

BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

Agar penelitian dapat berjalan dengan baik, maka diperlukan landasan bagi jalannya penelitian berupa teori-teori yang telah ada. Dalam penelitian ini, akan dijelaskan secara singkat tentang kecerdasan buatan dan jenis-jenis cabang ilmu Kecerdasan Buatan, yaitu Jaringan Saraf Tiruan, Sistem Pakar dan Logika Fuzzy.

2.1.1. Kecerdasan Buatan (Artificial Intelligence)

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud disini merujuk pada mesin yang mampu berfikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan manusia. Alan Turing, ahli matematika berkebangsaan Inggris yang dijuluki bapak komputer modern dan pembongkar sandi Nazi dalam era Perang Dunia II 1950, menetapkan definisi Artificial Intelligence “Jika komputer tidak dapat dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan (Sutojo et al., 2011: 3).

Cerdas berarti memiliki pengetahuan, pemahaman dan penalaran untuk membuat keputusan dan mengambil tindakan untuk membuat sebuah mesin

menjadi cerdas (dapat bertindak seperti manusia) maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar. Kecerdasan buatan memungkinkan komputer untuk berfikir 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.

Menurut John Mc Carthy dari Stanford dalam (Sutojo et al., 2011: 2) mendefinisikan kecerdasan sebagai “kemampuan untuk mencapai sukses dalam menyelesaikan suatu permasalahan lebih jauh lagi, berikut adalah beberapa definisi mengenai kecerdasan buatan yang dapat diketahui, yaitu (Sutojo, 2011: 2):

1. Herbert Alexander Simon (June 15, 1916-February 9, 2001):

Kecerdasan buatan (Artificial Intelligence) merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

2. Rich and Knight (1991):

Kecerdasan buatan (AI) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

3. Encyclopedia Britannica:

Kecerdasan buatan (AI) merupakan cabang ilmu komputer yang dalam merepresentasi pengetahuan lebih banyak menggunakan bentuk simbol-

simbol daripada bilangan dan memproses informasi berdasarkan metode heuristik atau dengan berdasarkan sejumlah aturan.

4. Menurut Winston dan Prendergast (1984), tujuan kecerdasan buatan adalah:
 - a. Membuat mesin menjadi lebih pintar (tujuan utama)
 - b. Memahami apa itu kecerdasan (tujuan ilmiah)
 - c. Membuat mesin lebih bermanfaat (tujuan entrepreneurial)

Berdasarkan definisi ini, maka kecerdasan buatan menawarkan media maupun uji teori tentang kecerdasan. Teori-teori ini nantinya dapat dinyatakan dalam bahasa pemrograman dan eksekusinya dapat dibuktikan pada komputer nyata. Dari sini dapat dikatakan bahwa: cerdas adalah memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Jadi, agar mesin bisa cerdas (bertindak seperti manusia) maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar (Sutojo et al., 2011: 2).

Jika dibandingkan dengan kecerdasan alami (kecerdasan yang dimiliki oleh manusia), kecerdasan buatan memiliki keuntungan komersial, antara lain Turban (1992) dalam (Sutojo et al., 2011: 10):

1. Kecerdasan buatan lebih bersifat permanen.
2. Kecerdasan buatan lebih mudah diduplikasi dan disebar.
3. Kecerdasan buatan lebih murah dibandingkan kecerdasan alami.
4. Kecerdasan buatan lebih bersifat konsisten.
5. Kecerdasan buatan dapat didokumentasi.
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami.

7. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibandingkan dengan kecerdasan alami.

Persoalan-persoalan yang ditangani oleh kecerdasan buatan makin lama makin berkembang sehingga memungkinkan bagi kecerdasan buatan untuk merambah ke bidang ilmu yang lain. Hal ini disebabkan karakteristik cerdas sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi (Sutojo et al., 2011: 12). Berikut beberapa lingkup kecerdasan buatan (Sutojo et al., 2011: 13):

1. Sistem pakar, adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan masalah.
2. Pengolahan Bahasa alami (*Natural language processing*), adalah pembuatan program yang memiliki kemampuan untuk memahami Bahasa manusia. Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin di komunikasikan antarmanusia. Bentuk utama representasinya adalah berupa suara atau ucapan (*Spoken language*), tetapi sering pula dinyatakan dalam tulisan.
3. Pengenalan ucapan (*Speech recognition*), adalah suatu pengembangan teknik dan sistem yang memungkinkan computer untuk menerima masukan berupa kata yang di ucapkan.
4. Robotika dan sistem sensor (*robotick and sensory system*), adalah ilmu pengetahuan dan teknologi rekayasa robot, desain, manufaktur, aplikasi, dan disposisi struktural.

5. *Computer vision*, adalah salah satu cabang pengetahuan yang mempelajari bagaimana computer mengenali objek yang diamati atau diobservasi. *computer vision* mencoba untuk menginterpretasikan gambar atau objek-objek terlihat melalui komputer dan mendeskripsikannya sehingga menghasilkan informasi yang berguna.
6. *Intelligen computer-aided instruction*, adalah komputer dapat digunakan sebagai tutorial yang dapat melatih dan mengajar.
7. *Game playing*, dalam kamus bahasa Indonesia, “*game*” adalah permainan, yaitu kegiatan yang kompleks yang didalamnya terdapat peraturan, *play*, dan budaya. *Game* di klasifikasikan dalam beberapa kelompok berikut:
 - a. *Board games* (Permainan papan)
 - b. *Card games* (Permainan kartu)
 - c. *Athletic games* (Permainan alletik)
 - d. *Children games* (Permainan anak)
 - e. *Computer games* (Permainan computer)
 - f. *Game online*

Selain tujuh subdisplin di atas adapun subdisiplin yang populer dalam kecerdasan buatan, yaitu *fuzzy logic* dan jaringan saraf tiruan:

1. *Fuzzy logic*, pertama kali diperkenalkan oleh Lotfi A.Zadeh, seorang propesor dari *Universitas of California*. *Fuzzy logic* memiliki derajat keanggotaan dalam rentang 0 (nol) hingga 1 (satu), berbeda dengan logika digital atau diskrit yang hanya memiliki dua nilai yaitu 1 (satu) atau 0 (nol). Logika fuzzy digunakan untuk menerjemahkan suatu besaran yang

diekspresikan menggunakan bahasa (*linguistic*). Himpunan fuzzy memiliki dua atribut, yaitu:

- a. *Linguistic*, yaitu nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami misalnya DINGIN, SEJUK, PANAS mewakili variabel temperatur.
 - b. *Numeris*, suatu nilai yang menunjukkan ukuran dari suatu variabel, misalnya 10, 35, 40, dan sebagainya (Sutojo et al., 2011:211).
2. Jaringan saraf tiruan adalah paradikma pengelola informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradikma 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 JST antara lain:
1. Belajar adaptive, kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
 2. *Self-organization*, sebuah JST dapat membuat organisasi sendiri atau representasi dari informasi yang di terimanya selama waktu belajar.
 3. *Real time operation*, perhitungan JST dapat dilakukan secara parallel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini (Sutojo et al., 2011: 284).

Selain mempunyai kelebihan tersebut, JST juga mempunyai kelemahan-kelemahan berikut:

- a. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi tinggi.
- b. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatik, operasi logika, dan simbolis.
- c. Untuk beroperasi JST butuh pelatihan sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama (Sutojo et al., 2011: 284).

2.1.2. Sistem Pakar (*Expert System*)

2.1.2.1. Definisi Sistem Pakar (*Expert System*)

Salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia adalah sistem pakar. Pemecahan masalah-masalah yang kompleks biasanya hanya dilakukan oleh seorang pakar. Sistem pakar mencoba mencari penyelesaian yang memuaskan, yaitu sebuah penyelesaian yang cukup bagus agar pekerjaan dapat berjalan walaupun itu bukan penyelesaian yang optimal (Hartati & Iswanti, 2008: 2). Beberapa definisi sistem pakar disajikan pada tabel 2.1

Tabel 2.1 Defenisi Sistem Pakar

Sumber	Definisi
Martin dan Oxman (1998)	Sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam pemecahan masalah, yang biasanya hanya dapat diselesaikan seorang pakar dalam bidang tertentu.
Iqnizio (1991)	Sistem pakar merupakan bidang yang didirikan oleh sistem berbasis pengetahuan (Knowledge base system), memungkinkan komputer dapat berpikir dan mengambil kesimpulan dari sejumlah kaidah.
Turban dan Aronson (2001)	Sistem yang menggunakan pengetahuan manusia yang dimasukkan kedalam komputer untuk dapat memecahkan masalah-masalah yang biasa dikerjakan oleh seorang pakar.
Giarrantano dan Riley (2005)	Salah satu cabang kecerdasan buatan yang menggunakan pengetahuan-pengetahuan khusus yang dimiliki oleh seorang ahli untuk menyelesaikan suatu masalah tertentu.

Sumber: (Hartati & Iswanti, 2008: 3)

2.1.2.2. Kategori Permasalahan Sistem Pakar

Biasanya aplikasi sistem pakar menyentuh beberapa area permasalahan berikut menurut (Sutojo et al., 2011: 162):

1. Interpretasi, menghasilkan deskripsi situasi berdasarkan data-data masukan.
2. Prediksi, memperkirakan akibat yang mungkin terjadi dari situasi yang ada.
3. Diagnosis, menyimpulkan suatu keadaan berdasarkan masalah-masalah yang diberikan (symptoms).
4. Desain, melakukan perancangan berdasarkan kendala-kendala yang diberikan.
5. *Planning*, merencanakan tindakan-tindakan yang akan dilakukan.
6. *Monitoring*, membandingkan hasil pengamatan dengan proses perencanaan.
7. *Debugging*, menentukan penyelesaian dari suatu kesalahan sistem.
8. *Reparasi*, melaksanakan rencana perbaikan.
9. *Instruction*, melakukan instruksi untuk diagnosis, debugging, dan perbaikan kinerja.
10. Kontrol, melakukan kontrol terhadap hasil interpretasi, diagnosis, debugging, monitoring, dan perbaikan tingkah laku sistem.

2.1.2.3. Manfaat dan Kekurangan Sistem Pakar

Menurut (Sutojo et al., 2011: 160) sistem pakar menjadi sangat populer karena sangat banyaknya kemampuan dan manfaat yang diberikan, di antaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.

2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan proses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional. Sistem pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespons dengan: “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawabannya.
10. Bisa digunakan sebagai media pelengkap dan pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya:

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.

3. Sistem pakar tidak 100% bernilai benar.

2.1.2.4. Komponen Sistem Pakar

Menurut Giarratano dan Riley (2005) dalam (Hartati & Iswanti, 2008: 3) sistem pakar sebagai sebuah program yang difungsikan untuk menirukan pakar manusia harus bisa melakukan hal-hal yang dapat dikerjakan oleh seorang pakar. Untuk membangun sistem yang seperti itu maka komponen-komponen yang harus dimiliki adalah sebagai berikut:

1. Antarmuka pengguna (*user interface*),
2. Basis pengetahuan (*knowledge base*),
3. Mekanisme inferensi (*inference machine*),
4. Memori kerja (*working memory*).

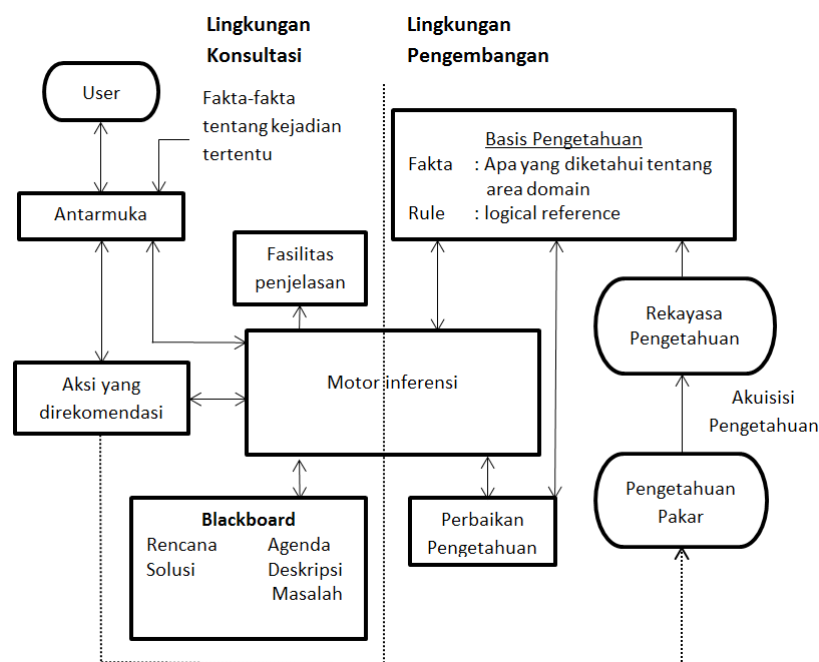
Sedangkan untuk menjadikan sistem pakar menjadi lebih menyerupai seorang pakar yang berinteraksi dengan pemakai, maka dilengkapi dengan fasilitas berikut:

1. Fasilitas penjelasan (*explanation facility*),
2. Fasilitas akuisisi pengetahuan (*knowledge acquisition facility*).

2.1.2.5.1. Struktur Sistem Pakar

Menurut (Sutojo et al., 2011:166) 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-komponen yang penting dalam sebuah sistem pakar.



Gambar 2.1 Komponen-komponen sistem pakar
Sumber: (Sutojo et al., 2011:166)

Keterangan:

1. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa di proses oleh computer dan menaruhnya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan). Sumber-sumber

pengetahuan bisa diperoleh dari pakar, buku, dokumen multimedia, basis data, laporan riset khusus, dan informasi yang terdapat di Web.

2. Basis Pengetahuan (*Knowledge Base*)

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

- a. Fakta, misalnya situasi, kondisi atau permasalahan yang ada.
- b. *Rule* (Aturan), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

3. 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.

4. Daerah Kerja (*Blackboard*)

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

- a. Rencana: bagaimana menghadapi masalah.
- b. Agenda: aksi-aksi potensial yang sedang menunggu untuk dieksekusi.
- c. Solusi: calon aksi yang akan dibangkitkan.

5. Antarmuka Pengguna (*User Interface*)

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

6. 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.

7. Sistem Perbaikan Pengetahuan (*knowledge refining system*)

Berfungsi 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 seperti itu diperlukan oleh program agar dapat menganalisa alasan-alasan kesuksesan dan kegagalannya dalam mengambil kesimpulan. Dengan cara ini basis pengetahuan yang lebih baik dan penalaran yang lebih efektif akan dihasilkan.

8. Pengguna (*User*)

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

2.1.2.6. Representasi Pengetahuan

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

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

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada

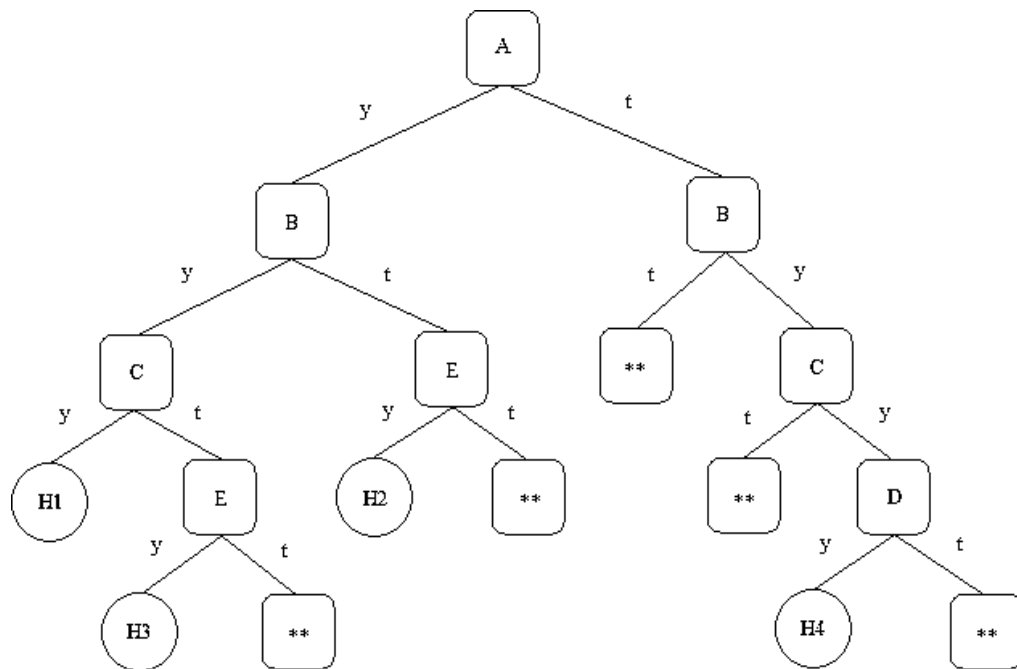
kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan (deteksi) (Hartati & Iswanti, 2008: 25).

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 & Iswanti, 2008: 26).

Tabel 2.2 Tabel Keputusan

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

Sumber: (Hartati & Iswanti, 2008: 32)



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

Keterangan:

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

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

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

D = *evidence* D, H4 = hipotesa 4

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

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

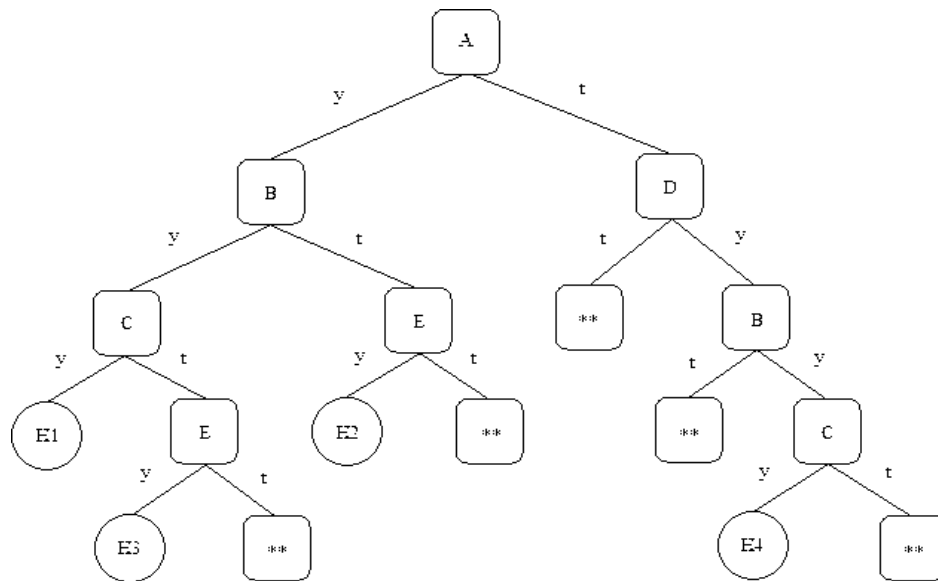
melihat pohon keputusan pada gambar 2.2 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki evidence A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak” maka sistem akan menanyakan evidence B. Ini berarti jawaban pengguna tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.3.

Tabel 2.3 *Alternatif Pohon Keputusan*

Hipotesa ence	Evid	Hipotesa 1		Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>		Ya		Ya	Ya	Tidak
<i>Evidence D</i>		Tidak		Tidak	Tidak	Ya
<i>Evidence B</i>		Ya		Tidak	Ya	ya
<i>Evidence C</i>		Ya		Tidak	tidak	Ya
<i>Evidence E</i>		Tidak		Ya	ya	Tidak

Sumber: (Hartati & Iswanti, 2008:34)

Berdasarkan tabel 2.3 dapat dihasilkan pohon keputusan sebagai berikut:



Gambar 2.3 Alternatif Pohon Keputusan
Sumber: (Hartati & 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.

2.1.2.7. Mesin Inferensi (*Inference Engine*)

Menurut (Hartati & Iswanti, 2008:5) mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, biasa dikatakan sebagai mesin pemikir (*Thinking Machine*). Pada prinsipnya mesin inferensi inilah yang akan mencari solusi dari suatu permasalahan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar, proses inferensi dilakukan dengan suatu modul yang disebut Mesin Inferensi (*Inference Engine*). Ada dua metode inferensi yang penting dalam sistem pakar, yaitu *Forward Chaining* (runut maju) dan *Backward Chaining* (runut mundur).

2.1.2.8 *Forward Chaining* (Runut Maju)

Menurut (Sutojo et al., 2011: 171) *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.

Konsep ini dapat juga disebut sebagai pencarian yang dimotori data (*data driven search*). Runut maju melakukan proses peruntutan (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 deteksi sehingga dapat dikatakan jalannya penalaran runut maju dimulai dari pengamatan menuju deteksi. 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 & Iswanti, 2008: 45).

2.2. Variabel

2.2.1. Sejarah Sepeda Motor

Sejarah motor sudah cukup lama sebenarnya, awalnya dimulai kira-kira akhir abad ke-19. Di zaman itu, orang-orang yang kreatif mulai mencoba

memasangkan mesin ke sepeda. Inovasi pun berlanjut hingga sekarang, hampir 100 tahun lebih. Hasilnya, motor yang super canggih sudah lazim ditemui, jauh lebih hebat dibandingkan kondisi awal sepeda motor di awal mulanya. Dua orang kreatif fan inovatif yang termasuk dalam golongan orang-orang pertama yang membuat sepeda motor adalah William S. Harley dan Arthur Davidson di Amerika Serikat. Harley dan Davidson ini nantinya akan mengembangkan sebuah sepeda motor yang sampai sekarang masih terkenal, yaitu Harley Davidson.

Ada lagi seorang insinyur berkewarganegaraan Jerman yang pada tahun 1902 mencoba melakukan eksperimen seputar sepeda motor. Nama insinyur ini adalah Mauritz Johann Schulte. Tetapi berbeda dengan Harley dan Davidson yang membuat pabrik sendiri, insinyur ini menjual hasil eksperimennya ke pabrik Triumph Cycle Co. Ltd yang ada di Inggris. Setelah kedua perintis tersebut, maka otomatis muncul banyak pabrik sepeda motor pada awal abad ke-20 di negara-negara maju. Namun, perkembangan pesat sepeda motor baru terjadi setelah masuknya motor hasil pabrik dari negeri Asia, yaitu Jepang. Jepang dengan pabrik sepeda motor Honda-nya mulai melakukan inovasi dengan cara membuat sepeda motor murah, irit, dan bandel mesinnya, untuk kebutuhan transportasi penduduk Jepang setelah perang dunia kedua. Hingga sekarang, tercatat banyak pabrik motor di dunia yang menjadikan sepeda motor Jepang sebagai contoh untuk mendesain sepeda motor mereka (Akbar Ali dan Eko P.U., 2016).

2.2.2. Asal Mula Tenaga

Untuk dapat meluncur di jalan raya, sepeda motor Suzuki Satria FU 150 MF memerlukan tenaga gerak untuk melawan gesekan ban dengan jalan, tahanan udara, dan gesekan antar bagian-bagian kendaraan yang berputar. Alat yang dapat membangkitkan daya gerak disebut penggerak utama, yang juga dikenal dengan nama mesin atau motor. Mesin sepeda motor Suzuki Satria FU 150 MF menghasilkan daya gerak apabila terjadi proses pembakaran antara campuran udara dan bensin di dalam mesinnya. Tenaga yang berasal dari pembakaran itulah yang digunakan mesin untuk menggerakkan sepeda motor. Mesin berfungsi untuk mengubah tenaga dari hasil pembakaran campuran udara dan bensin di dalam suatu ruang bakar menjadi tenaga gerak atau tenaga mekanik. Karena mesin itu menggunakan bahan bakar bensin maka disebut motor bensin (Marsudi, 2010: 1)

2.2.3. Prinsip Kerja sepeda motor Suzuki Satria FU 150 MF

Sepeda motor ini menggunakan kopling memiliki konstruksi yang terdiri dari katup hisap, katup buang, silinder, torak, cincin torak, batang torak, poros engkol, dan roda penerus. Didalam motor bensin 4-tak, campuran udara dan bensin dibakar untuk memperoleh tenaga panas. Jadi supaya motor bensin dapat bekerja, motor bensin 4-tak harus melakukan 4 proses kerja, yaitu:

1. Menghisap campuran udara dan bensin ke dalam silinder.
2. Mengompresikan gas campuran udara dan bensin agar diperoleh tekanan hasil pembakaran yang tinggi.

3. Meneruskan gaya tekan hasil pembakaran sedemikian rupa sehingga dapat diubah menjadi tenaga gerak.
4. Membuang gas sisa pembakaran dari ruang pembakaran.

Keempat proses diatas terjadi dalam suatu proses kerja yang disebut siklus. Jadi suatu siklus kerja motor bensin terdiri dari empat proses kerja yang mana masing-masing kerja dilakukan oleh langkah torak penuh. Karena satu siklus motor bensin terdiri dari empat langkah torak maka motor bensin ini disebut motor bensin 4 langkah atau motor bensin 4-tak (Marsudi, 2010: 1).

2.2.4. Bagian Utama Sepeda Motor Satria FU 150 MF

Menurut Bapak Teguh Wiyono selaku kepala mekanik di Bengkel showroom Suzuki Roda Makmur dan telah bersertifikat sebagai kepala teknisi (2003), ada tiga bagian utama Sepeda Motor Suzuki Satria FU MF, yaitu:

1. Komponen Mesin

Komponen yang terdapat pada bagian mesin

- a. *Bearing*
- b. *Piston*
- c. *Kampas kopling*
- d. *Head silinder*

2. Komponen Sistem Kelistrikan

Komponen yang terdapat pada bagian kelistrikan

- a. *Busy*

b. Spull

c. Kabel body

3. Komponen Sistem Injeksi

a. Injector

b. Pompa Injeksi

2.3. Software Pendukung

2.3.1. UML (*Unified Modelling Language*)

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

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

UML mendefinisikan diagram-diagram sebagai berikut:

1. *Use case diagram*

2. *Class diagram*

3. *Activity diagram*

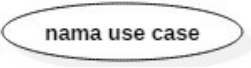
4. *Sequence diagram*

2.3.1.1. Use Case Diagram

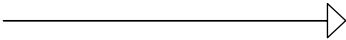
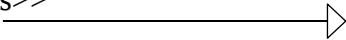
Menurut (A.S & Shalahuddin, 2011, p.130) Diagram *Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara garis besar, *use case* digunakan untuk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

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

Tabel 2.4 Simbol Use Case Diagram

Simbol	Deskripsi
<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/actor 	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/association 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
Ekstensi/extend <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu. Arah panah mengarah pada <i>use case</i> yang ditambahkan.

Tabel 2.4 Lanjutan

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

Sumber: (A.S, Rosa & M. Shalahuddin, 2014: 162)

2.3.1.2. Class Diagram

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

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

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusiketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case*

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

4. Kelas yang diambil dari pendefinisian data

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

Tabel 2.5 Simbol Class Diagram

Simbol	Deskripsi
Kelas Nama_kelas +atribut +operasi()	Kelas pada struktur sistem
Antarmuka / <i>interface</i>	Sama dengan konsep interface dalam

 Nama_interface	pemrograman berorientasi objek.
--	---------------------------------

Tabel 2.5 Lanjutan

Asosiasi / association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>agregation</i> 	Semua-bagian (<i>whole-part</i>)

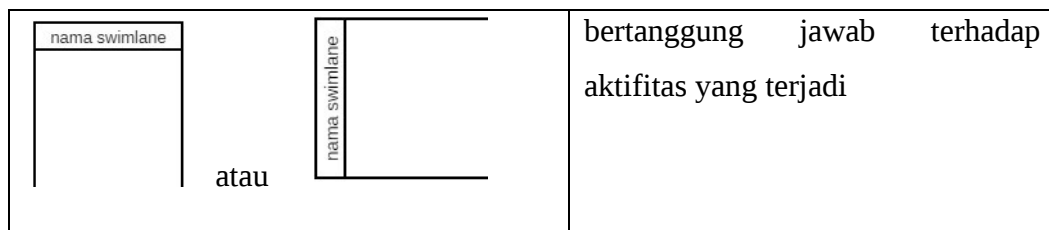
Sumber: (A.S, Rosa & M. Shalahuddin, 2014: 123)

2.3.1.3. Activity Diagram

Menurut (A.S & Shalahuddin, 2011: 161) Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diketahui bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.6 Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktifitas memiliki sebuah status awal
Aktifitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
<i>Swimlane</i>	Memisahkan organisasi bisnis yang

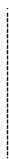
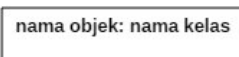


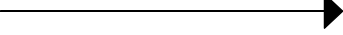
Sumber: (A.S. & Shalahuddin, 2011: 162)

2.3.1.4. Sequence Diagram

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

Tabel 2.7 Simbol Sequence Diagram

Simbol	Deskripsi
Aktor/<i>actor</i>  nama aktor	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 <i>frase</i> nama <i>actor</i>
Garis hidup/<i>lifeline</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>
Objek 	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>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat
---	--

Sumber: (A.S. & Shalahuddin, 2011: 166)

2.3.2. *StarUML*



Gambar 2.4 Logo StarUML

(Sumber: Sumber: <http://staruml.sourceforge.net/image/staruml-logo.jpg>)

StarUML merupakan salah satu *CASE* (*Computer-Aided Software Engineering*) *tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upper CASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak (A.S. & Shalahuddin, 2011: 122)

2.3.3. Android



Gambar 2.5 Logo Android
Sumber: (Tim EMS, 2015: 2)

Android merupakan sebuah sistem operasi perangkat lunak *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Ada dua jenis distributor sistem operasi android yaitu yang mendapat dukungan penuh *google* atau *google mail services* (GSM) dan yang kedua adalah yang benar-benar bebas dari diistribusinya tanpa dukungan langsung dari *google* *Open Handset Distribution* (OHD). Fitur yang ada pada android adalah (Tim EMS, 2015:1-9)

1. Android *run-time*, terdiri atas *library java* dan *dalvik virtual machine*.
2. *Open graphics library*, yaitu *application program interfaces* yang digunakan untuk membuat grafis 2D dan 3D.
3. *Webkit* yaitu *engine* dari *web browser* yang digunakan untuk menampilkan isi *website* dan menyederhanakan tampilan proses *loading*.
4. *SQLite*, yaitu *engine* dari *relasional database* yang dapat diintegrasikan dengan aplikasi.

Menurut (Tim EMS, 2015:1) Kelebihan dari penggunaan sistem Android, Yaitu:

1. *Multitasking* dapat menjalankan aplikasi secara bersamaan.

2. Terdapat notifikasi ketika ada panggilan atau sms, yaitu ketika ada sms atau *email* yang masuk, akan terdapat notifikasi pada ponsel.
3. Dukungan ribuan aplikasi terpercaya melalui situs *google play*, yaitu berfungsi mendapatkan berbagai aplikasi yang diperlukan.
4. Penggunaan *wedget* pada *home screen*, memudahkan dan,
5. mempercepat penggunaan ketika membuka aplikasi.

2.3.4. Java



Gambar 2.6 Logo Java
Sumber: (Hariyanto, 2014: 2)

Java adalah bahasa pemrograman yang dapat dijalankan di komputer maupun telepon genggam. Aplikasi aplikasi java umumnya dikompilasi kedalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai mesin virtual java (JVM) (Tim EMS, 2015: 53).

2.3.5. Eclipse



Gambar 2.7 Logo Eclipse
Sumber: (Tim EMS, 2015: 37)

Eclipse merupakan *tools* editor yang berdiri sendiri yang dipakai dalam pemograman Android, baik berbasis java ataupun pemaketan dengan *HTML* dengan *Cordova*, bisa menggunakan eclipse (Tim EMS, 2015: 35).

2.4. Penelitian Terdahulu

Penelitian yang pertama dilakukan oleh (F. Maria Shusanti, n.d.) dengan judul “**APLIKASI KERUSAKAN SEPEDA MOTOR BEBEK 4-TAK DENGAN METODE FORWARD CHAINING**” dengan ISSN: 2338-4018.

Penelitian ini akan menggambarkan serta menjelaskan sistem pakar yang akan digunakan untuk mendeteksi kerusakan sepeda motor 4-tak dengan menggunakan metode backward chaining dengan jelas. Peneliti akan menggunakan metode backward chaining sebagai metode inferensi untuk sistem pakar ini. Penjelasan mengenai sistem pakar untuk mendeteksi kerusakan sepeda motor 4-tak dengan menggunakan metode backward chaining. Adapun penelitian penelitian ini diharapkan dapat memberi manfaat, diantaranya:

1. Meningkatkan kinerja dari suatu bengkel dalam mengatasi permintaan konsumen yang meningkat.
2. Diharapkan dapat juga digunakan sebagai sarana pelatihan bagi tenaga ahli atau pakar dalam mendeteksi kerusakan sepeda motor 4-tak. Penelitian yang kedua dilakukan oleh Anggraheni Rukmana dan Siska Iriani (Jurnal, Science, & Unsa,

1979) dengan judul “ **Analisis dan Perancangan Sistem Pakar Untuk Mendiagnosis Kerusakan Sepeda Motor Non Injeksi Pada Bengkel Gemilang Jaya Motor Kabupaten Pacitan**” dengan ISSN: 1979-9330. Penelitian ini untuk Mendeteksi Kerusakan pada Sepeda Motor Non Injeksi ini merupakan suatu system untuk mempermudah pemilik motor mendekteksi kerusakan pada motor.

Sehingga

pemilik dapat mengetahui lebih dini kerusakan pada sepeda motor dan dapat melakukan tindakan awal sebelum ditindak lanjuti oleh mekanik ataupun dapat menangani kerusakan – kerusakan ringan pada sepeda motor. Dalam penelitian ini selain memaparkan kajian teori yang digunakan sebagai dasar penyusunan, juga akan dibahas mengenai pembuatan sistem pakar dan implementasi sistem, sehingga nanti dapat mempermudah cara kerja secara efektif dan efisien.

Penelitian yang ketiga dilakukan oleh mharia shushanti (Penelitian, n.d.) dengan judul “**Sistem Pakar Untuk Mendeteksi Kerusakan Pada Sepeda Motor 4-tak Dengan Menggunakan Metode Backward Chaining**” dengan ISSN: 2087-2062. Penelitian ini akan menghasilkan jenis-jenis kerusakan yang terjadi pada sepeda motor serta solusi penanganan dari kerusakan tersebut. Adapun metode pengumpulan data yang dilakukan oleh penulis adalah metode wawancara, metode observasi dan metode studi pustaka. Tujuan dari penelitian ini adalah membantu menganalisa kerusakan mesin sepeda motor bagi mekanik pemula dan para siswa yang sedang prakerin, hasilnya tercipta sebuah aplikasi yang dapat mendeteksi kerusakan mesin sepeda motor untuk membantu siswa yang sedang melaksanakan prakerin dalam mendeteksi kerusakan mesin.

Penelitian keempat Sandi Kosasi (Kosasi, n.d.) “**Pembuatan aplikasi Deteksi Kerusakan Motor Matic Dengan Case-Based Reasoning**” dengan ISSN: 2354-5771. Penelitian ini dapat memberikan kemudahan melakukan diagnosis awal secara mandiri. Case-based reasoning memiliki kemampuan dapat memberikan hasil diagnosis yang lebih akurat berdasarkan kejadian terdahulu dan dapat direvisi kembali dalam memecahkan permasalahan terbaru. Metode perancangan aplikasinya menggunakan reuse-based yang meliputi enam tahap yaitu spesifikasi persyaratan, analisis komponen, modifikasi persyaratan, integrasi design sistem dengan reuse, pengembangan dan integrasi, serta validasi sistem. Tujuan penelitian untuk melakukan deteksi kerusakan mesin sepeda motor matic dan memberikan solusi awal mengenai kondisi kerusakan dan pencegahannya melalui media situs web. Hasil pengujian memperlihatkan aplikasi ini memiliki kemampuan mendeteksi kerusakan dan memberikan solusi penyelesaian masalah dari pengguna dengan rata-rata nilai similaritas antara 0,62 dan 0,7 dengan nilai keakuratan solusi dari pakar sebesar 80% dan 90%.

Penelitian kelima oleh (Ienaco- et al., 2014) dengan judul “**SISTEM PAKAR DIAGNOSIS KERUSAKAN SEPEDA MOTOR NON MATIC**” dengan ISSN: 2337-4349. Penelitian ini bertujuan untuk merancang sistem pakar dibidang otomotif khususnya sepeda motor non matic. Perencanaan sistem dalam membuat knowledge base memakai pohon keputusan dan kaidah produksi sebagai representasi pengetahuan. Pembuatan metode inferensi memakai metode forward chaining yang telah dimodifikasi sehingga sesuai dengan permasalahan. Implementasi program sistem pakar ini menggunakan visual studio 2013. aplikasi

ini akan menghasilkan jenis-jenis kerusakan yang terjadi padasepeda motor non matic serta penanganan dari kerusakan tersebut. Pengujian aplikasi terdiri atas dua jenis pengujian, yaitu itu: pengujian akurasi dan variasi serta pengujian *user friendly* dan fleksibilitas. Akurasi dan variasi diujidengan melakukan analisis terhadap hasil dari aplikasi. Pengujian *user friendly* dan fleksibilitas menggunakan metode wawancara terhadap teknisi. Hasil dari keseluruhan pengujian ini dapat disimpulkan bahwa program sudah cukup baik dalam mendeteksi kerusakan mesin secara umum yang terjadi pada sepeda motor non matic serta disertakan cara perbaikan. Dari penjelasan tersebut dapat diperoleh kesimpulan bahwa sistem ini mempermudah pengguna dalam menggunakan aplikasi ini.

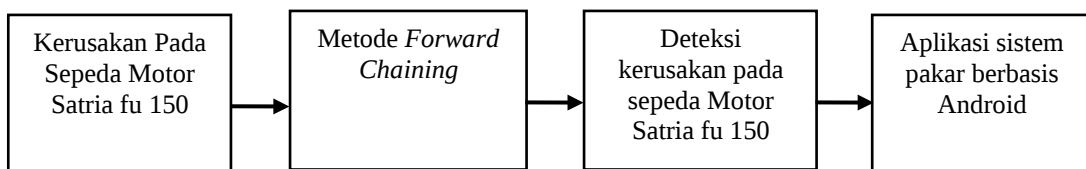
2.5. Kerangka Pemikiran

Menurut Sekaran (2006) dalam Sudaryono (2015: 21) mengemukakan bahwa kerangka berpikir adalah model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah penting. Secara teoritis kerangka berpikir yang baik akan menjelaskan peraturan antar variabel yang akan diteliti. Kerangka berpikir di dalam penelitian perlu dikemukakan apabila penelitian berkaitan dengan dua variabel atau lebih. Apabila penelitian hanya membahas satu variabel atau lebih secara mandiri, di samping mengemukakan deskripsi teoritis untuk masing-masing variabel, peneliti juga harus menyatakan argumentasi terhadap besaran variabel yang diteliti Sukmadinata (2006) dalam (Sudaryono, 2015: 20): . Selanjutnya, Sekaran (2006)

dalam (Sudaryono, 2015: 21) mengemukakan bahwa kerangka berpikir yang baik memuat hal-hal berikut:

1. Variabel-variabel yang akan diteliti harus dijelaskan.
2. Diskusi dalam kerangka berpikir harus dapat menunjukkan dan menjelaskan hubungan atau keterkaitan antarvariabel yang diteliti, serta teori yang mendasarinya.
3. Diskusi juga harus dapat menunjukkan dan menjelaskan apakah hubungan antarvariabel itu positif atau negatif, berbentuk simetris, kausal, atau interaktif (timbang balik atau umpan balik).
4. Kerangka berpikir selanjutnya perlu dinyatakan dalam bentuk diagram (paradigma penelitian) sehingga pihak lain dapat memahami kerangka berpikir yang dikemukakan dalam penelitian.

Berikut adalah kerangka pemikiran dari penelitian ini.



Gambar 2.8 Kerangka Pemikiran
Sumber: Olahan Peneliti (2017)