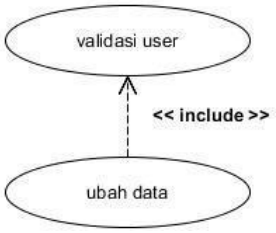
Berikut adalah simbol-simbol yang ada pada diagram *use case* (Rosa A.S dan M. Shalahuddin, 2011: 131-133):

Tabel 2.3 Simbol *Use Case Diagram*

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama *use case*
Aktor / *Actor* 	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 alah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / *association*	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor
Ekstensi / *extend* 	*Case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misal arah panah mengarah pada *use case* yang ditambahkan
	
Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya

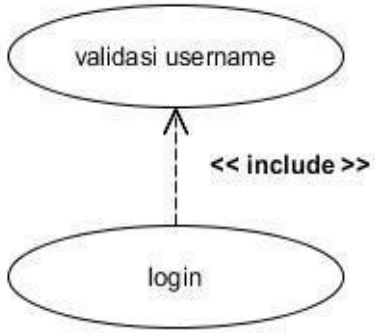
Tabel 2.3 Lanjutan

	 Arah panah mengarah pada *use case* yang menjadi generalisasinya (umum)

Menggunakan / *include / uses*  	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini Ada dua sudut pandang yang cukup besar mengenai *include* di *use case*: 1. *Include* berarti *use case* yang ditambahkan akan selalu di panggilsaat *use case* tambahan dijalankan misal pada kasus berikut: 2. *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus  berikut :

Tabel 2.3 Lanjutan

	Kedua interprestasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan

	interprestasi yang dibutuhkan. 

Sumber: Rosa A.S dan M. Shalahuddin, 2011: 131-133)

2.3.1.2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. 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 A.S dan M. Shalahuddin, 2011: 134).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa A.S dan M. Shalahuddin, 2011: 134):

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 A.S dan M. Shalahuddin, 2011: 134-135):

Tabel 2.4 *Activity Diagram*

Simbol	Deksripsi
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 */ decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan */ join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memilih sebuah status akhir

Tabel 2.4 Lanjutan

Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Rosa A.S dan M. Shalahuddin (2011: 134-135)

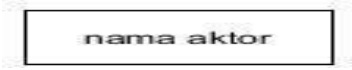
2.3.1.3 Sequence Diagram

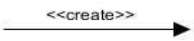
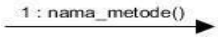
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeksripsikan waktu hidup objek dan *message* yang dikirmkan dan diterima antarobjek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* berserta metode-metode yang dimiliki kelas yang diinstasiasi menjadi objek itu (Rosa A.S dan M. Shalahuddin, 2011: 137).

Banyaknya diagram sekuen yan 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 sehinggan semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (Rosa A.S dan M. Shalahuddin, 2011: 138).

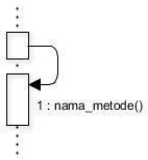
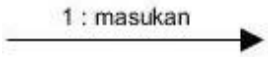
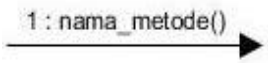
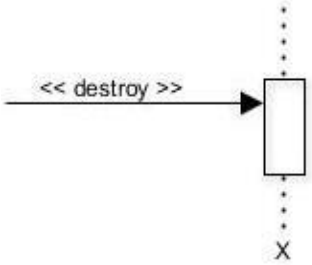
Berikut simbol-simbol yang ada pada diagram sekuen (Rosa A.S dan M. Shalahuddin, 2011: 138-139):

Tabel 2.5 *Sequence Diagram*

Simbol	Deksripsi
Aktor  Atau  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 alah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Garis hidup */ lifeline* 	Menyatakan kehidupan suatu objek

Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe *create* 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe *call* 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,

Tabel 2.5 Lanjutan

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

Sumber: Rosa A.S dan M. Shalahuddin (2011: 138-139):

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.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas sebagai berikut (Rosa A.S dan M. Shalahuddin, 2011: 122):

1. Kelas *main*

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case*

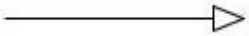
   Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

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 A.S dan M. Shalahuddin, 2011: 123-124):

Tabel 2.6 *Diagram Class*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / *interface* 	Sama dengan konsep *interface* dalam pemrograman beorientasi objek
Asosiasi / *association*	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicy*
Asosiasi Berarah / *directed association* 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi 	Relasi antarkelas dengan makna generalisasi=spesialisasi (umum-khusus)
Kebergantungan / *dependency* 	Relasi antarkelas dengan makn generalisasi-spesialisasi (umum-khusus)
Agregasi / *aggregation* 	Relasi antarkelas dengan makna semua-bagian (*whole-part*).

Sumber: Rosa A.S dan M. Shalahuddin (2011: 123)

2.3.2 XAMPP (*XApache MySQL PHP Perl*)

Menurut Sidik dan Pohan (2009: 6) *server web* adalah komputer yang digunakan untuk menyimpan dokumen-dokumen *web* yang akan melayani permintaan dokumen *web* dari kliennya. *XAMPP* adalah sebuah perangkat lunak *web server apache* yang didalamnya sudah tersedia *databaseserver MySQL* dan

dapat mendukung pemrograman *PHP*.*XAMPP* merupakan *software* yang mudah digunakan , gratis, dan mendukung instalasi di *Linux* dan *Windows*.

Gambar 2.13 Logo *XAMPP*
(Sumber: https: Aplikasi *XAMPP*)

2.3.3 PHP: Hypertext Preprocessor(PHP)

PHP adalah proyek *open source*, yang berarti memiliki akses ke kode sumber dan tidak membutuhkan biaya untuk menggunakan, mengubah, dan mendistribusikannya. Awalnya kepanjangan dari *PHP* adalah *Personal Home Page*, tetapi mengalami perubahan sejalan dengan konvensi penamaan *GNU* rekursif (*GNU = Gnu's Not Unix*) dan sekarang kepanjangan dari *PHP*adalah*PHP*: *Hypertext Preprocessor* (Welling dan Thomson, 2009: 3)

Gambar 2.14 Logo *PHP*
(Sumber: Welling dan Thomson, 2009: 3)

2.3.4 HTML

HTML merupakan singkatan dari *Hyper Text Markup Language. HTML* bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website*. Menurut sumber yang penulis kutip dari *Wikipedia, HTML* digunakan 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 (Agus Saputra, 2012:1).

Gambar 2.15 Logo *HTML*
(Sumber: Agus Saputra, 2012:1)

2.3.5 CSS (Cascading Style Sheet)

CSS merupakan bahasa pemrograman web yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam *web*lebih rapi, terstruktur dan seragam. *CSS* merupakan salah satu pemrograman wajib *html* yang harus dikuasai oleh para setiap pemrograman *web*, terlebih lagi itu adalah *Web Designer*.

Gambar 2.7 Logo *CSS*
(Sumber: Agus Saputra, 2012:27-29)

Tujuan utama dari *CSS* adalah untuk memisahkan konten utama dengan tampilan dokumen lainnya. *Web* yang menggunakan *CSS* akan lebih ringan dan mudah untuk dibuka dibandingkan dengan *web* tidak menggukan *CSS*. Dengan menggunakan *CSS*, akan banyak keuntungan yang dapat kita peroleh, diantaranya (Agus Saputra, 2012:27-29):

1. Memisahkan pembuatan dokumen (CSS dan HTML).
2. Mempermudah dan mempersingkat pembuatan dan pemeliharaan dokumen web.
3. Akses web lebih cepat saat di-loading (mempercepat pembacaan HTML).
4. Fleksibel, interaktif, tampilan lebih menarik dan nyaman dipandang.
5. Lebih kecil ukuran file sehingga *bandwith* yang digunakan juga otomatis menjadi lebih kecil.
6. Dapat digunakan pada semua *web browser*.

2.3.6 JavaScript dan jQuery

Menurut Sidik dan Pohan (2009: 267) *JavaScript* merupakan modifikasi dari bahasa *C++* dengan pola penulisan yang lebih sederhana.

Gambar 2.17 Logo *JavaScript*
(Sumber: Sidik dan Pohan, 2009: 267)

Beberapa hal penting dalam *JavaScript* adalah (Sidik dan Pohan, 2009: 267):

1. Menggunakan blok awal "{" dan blok akhir "}"
2. *Automatic conversion* dalam pengoperasian tipe data yang berbeda
3. *Sensitive case*, yaitu membedakan antara huruf kecil dan huruf capital sehingga harus berhati-hati dalam menggunakan nama variabel , fungsi, dan lain-lain.
4. *Extension* umumnya menggunakan "*.js"
5. Setiap *statement* dapat diakhiri tanda baca *semi colon* (;) dapat juga tidak
6. Jika tidak didukung oleh *browser* versi lama, *script* dapat disembunyikan diantara *tag* "<!--" dan "-->"
7. Jika program dalam satu baris terlalu panjang dapat disambung ke baris berikutnya menggunakan karakter *backslash* (\)

2.3.7 MySQL

MySQL adalah *RDBMS* (*Relational Database Management System*) yang sangat cepat dan kuat. Sebuah *database* memungkinkan untuk menyimpan,

mencari, mengurutkan, dan mengambil data secara efisien. *MySQL server* bertugas mengendalikan akses ke data yaitu untuk memastikan bahwa beberapa pengguna dapat bekerja dengan data-data itu secara konkuren, untuk memberikan akses yang cepat terhadap data, dan memastikan bahwa hak akses diberikan hanya kepada orang yang berwenang. *MySQL* adalah *multiuser* dan *multithread server* yang menggunakan *SQL (Sturctured Query Language*) sebagai standar bahasa pengelolaan *database* (Welling dan Thomson, 2009: 3).

Gambar 2.9 Logo *MySQL*
(Sumber: Welling dan Thomson, 2009: 3)

MySQL merupakan salah satu database kelas dunia yang sangat cocok bila dipa dukan dengan bahasa pemrograman PHP. MySQL bekerja menggunakan bahasa SQL (*Structure Query Language*) yang merupakan bahasa standar yang digunakan untuk manipulasi database. Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus) (Agus Saputra, 2012:77-78).

2.3.8 Notepad++

Gambar 2.10 Logo *Notepad++*
(Sumber: Gilmore ,2010: 36)

Notepad++ merupakan editor teks *open source* yang matang dan diakui sebagai pengganti *Notepad*.*Notepad++* tersedia untuk platform *Windows* yang dapat digunakan untuk menulis kode dengan beberapa pilihan bahasa (pemrograman).*Notepad++* menawarkan beragam kenyamanan fitur yang diharapkan dari setiap kemampuan *IDE*(*Integrated Development Environment*), termasuk kemampuan untuk menunjukkan baris tertentu dari suatu dokumen sebagai referensi yang mudah; sintaks, tanda kurung, *indentation highlighting*, fasilitas pencarian yang tangguh, *macro recording* untuk tugas-tugas seperti memasukkan *template* komentar, dan sebagainya. Salah satu kelebihan *Notepad++* adalah dukungan dasar untuk *auto-completion* dari nama fungsi yang ditawarkan sehingga akan mengurangi beberapa proses pengetikan kode (Gilmore ,2010: 36).

2.4 Penelitian Terdahulu

Sebagai bahan pertimbangan dalam penelitian ini, maka penulis mencantumkan beberapa penelitian yang diambil dari beberapa jurnal ilmiah, yaitu:

1. Nama Jurnal : Perancangan Aplikasi Sistem Pakar Diagnosa Kerusakan Hardware Komputer Metode Forward Chaining

   Nama Penulis : Jati Sasongko

   Volume / ISSN : Vol.3 September 2016, pp. 219~233 / ISSN: 2355-6579

   Kesimpulan : Dengan adanya Sistem Pakar Troubleshooting PC (Personal Computer), sangat membantu bagi pengguna komputer yang ingin memperbaiki kerusakan secara mandiri. Sistem Pakar Trouble Shooting PC (Personal Computer) ini sangat praktis bagi pengguna yang ingin mengembangkan bakat menjadi seorang teknisi yang handal. Karena disertai manual perbaikan komputer yang cukup mudah untuk dipraktekkan.

2. Nama Jurnal : Perancangan Sistem Pakar Troubleshooting Personal Computer

   Nama Penulis : Ali Akbar Rismayadi

   Volume / ISSN : XII, No.1, Januari 2007 : 37-50 / ISSN : 0854- 9524

   Kesimpulan : Pembangunan aplikasi sistem pakar diagnosa kerusakan hardware komputer ini dilakukan dengan menggunakan model inferensi forward chaining. Diagnosa kerusakan hardware komputer ini dapat dikembangkan dengan penggunaan aplikasi berbasis android untuk memudahkan identifikasi kerusakan hardware bagi masyarakat umum. Dari hasil implementasi Masih terdapat

kekurang dimana dari 24 hasil pengujian hanya 17 yang mendapatkan hasil seusia dengan yang di harapkan. Dan sebanyak 7 pengujian mendapatkan hasil yang belum valid.

3. Nama Jurnal : Sistem Pakar Kerusakan pada Perangkat Keras (Hardware) di SMA Negeri 11 Kabupaten Tangerang

   Nama Penulis : Joko Dwi Raharjo, M. Sofjan, Eksas Sugama

   Volume / ISSN : ISSN : 2088 – 1762 Vol. 4 No. 1 / Maret 2014

   Kesimpulan : Sistem penanganan yang berjalan saat ini jika terjadi kerusakan hardware komputer di SMA Negeri Kabupaten Tangerang sudah cukup baik karena dalam penanganannya maintanance cek keruakan dengan satu persatu hardware, dan hasil kerusakan dicatat untuk laporan riwayat hardware tersebut menggunakan aplikasi MS. excel. Pencatatan dilakukan berdasarkan kerusakan yang terjadi. Selanjutnya diperoses sebagai bahan evaluasi agar tidak terjadi kesalahan yang sama pada waktu berikutnya.

4. Nama Jurnal : Sistem Pakar Diagnosis Kerusakan Pada Televisi (Tv) Tabung Menggunakan Metode Forward Chaining

   Nama Penulis : Maisyaroh dan Ridwan Septianto

   Volume / ISSN : Vol. XIII, No. 2 September 2016

   Kesimpulan : Dari hasil penelitian di lapangan, proses diagnosis kerusakan televisi berwarna masih dilakukan secara manual dan tidak adanya panduan resmi yang menjadi acuan. “Sistem Pakar Diagnosis Kerusakan Pada Televisi Tabung”

yang berbasis aplikasi desktop merupakan penyelesaian dari masalah yang terjadi saat ini dalam hal proses diagnosis kerusakan pada televisi berwarna model tabung.

5. Nama Jurnal : Sistem Pakar Identifikasi Kerusakan Pada Mobil
   Nama Penulis : Ramadiani dan Nurbasar
   Volume / ISSN : Vol 6 No. 1 Febuari 2011 No. 29
   Kesimpulan : Perangkat lunak ini menampilkan informasi mengenai kategori kerusakan, jenis kerusakan, ciri kerusakan, mesin inferens, solusi, dan daftar istilah kerusakan otomotif khususnya mobil. Sistem pakar yang telah dibangun ini akan sangat membantu kelancaran bagi pengguna atau pemilik roda empat, yang belum mengetahui tentang kerusakan-kerusakan pada mobil. Kemudian selian itu juga pemilik mobil dapat mengetahui istilah-istilah dari automotif.

6. Nama Jurnal : Sistem Pakar Kerusakan Hardware Komputer Dengan Metode Forward Chaining (Studi Kasus: Benhur Sungai Penuh)
   Nama Penulis : Nency Extise Putri
   Volume / ISSN : Vol.18 No.2 Agustus 2016 / 1693-752X
   Kesimpulan : Dengan penggunaan aplikasi ini, user dapat mengetahui secara cepat kerusakan komputernya. Pengolahan kerusakan hardware komputer dengan sistem pakar ini menghasilkan ringakasan-ringkasan tentang

kerusakan,gejala, dan solusi sehingga mempercepat user megatasi masalah yang terjadi. Dan Aplikasi yang dibuat untuk memudahkan para user dalam mengatasi kerusakan hardware.

7. Nama Penulis : Toto Sudiyanto

   Nama Jurnal : Sistem Pakar Untuk Mendeteksi Kerusakan Hardware Komputer Dengan Menggunakan Metode Penelusuran Backward Chaining

   Volume / ISSN : Vol 1-No. 1 Edisi Juli 2011

   Kesimpulan : Dari berbagai macam percobaan yang telah dilakukan dapat ditarik kesimpulan bahwa dengan teknik backward chaining (pelacakan kebelakang) dapat memecahkan masalah dengan pernyataan objek yang cukup banyak dan ini memberi kemudahan bagi orang awam yang sekalipun dalam penggunaan sistem pakar tersebut tidak menghadirkan langsung pakar yang bersangkutan, dan yang menjadi dasar dari pembuatan sistem ini yaitu: Keterbatasan teknisi, Tidak semua user mengerti tentang troubleshooting hardware komputer dan Biaya konsultasi ke pakar yang relatif mahal dan keterbatasan waktu

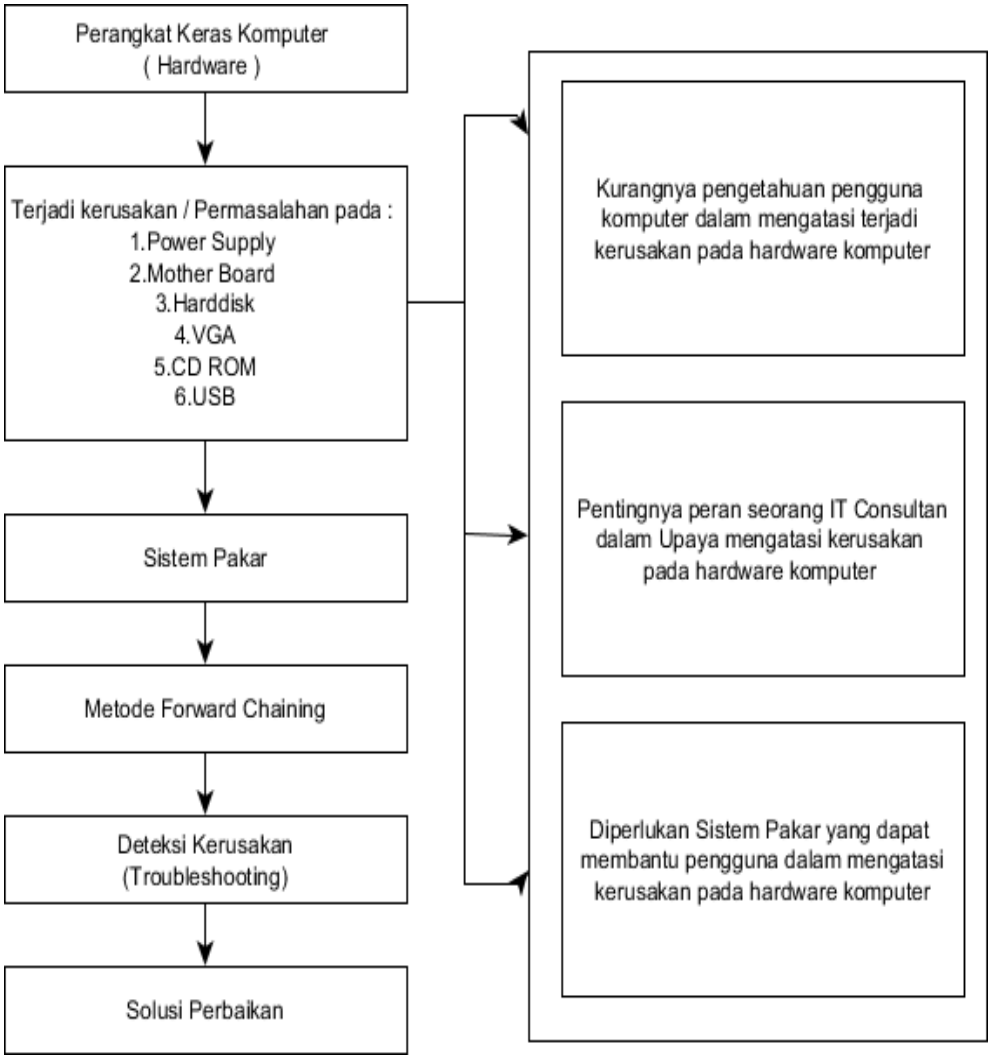
2.4 Kerangka Pemikiran

Menurut Hamdi dan Bahrudin (2014:32) melalui uraian dalam kerangka berpikir, peneliti dapat menjelaskan secara *konprehensif* variabel-variabel apa saja yang dan diteliti dan dari teori apa variabel-variabel itu diturunkan, serta mengapa variabel-variabel itu saja yang diteliti. Uraian dalam kerangka berpikir harus mampu menjelaskan dan menegaskan secara *komprehensif* asal-usul variabel yang diteliti, yang sinyalemennya telah dikemukakan dalam rumusan masalah dan identifikasi masalah semakin jelas asal-usulnya.

Dengan demikian, uraian atau paparan yang harus dilakukan dalam kerangka berpikir adalah paduan antara asumsi-asumsi teoritis dan asumsi-asumsi logika dalam menjelaskan atau memunculkan *variabel-variabel* yang diteliti serta bagaimana kaitan diantara variabel-variabel tersebut, ketika dihadapkan pada kepentingan untuk mengungkapkan fenomena atau masalah yang diteliti. Didalam menulis kerangka berpikir, ada 3 kerangka yang perlu dijelaskan, yakni: **kerangka teoritis, kerangka konseptual, dan kerangka operasional.**

1. Kerangka *Teoritis* adalah uraian yang menegaskan tentang teori apa saja yang dijadikan landasan serta asumsi-asumsi teoritis yang mana dari teori tesebut yang akan digunakan untuk menjelaskan fenomena yang diteliti.
2. Kerangka *Konseptual* adalah uraian yang menjelaskan konsep-konsep apa saja yang terkandung dalam asumsi-asumsi teoritis yang akan digunakan untuk *mengabstraksikan* (mengistilahkan) unsur-unsur yang terkandung didalam fenomena yang akan diteliti dan bagaimana hubungan di antara konsep-konsep tersebut

3. Kerangka *Operasional* adalah penjelasan tentang variabel-variabel apa saja yang diturunkan dari konsep-konsep terpilih tadi dan bagaimana hubungan di antara variabel-variabel tersebut, serta hal-hal apa saja yang dijadikan indikator untuk mengukur variabel-variabel tersebut.



Gambar 2.20 Kerangka Pemikiran
(Sumber: Data Penelitian,2018)

Dari Gambar 2.20 Kerangka Pemikiran, penelitian dimulai dari Perangkat keras komputer (*Hardware*) dimana terdapat beberapa permasalahan yang telah dapat di identifikasiakan yaitu:

1. Kurangnya pengetahuan pengguna komputer dalam mengatasi terjadi kerusakan pada hardware komputer.
2. Pentingnya peran seorang IT *Consultan* dalam upaya mengatasi kerusakan pada hardware komputer.
3. Diperlukan sebuah sistem pakar yang dapat membantu pengguna komputer dalam mengatasi kerusakan pada hardware komputer.

Dari identifikasi masalah yang telah didapatkan, dibuatlah Sistem pakar dengan menggunakan Metode *Forward Chaining*. Dimana Metode *Forward Chaining* merupakan penalaran dimulai dari fakta-fakta terlebih dahulu untuk menguji kebenaran hipotesis. Dari fakta-fakta yang telah ditentukan didapatkan jenis kerusakan yang terjadi pada *hardware* komputer, kemudian dari jenis tersebut diberikan solusi/saran agar kerusakan tersebut dapat diatasi.