

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

A. Sistem Pakar

Menurut Kusrini (2008:3) “Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut”.

Sedangkan menurut Durkin dalam Nita dan Rahmat (2012:1) “Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan seorang pakar”.

Ciri dan karakteristik sistem pakar adalah sebagai berikut :

1. Pengetahuan sistem pakar merupakan suatu konsep, bukan berbentuk numeris. Hal ini dikarenakan komputer melakukan proses pengolahan data secara numerik sedangkan keahlian seorang pakar adalah fakta dan aturan-aturan.
2. Informasi dari sistem pakar tidak selalu lengkap, subjektif, tidak konsisten, subjek terus berubah dan tergantung pada kondisi lingkungan sehingga keputusan yang diambil bersifat tidak pasti dan tidak mutlak “ya” atau “tidak” akan tetapi menurut ukuran kebenaran tertentu. Oleh karena itu dibutuhkan kemampuan sistem untuk belajar secara mandiri dalam menyelesaikan masalah-masalah dengan pertimbangan-pertimbangan khusus.

3. Kemungkinan solusi sistem pakar terhadap suatu permasalahan adalah bervariasi dan mempunyai banyak pilihan jawaban yang dapat diterima.
4. Perubahan atau pengembangan pengetahuan dalam sistem pakar dapat terjadi setiap saat bahkan sepanjang waktu sehingga diperlukan kemudahan dalam memodifikasi sistem untuk menampung jumlah pengetahuan yang semakin besar dan semakin bervariasi.
5. Pandangan dan pendapat setiap pakar tidaklah selalu sama, oleh karena itu tidak ada jaminan bahwa setiap solusi sistem pakar merupakan jawaban yang pasti benar. Setiap pakar akan memberikan pertimbangan-pertimbangan berdasarkan faktor subyektif.
6. Keputusan merupakan bagian terpenting dari sistem pakar. Sistem pakar harus memberikan solusi yang akurat berdasarkan masukan pengetahuan meskipun solusinya sulit sehingga fasilitas informasi sistem harus selalu diperlukan.

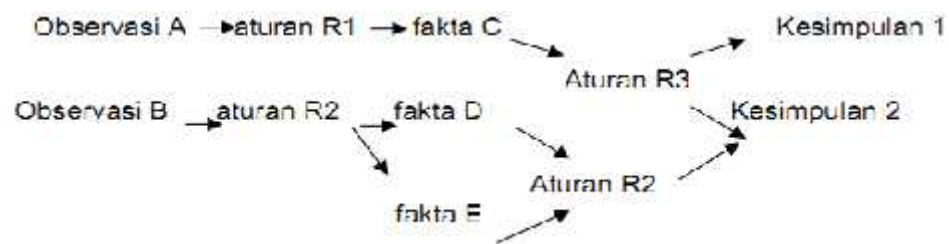
B. Mesin Inferensi (*Inference Method*)

Menurut Kusriani (2008:8) Metode inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis atau implikasi berdasarkan informasi yang tersedia, dalam hal ini akan digunakan metode inferensi dalam pengambilan kesimpulan.

Terdapat dua metode inferensi yang penting dalam pengambilan sistem pakar untuk menarik kesimpulan, yaitu :

1. *Forward Chaining* / Runut Maju

Runut maju adalah pendekatan yang dimonitori data (data driven). Dalam pendekatan ini pelacakan dimulai dari informasi masukan dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN.



Gambar II.1. Diagram Pelacakan Kedepan

(Sumber : Kusrini, Aplikasi Sistem Pakar, 2008)

2. *Backward Chaining* / Runut Balik

Runut Balik adalah Pendekatan ke depan adalah pendekatan yang dimonitori oleh tujuan (*goal driven*). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan untuk kesimpulannya, selanjutnya proses pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya.



Gambar II.2. Diagram Pelacakan Kebelakang

(Sumber : Kusrini, Aplikasi Sistem Pakar, 2008)

C. Metode Pengembangan Sistem

Menurut Kusrini dan Andri Koniyo (2007:44) “Metode pengembangan sistem yang digunakan didasarkan pada pendekatan-pendekatan sistem terstruktur, modular dan berkembang”.

Adapun pendekatan-pendekatan tersebut adalah sebagai berikut :

1. Pendekatan Sistem (*System Approach*)

Pendekatan sistem memperhatikan sistem informasi sebagai satu kesatuan yang terintegrasi untuk masing-masing kegiatan atau aplikasinya. Pendekatan sistem ini juga menekankan pada pencapaian sasaran keseluruhan organisasi, tidak hanya pada sasaran sistem informasi saja.

2. Pendekatan Terstruktur (*Structure Approach*)

Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem sehingga hasil akhir dari sistem yang dikembangkan adalah sistem yang strukturnya didefinisikan dengan baik dan jelas.

3. Pendekatan Modular (*Moduler Approach*)

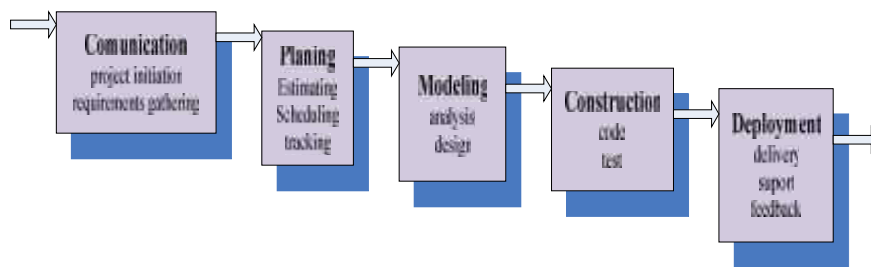
Pendekatan modular berusaha memecah sistem yang rumit menjadi beberapa bagian atau modul yang sederhana sehingga sistem lebih mudah dipahami dan dikembangkan.

4. Pendekatan Berkembang (*Evolutionary Approach*)

Pendekatan berkembang menerapkan teknologi canggih hanya untuk aplikasi-aplikasi yang diperlukan pada saat itu saja dan akan terus dikembangkan pada periode-periode berikutnya, memenuhi kebutuhan sesuai perkembangan teknologi.

D. Teori Model Air Terjun (*Waterfall Model*)

Dalam penulisan skripsi ini, penulis menggunakan proses *Waterfall Model* sebagai pola pengembangan sistem. Menurut Pressman (2010:46) “*Waterfall Model* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*”. Adapun tahapannya sebagai berikut:



Sumber (Pressman, 2010:46)

Gambar II.3.

The Waterfall Model

1. *Communication*

Pada tahap ini akan dilakukan inisiasi proyek, seperti menganalisis masalah yang ada dan tujuan yang akan dicapai. Selain itu dilakukan juga *requirements gathering*, dimana akan dikumpulkan *requirement* dari *user* melalui analisis kuesioner.

2. *Planning*

Tahap ini merupakan tahap dimana akan dilakukan estimasi mengenai kebutuhan-kebutuhan yang diperlukan untuk membuat sebuah sistem. Selain itu, penjadwalan dalam proses pengerjaan juga ditentukan pada tahap ini.

3. *Modeling*

Kemudian mulai masuk pada tahap perancangan dimana perancang menerjemahkan kebutuhan sistem kedalam representasi untuk menilai kualitas sebelum tahap selanjutnya dikerjakan. Tahap ini lebih difokuskan pada atribut program, seperti struktur data, arsitektur perangkat lunak dan detail prosedur.

4. *Construction*

Tahap ini merupakan tahap dimana perancangan diterjemahkan ke dalam bahasa yang dimengerti oleh mesin. Setelah itu dilakukan *testing* atau pengujian terhadap sistem yang telah dibuat.

5. *Deployment*

Setelah proses pengkodean dan pengujian selesai, dilakukan pengiriman yang artinya implementasi kepada masyarakat luas. Pada tahap ini juga dilakukan pemeliharaan, perbaikan dan pengembangan agar sistem tersebut tetap dapat berjalan sebagaimana fungsinya.

E. Konsep Dasar Program

Program adalah kata, ekspresi, pernyataan atau kombinasinya yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer. Bahasa pemrograman merupakan prosedur atau tata cara penulisan program. Pemrograman merupakan proses mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dengan menggunakan suatu bahasa pemrograman.

Menurut Kusriani dan Andri (2007:281) “Pemrograman Terstruktur yaitu suatu bentuk pemrograman yang terorganisir, yang programnya mudah dipahami dan dapat dimodifikasi secara benar”. Ciri Teknik Pemrograman Terstruktur antara lain:

1. Mengandung algoritma pemecahan masalah yang tepat, benar, sederhana, standar dan efektif.
2. Memiliki struktur logika dan struktur program yang benar dan mudah dipahami serta menghindari penggunaan instruksi *GOTO*.
3. Membutuhkan biaya testing, pemeliharaan dan pengembangan yang rendah.
4. Memiliki dokumentasi yang baik.

Sedangkan standar program yang baik antara lain:

- a. Standar Teknik Pemecahan Masalah Standar teknik dalam pemecahan masalah diantaranya dengan:

1. Teknik *Top-Down*

Suatu masalah yang kompleks dibagi-bagi kedalam beberapa tingkatan kelompok masalah hingga sub bagian yang paling kecil, kemudian disusun langkah-langkah untuk menyelesaikannya secara detail.

2. Teknik *Bottom-Up*

Teknik pemecahan masalah yang sudah mulai ditinggalkan karena sulit melakukan standarisasi proses dari prosedur-prosedur yang sudah ada untuk digabungkan menjadi satu kesatuan.

- b. Standar Penyusunan Program

Terdapat beberapa standar dalam penyusunan program, yaitu:

1. Kebenaran logika dan penulisan
2. Waktu minimum untuk penulisan program
3. Kecepatan maksimum eksekusi program
4. Ekspresi penggunaan memori
5. Kemudahan merawat dan mengembangkan program
6. *User friendly*
7. Portabilitas
8. Pemrograman modular.

- c. Standar Perawatan Program

1. Dokumentasi
2. Penulisan instruksi

F. Peralatan Pendukung (*Tools Sytem*)

Merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, diagram-diagram yang menunjukkan secara tepat arti dan fungsinya. Adapun peralatan pendukung sistem (*tools system*) yang dijelaskan sebagai model sistem yang akan dirancang adalah sebagai berikut:

1. UML (*Unified Modeling Language*)

Menurut Nugroho (2010:6) UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berorientasi objek. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Pada tahap ini penulis akan menjelaskan tentang diagram-diagram UML :

a. *Use Case Diagram*

Use Case diagram adalah permodelan untuk kelakuan (*behavioe*) 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 didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

b. *Activity Diagram*

Activity diagram atau diagram aktifitas menggambarkan alir kerja (*workflow*) atau aktivitas dari sebuah sistem atau menu yang terdapat pada perangkat lunak, yang perlu diperhatikan disini adalah bahwa diagram

aktivitas menggambarkan aktivitas sistem bukan yang dilakukan aktor, melainkan aktivitas yang dilakukan oleh sistem.

c. *Component Diagram*

Component diagram atau diagram komponen dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem.

d. *Deployment diagram*

Diagram deployment ini memperlihatkan konfigurasi komponen dalam proses eksekusi aplikasi atau aplikasi dijalankan (*runtime*). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya.

2. ERD (*Entity Relationship Diagram*)

a. Pengertian ERD (*Entity Relationship Diagram*)

Menurut Yuhefizar (2008:17), Diagram E-R adalah diagram yang digunakan untuk menggambarkan secara sistematis hubungan antar *entity-entity* yang ada dalam suatu sistem *database* menggunakan simbol-simbol sehingga lebih mudah dipahami”. Simbol-simbol yang boleh digunakan adalah :

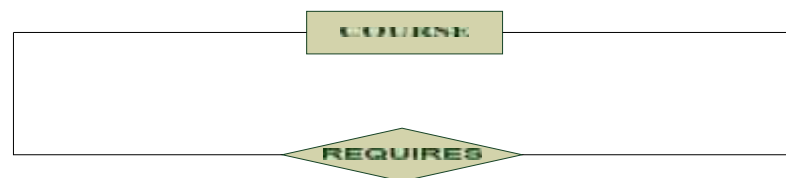
1. Persegi panjang, berfungsi untuk menyatakan suatu *entity*.
2. *Elips*, berfungsi untuk menyatakan *attribute*, jika diberi garis bawah menandakan bahwa *attributetersebut* merupakan *attribute / field* kunci.
3. Belah ketupat, menyatakan jenis relasi
4. Garis penghubung antara relasi dengan *entity* dan antara *entity* dengan *attribute*.

b. Jenis Kerelasian Antar Entitas (*Relationship*)

Terdapat tiga macam derajat dari *relationship*, yaitu:

1. Derajat Satu (*Unary Degree*)

Bila satu *entity* mempunyai relasi terhadap dirinya sendiri.

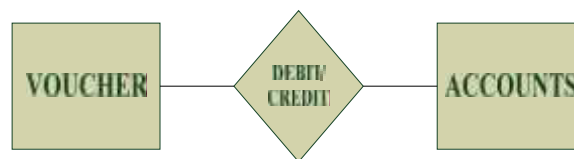


Sumber (Mittal dan Jain, 2010:20.11)

Gambar II.4. Contoh Derajat Satu (*Unary Degree*)

2. Derajat Dua (*Binary Degree*)

Bila satu relasi menghubungkan dua *entity*.

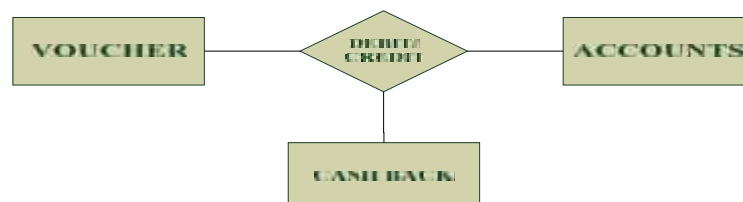


Sumber (Mittal dan Jain, 2010:20.11)

Gambar II.5. Contoh Derajat Dua (*Binary Degree*)

3. Derajat Tiga (*Ternary Degree*)

Bila satu *entity* menghubungkan lebih dari dua *entity*



Sumber (Mittal dan Jain, 2010:20.11)

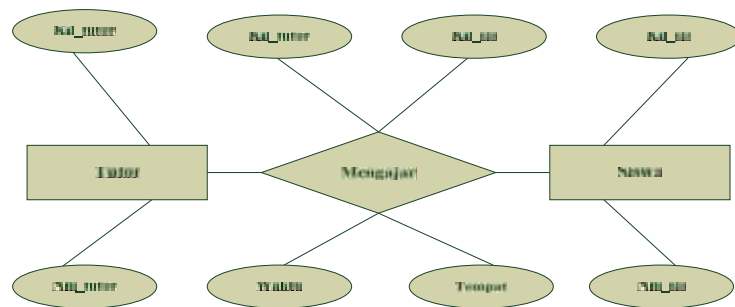
Gambar II.6. Contoh Derajat Tiga (*Ternary Degree*)

c. Cardinality Ratio

Terdapat empat jenis *Cardinality Ratio*, yaitu:

1. 1 : 1 (*One-To-One*)

Sebuah *entity* A diasosiasikan pada sebuah *entity* B, dan sebuah *entity* B diasosiasikan dengan paling banyak sebuah *entity* A.

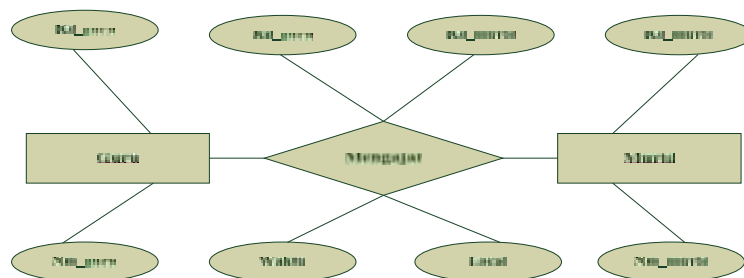


Sumber (Yuhefizar, 2008:18)

Gambar II.7. Contoh relasi 1 : 1 (*One-To-One*)

2. 1 : N (*One-To-Many*)

Sebuah *entity* A diasosiasikan dengan sejumlah *entity* B, tetapi *entity* B dapat diasosiasikan paling banyak satu *entity* A.

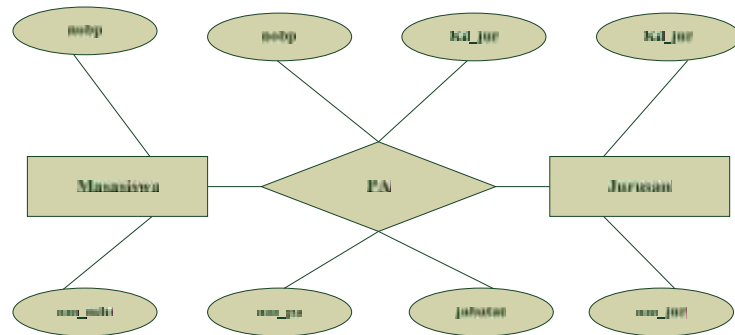


Sumber (Yuhefizar, 2008:19)

Gambar II.8. Contoh relasi 1 : N (*One-To-Many*)

3. N : 1 (*Many-To-One*)

Suatu *entity* A dapat diasosiasikan dengan paling banyak sebuah *entity* B, tetapi *entity* B dapat diasosiasikan dengan sejumlah *entity* di A.

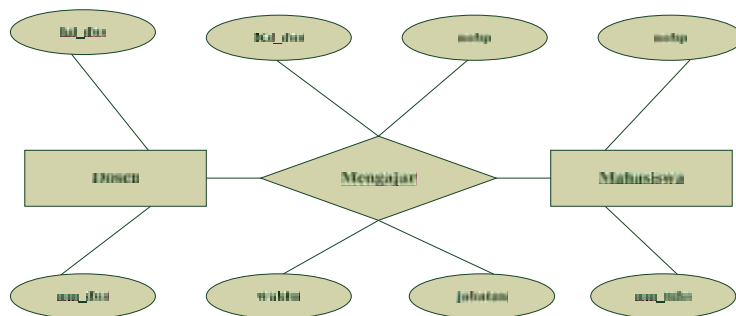


Sumber (Yuhefizar, 2008:20)

Gambar II.9. Contoh relasi N : 1 (*Many-To-One*)

4. M : N (*Many-To-Many*)

Suatu *entity* A dapat diasosiasikan dengan sejumlah *entity* B dan *entity* B dapat diasosiasikan dengan sejumlah *entity* di A.



Sumber (Yuhefizar, 2008:21)

Gambar II.10. Contoh relasi M : N (*Many-To-Many*)

d. *Participation Constraint*

Terdapat dua macam *Participation Constraint*, yaitu:

1. *Total Participation*

Keberadaan suatu *entity* tergantung pada hubungannya dengan *entity* lain.

2. *Partial Participation*

Keberadaan suatu *entity* tidak tergantung pada hubungannya dengan *entity* lain.

e. Tahapan pembuatan ERD

1. Mengidentifikasi dan menetapkan seluruh *entity* yang akan terlibat.
2. Menentukan atribut setiap *entity*.
3. Menentukan atribut *primary key* dari setiap *entity*.
4. Menentukan relationship antar *entity*.
5. Menentukan *Cardinality Ratio*.
6. Menentukan *Participation Constraint*.

G. Mesin

Menurut Yulirianto (2014:1) memberi batasan bahwa “mesin adalah suatu media atau alat untuk mengubah energi seperti air, panas, angin, listrik, atom menjadi energi gerak (*mechainical energy*)”. Beberapa komponen mesin mobil, antara lain:

1. Blok Silinder, merupakan komponen utama dari sebuah mesin kendaraan. Pada bagian blok silinder ini terdapat beberapa buah silinder mesin dan disetiap silinder mesin ini memiliki satu buah piston. Komponen ini berfungsi untuk memaksimalkan kinerja kendaraan.

2. Silinder, komponen ini memiliki fungsi untuk memindahkan dan membentuk panas ke tenaga mekanik, sehingga bila anda ingin tenaga mesin anda menjadi maksimal, usahakan agar menjaga agar tidak ada kebocoran pada gas yang dibakar pada bagian antara piston dan silinder.
3. Kepala Silinder, komponen ini dipasangkan pada blok silinder bagian atas dengan cara dibaut dan diberikan gasket. Biasanya kepala silinder ini terbuat dari campuran besi tuang dengan aluminium yang dapat membatasi pemuaian.
4. Karter, komponen ini terletak dibagian bawah blok silinder. Fungsi dari karter adalah sebagai penampung oli mesin.
5. Torak atau Piston, salah satu komponen mesin mobil ini memang diharuskan untuk dapat bekerja dengan kecepatan maksimal sehingga harus memiliki karakteristik komponen yang tahan terhadap tingkat suhu yang panas.
6. Batang Piston, komponen ini berfungsi menghubungkan antara piston dengan poros engkol.
7. Poros Engkol, komponen ini berfungsi sebagai pengubah dari gerakan lurus piston yang ada pada silinder menjadi gerak memutar melalui batang piston.

2.2. Penelitian Terkait

Menurut Sofyan dkk (2015:1) Sering kali melihat orang mengeluh, atau mengalami kebingungan jika mobil yang dikendarainya tiba-tiba mogok ditengah jalan. Hal ini disebabkan karena pengendara kurang paham bahkan tidak tahu bagaimana cara melakukan tindakan perbaikan. Kerusakan yang dialami pengendara ditangani oleh pakar atau montir ahli. Diagnosa kerusakan dilakukan oleh seorang montir. Hal ini mendorong para ahli untuk mengembangkan komputer baru agar mendukung kerja manusia. Perkembangan komputer sekarang ini telah mengalami banyak perubahan yang sangat pesat, seiring dengan kebutuhan manusia yang semakin banyak dan kompleks. Begitu juga dengan aplikasi-aplikasi yang dibuat dalam rangka membantu manusia dalam bekerja, maka dibutuhkan suatu cara atau sistem yang dapat mendiagnosa kerusakan yang terjadi pada mobil tersebut. Salah satunya adalah dengan memanfaatkan teknologi kecerdasan buatan. Salah satu sub bidang kecerdasan buatan adalah “*Expert System*” atau sistem pakar. Sistem pakar mampu meniru kerja seorang pakar dalam berbagai bidang, seperti diagnosa kerusakan mobil, diagnosa kerusakan motor dan lainnya. Dengan memanfaatkan teknologi sistem pakar dapat dihasilkan suatu sistem untuk mediganosa kerusakan mobil.

Menurut Turnawan dkk (2014:1) Kerusakan pada mobil baru disadari oleh pemilik ketika mobil beroperasi tidak seperti biasanya. Perawatan secara berkala dibutuhkan setiap mobil untuk mendeteksi kerusakan apa yang terjadi pada mobil tersebut. Banyak penyebab dan tanda-tanda saat mobil tidak beroperasi dengan baik, beberapa diantaranya misalnya lampu mobil tidak menyala dan tidak mempunyai gambaran mengapa hal tersebut terjadi, hal inilah yang mendorong pembangunan sistem pakar dengan menggunakan metode *forward chaining* untuk mendeteksi kerusakan mobil.

Berdasarkan jurnal diatas untuk kerusakan yang lebih spesifik pada mesin mobil transportasi umum juga membutuhkan pengetahuan dalam memperbaikinya. Baik itu pengetahuan tentang gejala-gejala kerusakannya sampai solusi untuk memperbaikinya. Untuk lebih memberi kemudahan dalam penanganannya, maka perlu dibuatnya sebuah program sistem pakar untuk mendiagnosis kerusakan pada mesin mobil transportasi umum dengan metode *forward chaining*.