

IMPLEMENTASI SISTEM PAKAR DI BIDANG OTOMOTIF UNTUK MENDIAGNOSA KERUSAKAN SEPEDA MOTOR MANUAL NON INJEKSI MENGGUNAKAN METODE *CERTAINTY FACTOR* BERBASIS WEB

Zaenal Abidin

Program Studi Informatika Universitas Nurtanio Bandung

Jl. Pajajaran 219 Husein Sastranegara Bandung

Email: zaenal1309@gmail.com

Sepeda motor merupakan alat transportasi yang banyak di minati oleh masyarakat Indonesia, namun tidak sedikit pengendara sepeda motor yang tidak mengetahui kondisi sepeda motornya sendiri.

Berdasarkan permasalahan tersebut, maka perlu dilakukan analisis, perancangan, dan pengimplementasian sistem pakar yang dapat mendiagnosa kerusakan sepeda motor, Analisis dilakukan dengan cara mewawancarai para pakar di bidang sepeda motor agar data yang di analisis bisa akurat.

Sistem pakar yang di kembangkan dalam penyusunan tugas akhir ini menggunakan metode certainty factor yaitu metode untuk mengelola ketidakpastian dalam sistem berdasarkan aturan

Hasil akhir perancangan tersebut menghasilkan Sistem pakar diagnosa kerusakan sepeda motor manual noninjeksi berbasis web yang diharapkan bermanfaat bagi seluruh masyarakat yang memiliki sepeda motor manual non injeksi.

Kata Kunci: sistem, pakar, sepeda, motor, *certainty*, *factor*

1. PENDAHULUAN

1.1 Latar Brlakang Masalah

Dalam menciptakan tenaga ahli atau seorang pakar diperlukan waktu dan biaya yang tidak sedikit. Salah satu cara untuk menanggulangi permasalahan ini adalah dengan menciptakan suatu sistem pakar (*expert sistem*).

Sebagian besar masyarakat sekarang telah menjadikan sepeda motor sebagai sarana transportasi utama dalam melakukan aktifitas sehari-hari. Tetapi banyak pengendara sepeda motor yang tidak mengetahui kendala kerusakan yang dialami oleh sepeda motor tersebut, Masalah bagi pengendara yang tidak mengetahui jenis kerusakan, akan sangat fatal apabila jenis kerusakan tersebut tidak segera ditangani.

Saat ini mekanik membutuhkan waktu yang cukup lama untuk mendiagnosa kerusakan sepeda motor, bahkan mekanik kadang-kadang melakukan kesalahan diagnosa karena faktor kelupaan atau faktor lainnya.

Dengan menggunakan metode sistem pakar, diharapkan kemampuan seorang pakar yang

ahli dalam bidang otomotif, khususnya sepeda motor bisa disubstitusikan ke komputer dalam bentuk program sehingga dapat digunakan sebagai alat bantu bagi mekanik dalam mendiagnosa kerusakan atau bagi para pengguna sepeda motor yang awam

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat diidentifikasi masalah, yaitu:

1. Banyak pengendara sepeda motor yang tidak mengetahui kendala kerusakan yang dialami oleh sepeda motornya?
2. Faktor kelupaan atau faktor lainnya pada mekanik pada saat menangani kerusakan sepeda motor sehingga melakukan kesalahan diagnosa kerusakan?

1.3 Rumusan Masalah

Perumusan masalah yang mendasari pembuatan Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana cara membantu mekanik dalam memudahkan mendiagnosa

kerusakan sepeda motor manual *non injeksi*?

2. Bagaimana pembuatan aplikasi sistem pakar untuk mendiagnosa kerusakan sepeda motor manual *non injeksi*?

1.4 Tujuan Penelitian

Berdasarkan permasalahan yang di teliti, maka tujuan dari penulisan tugas akhir ini adalah:

1. Sistem yang dirancang dapat digunakan oleh para pengguna sepeda motor yang awam untuk mengetahui kerusakan sepeda motor yang di miliknya dengan segera tanpa bantuan seorang pakar.
2. Sistem yang dirancang dapat mempercepat dan mengurangi kesalahan mekanik dalam mendiagnosa kerusakan sepeda motor.

2. LANDASAN TEORI

2.1 Sistem Pakar

Sistem pakar adalah suatu program komputer yang dirancang untuk mengambil keputusan seperti keputusan yang diambil oleh seorang atau beberapa orang pakar. Menurut Marimin (1992), sistem pakar adalah sistem perangkat lunak komputer yang menggunakan ilmu, fakta, dan teknik berpikir dalam pengambilan keputusan untuk menyelesaikan masalah-masalah yang biasanya hanya dapat diselesaikan oleh tenaga ahli dalam bidang yang bersangkutan.

Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu.

2.2 Modul Penyusun Sistem Pakar

Suatu sistem pakar disusun oleh tiga modul utama (Staugaard, 1987), yaitu:

- a. Modul Penerimaan Pengetahuan (*Knowledge Acquisition Mode*)
Sistem berada pada modul ini, pada saat Ia menerima pengetahuan dari pakar. Proses mengumpulkan pengetahuan-pengetahuan yang akan digunakan untuk pengembangan sistem, dilakukan dengan bantuan *knowledge engineer*. Peran

knowledge engineer adalah sebagai penghubung antara suatu sistem pakar dengan pakarnya.

- b. Modul Konsultasi (*Consultation Mode*)
Pada saat sistem berada pada posisi memberikan jawaban atas permasalahan yang diajukan oleh user, sistem pakar berada dalam modul konsultasi. Pada modul ini, *user* berinteraksi dengan sistem dengan menjawab pertanyaan-pertanyaan yang diajukan oleh sistem.
- c. Modul Penjelasan (*Explanation Mode*)
Modul ini menjelaskan proses pengambilan keputusan oleh sistem (bagaimana suatu keputusan dapat diperoleh).

2.3 Struktur Sistem Pakar

Komponen utama pada struktur sistem pakar (Hu et al, 1987) meliputi:

- a. Basis Pengetahuan (*Knowledge Base*)
Basis pengetahuan merupakan inti dari suatu sistem pakar, yaitu berupa representasi pengetahuan dari pakar. Basis pengetahuan tersusun atas fakta dan kaidah. Fakta adalah informasi tentang objek, peristiwa, atau situasi. Kaidah adalah cara untuk membangkitkan suatu fakta baru dari fakta yang sudah diketahui. Menurut Gondran (1986) dalam Utami (2002), basis pengetahuan merupakan representasi dari seorang pakar, yang kemudian dapat dimasukkan kedalam bahasa pemrograman khusus untuk kecerdasan buatan (misalnya PROLOG atau LISP) atau shell sistem pakar (misalnya EXSYS, PC-PLUS, CRYSTAL, dsb.)
- b. Mesin Inferensi (*Inference Engine*)
Mesin inferensi berperan sebagai otak dari sistem pakar. Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi penalaran dan strategi pengendalian. Strategi penalaran terdiri dari strategi penalaran pasti (*Exact Reasoning*) dan strategi penalaran tak pasti (*Inexact*

Reasoning). *Exact reasoning* akan dilakukan jika semua data yang dibutuhkan untuk menarik suatu kesimpulan tersedia, sedangkan *inexact reasoning* dilakukan pada keadaan sebaliknya. Strategi pengendalian berfungsi sebagai panduan arah dalam melakukan prose penalaran. Terdapat tiga teknik pengendalian yang sering digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik pengendalian tersebut.

d. Basis Data (*Database*)

Basis data terdiri atas semua fakta yang diperlukan, dimana fakta-fakta tersebut digunakan untuk memenuhi kondisi dari kaidah-kaidah dalam sistem. Basis data menyimpan semua fakta, baik fakta awal pada saat sistem mulai beroperasi, maupun fakta-fakta yang diperoleh pada saat proses penarikan kesimpulan sedang dilaksanakan. Basis data digunakan untuk menyimpan data hasil observasi dan data lain yang dibutuhkan selama pemrosesan.

e. Antarmuka Pemakai (*User Interface*)

Fasilitas ini digunakan sebagai perantara komunikasi antara pemakai dengan sistem

2.4 Metode Certainty Factor (CF)

Faktor kepastian (*Certainty Factor*) menyatakan kepercayaan dalam sebuah kejadian atau fakta atau hipotesis berdasarkan bukti atau penilaian pakar. *Certainty factor* menggunakan suatu nilai untuk mengasumsi derajat keyakinan seorang pakar terhadap suatu data.

2.5 Formula Certainty Factor

Certainty factor memperkenalkan konsep keyakinan dan ketidakyakinan yang kemudian diformulakan dalam rumusan dasar sebagai berikut:

$$CF [H,E] = MB[H,E] - MD[H,E]$$

Keterangan:

CF[H,E] : *Certainty factor* hipotesa yang dipengaruhi oleh *evidence* e diketahui dengan pasti

MB[H,E] : *Measure of belief* terhadap hipotesa H, jika diberikan *evidence* E (antara 0 dan 1)

MD : *Measure of Disbelief* (Nilai Ketidakpercayaan)

P : *Probability*

E : *Evidence* (Peristiwa/Fakta)

Formula dasar digunakan apabila belum ada nilai CF untuk setiap gejala yang menyebabkan penyakit. Kombinasi *certainty factor* yang digunakan untuk mengdiagnosa penyakit adalah (Turban: 2005):

1. *Certainty Factor* untuk kaidah dengan premis/gejala tunggal (*single premis rules*):

$$CF \text{ gejala} = CF[\text{user}] * CF[\text{pakar}]$$

2. Apabila terdapat kaidah dengan kesimpulan yang serupa (*similiary concluded rules*) atau lebih dari satu gejala, maka CF selanjutnya dihitung dengan persamaan:

$$CF \text{ combine} = CF \text{ old} + CF \text{ gejala} * (1 - CF \text{ old})$$

3. Sedangkan untuk menghitung persentase terhadap penyakit, digunakan persamaan:

$$CF \text{ persentase} = CF \text{ combane} * 100$$

Untuk menentukan keterangan faktor keyakinan dari pakar, dilihat dari *CF combine* dengan berpedoman dari tabel interpretasi (*term*) *certainty factor*.

2.6 Proses Hitung Manual atau Implementasi Metode Certainty Factor (CF)

Pada sesi diagnosa penyakit, user diberikan pilihan interpretasi yang masing-masing memiliki nilai CF sebagai berikut:

Tabel 2.1 Interpretasi Sesi Diagnosa Penyakit

No	Certainty Term	CF akhir
1	Tidak Yakin	0
2	Kurang Yakin	0.2
3	Kurang Yakin	0.3
4	Sedikit Yakin	0.4
5	Sedikit Yakin	0.5
6	Cukup Yakin	0.6
7	Cukup Yakin	0.7
8	Yakin	0.8
9	Yakin	0.9
9	Sangat Yakin	1.0

Proses penghitungan prosentase keyakinan diawali dengan pemecahan sebuah kaidah (rule) yang memiliki gejala majemuk, menjadi kaidah-kaidah (rules) yang memiliki gejala tunggal. Kemudian masing-masing rule baru

dihitung CF nya dengan menggunakan persamaan 2:

$$CF \text{ gejala} = CF(\text{user}) * CF(\text{pakar})$$

Namun apabila terdapat lebih dari satu gejala, maka CF penyakit dihitung dengan menggunakan persamaan 3:

$$CF \text{ combine} = CF \text{ old} + CF \text{ gejala} * (1 - CF \text{ old})$$

Sebagai contoh kasus adalah proses pemberian nilai CF pada setiap gejala hingga perolehan prosentase keyakinan untuk penyakit paru-paru Pneumonia Bakterialis.

Kaidah 1:

IF Demam Tinggi (Gejala 1)
AND Nyeri Dada (Gejala 2)
AND Napas Pendek-Pendek (Gejala 3)
AND Batuk Yang Dimulai Dengan Cepat (Gejala 4)
THEN Pneumonia Bakterialis

Langkah pertama, pakar menentukan nilai CF untuk masing-masing gejala. Adapun nilai CF yang diberikan pakar misalnya:

CF pakar (Gejala 1) = 0.4
CF pakar (Gejala 2) = 0.7
CF pakar (Gejala 3) = 0.7
CF pakar (Gejala 4) = 0.6

Misalkan user memilih jawaban sebagai berikut:

Gejala 1 = (Ya) Cukup Yakin = 0.6
Gejala 2 = (Ya) Kurang Yakin = 0.2
Gejala 3 = (Ya) Yakin = 0.8
Gejala 4 = (Ya) Sedikit Yakin = 0.4

Kaidah awal (kaidah 1) yang memiliki 4 gejala dipecah menjadi kaidah yang memiliki gejala tunggal, sehingga menjadi:

Kaidah 1.1:
IF Demam Tinggi (Gejala 1)
THEN Pneumonia Bakterialis

Kaidah 1.2:
IF Nyeri Dada (Gejala 2)
THEN Pneumonia Bakterialis

Kaidah 1.3:
IF Napas Pendek-Pendek (Gejala 3)

THEN Pneumonia Bakterialis

Kaidah 1.4:

IF Batuk Yang Dimulai Dengan Cepat (Gejala 4)

THEN Pneumonia Bakterialis

Proses perhitungan manual untuk kaidah 1.1:

$$CF \text{ gejala1} = CF(\text{user}) * CF(\text{pakar}) \\ = 0.6 * 0.4 \\ = 0.240$$

Proses perhitungan manual untuk kaidah 1.2:

$$CF \text{ gejala2} = CF(\text{user}) * CF(\text{pakar}) \\ = 0.2 * 0.7 \\ = 0.140$$

Proses perhitungan manual untuk kaidah 1.3:

$$CF \text{ gejala3} = CF(\text{user}) * CF(\text{pakar}) \\ = 0.8 * 0.7 \\ = 0.560$$

Proses perhitungan manual untuk kaidah 1.4:

$$CF \text{ gejala4} = CF(\text{user}) * CF(\text{pakar}) \\ = 0.4 * 0.6 \\ = 0.240$$

Dikarenakan terdapat lebih dari satu gejala, maka untuk menentukan CF penyakit selanjutnya digunakan persamaan 3:

$$CF \text{ combine1} (CF \text{ gejala1}, CF \text{ gejala2}) = \\ CF \text{ gejala1} + CF \text{ gejala2} * (1 - CF \text{ gejala1}) \\ = 0.240 + 0.140 * (1 - 0.240) \\ CF \text{ old1} = 0.289 \\ CF \text{ combine2} (CF \text{ old1}, CF \text{ gejala3}) = CF \\ \text{old1} + CF \text{ gejala3} * (1 - CF \text{ old1}) = 0.289 \\ + 0.560 * (1 - 0.289) \\ CF \text{ old2} = 0.604 \\ CF \text{ combine3} (CF \text{ old2}, CF \text{ gejala4}) = CF \\ \text{old2} + CF \text{ gejala4} * (1 - CF \text{ old2}) = 0.604 \\ + 0.240 * (1 - 0.604) \\ CF \text{ old3} = 0.334$$

Keterangan:

CFold terakhir merupakan CFpenyakit, berdasarkan hasil perhitungan CF di atas, maka CF penyakit adalah 0.334. Selanjutnya hitung persentase keyakinan terhadap penyakit dengan persamaan

$$\text{Persentase} = CF \text{ penyakit} * 100 \\ = 0.334 * 100 \\ = 33.4 \%$$

Berdasarkan hasil perhitungan, maka keterangan tingkat keyakinan berdasarkan tabel interpretasi adalah sedikit yakin.

3. ANALISIS DAN PERANCANGAN

3.1 Analisa Permasalahan

Sepeda motor merupakan kendaraan bermotor beroda dua, atau tiga tanpa rumah-rumah, baik dengan atau tanpa kereta samping. Berdasarkan dari segi transmisinya sepeda motor terbagi jadi dua yaitu transmisi manual dan transmisi otomatis.

Transmisi manual biasanya digunakan pada motor bebek dan motor sport, tipe transmisi manual ini mengharuskan untuk mengatur perpindahan gear ketika ingin menambah atau mengurangi tenaga mesin motor. Sedangkan transmisi otomatis biasanya digunakan pada motor matic, tipe transmisi otomatis ini memudahkan Anda dalam berkendara karena Sistem penggerak roda yang berjalan secara otomatis atau biasa disebut dengan *Continuously Variable Transmission* (CVT) akan menyebabkan perpindahan gear tanpa perlu dipicu secara manual.

Seiring perkembangan teknologi di bidang otomotif sistem bahan bakar pada sepeda motor terbagi jadi dua yaitu sistem *injeksi* dan sistem karburator (*noninjeksi*).

Perbedaan antara sistem karburator (*non injeksi*) dan *injeksi* adalah hanya pada proses penghisapan bensin ke ruang bakar. Pada Sistem *Injeksi* sudah menggunakan peranti elektronik seperti injektor, yang tugasnya menyemprotkan bensin ke ruang bakar. Sedangkan pada karburator yang masih mengandalkan hisapan yang diperoleh dari pergerakan piston pada silinder. Sistem *injeksi* bisa menyeimbangkan volume bensin yang disemprotkan ke ruang bakar dengan kebutuhan mesin, sehingga didapat hasil pembakaran yang efisien. Jadi Sistem *Injeksi* sudah tentu akan lebih irit.

Sebagian besar masyarakat sekarang telah menjadikan sepeda motor sebagai sarana transportasi utama dalam melakukan aktifitas sehari-hari. Tetapi banyak pengendara sepeda motor yang tidak mengetahui kendala kerusakan yang dialami oleh sepeda motor tersebut, Masalah bagi pengendara yang tidak mengetahui Jenis kerusakan, akan sangat fatal apabila Jenis kerusakan tersebut tidak segera ditangani.

Saat ini mekanik membutuhkan waktu yang cukup lama untuk mendiagnosa kerusakan sepeda motor, bahkan mekanik kadang-kadang melakukan kesalahan diagnosa karena faktor kelupaan atau faktor lainnya.

Dengan menggunakan metode sistem pakar, diharapkan kemampuan seorang pakar yang ahli dalam bidang otomotif, khususnya sepeda motor bisa disubstitusikan ke komputer dalam bentuk program sehingga dapat digunakan sebagai alat bantu bagi mekanik dalam mendiagnosa kerusakan atau bagi para pengguna sepeda motor yang awam. Dalam mendiagnosa kerusakan sepeda motor sering kali dihadapkan oleh ketidakpastian dalam menganalisis sebuah informasi. Oleh karena itu penerapan metode *certainy factor* cocok untuk diterapkan dalam sistem pakar ini, karena metode ini dapat mengakomodasi ketidakpastian menjadi sebuah nilai yang menggambarkan tingkat keyakinan seorang pakar.

Sistem pakar yang akan dibangun merupakan sebuah aplikasi sistem pakar berbasis web, sebab dengan mengembangkan aplikasi berbasis web dapat membantu pemilik sepeda motor dalam melakukan diagnosis kerusakan sepeda motornya dimana saja dan kapan saja

3.2 Perancangan Sistem

Dalam melakukan perancangan sistem pakar ada beberapa tahap yang harus dilakukan, agar aplikasi yang dibuat dapat berfungsi sesuai dengan yang diharapkan.

3.2.1 Desain Arsitektur

Desain arsitektur dapat dilihat pada Gambar 3.1 yang menggambarkan hubungan antara elemen-elemen utama dari sistem pakar diagnosis kerusakan sepeda motor manual *non injeksi* yang akan dibuat.



Gambar 3.1 Blok Diagram Sistem Pakar Kerusakan Sepeda Motor Manual Noninjeksi

Penjelasan dari desain arsitektur untuk sistem pakar diagnosa kerusakan sepeda motor manual non injeksi adalah sebagai berikut:

- a. *User*
User dalam sistem pakar ini adalah orang yang memasukkan jawaban dari pertanyaan-pertanyaan berupa fakta-fakta gejala yang terjadi pada sepeda motor. Nilai dari jawaban tersebut nantinya akan diolah untuk mendapatkan suatu kesimpulan.
- b. *Konversi Nilai*
Proses konversi nilai merupakan proses dalam melakukan konversi jawaban dari *user* menjadi sebuah nilai tertentu yang nantinya akan diolah dalam proses *inferensi*.
- c. *Knowledge Base*
Knowledge base berisi kumpulan dari fakta-fakta mengenai situasi, kondisi atau permasalahan yang ada; dan aturan-aturan yang digunakan sebagai acuan dalam menggunakan pengetahuan untuk menyelesaikan masalah yang ada. Dalam sistem pakar diagnosa kerusakan sepeda motor manual *non injeksi* ini, fakta dan aturan yang ada telah di desain berupa data gejala kerusakan, data kerusakan, dan data saran pengobatan terhadap penyakit kulit.
- d. *Inference Engine*
Mesin Inferensi adalah sebuah program untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan *rule*, model, dan fakta yang disimpan dalam *knowledge base* untuk mencapai solusi atau kesimpulan. Dalam sistem pakar ini dapat dilihat dalam perhitungan *certainty factor*.

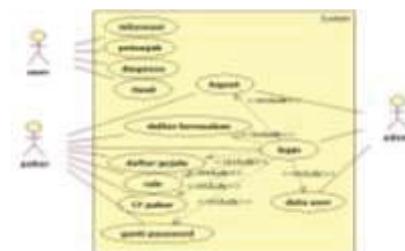
Proses dalam mesin inferensi ini dimulai dengan inputan jawaban pertanyaan gejala-gejala yang dialami sepeda motor. Setelah semua pertanyaan dijawab, maka jawaban tersebut akan dikonversi dari sebuah "*certain term*" menjadi sebuah nilai CF. Nilai hasil konversi nantinya akan dikalikan dengan nilai CF pakar. Setelah mendapatkan hasil perkalian nilai CF gejala, maka akan dilakukan verifikasi apakah nilai tersebut harus dikombinasi untuk menghasilkan sebuah nilai atau tidak.

Setelah semua hasil perhitungan nilai CF gejala didapatkan, maka proses selanjutnya melakukan perkalian dengan nilai CF *rule* dari setiap kerusakan sepeda motor. Nilai CF kerusakan pertama diperoleh dari perhitungan nilai CF berdasarkan CF *rule* kerusakan pertama. Proses perhitungan nilai CF dari kerusakan diawali dengan mencari nilai CF dari gejala pertama dan gejala kedua, setelah itu kedua nilai CF tersebut akan dikombinasikan. Hasil dari nilai CF kombinasi pertama nantinya akan dikombinasikan dengan nilai CF *rule* dari gejala berikutnya. Proses perhitungan nilai CF kombinasi akan diulang sampai dengan gejala terakhir dari kerusakan pertama. Setelah perhitungan nilai CF dari kerusakan pertama selesai, maka proses perhitungan akan dilanjutkan sampai dengan kerusakan terakhir. Apabila nilai semua kerusakan ditemukan, maka proses perhitungan telah selesai dan akan menampilkan nilai CF dari semua kerusakan sepeda motor beserta saran perbaikan yang harus dilakukan.

- e. *Output*
Output merupakan hasil kesimpulan dari sistem yang menunjukkan jawaban dari gejala atau fakta mengenai kerusakan sepeda motor yang telah di *input*-kan. *Output* yang dihasilkan sistem pakar ini merupakan hasil diagnosis kerusakan sepeda motor manual noninjeksi beserta saran perbaikan yang harus dilakukan

3.2.2 Use Case Diagram

Model use case menjelaskan mengenai aktor-aktor yang terlibat dengan perangkat lunak yang dibangun beserta proses-proses yang ada didalamnya. Diagram *use case* dari Aplikasi Sistem pakar diagnosa kerusakan sepeda motor manual *non injeksi* adalah sebagai berikut:



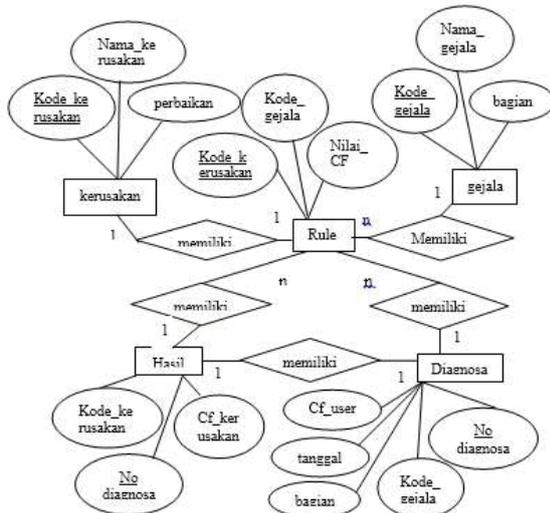
Gambar 3.2 Use Case Diagram SP Kerusakan Motor

Tabel 3.1 Definisi Aktor

No	Aktor	Deskripsi
1	User	Mencari solusi melalui sistem aplikasi sistem pakar diagnosa kerusakan sepeda motor manual <i>non injeksi</i> .
2	Pakar	Melakukan tugas <i>login</i> dan memanipulasi (Insert, Update, dan Delete) semua basis pengetahuan dari sistem pakar diagnosa kerusakan sepeda motor manual <i>non injeksi</i> .
3	Admin	Melakukan tugas <i>login</i> dan memanipulasi (Insert, Update, dan Delete) data <i>user</i>

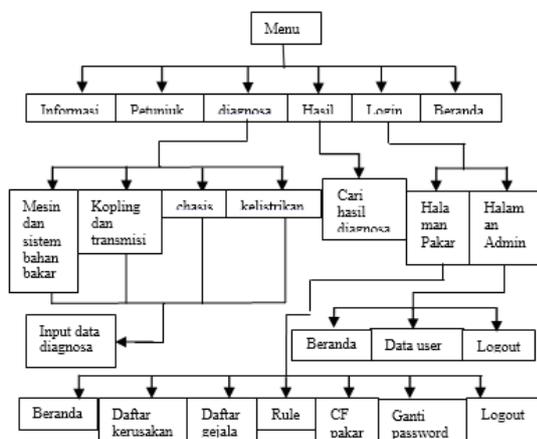
3.2.3 Entity Relationship Diagram (ERD)

Entity Relationship diagram (ERD) adalah diagram yang digunakan untuk merancang suatu basis data, untuk memperlihatkan hubungan atau relasi antar entitas atau objek yang terlihat beserta atributnya. Berikut ini merupakan Entity Relationship diagram (ERD) basis data dari sistem pakar diagnosa kerusakan sepeda motor manual noninjeksi.



Gambar 3.3 ERD Sistem Pakar Diagnosa Kerusakan Sepeda Motor Manual Noninjeksi

3.2.3 Struktur Navigasi Menu



Gambar 3.4 Struktur Navigasi Menu

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi sistem merupakan kelanjutan dari kegiatan perancangan sistem dan dapat disebut sebagai usaha untuk mewujudkan sistem yang dirancang. Langkah-langkah dari proses implementasi sistem adalah urutan dari kegiatan awal sampai kegiatan yang dilakukan dalam mewujudkan sistem yang dirancang.

4.1.1 Batasan Implementasi

Dalam mengimplementasikan perangkat lunak, pemantauan, dan pengendalian aplikasi sistem pakar diagnosa kerusakan sepeda motor manual noninjeksi berbasis web ini terdapat beberapa hal yang menjadi batasan implementasi, yaitu:

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang harus disiapkan dalam implementasi sistem pakar diagnosa kerusakan sepeda motor manual non-injeksi berbasis web yaitu melakukan *development tools* yang di lakukan diantaranya:

1. *Notepad++*
Notepad++ yang digunakan sebagai *text editor* untuk membuat *source code* PHP dalam pengembangan sistem pakar diagnosa kerusakan sepeda motor manual noninjeksi berbasis web ini.
2. *Xampp*
Xampp yang digunakan sebagai server yang berdiri sendiri (*localhost*), MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP.
3. *Mozilla Firefox*
Mozilla Firefox adalah *web browser* digunakan untuk membuka halaman web yang di buat.

4.1.3 Implementasi Perangkat Keras

Perangkat keras yang digunakan di sistem pakar diagnosa kerusakan sepeda motor manual noninjeksi berbasis web ini berdasarkan spesifikasi kebutuhan yang harus dipenuhi untuk menjalankan sistem pakar tersebut di antaranya:

1. Harddisk minimal 20 GB,
2. RAM 512 MB,
3. Processor minimum Pentium IV,
4. Monitor color 14'',
5. Mouse,
6. Keyboard,

7. Printer (opsional)

4.1.4 Implementasi Antarmuka

Sistem pakar diagnosa kerusakan sepeda motor manual noninjeksi berbasis web ini terdiri dari tampilan-tampilan dan proses yang berhubungan dengan sistem, untuk memperjelas sistem pakar tersebut maka dapat dilihat pada gambar berikut:

1. Index (halaman utama)

Halaman utama terdapat menu beranda, login, informasi, petunjuk, diagnose dan hasil.



Gambar 4.1 Halaman Index

2. Halaman diagnosa

Halaman diagnosa menampilkan pilihan bagian sepeda motor yang akan didiagnosa



Gambar 4.2 Halaman Diagnosa

3. Halaman input data diagnosa

Halaman input data diagnose menampilkan pertanyaan yang harus diisi oleh user sesuai dengan kondisi sepeda motor yang akan didiagnosa



Gambar 4.3 Input Data Diagnosa

4. Halaman hasil diagnosa

Halaman hasil menampilkan hasil diagnose.



Gambar 4.4 Hasil Diagnosa

5. Login

Halaman menampilkan form login untuk memasuki halaman utama pakar atau halaman utama admin sesuai level user yang di masukan.



Gambar 4.5 Login

6. Pakar index (halaman utama pakar)

Halaman utama pakar bisa diakses setelah melakukan login dengan level user pakar, pada halaman ini terdapat menu beranda, logout, daftar kerusakan, daftar gejala, rule, CF pakar, dan ganti password



Gambar 4.6 Pakar Index

7. Admin index (halaman utama admin)

Halaman utama admin bisa diakses setelah melakukan login dengan level user admin,

pada halaman ini terdapat menu beranda, logout, dan data user.



Gambar 4.7 Admin Index

4.2 Pengujian

Pengujian merupakan bagian yang penting dalam siklus pengembangan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Pengujian perangkat lunak ini menggunakan metode pengujian *black box*. Pengujian *black box* ini tidak perlu tahu apa yang sesungguhnya terjadi dalam sistem atau perangkat lunak, yang diuji adalah masukan serta keluarannya. Dengan berbagai masukan yang diberikan, apakah sistem atau perangkat lunak memberikan keluaran seperti yang kita harapkan atau tidak.

4.2.1 Rencana Pengujian

Pengujian sistem pakar ini menggunakan data uji berupa sebuah data masukan dari *user*.

Tabel 4.1 Rencana Pengujian Halaman *User*

No	Kode Use Case	Nama Use Case	Requirement yang di uji	Butir uji	Jenis pengujian
1	UC001	Informasi	Halaman informasi	User mengakses halaman informasi	<i>Black Box</i>
2	UC002	Petunjuk	Halaman petunjuk	User mengakses halaman petunjuk	<i>Black Box</i>
3	UC003	Diagnosa	Halaman diagnosa	User melakukan diagnosa kerusakan	<i>Black Box</i>
4	UC004	Hasil	Halaman hasil	User melakukan pencarian hasil diagnosa	<i>Black Box</i>

Tabel 4.2 Rencana Pengujian Halaman Pakar

No	Kode Use Case	Nama Use Case	Requirement yang di uji	Butir uji	Jenis pengujian
1	UC005	<i>Login</i>	Halaman <i>login</i>	Pakar melakukan login untuk mengakses halaman utama pakar	<i>Black Box</i>

2	UC006	<i>Logout</i>	Tombol <i>logout</i>	Pakar melakukan <i>logout</i>	<i>Black Box</i>
3	UC007	Data kerusakan	Tambah data kerusakan	Pakar melakukan penambahan data kerusakan	<i>Black Box</i>
			Hapus data kerusakan	Pakar melakukan penghapusan data kerusakan	
			Edit data kerusakan	Pakar melakukan edit data kerusakan	
4	UC008	Data gejala	Tambah data gejala	Pakar melakukan penambahan data gejala	<i>Black Box</i>
			Hapus data gejala	Pakar melakukan penghapusan data gejala	
			Edit data gejala	Pakar melakukan edit data gejala	
5	UC009	Rule	Buat rule kerusakan dan gejala	Pakar membuat rule antara kerusakan dan gejala	<i>Black Box</i>
			Edit rule	Pakar melakukan edit rule	
6	UC010	CF pakar	Memasukkan nilai CF pakar	Pakar memasukkan nilai CF pakar	<i>Black Box</i>
			Edit nilai CF pakar	Pakar melakukan edit nilai CF pakar	
7	UC011	Ganti password	Merubah password	Pakar melakukan perubahan password	<i>Black Box</i>

Tabel 4.3 Rencana Pengujian Halaman Admin

No	Kode Use Case	Nama Use Case	Requirement yang di uji	Butir uji	Jenis pengujian
1	UC005	<i>Login</i>	Halaman <i>login</i>	admin melakukan login untuk mengakses halaman utama admin	<i>Black Box</i>
2	UC006	<i>Logout</i>	Tombol <i>logout</i>	Admin melakukan <i>logout</i>	<i>Black Box</i>
3	UC012	Data <i>user</i>	Tambah data <i>user</i>	Admin melakukan penambahan data <i>user</i>	<i>Black Box</i>
			Edit data <i>user</i>	Admin melakukan edit data <i>user</i>	
			Hapus data <i>user</i>	Admin melakukan penghapusan data <i>user</i>	

Tabel 4.4 Rencana Pengujian Kesesuaian dengan Pakar

No	Requirement yang di uji	Butir uji	Jenis pengujian
1	Diagnosa bagian mesin dan sistem bahan bakar	Pakar melakukan diagnosa bagian mesin dan sistem bahan bakar secara manual hasilnya akan dibandingkan dengan hasil dari sistem.	<i>Black Box</i>
2	Diagnosa bagian kopling dan transmisi	Pakar melakukan diagnosa bagian kopling dan transmisi secara manual hasilnya akan dibandingkan dengan hasil dari sistem.	<i>Black Box</i>
3	Diagnosa bagian chasis	Pakar melakukan diagnosa bagian chasis secara manual hasilnya akan dibandingkan dengan hasil dari sistem.	<i>Black Box</i>

4	Diagnosa bagian kelistrikan	Pakar melakukan diagnosa bagian kelistrikan secara manual hasilnya akan di bandingkan dengan hasil dari sistem.	Black Box
---	-----------------------------	-----------------------------------------------------------------------------------------------------------------	-----------

4.2.2 Kasus dan Hasil Pengujian

Berikut beberapa kasus pengujian dan hasil dari pengujiannya:

Tabel 4.5 Hasil Pengujian Halaman User

Kode Use Case	Butir uji	Skenario uji	Hasil yang diharapkan	Hasil pengujian
UC001	User mengakses halaman informasi	User mengakses halaman informasi (jika benar)	Sistem menampilkan halaman informasi	Sesuai
	User mengakses halaman informasi	User mengakses halaman informasi (jika salah)	Sistem tidak menampilkan halaman informasi	Sesuai
UC002	User mengakses halaman petunjuk	User mengakses halaman petunjuk (jika benar)	Sistem menampilkan halaman petunjuk	Sesuai
	User mengakses halaman petunjuk	User mengakses halaman petunjuk (jika salah)	Sistem tidak menampilkan halaman petunjuk	Sesuai
UC003	User melakukan diagnosa kerusakan	User melakukan diagnosa kerusakan (jika benar)	Data diagnosa dan data hasil diagnosa tersimpan di database dan menampilkan hasil diagnosa	Sesuai
	User melakukan diagnosa kerusakan	User melakukan diagnosa kerusakan (jika salah)	Data diagnosa dan data hasil diagnosa tidak tersimpan di database dan tidak menampilkan hasil diagnosa	Sesuai
UC004	User melakukan pencarian hasil diagnosa	User melakukan pencarian hasil diagnosa (jika benar)	Tampil hasil diagnosa yang di cari	Sesuai
	User melakukan pencarian hasil diagnosa	User melakukan pencarian hasil diagnosa (jika salah)	Tampil konfirmasi data tidak ditemukan	Sesuai

Tabel 4.6 Hasil Pengujian Halaman Pakar

Kode Use Case	Butir uji	Skenario uji	Hasil yang diharapkan	Hasil pengujian
UC005	Pakar melakukan login untuk mengakses halaman utama pakar	Pakar melakukan login untuk mengakses halaman utama pakar (jika benar)	Tampil halaman utama pakar	Sesuai
	Pakar melakukan login untuk mengakses halaman utama pakar	Pakar melakukan login untuk mengakses halaman utama pakar (jika salah)	Tampil konfirmasi gagal login	Sesuai

UC006	Pakar melakukan logout	Pakar melakukan logout (jika benar)	Keluar dari halaman pakar	Sesuai
	Pakar melakukan logout	Pakar melakukan logout (jika salah)	Tidak keluar dari halaman pakar	Sesuai
UC007	Pakar melakukan penambahan data kerusakan	Pakar melakukan penambahan data kerusakan (jika benar)	Data tersimpan dalam database dan tampil data kerusakan	Sesuai
		Pakar melakukan penambahan data kerusakan (jika salah)	Data tidak tersimpan dalam database dan tidak tampil data kerusakan	Sesuai
	Pakar melakukan penghapusan data kerusakan	Pakar melakukan penghapusan data kerusakan (jika benar)	Tampil konfirmasi untuk menghapus dan data kerusakan terhapus dari database	Sesuai
		Pakar melakukan penghapusan data kerusakan (jika salah)	Tampil konfirmasi data tidak bisa di hapus	Sesuai
Pakar melakukan edit data kerusakan	Pakar melakukan edit data kerusakan (jika benar)	Tampil konfirmasi perubahan data berhasil di simpan dan data di database berubah	Sesuai	
	Pakar melakukan edit data kerusakan (jika salah)	Data di database tidak berubah	Sesuai	
UC008	Pakar melakukan penambahan data gejala	Pakar melakukan penambahan data gejala (jika benar)	Data tersimpan dalam database dan tampil data gejala	Sesuai
		Pakar melakukan penambahan data gejala (jika salah)	Data tidak tersimpan dalam database dan tidak tampil data gejala	Sesuai
	Pakar melakukan penghapusan data gejala	Pakar melakukan penghapusan data gejala (jika benar)	Tampil konfirmasi untuk menghapus dan data gejala terhapus dari database	Sesuai
		Pakar melakukan penghapusan data gejala (jika salah)	Tampil konfirmasi data tidak bisa di hapus	Sesuai
	Pakar melakukan edit data gejala	Pakar melakukan edit data gejala (jika benar)	Tampil konfirmasi perubahan data berhasil di simpan dan data di database berubah	Sesuai
		Pakar melakukan edit data gejala (jika salah)	Data di database tidak berubah	Sesuai

UC009	Pakar membuat rule antara kerusakan dan gejala	Pakar membuat rule antara kerusakan dan gejala (jika benar)	Data rule tersimpan dalam database dan tampil data rule	Sesuai
		Pakar membuat rule antara kerusakan dan gejala (jika salah)	Data rule tidak tersimpan dalam database dan tidak tampil data rule	Sesuai
	Pakar melakukan edit rule	Pakar melakukan edit rule (jika benar)	Tampil konfirmasi perubahan data berhasil di simpan dan data di database berubah	Sesuai
		Pakar melakukan edit rule (jika salah)	Data di database tidak berubah	Sesuai
UC010	Pakar memasukkan nilai CF pakar	Pakar memasukkan nilai CF pakar (jika benar)	Nilai CF pakar tersimpan dalam database dan tampil nilai CF pakar	Sesuai
		Pakar memasukkan nilai CF pakar (jika salah)	Nilai CF pakar tidak tersimpan dalam database dan tidak tampil nilai CF pakar	Sesuai
	Pakar melakukan edit nilai CF pakar	Pakar melakukan edit nilai CF pakar (jika benar)	Tampil konfirmasi perubahan data berhasil di simpan dan data di database berubah	Sesuai
		Pakar melakukan edit nilai CF pakar (jika salah)	Data di database tidak berubah	Sesuai
UC011	Pakar melakukan perubahan password	Pakar melakukan perubahan password (jika benar)	Password di database berubah dan tampil konfirmasi password berhasil di rubah	Sesuai
		Pakar melakukan perubahan password (jika salah)	Tampil peringatan pada setiap kesalahan	Sesuai

Tabel 4.7 Hasil Pengujian Halaman Admin

Kode Use Case	Butir uji	Skenario uji	Hasil yang diharapkan	Hasil pengujian
UC005	admin melakukan login untuk mengakses halaman utama admin	admin melakukan login untuk mengakses halaman utama admin (jika benar)	Tampil halaman utama admin	Sesuai
		admin melakukan login untuk mengakses halaman utama admin (jika salah)	Tampil konfirmasi gagal login	Sesuai

UC006	Admin melakukan logout	Admin melakukan logout (jika benar)	Keluar dari halaman admin	Sesuai
		Admin melakukan logout (jika salah)	Tidak keluar dari halaman admin	Sesuai
UC012	Admin melakukan penambahan data user	Admin melakukan penambahan data user (jika benar)	Data tersimpan dalam database dan tampil data user	Sesuai
		Admin melakukan penambahan data user (jika salah)	Data tidak tersimpan dalam database dan tidak tampil data user	Sesuai
	Admin melakukan edit data user	Admin melakukan edit data user (jika benar)	Tampil konfirmasi perubahan data berhasil di simpan dan data di database berubah	Sesuai
		Admin melakukan edit data user (jika salah)	Data di database tidak berubah	Sesuai
	Admin melakukan penghapusan data user	Admin melakukan penghapusan data user (jika benar)	Tampil konfirmasi untuk menghapus dan data user terhapus dari database	Sesuai
		Admin melakukan penghapusan data user (jika salah)	Data di database tidak terhapus	Sesuai

Tabel 4.8 Hasil Pengujian Kesesuaian dengan Pakar

Butir uji	Skenario uji	Hasil dari pakar	Hasil pengujian
Pakar melakukan diagnosa bagian mesin dan sistem bahan bakar dengan gejala Bila gas diputar/di perbesar mesin mati atau bahan bakar berputar secara kasar(yakin), Bila gas diputar penuh,putaran mesin tidak naik dengan cepat dan respon dari gas lambat(yakin), dan Mesin kehilangan dayanya(yakin)	Pakar melakukan diagnosa bagian mesin dan sistem bahan bakar dengan gejala Bila gas diputar/di perbesar mesin mati atau bahan bakar berputar secara kasar(yakin), Bila gas diputar penuh,putaran mesin tidak naik dengan cepat dan respon dari gas lambat(yakin), dan Mesin kehilangan dayanya(yakin)	Kerusakan pada karburator	Sesuai
	Pakar melakukan diagnosa bagian kopling dan tranmisi secara manual hasilnya akan di bandingkan dengan hasil dari sistem	Pakar melakukan diagnosa bagian kopling dan tranmisi dengan gejala Kopling selip (yakin), Kopling menahan (yakin), dan Kopling sukar untuk bekerja (yakin)	Kerusakan pada kopling

Pakar melakukan diagnosa bagian chasis secara manual hasilnya akan di bandingkan dengan hasil dari sistem	Pakar melakukan diagnosa bagian chasis dengan gejala Batang kemudi mempunyai kelonggaran berlebihan(yakin), Batang kemudi kaku (yakin), dan Sepeda motor menarik ke salah satu sisi atau kecenderungan membelok (yakin)	Kerusakan pada sistem kemudi, sistem suspensi depan dan tekanan ban harus di cek	Sesuai
Pakar melakukan diagnosa bagian kelistrikan secara manual hasilnya akan di bandingkan dengan hasil dari sistem	Pakar melakukan diagnosa bagian kelistrikan dengan gejala Lampu utama rusak (yakin), Lampu rem tidak dapat menyala (yakin), dan sekering sering putus terus (yakin)	Cek lampu, cek semua kabel kemungkinan ada yang konslet, cek juga semua saklar, dan periksa aki	Sesuai

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari skripsi yang berjudul: “Implementasi Sistem Pakar di Bidang Otomotif Untuk Mendiagnosa Kerusakan Sepeda Motor Manual Non Injeksi Menggunakan Metode Certainty Factor Berbasis Web” adalah sebagai berikut:

1. Sistem pakar ini bisa menjadi alat bantu bagi pengendara sepeda motor untuk mengetahui kendala kerusakan yang dialami oleh sepeda motornya.
2. Sistem pakar ini juga bisa menjadi alat bantu mekanik sepeda motor untuk menghindari kelupaan atau faktor lainnya saat mendiagnosa kerusakan sepeda motor.

5.2 Saran

Dalam pembuatan Sistem Pakar Untuk Mendiagnosa Kerusakan Sepeda Motor Manual Non Injeksi Menggunakan Metode Certainty Factor Berbasis Web ini masih banyak hal yang dapat dikembangkan, seperti:

1. Sistem pakar yang sudah dibuat hanya untuk mendiagnosa kerusakan sepeda motor manual dengan sistem pembakaran karburator (*non injeksi*), kedepannya bisa di kembangkan sehingga bisa mendiagnosa semua jenis sepeda motor
2. Sistem pakar ini hanya menggunakan metode *certainty factor*, kedepannya bisa di kembangkan dengan menggunakan lebih dari satu metode sehingga bisa di bandingkan metode mana yang lebih baik dalam mendiagnosa kerusakan sepeda motor.
3. Penambahan fitur untuk menghitung kemungkinan biaya perawatan kerusakan

sepeda motor yang telah di diagnosa oleh sistem.

DAFTAR PUSTAKA

1. Barasa Haidi. (2013). “*Pengertian Xampp*”. [Online]. Tersedia: <https://haidibarasa.wordpress.com/2013/07/06/pengertian-xampp/>. [23 Mei 2017]
2. Daryanto. 2002. *Teknik Reparasi Dan Perawatan Sepeda Motor*. Jakarta : Bumi Aksara
3. Deviachrista. (2013). “*Pengertian UML (Unified modeling language) dan Jenis Diagram UML*”. [Online]. Tersedia: <https://deviachrista.blogspot.co.id/2013/05/pengertian-uml-unified-modeling.html>. [23 Mei 2017]
4. Dwi Arta Permana Bayu. (2014). “*Fungsional Dari Notepad++*”. [Online]. Tersedia: <https://bayudwiarta.wordpress.com/2014/10/16/fungsional-dari-notepad/>. [22 Mei 2017]
5. Fowler, Martin. 2005. *UML Distilled*. Edisi 3. Yogyakarta: Andi
6. Hastomo Widi. (2013). “*Pengertian Dan Kelebihan Database MySQL*”. [Online]. Tersedia: <http://hastomo.net/php/pengertian-dan-kelebihan-database-mysql/>. [21 Mei 2017]
7. Rahman Taufiq. (2014). “*Pengertian Dari PHP*”. [Online]. Tersedia: <https://rahman371.wordpress.com/2014/09/19/pengertian-dari-php/>. [21 Mei 2017]
8. Reka yanti. (2015). “*Metode Certainty Factor (CF) Serta Proses Hitung Manual Contoh Kasus Diagnosa Penyakit Paru-Paru*”. [Online]. Tersedia: <http://rekayantianwar.blogspot.com/2015/09/metode-certainty-factor-cf-serta-proses.html?m=1>. [21 Mei 2017]
9. Yuliana Esti. (2011). “*Pengertian Sistem Pakar*”. [Online]. Tersedia: <http://teknikinformatika-esti.blogspot.co.id/2011/06/pengertian-sistem-pakar.html>. [21 Mei 2017]