

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Landasan teori perlu diterapkan pada penelitian agar penelitian punya suatu pondasi yang kokoh, bukan hanya sekedar perbuatan uji coba. Pentingnya teori dasar merupakan ciri bahwa peneliti merupakan hal yang ilmiah untuk memperoleh data. Kerlinger menyatakan bahwa teori adalah seperangkat konstruk (konsep), definisi, dan proposisi yang fungsi untuk melihat fenomena secara merata lewat spesifikasi hubungan antar variabel hingga bisa dapat berfungsi untuk menjelaskan dan mengetahui fenomena. Menurut pendapat ahli, teori merupakan generalisasi atau kumpulan generalisasi yang bisa di gunakan pada berbagai fenomena secara sama. Sudaryono, (2015: 13)

2.1.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan atau *artificial intelligence* merupakan salah satu bagian ilmu computer yang membuat agar mesin (computer) dapat melakukan pekerjaan seperti dan sebaik yang di lakukan oleh manusia. Pada awal di ciptakannya, computer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran computer semakin mendominasi kehidupan umat manusia. Computer tidak lagi hanya di gunakan sebagai alat hitung, lebih

dari itu, computer di harapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia. Kusumadewi, (2003:1)

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan di peroleh dari belajar. Semakin banyak bekal pengetahuan yang di miliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga di beri akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki, tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pegalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian pula, dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik. Agar computer bisa bertindak seperti dan sebaik manusia, maka computer uuga harus diberi bekal pengetahuan, dan mempunyai kemampuan untuk menalar. Untuk itu pada *artificial intelligence*, akan mencoba untuk memberikan beberapa metode untuk membekali computer dengan kedua komponen tersebut agar computer bisa menjadi mesin yang pintar. Kusumadewi, (2003: 1)

2.1.2 Sistem Pakar atau *Expert System*

System pakar (expert system) adalah system yang melakukan mengadopsi pengetahuan manusia ke dalam computer, supaya computer bisa menyiapkan masalah seperti yang di lakukan oleh orang keahlian. System pakar yang baik di

rancang supaya bisa dapat menyiapkan suatu permasalahan tertentu dengan menggunakan kerja dari orang yang ahli. Dengan system pakar ini, crew kapal dapat memperbaiki mesin yang rusak pada masalah yang cukup sulit yang sebenarnya untuk bisa di selesaikan dengan bantuan orang yang ahli. system pakar ini juga bisa menjalankan pekerjaannya sebagai asisten yang mempunyai pengalaman Kusumadewi, (2003: 109)

Berikut merupakan definisi system pakar :

1. Pendapat durkin : system pakar adalah suatu program computer yang di rancang untuk menerapkan kemampuan dalam menyelesaikan masalah yang di buat oleh orang yang ahli.
2. Pendapat ignizio : system pakar adalah suatu model dan aturan yang saling berkaitan, pada suatu domain tertentu, yang mana tingkat ahlinya bisa di bandingkan dengan ahlian seorang pakar.
3. Pendapat giarratano dan riley : system pakar adalah suatu system computer yang dapat menjiplak atau mengikuti keahlian seorang pakar.

2.1.2.1 Sejarah system pakar

Es mulai dikembangkan pada pertengahan tahun 1960-an oleh *Artificial Intelligence Corporation*. Periode penelitian *artificial Intelligence* ini didominasi oleh suatu keyakinan bahwa nalar yang digabung dengan computer canggih akan menghasilkan prestasi pakar atau bahkan manusia super. Suatu usaha ke arah ini adalah *General Purpose Problem-Solver* (GPS). GPS yang berupa sebuah prosedur yang di kembangkan oleh Allen Newell, John Cliff Shaw, dan Herbert

Alexander Simon dari *Logic Theorist* merupakan sebuah percobaan untuk menciptakan mesin yang cerdas. GPS sendiri merupakan sebuah predecessor menuju *Expert System* (ES). GPS berusaha untuk menyusun langkah-langkah yang di butuhkan untuk mengubah situasi awal menjadi state tujuan yang telah di tentukan sebelumnya. Kusrini, (2008: 12)

Pada pertengahan tahun 1960-an, terjadi pergantian dari program serba bisa (*General-Purpose*) ke program yang spesialis (*Special-Purpose*) dengan di kembangkannya Dendral oleh E. Feigenbaum dari Universitas Stanford dan kemudian diikuti oleh MYCIN. Pembuatan Dendral mengarah pada konklusi-konklusi berikut : GPS terlalu lemah untuk di gunakan sebagai dasar untuk membangun ES yang berunjuk kerja tinggi. Pemecahan masalah manusia adalah baik hanya jika beroperasi dalam domain yang sangat sempit. ES harus di update secara berkala untuk informasi baru. Update semacam ini dapat efisien apabila menggunakan representasi pengetahuan berbasis rule. Kusrini, (200: 13)

Problem yang kompleks membutuhkan pengetahuan yang banyak sekali tentang area problem. Pada pertengahan tahun 1970-an, beberapa ES mulai muncul. Sebuah pengetahuan kunci yang di pelajari saat itu adalah kekuatan dari ES formalisme-formalisme khusus dan pola penarikan kesimpulan yang digunakannya Kusrini, (2006: 13)

Awal 1980-an, teknologi ES yang mula-mula di batasi oleh suasana akademis mulai muncul sebagai aplikasi komersil, khususnya XCON, XSEL (dikembangkan dari R-1 pada digital Equipment Corp.) dan CATS-1 (dikembangkan oleh General Electric). Kusrini, (2006: 13)

System pakar untuk melakukan diagnosis kesehatan telah di kembangkan sejak pertengahan tahun 1970. System pakar untuk melakukan diagnosis pertama di buat oleh Bruce Burchanan dan Edward Shortliffe di Stanford University. System pakar di beri nama MYCIN Kusrini, (2006: 13)

MYCIN merupakan program interaktif yang melakukan diagnosis penyakit minigitis dan infeksi bacremia serta memberikan rekomendasi terapi antimikrobia. MYCIN mampu memberikan penjelasan atas penalarannya secara detail. Dalam uji coba, dia mampu menunjukkan kemampuannya seperti seorang spesialis. Meskipun MYCIN tidak pernah digunakan secara rutin oleh dokter, MYCIN merupakan referensi yang bagus dalam penelitian kecerdasan buatan yang lain. Kusrini, (2006: 13)

2.1.2.2 Pemakaian system pakar

System pakar dapat digunakan oleh :

1. Orang awam yang bukan pakar untuk meningkatkan kemampuan mereka dalam memecahkan masalah.
2. Pakar sebagai asisten yang berpengetahuan.
3. Memperbanyak atau menyebarkan sumber pengetahuan yang semakin langka.

System pakar merupakan program yang dapat menggantikan keberadaan seorang pakar alasan mendasar mengapa ES 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 pension atau pergi.
4. Menghadirkan/menggunakan jasa seorang pakar memerlukan biaya yang mahal.
5. Kepakaran di butuhkan juga pada lingkungan yang tidak bersahabat (*hostile enviroentment*).Kusrini, (2006: 14)

2.1.2.3 ciri-ciri system pakar

1. terbatas pada bidang yang spesifik.
2. dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. dapat mengemukakan rangkaian alasan yang di berikannya dengan cara yang dapat dipahami.
4. berdasarkan pada *rule* atau kaidah tertentu.
5. dirancang untuk dapat dikembangkan secara bertahap.
6. outputnya bersifat nasihat atau anjuran.
7. output tergantung dari dialog dengan user.
8. *knowledge base* dan *inference engine* terpisah. Kusrini, (2006: 14)

2.1.2.4 manfaat system pakar

1. membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.

3. meningkatkan kualitas.

4. memiliki kemampuan untuk memecahkan masalah yang kompleks. Kusrini, (2006: 14)

2.1.2.5 konsep dasar system pakar

System pakar pertama kali di kembangkan oleh komunitas AI pada pertengahan tahun 1960. System pakar yang muncul pertama kali adalah *General-purpose Problem Solver* (GPS) yang di kembangkan oleh Newel dan simon. GPS (dan program-program yang serupa) ini mengalami kegagalan dikarenakan cakupannya terlalu luas sehingga terkadang justru meninggalkan pengetahuan-pengetahuan penting yang seharusnya di sediakan. Kusumadewi, (2003:109)

a. Keuntungan system pakar

Secara garis besar, banyak manfaat yang dapat di ambil dengan adanya system pakar, antara lain :

1. Menungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Meningkatkan kualitas.

b. Kelemahan system pakar

Disamping memiliki beberapa keuntungan, system pakar juga memiliki beberapa kelemahan, antara lain :

1. Biaya yang di perlukan untuk membuat dan memeliharanya sangat mahal.

2. Sulit di kembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
3. System pakar tidak 100 % bernilai benar.

c. Kepakaran (*expertise*)

Pakar merupakan pengetahuan yang di dapat pada pelatihan, membacandan jejak.pakar inilaheyang membuat para keahlian bisa mengambil keputusan lebih baik dari pada seorang yang bukan merupakan keahlian Sutojo et al., (2011:163).kepakaran itu sendiri meliputi pengetahuan tentang:

1. Fakta-fakta tentang bidang permasalahan tertentu.
2. Teori-teori tentang bidang permasalahan tertentu.
3. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya.
4. Aturan *heuristic* yang harus di kerjakan dalam suatu situasi tertentu.
5. Strategi global untuk memecahkan permasalahan.
6. Pengetahuan tentang pengetahuan (*meta knowledge*).

d. Pakar (*expert*)

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topic permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang di dapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya, Sutojo et

al., (2011: 163) jadi seseorang pakar harus mampu melakukan kegiatan-kegiatan berikut:

1. Mengenali dan memformulasikan permasalahan.
2. Memecahkan permasalahan secara cepat dan tepat.
3. Menerangkan pemecahannya.
4. Belajar dari pengalaman.
5. Merestrukturisasi pengetahuan.
6. Memecahkan aturan-aturan.
7. Menentukan relevansi.

e. Pemindahan kepakaran (*transferring expertise*)

Tujuan dari *system* pakar adalah memindahkan kepakaran dari seorang pakar ke dalam computer, kemudian di transfer kepada orang lain yang bukan pakar. Sutojo et al., (2011: 164) Proses ini melibatkan empat kegiatan, yaitu :

1. Akuisisi pengetahuan (dari pakar atau sumber lain)
2. Representasi pengetahuan (pada computer)
3. Inferensi pengetahuan
4. Pemindahan pengetahuan ke pengguna

f. Inferensi (*inferencing*)

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi di tampilkan pada suatu komponen yang di sebut mesin inferensi yang mencakup prosedur-prosedur dalam memecah masalah. Semua pengetahuan yang di miliki oleh seorang ahli di simpan di basis pengetahuan oleh system pakar. mesin inferensi adalah

meng ambil ringkasan berdasarkan basis pengetahuan yang di milikinya. Sutojo et al., (2011:164)

g. Aturan-aturan (*rule*)

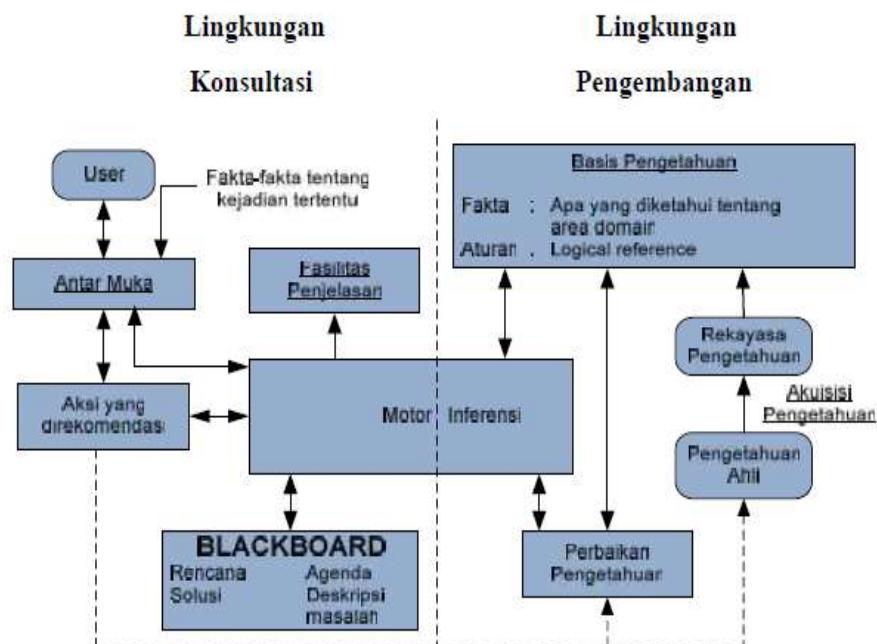
banyak software system pakar komersial adalah system yang berbasis rule (*rule-based system*), yaitu pengetahuan di simpan di dalam bentuk rule, untuk prosedur-prosedure memecah permasalahan. Sutojo et al., (2011: 165)

h. Kemampuan menjelaskan (*explanation capability*)

Fasilitas lain pada system pakar adalah kemampuannya untuk menerangkan nasihat atau merekomendasi untuk di berikannya. kejelasan di lakukan pada sub sistem yang di sebut subsistem penjelasan (*explanation*). Bagian dari system ini memungkinkan system untuk memeriksa penalaran yang di buatnya sendiri dan menjelaskan operasi-operasinya Sutojo et al., (2011: 165)

i. Struktur system pakar

Ada dua bagian terpenting di system pakar, yaitu lingkungan kembangan (*development environment*) dan lingkungan Tanya jawab (*consultation environment*). Lingkungan pengembangan di gunakan untuk membuat system pakar untuk membangun komponen-komponennya atau memperesentasikan pengetahuan ke dalam *knowledge base* (basis pengetahuan). tempat konsultasi di gunakan para user untuk konsultasi sehingga user mendapatkan pengetahuan atau nasehat dari system pakar layak nya berkonsultasi kepada seseorang ahli. Sutojo et al., (2011:166)



Gambar 2.1 struktur system pakar
Sumber : Sutojo et al., (2011: 167)

Komponen yang terdapat dalam struktur system pakar ini adalah: Basis pengetahuan (*knowledge base*), Mesin Inferensi (*Inference engine*), Daerah Kerja (*Blackboard*), Antarmuka pengguna (*User Interface*), Subsistem penjelasan (*Explanation subsystem / justifier*), Sistem perbaikan pengetahuan (*knowledge Refining system*), dan Pengguna (*user*). Sutojo et al., (2011:168)

1. Basis Pengetahuan (*knowledge base*)

Basis pengetahuan mengandung pengetahuan baik di perlukan untuk memahami, mem formulasikan, atau menyelesaikan masalah. Ilmu pengetahuan terdiri beberapa dua lapisan dasar, yaitu :

- Kenyataan, misalnya situasi, kondisi,dan masalahan yang di alami.

- b. Rule (aturan), untuk mengarahkan user pengetahuan ke memecahkan masalah.

2. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah sebuah perancangan yang berguna untuk memandu alur penalaran pada suatu kondisi berdasarkan pada basis pengetahuan, manipulasi dan mengarahkan kaidah, model, dan fakta harus di simpan di basis pengetahuan untuk mendapatkan solusi atau simpulan. Dalam proses nya, mesin inferensi menggunakan teknik kendalian, yaitu strategi yang bertujuan sebagai pemaantuan arah dalam melakukan proses pemikiran. Ada tiga teknik pengendalian yang di gunakan, yaitu *Forward chaining*, *Backward Chaining*, dan kumpulan dari dua teknik tersebut.

3. Daerah kerja (*BlackBoard*)

Untuk mencatat hasil dugaan sementara yang akan di jadikan sebagai acuan dan untuk menjelaskan sebuah tujuan yang sedang berlangsung, system pakar membutuhkan *Black board*, yaitu tempat untuk memori yang di gunakan untuk basis data. Tiga tipe keputusan yang bisa di rekam pada *blackboard*, yaitu :

- a. Rencana : cara menghadapi masalah.
- b. Agenda : perlakuan potensial yang sedang menunggu untuk di eksekusi.
- c. Solusi : user aksi yang akan di bangkitkan.

4. Antarmuka Pengguna (*User Interface*)

Di pakai sebagai media komunikasi antara *user* dan system pakar. berkomunikasi ini paling bagus bila di sajikan pada bahasa alamiah (*natural*

language) dan dilengkapi dengan grafik, menu, dan formulir elektronik. Pada bagian ini bisa terjadi percakapan antara system pakar dan user.

5. Subsystem Penjelasan (*Explanation Subsystem / Justifier*)

Berfungsi memberi keterangan kepada pengguna, bagaimana bentuk simpulan dapat di ambil. Kemampuan seperti ini paling berharga bagi pengguna untuk mengetahui alur pemindahan ke ahlian pakar maupun dalam pemecahan permasalahan.

6. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)

Kemampuan memperbaiki pengetahuan (*Knowledge Refining System*) dari user pakar di perlukan untuk menganalisis pengetahuan, belajar dari permasalahan yang sudah lama, dan selanjutnya perbaiki pengetahuannya sehingga dapat di pakai untuk masa mendatang. Kemampuan evaluasi diri seperti itu di perlukan oleh aplikasi agar dapat meng analisis alasan-alasan ke suksesan dan kegagalanya pada ambil kesimpulan. Dengan cara ini basis pengetahuan yang lebih maju dan mempercepat penalaran yang lebih efektifas akan di hasilkan.

7. Pengguna (*User*)

umumnya pengguna system pakar bukan lah seorang ahli (*non-expert*) yang membutuhkan solusi, saran, atau pembelajaran (*training*) dari berbagai masalahan yang timbul.

2.1.2.6 Representasi pengetahuan

Agar pengetahuan dapat digunakan dalam system, pengetahuan harus direpresentasikan dalam format tertentu yang kemudian di himpun dalam suatu basis pengetahuan. Cara system pakar merepresentasikan pengetahuan akan mempengaruhi perkembangan, efisiensi, dan perbaikan system Kusrini, (2006:23)

1. Pengertian pengetahuan

Pengetahuan merupakan kemampuan untuk membentuk model mental yang menggambarkan obyek dengan tepat dan merepresentasikan dalam aksi yang di lakukan terhadap suatu obyek. Kusrini, (2006: 23)

Pengetahuan dapat diklasifikasikan menjadi tiga, yaitu pengetahuan procedural (*procedural knowledge*), pengetahuan deklaratif (*declarative knowledge*), dan pengetahuan tacit (*tacit knowledge*). Pengetahuan procedural lebih menekankan pada bagaimana melakukan sesuatu. Pengetahuan deklaratif menjawab pertanyaan apakah sesuatu bernilai salah atau benar. Sedangkan pengetahuan tacit merupakan pengetahuan yang tidak dapat di ungkapkan dengan bahasa. Misalnya : bagaimana cara kita memindahkan tangan. Kusrini, (2006: 24)

2. Representasi pengetahuan

Representasi pengetahuan merupakan metode yang di gunakan untuk mengodekan pengetahuan dalam sebuah system pakar yang berbasis pengetahuan. Perepresentasi dimaksudkan untuk menangkap sifat-sifat penting problema dan membuat informasi itu dapat diakses oleh prosedur pemecahan problema Kusrini, (2006: 24)

Bahasa representasi harus dapat membuat seorang pemrogram mampu mengekspresikan pengetahuan yang di perlukan untuk mendapatkan solusi problema, dapat di terjemahkan ke dalam bahasa pemrograman dan dapat disimpan. Harus di rancang agar fakta-fakta dan pengetahuan lain yang terkandung di dalamnya dapat digunakan untuk penalaran. Kusrini, (2006: 24)

3. Model Representasi pengetahuan

Pengetahuan dapat direpresentasikan dalam bentuk yang sederhana atau kompleks, tergantung dari masalahnya Kusrini, (2006: 24). Beberapa model representasi pengetahuan yang penting, adalah :

a. Logika

Logika merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, system kaidah, dan prosedur yang membantu proses penalaran. Logika merupakan bentuk representasi pengetahuan yang paling tua, yang menjadi dasar dari teknik representasi *high level* Kusrini, (2006: 25)

b. Jaringan semantic (*semantic nets*)

Konsep jaringan semantic diperkenalkan pada tahun 1968 oleh Ross Quillian. Jaringan semantic merupakan teknik representasi kecerdasan buatan klasik yang di gunakan untuk informasi proposional adalah pernyataan yang mempunyai nilai benar atau salah. Informasi proposional merupakan bahasa deklaratif karena menyatakan fakta Kusrini, (2006: 26)

c. Object-Attribute-Value (OAV)

Object dapat berupa bentuk fisik atau konsep. *Attribute* adalah karakteristik atau sifat dari *Object* tersebut. *Values* (nilai)-besaran/nilai/takaran spesifik dari *Attribute* tersebut pada situasi tertentu, dapat berupa numeric,string atau Boolean. Kusrini, (2006:31)

d. Bingkai (frame)

Bingkai merupakan ruang-ruang (*slots*) yang berisi atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berupa kejadian, lokasi, situasi, ataupun elemen-elemen lainnya. Bingkai di gunakan untuk merepresentasikan pengetahuan deklaratif.Kusrini, (2006: 32)

e. Kaidah produksi

Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategis. Kaidah produksi dituliskan dalam bentuk jika-maka (if-then). Kaidah if-then menghubungkan antesedan (*antecedent*) dengan konsekuensi yang diakibatkannya. Berbagai struktur kaidah if-then yang menghubungkan obyek atau atribut adalah sebagai berikut:

JIKA premis Maka konklusi

JIKA masukan MAKA keluaran

JIKA kondisi MAKA tindakan

JIKA antesedan MAKA konsekuensi

JIKA data MAKA hasil

JIKA tindakan MAKA tujuan

Premis mengacupada fakta yang harus benar sebelum konklusi tertentu dapat di peroleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat di peroleh.kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat di ambil. Antesedan mengacu pada situasi yang terjadi sebelum konsekuensi dapat di amati. Data mengacu pada kegiatan yang harus di lakukan sebelum hasil dapat di harapkan. Tindakan mengacu pada kegiatan yang harus di lakukan sebelum hasil dapat di harapkan. Kusrini, (2006: 33)

2.1.2.7 Mesin Inferensi (*Inference Engine*)

Inferensi merupakan alur untuk mendapatkan informasi dari fakta yang di ketahui dan diasumsikan. Inferensi merupakan konklusifitas logis (*logical conclusion*) dan implikasi berdasarkan penyampaian yang tersedia (Kusrini, 2006:35). Dalam aplikasi pakar proses inferensi di lakukan dalam suatu panduan yang di sebut *Inference Engine* (Mesin Inferensi).

Pada system pakar berbasis *rule*, domain pengalaman direpresentasikan dalam sebuah perkumpulan *rule* bentuk IF-THEN, sedangkan data di representasikan dalam bentuk kumpulan fakta-fakta tentang kejadian saat ini. Mesin interferensi bandingkan masing- masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam *database*. Jika bagian IF (kondisi) dari *rule* cocok dengan fakta, maka *rule* dieksekusi dan

bagian THEN (aksi) diletakkan dalam *database* sebagai fakta baru yang ditambahkan Sutojo et al., (2011: 171)

dua metode inferensi yang penting dalam system :

1. Runut maju (*forward Chaining*)

Runut maju berarti memakai himpunan aturan kondisi-aksi. Dalam metode yang di terapkan, data di gunakan untuk digunakan aturan mana yang akan di jalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai di temukan suatu hasil Wilson, (1998).

Forward chaining adalah teknik pencarian yang di mulai dengan fakta yang di ketahui, 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) di tambahkan ke dalam database. Setiap kali pencocokan, di mulai dari rule teratas. Setiap rule hanya boleh dieksekusi sekali saja. proses pencocokan berhenti bila tidak ada lagi rule yang bisa di eksekusi Sutojo et al., (2011: 171)

Metode inferensi runut maju cocok di gunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*). Giarattano dan Riley, (1994).

2.1.2.8 Tabel Keputusan

Dalam proses pengambilan keputusan, semua informasi yang diperlukan disusun dalam bentuk ringkasan hasil yang disebut sebagai tabel hasil atau tabel keputusan. Tabel ini merupakan suatu matriks yang terdiri dari baris yang menunjukkan berbagai alternative pilihan/keputusan dan kolom yang menunjukkan nilai harapan untuk setiap alternative pilihan /keputusan pada berbagai keadaan atau situasi yang mungkin terjadi. Herjanto & Herfan, (2008: 26)

2.1.2.9 Pohon Keputusan

Pohon merupakan struktur penggambaran pohon secara hirarkis, struktur pohon terdiri dari node-node yang menunjukan obyek, dan *arc* (busur)yang menunjukan hubungan antar obyek. Kusumadewi, (2003: 28)

2.2 Variabel Penelitian

Secara teoritis variable dapat di definisikan sebagai atribut seseorang, atau obyek, yang mempunyai “variasi” antara satu orang dengan yang lain atau satu obyek dengan obyek yang lain Sugiyono, (2014: 38) variabel juga dapat merupakan atribut dari bidang keilmuan atau kegiatan tertentu. Tinggi, berat badan, sikap, motivasi, kepemimpinan, disiplin kerja, merupakan atribut dari setiap orang. Berat, ukuran, bentuk, dan warna merupakan atribut-atribut dari obyek. Struktur organisasi, model pendelegasian, kepemimpinan, pengawasan, koordinasi, produser dan mekanisme kerja, deskripsi pekerjaan, kebijakan, adalah

merupakan contoh variable dalam kegiatan administrasi Sugiyono, (2014: 38) Berdasarkan pengertian-pengertian di atas, maka dapat dirumuskan di sini bahwa variable penelitian adalah suatu atribut atau sifat atau nilai dari orang, obyek atau kegiatan yang mempunyai variasi tertentu yang di tetapkan oleh peneliti untuk di pelajari dan kemudian di tarik kesimpulannya.

2.2.1 Mesin Kapal Laut

Mesin kapal laut merupakan alat atau mesin yang di gunakan oleh kapal untuk menggerakkannya, mesin kapal laut berperan sangat penting dalam mendorong pertumbuhan ekonomi segala aktifitas yang lintas melalui laut semuanya menggunakan mesin sebagai sarana penggerak, untuk itu *crew* kapal harus tau bagaimana untuk mengoprasikan mesin tersebut dan mengetahui komponen-komponen mesin tersebut jika ada kerusakan yang di alami.

Indikator kerusakan yang terdapat pada mesin kapal laut adalah sebagai berikut :

1. Piston utama mesin

Piston merupakan komponen mesin yang berbentuk ruang bakar bersama – sama memakai *silinder* blok dan *silinder head*. Piston juga lah yang menjalankan gerakan naik turun untuk melakukan siklus kerja mesin, serta piston harus bisa jalan dan menghasilkan tenaga hasil pembakaran ke crankshaft. Jadi dapat kita lihat bahwa piston memiliki fungsi yang sangat penting dalam melakukan siklus kerja

mesin dan dalam menghasilkan tenaga pembakaran yang optimal. Budihartono & Andi, (2010 : 14).

2. Gasket

Gasket mempunyai peran yang sangat penting dalam serangkaian mesin, gasket ini berfungsi sebagai penyekat antara *head cylinder* dan *cylinder block* supaya tidak bocor, bisa mencegah kebocoran oli masuk ke saluran radiator atau bahkan sebaliknya air radiator dari jalur nya masuk ke ruang bakar. Gasket ini di ciptakan untuk mencegah kebocoran tersebut dalam segala situasi saat mesin hidup dalam kondisi dingin maupun kondisi mesin yang sangat panas. Budihartono & Andi, (2010 : 15)

3. Piston Ring

Piston ring merupakan komponen mesin yang di pasang menjadi satu dengan piston, tepatnya pada *ring groove* di bagian sisi atas piston. Piston ring untuk mesin 4 langkah umumnya ada 3, dua buah untuk *ring* kompresi dan 1 lagi untuk *ring oil*. Budihartono & Andi, (2010 : 15)

4. Sensor Oil mesin

Oli mesin merupakan komponen terpenting di dalam mesin, karena mesin harus dilumasi dengan cukup agar bisa bekerja secara maksimal. Agar oli mesin bekerja dengan efisien, oli harus mempunyai tekanan tertentu pada saat bersirkulasi. Di sinilah peran sensor *oil* mesin. Sensor *oil* mesin atau sensor tekanan oli bertugas untuk membaca tekanan oli di dalam mesin dan mengirim informasi ini ke *Speedometer* untuk diolah. Jika tekanan oli berada dibawah level yang seharusnya, maka akan muncul lampu

oli mesin di *dashboard* yang memunculkan tanda peringatan. Budihartono & Andi, (2010 : 14)

5. Payung klep

Payung klep berfungsi mengatur masuknya gas baru dan keluarnya gas buang sisa pembakaran pada mesin. Tugas dari *klep* sendiri sangat berat dan penting, karena apabila ada kebocoran/gangguan sedikit saja pada *klep* akan mengakibatkan tenaga mesin menjadi menurun karena terdapat kebocoran gas pada saat pembakaran. Budihartono & Andi, (2010 : 14)

6. Jarum minyak mesin

Jarum minyak mesin adalah penentuan untuk nyala atau berhentinya mesin, serta tenaga yang di hasilkan karena pada saat proses pembakaran jarum minyak mesin yang mengatur untuk kencang atau lambatnya putaran mesin tersebut dan berfungsi sangat penting untuk mesin. Budihartono & Andi, (2010 : 14)

7. Pompa pendingin mesin

Pompa pendingin mesin adalah sebuah pompa yang bertugas untuk menyedot air untuk di sirkulasi ke mesin tersebut atau di alirkan, guna pompa pendingin mesin adalah untuk menjaga suhu mesin dalam kondisi stabil jika di nyalakan pada jangka panjang. Pompa pendingin mesin sangat penting karena kalau pompa tidak berjalan lancar maka berakibat fatal ke mesin. Budihartono & Andi, (2010 : 15)

8. Kepala silinder(silinder head)

Kepala silinder (*Cylinder head*) adalah salah satu komponen utama mesin yang dipasangkan pada blok silinder dan di ikat menggunakan baut. Kepala silinder harus tahan terhadap temperatur dan tekanan yang tinggi selama engine bekerja. Oleh sebab itu umumnya kepala silinder dibuat dari besi tuang, Pada saat ini banyak engine yang kepala silindernya terbuat dari paduan aluminium. Kepala silinder yang terbuat dari paduan Aluminium memiliki kemampuan pendinginan lebih besar di Banding dengan yang terbuat dari besi tuang. Budihartono & Andi, (2010 : 14)

9. Platup

Platup adalah berfungsi sebagai penghubung antara kem dengan katup sebagai pengatur naik turunnya (buka-tutup) klep. temlar harus dibuat dengan kepresisian yang tinggi dan menggunakan material yang tepat sehingga dapat membuat pergerakan antara kem dan katup lebih presisi, sehingga *supply* bahan bakar dari jarum minyak mesin menuju ke ruang bakar menjadi efisien. Budihartono & Andi, (2010 : 14)

10. Karter

Carter (karter) atau bak oli atau penampung oli merupakan komponen pada mesin yang terletak pada bagian paling bawah, *Carter* sendiri berfungsi untuk menampung oli atau pelumas di dalam mesin sehingga oli pelumas selalu tersedia. Budihartono & Andi, (2010 : 14)

11. Metal Round

Metal adalah sebuah jenis bearing yang memiliki spesifikasi khusus kecepatan tinggi dan tekanan tinggi. Bearing ini dikenal dengan *Journal Bearing* yang bekerja dengan lapisan fluida yang memisahkan antara As dan *Bearing* ini. Fluida yang dipasok melalui lubang yang terdapat pada bearing ini kemudian mengisi celah diantara Metal dan As (*Crank Shaft*) biasanya bertekanan sehingga pada saat perputaran tidak terjadi sentuhan sama sekali antara As dan Permukaan Metal (*Journal Bearing*). Budihartono & Andi, (2010 : 14)

2.3 Software pendukung

Software pemograman biasanya digunakan untuk mempermudah para user membuat program (*programmer*) untuk menulis program yang kemudian di bentuk jadi sebuah objek yang bisa pakai oleh *system software* dalam bentuk aplikasi. *Software* pendukung yang di gunakan untuk merancang system aplikasi ini adalah:

2.3.1 UML (*Unified Modeling Language*)

Salah satu model yang saat ini paling banyak di gunakan adalah UML. UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak di gunakan di dunia industry untuk mendefinisikan *requirement*, membuat analis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek Rosa & Shalahuddin, (2011: 113)

Berikut adalah beberapa jenis diagram UML :

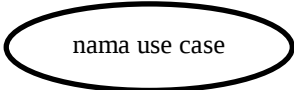
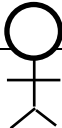
2.3.1.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk melakukan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan *system* 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.

Use case atau diagram *use case* merupakan permodelan 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 kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Ada dua (2) hal utama pada *use case* yaitu pendefinisian apa yang disebut *actor* dan *use case* Rosa & Shalahuddin, (2015:155).

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.1 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 / <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, jadi

	walaupun simbol dari actor adalah gambar orang; tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Asosiasi / <i>association</i> 	Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan aktor
Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang "ditambahkan" dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan
Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya

Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.3.1.2 Activity Diagram

Activity diagram menggambarkan 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 actor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut: Rosa & Shalahuddin, (2015:161)

- b. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- c. Urutan atau pengelompokkan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- d. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan”kasus ujinya.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.2 Simbol *Activity Diagram*

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

Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.3.1.3 *Sequence Diagram*

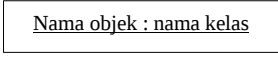
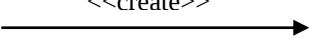
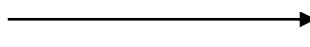
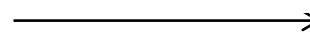
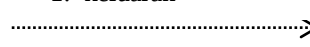
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antara objek. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

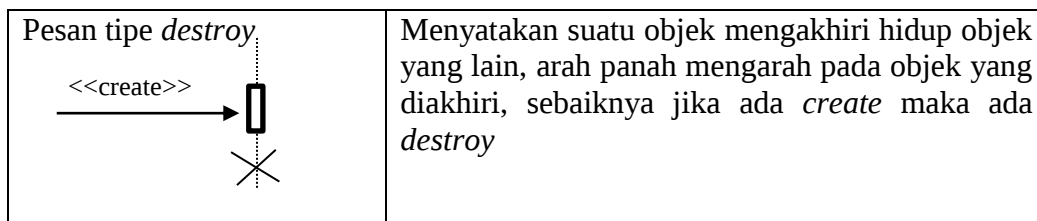
Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen

maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor / <i>actor</i>  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan system informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang; tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi dengan pesan
Pesan tipe <i>create</i> 	Objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian



Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.3.1.4 Class Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel-variabel yang dimiliki suatu kelas.
- 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 sehingga pembuat perangkat lunak atau programmer dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut Rosa & Shalahuddin, (2015):

1. Kelas main

Kelas yang memiliki fungsi awal di eksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (View)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

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

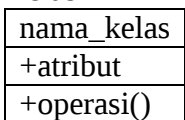
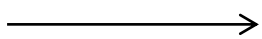
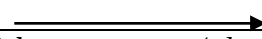
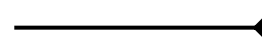
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

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

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

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.4 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	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>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole part</i>)

Sumber: Rosa, A. & shalahuddin, M. (2011)

2.3.2 Android Studio

Anda sudah mengenal apa itu android, merasakan berbagai fitur canggihnya, dan mengetahui keberagaman jenis aplikasi gratis maupun berbayar di play store. Sekarang saatnya keluar dari zona nyaman yang hanya mampu menjadi pengguna. Kini anda juga bias membuat hal serupa. Namun, dari manakah anda harus memulainya ? Firly, (2018:13)



Gambar 2.2 Logo Android Studio
Sumber : Firly, (2018:13)

Android studio merupakan *integrated development environment (IDE)* atau dalam artian lain adalah sebuah lingkungan pengembangan terintegrasi resmi yang memang di rancang khusus untuk pengembangan system operasi google android. Aplikasi yang satu ini , di bangun di atas sebuah perangkat lunak yang di namakan IntelliJ IDEA milik JetBrains. Bisa juga di bilang bahwa android studio merupakan pengganti dari *Eclipse android Development tools* atau ADT sebagai IDE utama dalam pengembangan aplikasi android yang asli. anda tentunya sudah mengetahui bahwa android merupakan platform *open source* yang bebas untuk di kembangkan oleh siapapun. Nah, inilah software yang akan anda gunakan untuk merancang sebuah aplikasi android anda sendiri. Firly, (2018: 13)

Di luncurkan pada 16 mei tahun 2013 dalam konferensi google I/O yang pada saat itu masih dalam tahap pratinjau akses versi 0.1 sebagai perintis, hingga pada akhirnya versi stabil 3.0 yang rilis pada pertengahan bulan oktober 2017 ini menjadi software terlaris di kalangan developer muda. Termasuk anda bukan ? ya, aplikasi ini dapat di gunakan di berbagai system operasi yaitu : window, Linux dan macOS. Firly, (2018: 14)

2.3.3 Java

Berisikan file-file kode sumber java, termasuk junit sebagai kode pengujian. File java berfungsi untuk menjalankan setiap fungsi yang ada pada tampilan ada display pada layar aplikasi android. Dengan kata lain java lah yang mengurus segala aktifitas yang terjadi pada sebuah aplikasi. Contoh, ketika anda membuat sebuah aplikasi kalkulator dasar. Proses penjumlahan lima tambah lima sama dengan sepuluh ada pada java. Yang patut di perhatikan, bahwa lembar kerja java selalu berhubungan dengan lembar kerja xml. Karena satu sama lain memiliki keterkaitan dalam proses pemrograman.

2.4 Penelitian Terdahulu

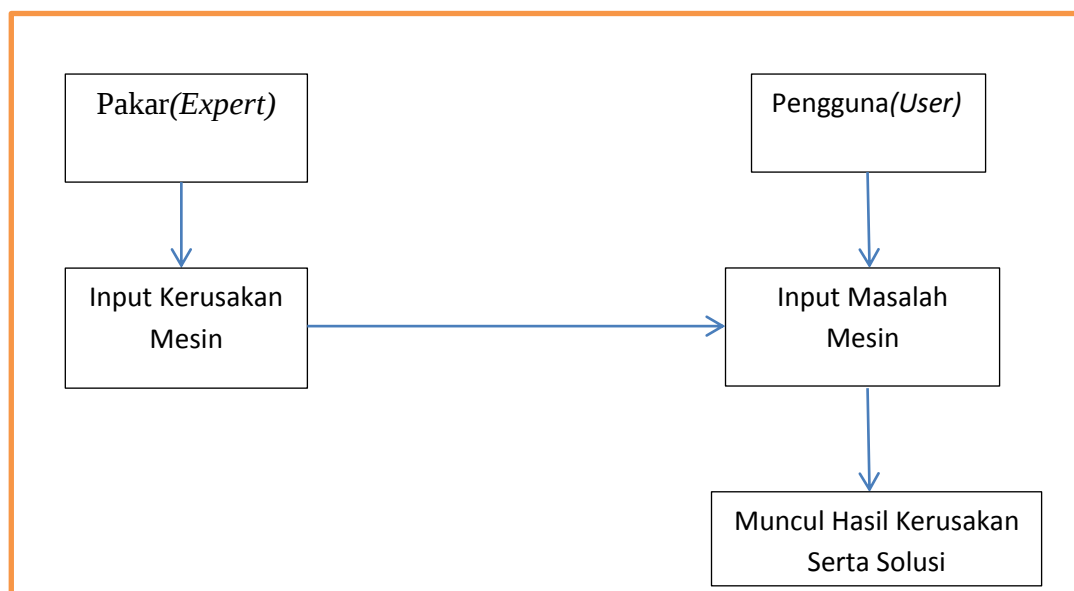
Penelitian terdahulu sangat penting sebagai dasar pijakan dalam rangka penyusunan penelitian ini. Kegunaanya untuk mengetahui hasil yang telah dilakukan oleh peneliti terdahulu.

Penulis	Judul	Issn/Vol
Anggri Sartika Wiguna dan Isman Harianto	Sistem Pakar Diagnosa Kerusakan Sepeda Motor Matic Injeksi Menggunakan Metode Forward Chaining Berbasis Android	e-ISSN : 2476-9754
Anggia Dasa Putri dan Dedy Suhendra	Sistem Pakar Untuk Mendeteksi Kerusakan Air Conditioner Menggunakan Metode Forward Chaining Berbasis Web	e-ISSN : 2527-9866
Muhammad Fauzi	Sistem Pakar Mendeteksi Kerusakan Keyboard Menggunakan Metode Forward Chaining	e-ISSN : 2548-9712
Rosmawati Tamin	Sistem Pakar Diagnosa Kerusakan Pada Printer Menggunakan Metode Forward Chaining	e-ISSN : 2442-4512
Januardi Nasir dan Jahro	Sistem Pakar Konseling Dan Psikoterapi Masalah Kepribadian Dramatik Menggunakan Metode Forward Chaining Berbasis Web	e-ISSN : 2477-2062

2.5 Kerangka Pemikiran

Uma Sekaran dalam bukunya *Business Research* (1992) mengemukakan bahwa, kerangka berfikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai factor yang telah diidentifikasi sebagai masalah yang penting. Sugiyono, (2014: 60)

Kerangka berfikir yang baik akan menjelaskan secara teoritis pertautan antar variable yang akan di teliti. Jadi secara teoritis perlu dijelaskan hubungan antar variable independen dan dependen. Bila dalam penelitian ada variable moderator dan intervening, maka juga perlu di jelaskan, mengapa variable itu ikut di libatkan dalam penelitian. Pertautan antar variable tersebut, selanjutnya di rumuskan ke dalam bentuk paradigm penelitian. Oleh karena itu pada setiap penyusunan paradigm penelitian harus di dasarkan pada kerangka berfikir. Sugiyono, (2014: 60)



Gambar 2.3 Kerangka Pemikiran

Sumber : Data Peneliti 2019

Berikut Merupakan Point-Point yang terdiri dalam kerangka pemikiran dapat di uraikan Sebagai berikut :

2.5.1 Pakar (*Expert*)

Merupakan seseorang yang ahli di bidang mesin untuk mengisi semua jenis kerusakan mesin serta solusi untuk memperbaiki mesin tersebut dan untuk mengupdate tentang kerusakan mesin terbaru.

2.5.2 User (*Pengguna*)

Merupakan Pengguna *system* untuk mengetahui kerusakan dan solusi yang diberikan oleh pakar mesin tersebut, user tidak dapat untuk menambah apapun atau mengupdate program.

2.5.3 Input Kerusakan Mesin

Untuk Menambahkan Kerusakan yang terjadi pada mesin ketika *user* memberitahu ada kerusakan yang baru.

2.5.4 Input Masalah Mesin

Apabila pengguna mesin mengalami masalah saat menggunakan mesin, maka dapat ke langkah ini untuk memberitahukan kepada pakar. Bahwa mesin terjadi kerusakan pada mesin yang mereka gunakan, serta gejala-gejala yang terjadi.

2.5.5 Muncul Hasil Kerusakan Serta Solusi

Dari gejala-gejala yang telah di input oleh pengguna, maka sistem akan menampilkan hasil mendeteksi kerusakan mesin tersebut yang di input oleh user atau pengguna.