

**PEMODELAN SISTEM PAKAR UNTUK MENGANALISIS
KERUSAKAN PADA MESIN KENDARAAN RODA 4
MENGUNAKAN METODE *FORWARD CHAINING* –
*DEMPSTER SHAFFER***

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun Oleh :
Zainur Rochim Efendi
NIM. 105060807111169



**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA
MALANG
2016**

PENGESAHAN

PEMODELAN SISTEM PAKAR UNTUK MENGANALISIS KERUSAKAN PADA
KENDARAAN RODA 4 MENGGUNAKAN METODE FORWARD CHAINING –
DEMPSTER SHAFER

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun Oleh :
Zainur Rochim Efendi
NIM: 105060807111169

Skripsi ini telah diuji dan dinyatakan lulus pada
30 Juni 2016

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Nurul Hidayat, S.Pd, M.Sc

NIP. 19680430 200212 1 001

Drs. Achmad Ridok, M.Kom

NIP. 19680825 199403 1 002

Mengetahui

Ketua Program Studi Informatika / Ilmu Komputer

Issa Arwani, S.Kom, M.Sc

NIP: 19830922 201212 1 003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta proses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang,

Zainur Rochim Efendi

NIM. 105060807111169

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pemodelan Sistem Pakar Untuk Menganalisis Kerusakan Mesin Pada Kendaraan Roda 4 Menggunakan Metode *Forward Chaining – Dempster Shafer*” ini dapat terselesaikan.

Dalam pelaksanaan dan penulisan tugas akhir ini penulis mendapatkan banyak bantuan dari berbagai pihak baik secara moril maupun materil. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Nurul Hidayat, S.Pd, M,Sc selaku dosen pembimbing I yang telah banyak memberi ide, pemikiran, bimbingan, ilmu, dan saran dalam penyusunan tugas akhir ini.
2. Drs. Achmad Ridok, M.kom selaku dosen pembimbing II yang telah memberikan masukan dan ilmu serta saran dalam menyelesaikan skripsi ini.
3. Ayah , Ibu dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian, dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
4. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D., selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
5. Issa Arwani, S.Kom, M.Sc selaku dosen pembimbing akademik yang telah memberikan bimbingan, ilmu, dan saran.
6. Segenap Bapak dan Ibu Dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.

7. Seluruh staff Akademik Program Studi Informatika / Ilmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama menulis menempuh studi dan selama penyelesaian skripsi ini.
8. Sahabat yang telah memberikan semangat kepada penulis Dian Hermansyah, Arroifi F Shodiq, Dwi Ardiansyah, Ridhofi Garish Iswara, Syailendra Orthega, Novan Listanto, Affandy, Alvin, Taufiq, Suryadi, Agung, Aan, Dini, Rani, Yana, Yani, Nike, Gaby, Rini, Cita, Risa dan yang tidak bisa disebutkan satu persatu.
9. Teman-teman Aerio lindonesia Club Surabaya (om diar, om windu, om nova, om mike, om dion, om anjik, om arian, om taufiq, om erik, om zaenal, om agus, om san dan teman-teman aic yang tidak bisa disebutkan satu persatu) yang telah memberikan semangat dan dukungan kepada penulis.
10. Teman-teman angkatan 2010 Program Studi Informatika / Ilmu Komputer yang telah memberikan segala bantuannya selama menempuh studi di Program Studi Informatika / Ilmu Komputer Universitas Brawijaya.
11. Semua pihak yang telah membantu dan berbagi ilmu dalam penyelesaian skripsi, yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan baik format penulisan maupun isinya. Oleh karena itu, saran dan kritik membangun dari para pembaca senantiasa penulis harapkan guna pengembangan diri. Semoga skripsi ini dapat memberikan manfaat bai semua pihak.

Malang,

Penulis

Zainur Rochim Efendi

ABSTRAK

Mobil menjadi salah satu kebutuhan utama untuk mempermudah aktivitas masyarakat sehari-hari. Kerusakan pada mobil mungkin terjadi seiring pemakaian mobil. Namun kepemilikan mobil tidak diimbangi dengan pengetahuan tentang kerusakan pada mesin mobil dan tindakan penanganan yang dapat dilakukan. Untuk membantu pemilik mobil dalam mengetahui kerusakan mesin yang muncul dan tindakan penanganan yang dapat dilakukan, maka diperlukan sistem diagnosa kerusakan mesin. Pada penelitian ini, sistem diagnosa kerusakan mesin mobil dikembangkan menggunakan algoritma Dempster-Shafer dengan pendekatan *forward chaining*. Basis pengetahuan memuat nilai densitas kerusakan mesin yang mungkin ada, untuk menentukan keputusan melalui perhitungan Dempster-Shafer. Berdasarkan pengujian kebutuhan pengguna (pengujian validasi perangkat lunak), persentasi pemenuhan kebutuhan sebesar 100%; sedangkan hasil pengujian akurasi sistem sebesar 92%. Hasil pengujian ini menunjukkan bahwa sistem pakar diagnosa kerusakan mesin mobil ini dapat digunakan untuk membantu pemilik mobil dalam mendiagnosa kerusakan mesin mobil.

Kata Kunci : Mesin Mobil, Sistem Pakar, Forward Chaining, Dempster Shafer



ABSTRACT

Cars became one of the main requirements to facilitate people's daily activities. Damage to the car may occur as the use of the car. However, car ownership is not matched by knowledge of the damage to the car's engine and response actions that can be done. To assist car owners in emerging engine damage, and response actions that can be done, it is necessary to diagnose system engine damage. In this research, diagnostics system damage car engines were developed using Dempster-Shafer algorithm with forward chaining approach. The knowledge base contains density values of engine damage that may exist, to determine the decision by calculating the Dempster-Shafer. Based on the testing needs of the user (software validation testing), the percentage of fulfillment of 100%; whereas the results of testing the accuracy of the system by 92%. These test results show that the diagnosis expert system damage car engines can be used to assist car owners in diagnosing damage to the engine.

Keywords: *Car engine ,Systems Experts, forward chaining, dempster shafer*



DAFTAR ISI

PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
KATA PENGANTAR	iii
ABSTRAK	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah.....	2
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan.....	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Kajian Pustaka.....	5
2.2 Pemodelan.....	5
2.2.1 Prinsip Pemodelan Sistem	6
2.2.2 Perangkat Pemodelan Sistem.....	6
2.3 Pemodelan Sistem Pakar	13
2.3.1 Struktur Sistem Pakar	14
2.3.2 Representasi Pengetahuan.....	16
2.3.3 Ciri-Ciri Sistem Pakar	17
2.3.4 Orang yang terlibat dalam sistem pakar	17
2.3.5 Kelebihan Sistem Pakar	17
2.4 Inferensi.....	18
2.5 Komponen Mobil	19
2.6 Kerusakan	19
2.7 Mekanisme Mesin	19
2.8 Mesin Inferensi.....	20
2.8.1 Foward Chaining.....	20
2.8.2 Backward Chaining	20
2.9 Teori Dempster-Shafer	21
2.10 Pengujian Sistem	24
BAB 3 METODOLOGI PENELITIAN	24
3.1 Studi Literatur	24
3.2 Pengumpulan Data	25
3.2.1 Analisa Kebutuhan Masukan	25
3.2.2 Analisa Kebutuhan Proses	27
3.2.3 Analisa Kebutuhan Keluaran	27
3.2.4 Perancangan Sistem Pakar	27
3.3 Implementasi Sistem	28
3.4 Pengujian dan Analisis	28
3.5 Penarikan Kesimpulan dan Saran	29
BAB 4 PERANCANGAN SISTEM	29
4.1 Analisis Kebutuhan Perangkat	30
4.1.1 Analisis Kebutuhan Masukan	30
4.1.2 Analisa Kebutuhan Proses	31
4.1.3 Analisa Kebutuhan Keluaran	31

4.2	Perancangan Perangkat Lunak	31
4.2.1	Entity Relationship Diagram (ERD)	31
4.2.2	Data Flow Diagram (DFD)	32
4.3	Perancangan Sistem Pakar	35
4.3.1	Akuisisi Pengetahuan.....	36
4.3.2	Basis Pengetahuan	39
4.3.3	Mesin Inferensi.....	39
4.3.4	Perhitungan Kasus Secara Manual	42
4.3.5	Daerah Kerja (Blackboard)	45
4.3.6	Perancangan Antarmuka	45
4.3.4	Perancangan Antarmuka Program	46
BAB 5 IMPLEMENTASI.....		50
5.1	Spesifikasi Sistem	51
5.1.1	Spesifikasi Perangkat Keras	51
5.1.2	Spesifikasi Perangkat Lunak	51
5.2	Batasan – Batasan Implementasi	51
5.3	Implementasi Algoritma <i>Dempster Shafer</i>	52
5.4	Implementasi Antarmuka.....	61
5.4.1	Implementasi Halaman Login	62
5.4.2	Implementasi halaman Sign Up	62
5.4.3	Implementasi Halaman Data User.....	63
5.4.4	Implementasi Halaman Gejala Kerusakan.....	63
5.4.5	Implementasi Halaman Data Kerusakan	64
5.4.6	Implementasi Halaman Basis Pengetahuan	64
5.4.7	Implementasi Halaman Riwayat Sistem	65
5.4.8	Implementasi Halaman Diagnosa.....	65
5.4.9	Implementasi Halaman Untuk User	66
BAB 6 PENGUJIAN DAN ANALISIS		67
6.1	Pengujian Sistem	67
6.1.1	Pengujian Validasi.....	67
6.1.2	Pengujian Akurasi	73
6.2	Analisa pengujian.....	82
6.2.1	Analisis Validasi	82
6.2.2	Analisis Akurasi.....	82
BAB 7 PENUTUP.....		84
7.1	Kesimpulan	84
7.2	Saran	84
DAFTAR PUSTAKA.....		85

DAFTAR TABEL

Tabel 2. 1 Simbol-simbol Konteks Diagram 1	7
Tabel 2. 2 Simbol-simbol Flowchart	11
Tabel 2. 3 Perbandingan kemampuan seorang pakar dengan sistem pakar	17
Tabel 2. 4 Range Belief dan Plausibility.....	22
Tabel 3. 1 Kebutuhan Data Penelitian.....	25
Tabel 3. 2 Daftar Kebutuhan Fungsional	26
Tabel 4. 1 Tabel Jenis-Jenis Kerusakan Mesin Kendaraan 4.....	37
Tabel 4. 2 Gejala-Gejala Kerusakan Mesin Kendaraan 4	37
Tabel 4. 3 Tabel Akusisi Kerusakan Mesin Kendaraan Roda 4.....	38
Tabel 4. 4 Tabel Nilai Densitas Kerusakan Mesin Kendaraan Roda 4.....	38
Tabel 4. 5 Aturan Kombinasi Untuk m3 Kasus 2	44
Tabel 5. 1 Spesifikasi Perangkat Keras.....	51
Tabel 5. 2 Spesifikasi Perangkat Lunak.....	51
Tabel 6. 1 Hasil Pengujian Validasi Fungsional Sistem.....	68
Tabel 6. 2 Tabel Pengujian Akurasi Hasil Diagnosa Sistem dengan Pakar	74
Tabel 6. 3 Hasil pengujian perubahan nilai densitas.....	78

DAFTAR GAMBAR

Gambar 2. 1 Terminator 1	8
Gambar 2. 2 Proses 1	8
Gambar 2. 3 Arus Data	9
Gambar 2. 4 Data Store 1	9
Gambar 2. 5 Simbol Entity	9
Gambar 2. 6 Simbol Relationship	9
Gambar 2. 7 Simbol Atribut	10
Gambar 2. 8 Struktur Sistem Pakar	15
Gambar 2. 9 Alur Metode Foward Chaining	20
Gambar 2. 10 Alur Metode Backward Chaining	21
Gambar 3. 1 Blok Diagram Metodologi Penelitian	24
Gambar 3. 2 Desain Umum Perancangan Sistem	27
Gambar 3. 3 Blok Diagram Implementasi Sistem	28
Gambar 3. 4 Blok Diagram Pengujian Validasi Sistem	28
Gambar 4. 1 Pohon Perancangan	29
Gambar 4. 2 Entity Relationship Diagram	32
Gambar 4. 3 Context Diagram	33
Gambar 4. 4 DFD Level 0	34
Gambar 4. 5 DFD Level 1 proses 1.0	34
Gambar 4. 6 Proses DFD level 1 proses 2.0	34
Gambar 4. 7 DFD Level 1 proses 3.0	35
Gambar 4. 8 Kerangka Arsitektur Sitem Pakar Kerusakan Kendaraan Roda 4	36
Gambar 4. 9 Diagram Alir Proses Pengambilan Keputusan Pakar dengan Sistem ...	Error!
Bookmark not defined.	
Gambar 4. 10 Mesin Inferensi Foward Chaining Dengan Metode Dempster-Shafer	40
Gambar 4. 11 Flowchart inferensi Dempster-shafer	40

Gambar 4. 12 Flowchart inferensi Dempster-shafer.....	41
Gambar 4. 13 Struktur Menu	45
Gambar 4. 14 Struktur Menu Pakar	45
Gambar 4. 15 Antarmuka Halaman Login	46
Gambar 4. 16 Antarmuka Halaman Register.....	46
Gambar 4. 17 Antarmuka Halaman Diagnosa	47
Gambar 4. 18 Antarmuka Riwayat Diagnosa	47
Gambar 4. 19 Antarmuka Gejala	48
Gambar 4. 20 Antarmuka data kerusakan	48
Gambar 4. 21 Antarmuka Basis Pengetahuan.....	49
Gambar 4. 22 Antarmuka Halaman Diagnosa Admin	49
Gambar 5. 1 Diagram Implementasi.....	50
Gambar 5. 2 Antarmuka Halaman Login Admin.....	62
Gambar 5. 3 Implementasi halaman sign up.....	62
Gambar 5. 4 Antarmuka Halaman Data User	63
Gambar 5. 5 Antarmuka Halaman Data Gejala Kerusakan	63
Gambar 5. 6 Antarmuka Data Kerusakan	64
Gambar 5. 7 Antarmuka Halaman Basis Pengetahuan	64
Gambar 5. 8 Antarmuka Halaman Riwayat Sistem.....	65
Gambar 5. 9 Antarmuka Halaman Diagnosa	65
Gambar 5. 10 Antar muka halaman untuk user	66
Gambar 6. 1 Pohon Pengujian dan Analisis.....	67

BAB I PENDAHULUAN

1.1 Latar Belakang

Saat ini alat transportasi merupakan kebutuhan utama masyarakat untuk melakukan aktivitas sehari-hari terutama mobil. Tidak sedikit masyarakat yang kini mempunyai mobil. Semakin banyaknya pengguna mobil tentunya mempengaruhi industri mesin untuk berlomba-lomba membuat mobil dengan harga, variasi dan keunggulan yang menarik. Namun semua itu tak luput dari adanya kendala atau kerusakan yang terjadi pada mobil. Namun dengan tingginya ketergantungan kegiatan sehari-hari dengan mobil tidak diimbangi dengan kemampuan masyarakat untuk menganalisa sejak dini permasalahan yang ada pada mobilnya, karena apabila kerusakannya semakin parah maka permasalahan itu akan membuat aktivitas para pengguna mobil akan terganggu sehingga akan membuat pengeluaran akan lebih banyak. Keadaan tersebut mengakibatkan pengguna mobil memiliki ketergantungan yang tinggi terhadap pakar (montir) mobil. Padahal tidak semua kerusakan harus dibawa ke pakarnya ataupun bengkel, sehingga membuat biaya dan waktu yang harus dikorbankan untuk melakukan perbaikan akan memakan waktu yang cukup lama. Dari kondisi-kondisi tersebut maka masyarakat umum mengalami kesulitan dalam mendeteksi kerusakan mobilnya.

Penerapan teknologi berbasis komputer pada kegiatan manusia diantaranya adalah dengan penerapan aplikasi Sistem Pakar (*Expert System*) adalah program berbasis pengetahuan yang menggunakan pengetahuan manusia, dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer, dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia. Implementasi sistem pakar saat ini banyak digunakan dalam kegiatan manusia karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar pada bidang tertentu dalam program komputer sehingga keputusan dapat diberikan dalam melakukan penalaran secara cerdas.

Pada penelitian yang dilakukan oleh Satwika yang meneliti tentang sistem pakar kerusakan mobil toyota menggunakan metode *forward chaining* menyimpulkan bahwa sistem pakar menggunakan metode runut maju ini dapat membantu pengguna dalam mengetahui kerusakan yang terdapat pada mobilnya dari gejala-gejala atau keluhan yang sering dialami oleh pengguna dengan mobilnya. Penggunaan bahasa pemrograman WEB yaitu PHP dapat memudahkan pengguna dalam mengakses sistem ini tanpa melakukan penginstalan terlebih dahulu. Namun, penelitian tersebut belum menerapkan metode *dempster shafer* untuk mendapatkan nilai presentase yang berguna untuk melihat seberapa besar tingkat kerusakan.

Dewasa ini *Dempster-shafer* telah berhasil diaplikasikan dalam permasalahan dunia nyata dan memberikan solusi yang lebih baik untuk kasus tertentu dibandingkan metode lain misalnya *Naïve Bayes* (NB) dan *Certainty*

Factor (CF). Penelitian Iswari Nur Hidayati (2010) memperlihatkan keunggulan *Dempster Shafer* dibanding metode *Naïve Bayes*, dimana *Dempster Shafer* dapat diaplikasikan untuk data-data multisensor dan atau multisumber termasuk data-data dari penginderaan jauh [HID-10]. Dilihat dari tingkat akurasi, *Dempster-shafer* memberikan akurasi yang baik. Pada penelitian mengenai diagnosa jenis penyakit *Diabetes Mellitus*, tingkat akurasi mencapai 96,67% [KUR-14].

Pada penelitian kali ini, penulis mencoba menggunakan Metode penalaran yaitu menggunakan teknik inferensi runut maju (*forward chaining*). Dimana pada *forward chaining* ini dimulai dengan informasi awal (gejala awal) dan bergerak maju untuk mencocokkan informasi selanjutnya sampai menemukan informasi yang sesuai dengan kaidah, lalu akan menyimpulkan berupa keterangan jenis kerusakan dan solusi. Metode *forward chaining* memiliki kelemahan karena tidak terdapat persentase hasil perhitungan atas kerusakan tersebut. Untuk itu digunakan metode lain yaitu metode *Dempster- Shafer*, Metode ini dapat digunakan untuk mencari persentase kemungkinan kerusakan mesin yang dialami mobil dengan mendiagnosa gejala yang dirasakan oleh pemilik kendaraan (*user*). Diharapkan dengan penggunaan metode ini dapat meminimalisir ketidaktepatan sehingga dapat menghasilkan diagnosa yang valid. Aplikasi yang dibuat nantinya menyediakan informasi penunjang serta membantu mengidentifikasi kerusakan pada mesin mobil yang mampu memberikan solusi mengenai cara mencegah dan cara mengatasinya. Di harapkan Sistem ini dapat membantu para pemilik kendaraan atau yang baru akan memiliki mobil untuk bisa mengetahui sejak dini kerusakan yang terjadi.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang diatas, maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana merancang pemodelan sistem pakar kerusakan mesin kendaraan roda 4 dengan metode *forward chaining – Dempster shafer*.
2. Bagaimana menguji sistem pakar untuk mendiagnosa kerusakan mesin kendaraan roda 4 dengan metode *forward chaining – Dempster shafer*.

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Memodelkan sistem pakar kerusakan mesin mobil dengan metode *forward chaining – Dempster shafer*.
2. Menguji sistem pakar kerusakan mesin mobil dengan metode *forward chaining – Dempster shafer*

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. Data penelitian ini didapat dari buku dan bengkel resmi Suzuki di UMC Ahmad Yani Surabaya.
2. Data yang digunakan pada penelitian ini berupa 6 jenis Kerusakan dan 16 gejala kerusakan mesin yang sering terjadi yang didapatkan dari pakar.

3. Gejala kerusakan mesin jenis mobil Suzuki dengan jenis mesin injeksi tahun pembuatan dari 2005 hingga 2010.
4. Aplikasi dibuat berbasis web dengan bahasa pemrograman PHP dan *database MySQL*.
5. Pengujian sistem dilakukan dengan dua tahap, yaitu pengujian validasi dan pengujian akurasi.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini diharapkan bisa membantu para pemilik kendaraan yang masih awam untuk mengetahui kerusakan mesin pada mobil yang dimiliki dan para mekanik untuk dengan segera bisa mengetahui kerusakan sehingga dapat segera memperbaiki atau mengganti part yang rusak.

1.6 Sistematika Penulisan

Agar dalam penulisan skripsi ini dapat lebih terarah, maka penulis berusaha menyusun secara sistematis sehingga diharapkan tahap-tahap pembahasan akan tampak jelas kaitannya antara bab yang satu dengan bab yang lainnya. Adapun isi dari masing-masing bab tersebut adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi uraian tentang latar belakang permasalahan, merumuskan inti permasalahan yang dihadapi, pembatasan masalah, menentukan tujuan, manfaat dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Bab ini membahas berbagai konsep dasar dan teori-teori penunjang yang berkaitan dengan topik penelitian yang dilakukan. Pada bab ini membahas pengertian jenis-jenis penyimpangan tumbuh kembang anak dan metode *Dempster-shafer* yang menjadi bahasan utama dalam penelitian ini.

BAB 3 METODOLOGI PENELITIAN

Menguraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literatur, perancangan sistem perangkat lunak, implementasi algoritma, pengujian dan analisa serta penulisan laporan.

BAB 4 PERANCANGAN

Bab ini membahas analisis kebutuhan dan perancangan aplikasi pemodelan sistem pakar untuk mendiagnosa kerusakan mesin kendaraan roda 4

BAB 5 IMPLEMENTASI SISTEM

Bab ini menjelaskan tentang implementasi metode *Forward chaining* – *Dempster shafer* untuk mendiagnosa p mendiagnosa kerusakan mesin kendaraan roda 4 sesuai dengan perancangan sistem yang telah dibuat.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini berisi penjelasan tentang strategi pengujian (unit, integrasi dan validasi) dan teknik pengujian (*black box*) yang dilakukan. Dijelaskan juga seluruh kasus uji berupa data masukan, prosedur uji dan analisis hasil pengujiannya. Pada bagian akhir dijelaskan pula analisis hasil pengujian keseluruhan.

BAB 7 PENUTUP

Bab ini memuat kesimpulan yang diperoleh dari hasil penelitian dan saran-saran yang perlu diperhatikan untuk pengembangan penelitian lebih lanjut.



BAB 2 TINJAUAN PUSTAKA

Bab ini berisi kajian pustaka dan pembahasan tentang dasar teori yang diperlukan untuk penelitian. Kajian pustaka membahas penelitian yang telah ada sebelumnya. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan. Adapun landasan teori tersebut adalah hasil penelitian terkait, pemodelan, sistem pakar, *forward chaining*, *dempster shafer*, dan komponen mobil.

2.1 Kajian Pustaka

Kajian pustaka dilakukan dengan cara melakukan analisa perbandingan beberapa penelitian terkait sistem pakar Kerusakan mesin kendaraan dengan menggunakan metode lain.

Penelitian pertama tentang "*sistem pakar kerusakan mesin mobil toyota menggunakan metode forward chaining*" menyimpulkan bahwa sistem pakar menggunakan metode runut maju ini dapat membantu pengguna dalam mengetahui kerusakan yang terdapat pada mobilnya dari gejala-gejala atau keluhan yang sering dialami oleh pengguna dengan mobilnya. Penggunaan bahasa pemrograman WEB yaitu PHP dapat memudahkan pengguna dalam mengakses sistem ini tanpa melakukan penginstalan terlebih dahulu. Namun, penelitian tersebut belum menerapkan metode *dempster shafer*.

Penelitian kedua adalah melakukan penelitian mengenai "*sistem pakar diagnosa penyakit ginjal dengan menggunakan metode Dempster shafer.*" Aplikasi sistem pakar ini menghasilkan keluaran berupa kemungkinan penyakit ginjal yang diderita berdasarkan gejala yang dirasakan oleh user. Sistem ini juga menampilkan besarnya kepercayaan gejala tersebut terhadap kemungkinan penyakit ginjal yang diderita oleh user. Besarnya nilai kepercayaan tersebut merupakan hasil perhitungan dengan menggunakan metode *dempster shafer* [SUL-08].

2.2 Pemodelan

Model adalah adalah rencana, representasi, atau deskripsi yang menjelaskan suatu objek, sistem, atau konsep, yang seringkali berupa penyederhanaan atau idealisasi. Bentuknya dapat berupa model fisik (maket, bentuk prototipe), model citra (gambar, komputerisasi, grafis), atau rumusan matematis. Sedangkan sistem adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi.

Pemodelan Sistem adalah suatu bentuk penyederhanaan dari sebuah elemen dan komponen yang sangat kompleks untuk memudahkan pemahaman dari informasi yang dibutuhkan [HEN-10].

Karakteristik dari Pemodelan Sistem adalah sebagai berikut [HEN-10]:

1. Dibuat dalam bentuk grafis dan tambahan keterangan secara tekstual.
2. Dapat diamati dengan pola *top-down* dan *partitioned*.
3. Memenuhi persyaratan minimal *redundancy*.
4. Dapat mempresentasikan tingkah laku sistem dengan cara yang transparan.

Pemodelan sistem dibuat dalam bentuk grafis atau bergambar sehingga dapat memudahkan *customer* dan dilengkapi juga dengan keterangan dari gambar atau grafis tersebut. Alur dari proses model tersebut dapat dilihat dan diamati dan dapat mempresentasikan proses dari pada sistem yang dibuat dan dapat dipahami oleh *customer*.

2.2.1 Prinsip Pemodelan Sistem

Prinsip dari Pemodelan sistem menurut Grady Booch, James Rumbaugh dan Ivar Jacobson adalah:

1. Memilih model apa yang digunakan, bagaimana masalahnya dan bagaimana juga dengan solusinya.
2. Setiap model dapat dinyatakan dalam tingkatan yang berbeda
3. Model yang terbaik adalah yang berhubungan dengan realitas.
4. Tidak pernah ada model tunggal yang cukup baik, setiap sistem yang baik memiliki serangkaian model kecil yang independen.

Prinsip pemodelan sistem tidak terlalu menitik beratkan pada bentuk model apa untuk merancang sebuah sistem. Bentuk model ini bebas, sesuai dengan keinginan. Contohnya berupa narasi, *prototype*, maupun gambar. Yang terpenting adalah harus mampu merepresentasikan visualisasi bentuk sistem yang diinginkan oleh *user*, karena sistem akhir yang dibuat bagi *user* akan diturunkan dari hasil model tersebut [FAH-14].

2.2.2 Perangkat Pemodelan Sistem

Dalam perancangan sebuah sistem langkah awal yang harus dilakukan adalah memodelkan sistem. Hal ini dilakukan untuk memfokuskan perhatian pada hal-hal penting dalam sistem tanpa terlibat terlalu jauh.

Ada beberapa macam perangkat pemodelan sistem yang dapat digunakan untuk memodelkan sistem, akan tetapi tidak mutlak untuk menggunakan semua perangkat pemodelan yang ada, artinya boleh menggunakan sebagian perangkat pemodelan dari beberapa perangkat yang ada. Perangkat yang digunakan untuk memodelkan sistem, diantaranya adalah [HEN-10]:

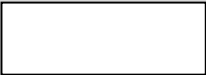
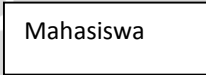
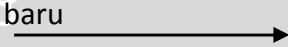
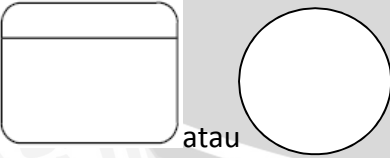
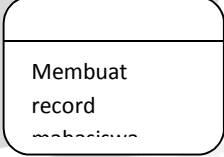
2.2.2.1 Context Diagram (Konteks Diagram)

Diagram konteks merupakan kejadian tersendiri dari suatu diagram alir data. Dimana satu lingkaran merepresentasikan seluruh sistem. Diagram konteks ini harus berupa suatu pandangan, yang mencakup masukan-masukan dasar, sistem-sistem dan keluaran.

Diagram konteks dimulai dengan penggambaran terminator, aliran data, aliran kontrol penyimpanan, dan proses tunggal yang menunjukkan keseluruhan sistem. Bagian termudah adalah menetapkan proses (yang hanya terdiri dari satu lingkaran) dan diberi nama yang mewakili sistem.

Nama dalam hal ini dapat menjelaskan proses atau pekerjaan berupa nama perusahaan yang dalam hal ini mewakili proses yang dilakukan keseluruhan organisasi. Terminator ditunjukkan dalam bentuk persegi panjang dan berkomunikasi langsung dengan sistem melalui aliran data atau penyimpanan eksternal. Antar terminator tidak diperbolehkan komunikasi langsung. Pada kenyataannya hubungan antar terminator dilakukan, tetapi secara definitif karena terminator adalah bagian dari lingkungan, maka tidak relevan jika dibahas dalam diagram konteks [HEN-10]. Simbol-simbol konteks diagram dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Simbol-simbol Konteks Diagram 1

Simbol	Arti	Contoh
	Terminator	
	Aliran Data/ <i>Data Flow</i>	
 atau	Proses	

Sumber: [HEN-10]

2.2.2.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data,

baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi.

DFD adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

Adapun komponen-komponen yang digunakan dalam penggambaran diagram alur data adalah sebagai berikut [HEN-10]:

1. Terminator (Sumber)

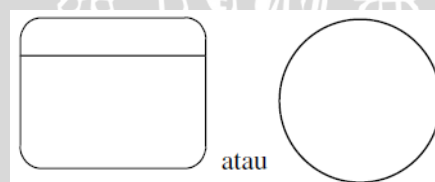
Setiap sistem akan mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. terminator merupakan kesatuan atau (*entity*) sumber atau tujuan di dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan input atau menerima output dari sistem. Simbol *context diagram* terminator dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Terminator 1

2. Proses

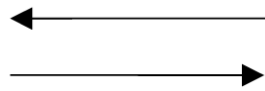
Suatu proses adalah kegiatan yang dilakukan atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses. Simbol proses pada *context diagram* dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Proses 1

3. Arus Data (*Data Flow*)

Suatu proses adalah kegiatan yang dilakukan atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar. Arus data menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. Arus data ini mengalir antara proses, penyimpanan data, dan kesatuan luar dari proses. Gambar 2.3 merupakan simbol arus data untuk *context diagram*.



Gambar 2. 3 Arus Data

4. Simpanan Data (*Data Store*)

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa file, arsip, tabel, dan agenda. *Data store* pada *context diagram* dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Data Store 1

2.2.2.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah sebuah diagram yang menggambarkan hubungan atau relasi antar entitas (*entity*), setiap entitas terdiri atas satu atau lebih atribut yang merepresentasikan seluruh kondisi atau fakta dari dunia nyata yang ditinjau. Dengan ERD dapat mentransformasikan keadaan dari dunia nyata ke dalam bentuk basis data.

Ada tiga macam komponen- komponen ERD yang digunakan yaitu:

1. *Entity*/objek data

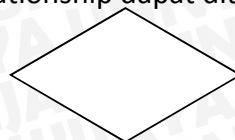
Entity adalah kumpulan objek atau suatu yang dapat dibedakan atau dapat diidentifikasi secara unik, kumpulan entitas yang sejenis disebut entity set. Penggambaran entitas pada ERD menggunakan simbol persegi panjang. Simbol *Entity*/objek data dapat dilihat pada Gambar 2.5.



Gambar 2. 5 Simbol Entity

2. Relationship

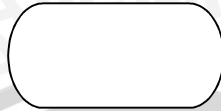
Relationship adalah Hubungan yang terjadi antara satu entitas atau lebih, kumpulan relationship yang sejenis disebut Relationship set. Hubungan digambarkan dengan bentuk belah ketupat, tiap belah ketupat diberi label kata kerja. Simbol ERD Relationship dapat ditunjukkan pada Gambar 2.6.



Gambar 2. 6 Simbol Relationship

3. Atribut

Atribut merupakan sifat atau karakteristik suatu entitas yang menyediakan penjelasan detail tentang entitas tersebut. Simbol Atribut dapat ditunjukkan pada Gambar 2.7.



Gambar 2. 7 Simbol Atribut

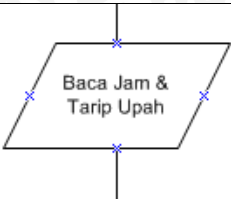
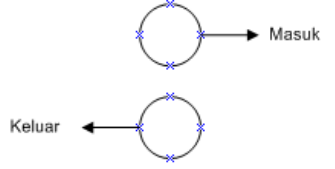
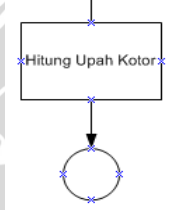
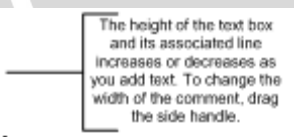
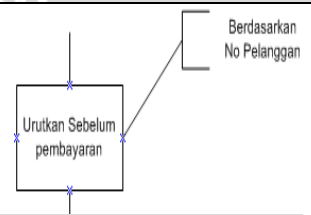
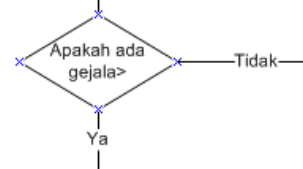
2.2.2.4 Flowchart (Bagan Alir)

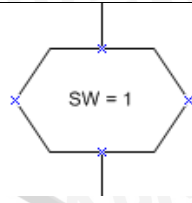
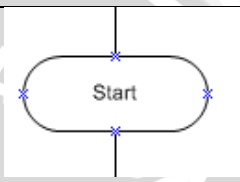
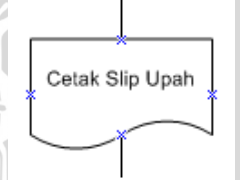
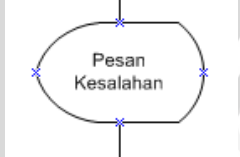
Flowchart adalah bagian (*chart*) yang menunjukkan alir (*flow*) didalam program atau prosedur sistem atau logika. *Flowchart* digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Flowchart* dibagi menjadi lima jenis yaitu [SUL-08]:

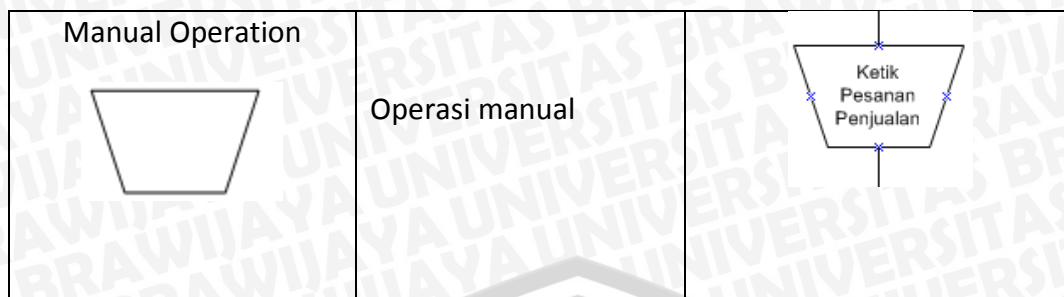
1. Bagan Alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur yang ada di dalam sistem tersebut. Bagan alir sistem mewujudkan apa yang dikerjakan di sistem.
2. Bagan Alir Dokumen (*document flowchart*) merupakan bagan yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.
3. Bagan Alir Skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem. Perbedaanya adalah bagan alir skematik selain menggunakan symbol-simbol bagan alir sistem juga menggunakan gambar-gambar komputer dan peralatan lain yang digunakan. Maksud menggunakan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir.
4. Bagan Alir Program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.
5. Bagan Alir Proses (*process flowchart*) merupakan bagan alir yang banyak digunakan oleh teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur

Pada Tabel 2.2 merupakan simbol-simbol *flowchart* yang biasanya digunakan adalah simbol-simbol standar yang dikeluarkan oleh ANSI dan ISO [SUL-08].

Tabel 2. 2 Simbol-simbol Flowchart

Simbol	Arti	Contoh
<i>Input/ Output</i> 	Merepresentasikan <i>Input</i> data atau <i>Output</i> data yang diproses atau Informasi.	
Proses 	Mempresentasikan operasi	
Penghubung 	Keluar atau masuk dari bagian lain flowchart yang sama	
Anak Panah 	Mempresentasikan alur kerja	
Penjelasan 	Digunakan untuk komentar tambahan	
Keputusan 	Keputusan dalam program	
Predefined Process	Rincian operasi berada di tempat lain	

		
Preparation 	Pemberian harga awal	
Terminal Points 	Awal/ akhr program	
Punched Card 	Input/ output yang menggunakan kartu berlubang	
Dokumen 	I/ O dalam format dicetak	
Display 	Output yang ditampilkan terminal pada	



Sumber: [SUL-08]

2.3 Pemodelan Sistem Pakar

Sistem pakar (*expert system*) adalah suatu sistem komputer berbasis pengetahuan yang menyamai kemampuan pengambilan keputusan dari seorang pakar (mekanik). Sistem pakar telah dibuat untuk memecahkan masalah-masalah dalam berbagai bidang, antara lain matematika, teknik, kedokteran, ilmu komputer, sampai bidang hukum. Walaupun sistem pakar sebagai sistem komputer yang dalam berbagai hal bekerjanya jauh lebih baik dari manusia atau ahli, tetapi kita tidak bisa menghilangkan begitu saja faktor manusia dan digantikan oleh sistem komputer, karena pada banyak situasi keahlian manusia tetap dibutuhkan, sebab kemampuan komputer terbatas (Arhami, 2004).

Pengetahuan dari suatu sistem pakar mungkin dapat direpresentasikan dalam sejumlah cara. Salah satu metode yang paling umum untuk merepresentasikan pengetahuan adalah dalam bentuk tipe aturan (*rule*) IF..THEN (jika..maka). Walaupun cara diatas sangat sederhana, namun banyak hal yang berarti dalam membangun sistem pakar dengan mengekspresikan pengetahuan pakar (mekanik) dalam bentuk aturan diatas. Konsep dasar dari suatu sistem pakar mengandung beberapa unsur atau elemen, yaitu keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan.

Keahlian merupakan suatu penguasaan pengetahuan dibidang tertentu yang didapatkan dari pelatihan, membaca atau pengalaman. Seorang ahli adalah seorang yang mempunyai pengetahuan tertentu dan mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan (*domain*), dan menentukan relevan atau tidaknya keahlian mereka.

Pengalihan keahlian dari para ahli untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, merupakan tujuan utama dari sistem pakar. Proses ini membutuhkan empat aktivitas, yaitu tambahan pengetahuan dari para ahli atau sumber-sumber lainnya, representasi pengetahuan kedalam komputer, inferensi pengetahuan dan pengalihan pengetahuan ke pengguna. Pengetahuan yang disimpan di komputer dinamakan dengan basis pengetahuan (*knowledge base*).

Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar (*reasoning*). Jika keahlian-keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basis data,

maka komputer harus dapat diprogram untuk membuat inferensi. Proses ini dibuat dalam bentuk motor inferensi (*inference engine*).

Sistem pakar menggunakan basis pengetahuan (*knowledge base*) sebagai dasar pemikirannya. *Knowledge base* tersebut terdiri dari heuristik dan sejumlah *rule-rule* dan aturan-aturan yang tersusun secara sistematis dan spesifik, juga relasi antara data dan aturan (*rule*) dalam pengambilan kesimpulan. *Knowledge base* tersebut disimpan dalam sebuah basis data pada *database*. Sedangkan sebagai pusat pemrosesannya adalah *inference engine*, yaitu suatu rancangan aplikasi yang berfungsi untuk memberikan pertanyaan dan menerima input dari user, kemudian melakukan proses logika sesuai dengan *knowledge base* yang tersedia, untuk selanjutnya menghasilkan *output* berupa suatu kesimpulan atau bisa juga berupa keputusan sebagai hasil akhir konsultasi.

Knowledge acquisition source berfungsi sebagai penterjemah dari *knowledge base* menjadi sebuah bahasa yang dapat dimengerti oleh *user*. Bagian ini diperlukan karena *knowledge base* yang disimpan dalam sebuah *database*, disimpan dalam format tertentu, sedemikian rupa sehingga *user* sulit mengartikannya.

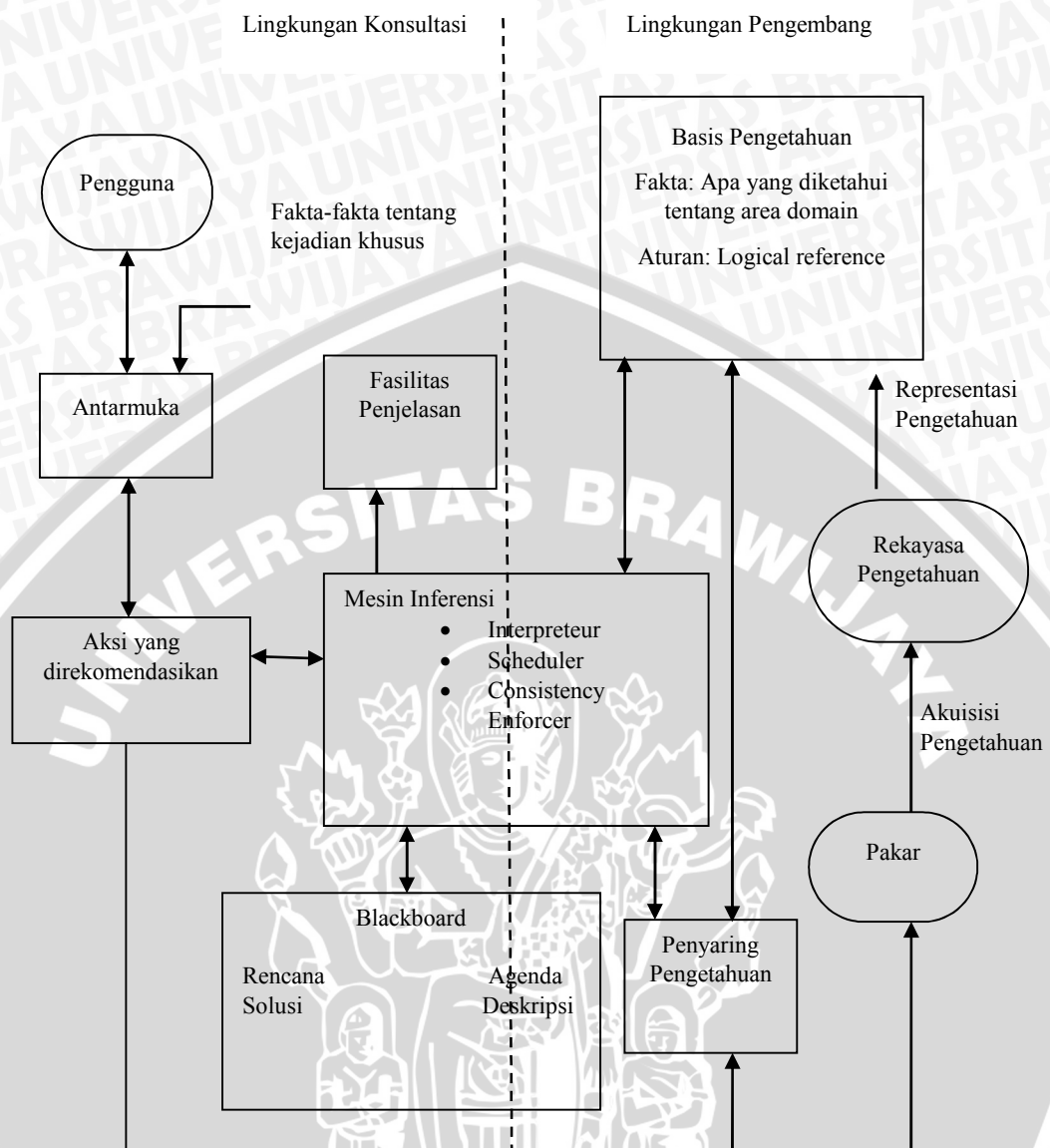
Disk (*working memory*) adalah sejumlah modul memory yang menyimpan informasi sementara dari suatu proses konsultasi. Setiap proses baru dijalankan, memory tersebut akan di set ke kondisi awal. Dalam menjalankan proses, memory tersebut menyimpan informasi dari *rule-rule* yang dipakai dalam *knowledge base*. Oleh karena itu, langkah-langkah perancangan sistem pakar sebaiknya mengikuti urutan sebagai berikut [ARH-04] :

- 1 Menentukan batasan-batasan atau bidang konsentrasi dari sebuah sistem pakar yang akan dirancang.
- 2 Memilih jenis keputusan apa yang diambil.
- 3 Membuat pohon keputusan (*decision tree*).
- 4 Menuliskan *IF-THEN rules*.
- 5 Merancang antarmuka pengguna (*user interface*).

2.3.1 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembang (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar.

Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam gambar 2.8 [ARH-04].



Gambar 2.8 Struktur Sistem Pakar

Sumber : [ARH-04]

1. Pengguna (User)

Pada umumnya pengguna sistem pakar adalah orang awam yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada.

2. Antarmuka Pengguna (User Interface)

User interface merupakan sebuah mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

3. Mesin Inferensi (*Inference Engine*)
Sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan basis pengetahuan yang ada, manipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan hingga dicapai suatu kesimpulan.
4. Daerah Kerja (*Blackboard*)
Blackboard adalah tempat untuk merekam hasil sementara untuk dijadikan keputusan dan untuk menjelaskan masalah yang terjadi. Tiga tipe keputusan yang direkam pada Blackboard meliputi: rencana, agenda, dan solusi.
5. Fasilitas Penjelasan (*Explanation Subsystem*)
Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan penalaran sistem kepada pemakai. Fasilitas penjas dapat menjelaskan perilaku sistem pakar dengan menjawab pertanyaan-pertanyaan.
6. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)
Sistem memiliki kemampuan menganalisa pengetahuan yang diperlukan dari seorang pakar dan juga untuk mengevaluasi diri sehingga mengetahui alasan kesuksesan dan kegagalan dalam mengambil keputusan.

2.3.2 Representasi Pengetahuan

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar [ARH-04]. Metode representasi yang cocok untuk pengetahuan bersifat deklaratif.

1. Logika (*Logic*)
Logika merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah, dan prosedur yang membantu proses penalaran.
2. Jaringan Semantik (*Semantic Nets*)
Jaringan semantik merupakan teknik representasi kecerdasan buatan klasik yang digunakan untuk informasi proposional. Yang dimaksud dengan informasi proposional adalah pernyataan yang mempunyai nilai benar atau salah.
3. 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.
Representasi yang cocok untuk pengetahuan prosedural (ada aksi dan reaksi) adalah kaidah produksi (*Production Rule*). Dimana kaidah produksi adalah kaidah yang menyediakan cara formal untuk mempresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dalam bentuk jika-maka (*if-then*). Kaidah *if-then* menghubungkan antiseden (*antecedent*) dengan konsekuensi yang diakibatkannya.

2.3.3 Ciri-Ciri Sistem Pakar

Sistem pakar memiliki ciri-ciri :

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada *rule* atau kaidah tertentu.
5. Outputnya bersifat nasihat atau anjuran dan tergantung dari dialog *user*.
6. *Knowledge base* dan *inference engine* terpisah.

2.3.4 Orang yang terlibat dalam sistem pakar

Untuk memahami perancangan sistem pakar, perlu dipahami mengenai siapa saja yang berinteraksi dengan sistem. Orang yang terlibat dalam sistem pakar antara lain :

1. Pakar (*domain expert*) adalah seseorang ahli yang dapat menyelesaikan masalah yang sedang diusahakan untuk dipecahkan oleh sistem.
2. Pembangun pengetahuan (*knowledge engineer*) merupakan seseorang yang menerjemahkan pengetahuan seorang pakar (mekanik) dalam bentuk deklaratif sehingga dapat digunakan oleh sistem pakar.
3. Pengguna (*user*) merupakan seseorang yang berkonsultasi dengan sistem untuk mendapatkan saran yang disediakan oleh pakar.
4. Pembangun sistem (*system engineer*) merupakan seseorang yang membuat antarmuka pengguna, merancang bentuk basis pengetahuan secara deklaratif dan mengimplementasikan mesin inferensi.

2.3.5 Kelebihan Sistem Pakar

Seorang pakar (mekanik) dengan sistem pakar mempunyai banyak perbedaan. Perbandingan kemampuan antara seorang pakar (mekanik) dengan sebuah sistem pakar dapat dilihat seperti pada tabel 2.3 .

Tabel 2. 3 Perbandingan kemampuan seorang pakar dengan sistem pakar

Faktor	Human Expert	Expert System
Time availability	Hari Kerja	Setiap Saat
Geografis	Lokal/tertentu	Dimana saja
Kemananan	Tidak tergantung	Dapat diganti
Perishable/dapat habis	Ya	Tidak
Performansi	Variable	Konsisten
Kecepatan	Variable	Konsisten
Biaya	Tinggi	Terjangkau

Dari tabel 2.3 dapat dikembangkan penjelasan lebih lanjut tentang kelebihan sistem pakar.

1. Sistem pakar bisa digunakan setiap hari menyerupai sebuah mesin sedangkan seorang pakar (mekanik) tidak mungkin bekerja terus menerus setiap hari tanpa beristirahat.
2. Sistem pakar merupakan suatu *software* yang dapat diperbanyak dan kemudian dibagikan keberbagai lokasi maupun tempat yang berbeda-beda untuk digunakan, sedangkan seorang pakar(mekanik) hanya berkerja pada satu tempat dan pada saat yang bersamaan.
3. Suatu sistem pakar dapat diberi pengamanan untuk menentukan siapa saja yang mempunyai hak akses untuk menggunakannya dan jawaban yang diberikan oleh sistem terbebas dari proses intimidasi atau ancaman, sedangkan seorang pakar (mekanik) bisa saja mendapat ancaman atau tekanan pada saat menyelesaikan permasalahan.
4. Pengetahuan (*knowledge*) yang disimpan pada sistem pakar tidak akan bisa hilang atau lupa, yang dalam hal ini tentu harus didukung oleh *maintenance* yang baik, sedangkan pengetahuan seorang pakar manusia lambat laun akan hilang karena meninggal, usia yang semakin tua, maupun menderita suatu penyakit.
5. Kemampuan memecahkan masalah pada suatu sistem pakar tidak dipengaruhi oleh faktor dari luar seperti intimidasi, perasaan kejiwaan, faktor ekonomi ataupun perasaan tidak suka. Akan tetapi sebaliknya dengan seorang pakar yang disebutkan dipengaruhi oleh faktor-faktor luar seperti yang disebutkan pada tabel 2.3 ketika sedang menyelesaikan atau memecahkan suatu masalah, sehingga dapat memunculkan jawaban yang berbeda-beda atas pertanyaan yang diajukan walaupun masalahnya sama. Atau dengan kata lain, seorang pakar boleh jadi tidak konsisten.
6. Umumnya kecepatan dalam memecahkan masalah pada suatu sistem pakar relatif lebih cepat dibandingkan oleh seorang pakar manusia. Hal ini sudah dibuktikan pada beberapa sistem pakar yang terkenal di dunia.
7. Biaya menggaji seorang pakar lebih mahal bila dibandingkan dengan penggunaan program sistem pakar (dengan asumsi bahwa program sistem pakar itu sudah ada).

2.4 Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar, proses inferensi dilakukan dalam suatu modul yang disebut *Inference Engine* (Mesin Inferensi).

Ketika representasi pengetahuan pada bagian *knowledge base* telah lengkap, atau paling tidak telah berada pada level yang cukup akurat, maka RP tersebut telah siap digunakan. *Inference engine* merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning*.

2.5 Komponen Mobil

Komponen-komponen mobil merupakan satu kesatuan yang menunjang sistem kerja mesin mobil. Mesin bergerak karena adanya sistem pembakaran dalam (*internal combustion engine*), yang mengubah energi kimia menjadi energi mekanis dengan membakar campuran bensin dan udara, sehingga terjadilah ledakan yang dapat menggerakkan mesin. Secara sederhana, proses kerja masing-masing komponen dalam menunjang kerja secara keseluruhan meliputi sistem pengisian dan pembakaran dalam, sistem transmisi, sistem kemudi, sistem suspensi, sistem rem, serta sistem pelumasan dan pendinginan.

2.6 Kerusakan

Pada umumnya, setiap mobil dilengkapi dengan buku petunjuk (*manual book*). Selain berisi spesifikasi teknis, manual book juga berisi petunjuk perawatan. Ironisnya, banyak pemilik mobil yang tidak mau membaca atau mempelajarinya, padahal itu merupakan bekal paling sederhana untuk merawat mobil. Selain itu, banyak pemilik mobil yang tidak mengetahui kondisi mobilnya, sehingga tidak segera melakukan perbaikan, saat mobilnya menunjukkan tanda-tanda kerusakan.

Biasanya pemilik mobil baru sadar jika sudah terjadi kerusakan yang cukup parah, seperti tidak bisa berjalan sama sekali alias mogok. Agar kondisi mobil selalu prima dan komponen-komponennya tahan lama, pemilik perlu melakukan perawatan yang intensif seperti mengganti kerusakan oli mesin secara berkala, mengisi bahan bakar di SPBU terpercaya, dan mengecek seluruh mobil secara rutin. Penggunaan mobil yang normal dan wajar akan memperpanjang usia pakai komponen-komponennya. Namun pemilik juga harus mengetahui bahwa pabrik mobil sudah mempertimbangkan usia pakai komponen, sehingga suatu saat akan rusak dan harus diperbaiki atau diganti dengan komponen yang baru.

Kerusakan pada mobil bisa dideteksi melalui berbagai tanda, seperti : suara mesin kasar, adanya tetesan oli atau cairan pendingin di tempat parkir, boros bensin dan oli, mesin kurang bertenaga, cairan pendingin berwarna putih karena tercampur dengan oli mesin, mengeluarkan asap putih atau hitam yang mengempul, kap mesin menganga, uap air menyembur keluar ke radiator, dan laju mobil tersendat-sendat.

Penelitian ini membahas gejala-gejala kerusakan pada mesin Injeksi mobil berbahan bakar bensin.

2.7 Mekanisme Mesin

Mekanisme mesin merupakan kumpulan beberapa komponen yang saling berhubungan, sehingga menghasilkan energi gerak. Agar dapat berputar, mesin harus didukung oleh beberapa sistem yang saling berhubungan. Oleh karena itu,

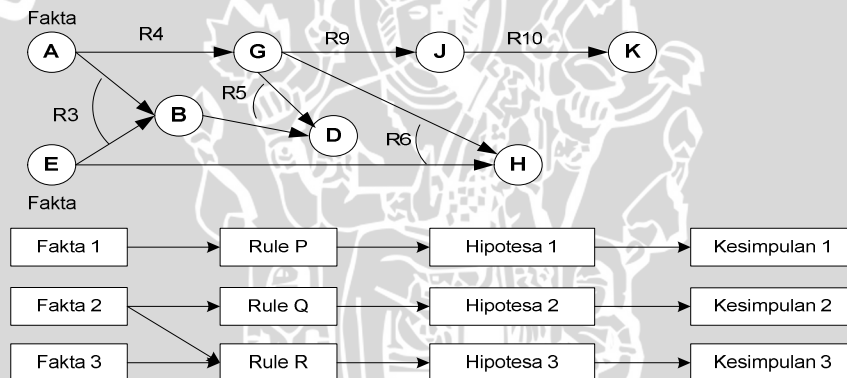
sistem-sistem pendukung mesin harus dapat bekerja dengan baik sehingga mesin dapat berputar secara berkesinambungan.

2.8 Mesin Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Proses inferensi dalam sistem pakar disebut inferensi. Berikut adalah dua jenis metode inferensi :

2.8.1 Foward Chaining

Teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian dicocokkan fakta tersebut dengan bagian *IF* dari aturan *IF-THEN*. Bila ada aturan yang sesuai dengan bagian *IF*, maka aturan tersebut dieksekusi. Bila aturan dieksekusi maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam basis data. Pencocokan dimulai dari aturan teratas dan setiap aturannya hanya boleh dieksekusi sekali. Alur metode inferensi *Forward Chaining* terlihat seperti Gambar 2.9.

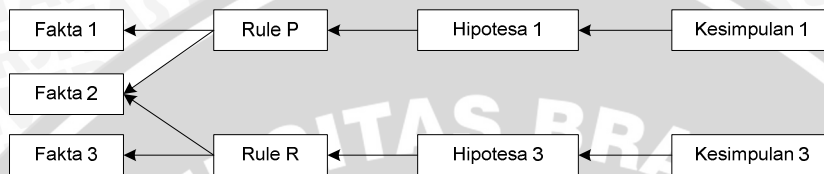
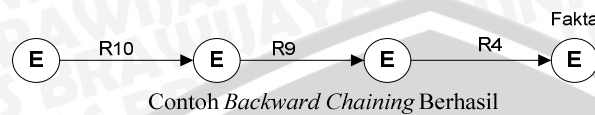
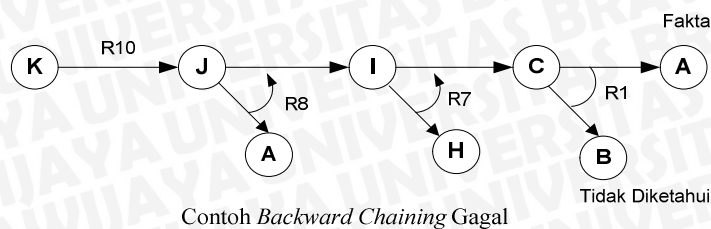


Gambar 2. 9 Alur Metode Foward Chaining

Sumber: [KUS-03]

2.8.2 Backward Chaining

Metode inferensi yang bekerja mundur kearah kondisi awal. Proses diawali dari goal (yang ada pada bagian *THEN* dari aturan *IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis dibagian *IF*. Jika cocok, maka aturan dieksekusi, kemudian hipotesis dibagian *THEN* ditempatkan dibasis data sebagai fakta baru. Jika tidak cocok simpan premis dibagian *IF* kedalam subgoal. Proses berakhir jika goal ditemukan atau tidak ada aturan yang bisa membuktikan kebenaran subgoal atau goal. Alur motede inferensi *Backward Chaining* terlihat seperti Gambar 2.10.



Gambar 2. 10 Alur Metode Backward Chaining

Sumber: [KUS-03]

2.9 Teori Dempster-Shafer

Metode *Dempster-Shafer* pertama kali diperkenalkan oleh Dempster, yang melakukan percobaan model ketidakpastian dengan range probabilitas sebagai probabilitas tunggal. Kemudian pada tahun 1976 Shafer mempublikasikan teori Dempster tersebut pada sebuah buku yang berjudul *Mathematical Theory of Evident* [SUL-08].

Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval:

[Belief, Plausibility]

Belief (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 (nol) maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. Giarratano dan Riley (1994) menjelaskan bahwa fungsi *belief* dapat diformulasikan sebagai persamaan 2.1.

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \quad (2.1)$$

Sedangkan *Plausibility* (Pls) dinotasikan sebagai persamaan 2.2

$$Pls(X) = 1 - Bel(X') = 1 - \sum_{Y \subseteq X'} m(Y) \quad (2.2)$$

dimana:

$Bel(X) = Belief(X)$

$Pls(X) = Plausibility(X)$

$m(X) = mass\ function\ dari\ (X)$

$m(Y) = mass\ function\ dari\ (Y)$

Plausibility juga bernilai 0 sampai 1, jika yakin akan X' maka dapat dikatakan *Belief* (X') = 1 sehingga dari rumus diatas nilai $Pls(X) = 0$. Beberapa kemungkinan range antara *Belief* dan *Plausibility* dapat dilihat pada Tabel 2.4:

Tabel 2. 4 Range Belief dan Plausibility

Kemungkinan	Keterangan
[1,1]	Semua Benar
[0,0]	Semua Salah
[0,1]	Ketidakpastian
[Bel,1] where $0 < Bel < 1$	Cenderung Mendukung
[0,Pls] where $0 < Pls < 1$	Cenderung Menolak
[Bel,Pls] where $0 < Bel \leq Pls < 1$	Cenderung Mendukung dan Menolak

Pada teori *Dempster-Shafer* juga dikenal adanya *frame of discernment* yang dinotasikan dengan Θ . *Frame of discernment* ini merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment* yang ditunjukkan pada Persamaan 2.3.

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_n\} \quad (2.3)$$

dimana:

$\Theta = Frame\ of\ discernment\ atau\ environment$

$\theta_1, \dots, \theta_n = Elemen\ atau\ unsur\ bagian\ dalam\ environment$

Environment mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori *Dempster-Shafer* disebut dengan *power set* dan dinotasikan dengan $P(\Theta)$, setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1.

$$m = P(\Theta) \quad [0,1] \quad \rightarrow$$

sehingga dapat dirumuskan seperti persamaan 2.4.

$$\sum_{X \in P(\Theta)} m(X) = 1 \approx \sum_{X \in P(\Theta)} m(X) = 1 \quad (2.4)$$

dengan :

$P(\theta)$ = power set

$m(X)$ = mass function dari (X)

Dalam teori *Dempster-Shafer*, *disbelief* dalam *environment* biasanya dinotasikan $m(\theta)$. Sedangkan *mass function* (m) dalam teori *Dempster-Shafer* adalah tingkat kepercayaan dari suatu *evidence* (gejala), sering disebut dengan *evidencemeasure* sehingga dinotasikan dengan (m).

Pada aplikasi sistem terdapat sejumlah *evidence* yang akan digunakan pada faktor ketidakpastian dalam pengambilan keputusan untuk diagnosa suatu permasalahan. Untuk mengatasi sejumlah *evidence* tersebut pada teori *Dempster-Shafer* menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination*. Dapat dilihat pada persamaan 2.5.

$$m1 \oplus m2(Z) = \sum_{X \cap Y = Z} m1(X)m2(Y) \quad (2.5)$$

dimana:

$m1 \quad m2(Z)$ = mass function dari evidence (Z)

$m1(X)$ = mass function dari evidence (X)

$m2(Y)$ = mass function dari evidence (Y)

= operator direct sum

secara umum formulasi untuk *Dempster's Rule of Combination* adalah seperti yang terdapat pada persamaan 2.6 (Sentz, 2002).

$$m1 \quad m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1-k} \quad (2.6)$$

dimana:

k = Jumlah *evidential conflict*.

Besarnya jumlah *evidential conflict* (k) dirumuskan dengan persamaan 2.7 [SEN-02].

$$k = \sum_{X \cap Y = \emptyset} m1(X)m2(Y) \quad (2.7)$$

Sehingga bila Persamaan (2.7) disubstitusikan ke Persamaan (2.6) akan menjadi persamaan 2.8.

$$m1 \quad m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1 - \sum_{X \cap Y = \emptyset} m1(X)m2(Y)} \quad (2.8)$$

dimana:

$m1 \quad m2(Z)$ = mass function dari evidence (Z)

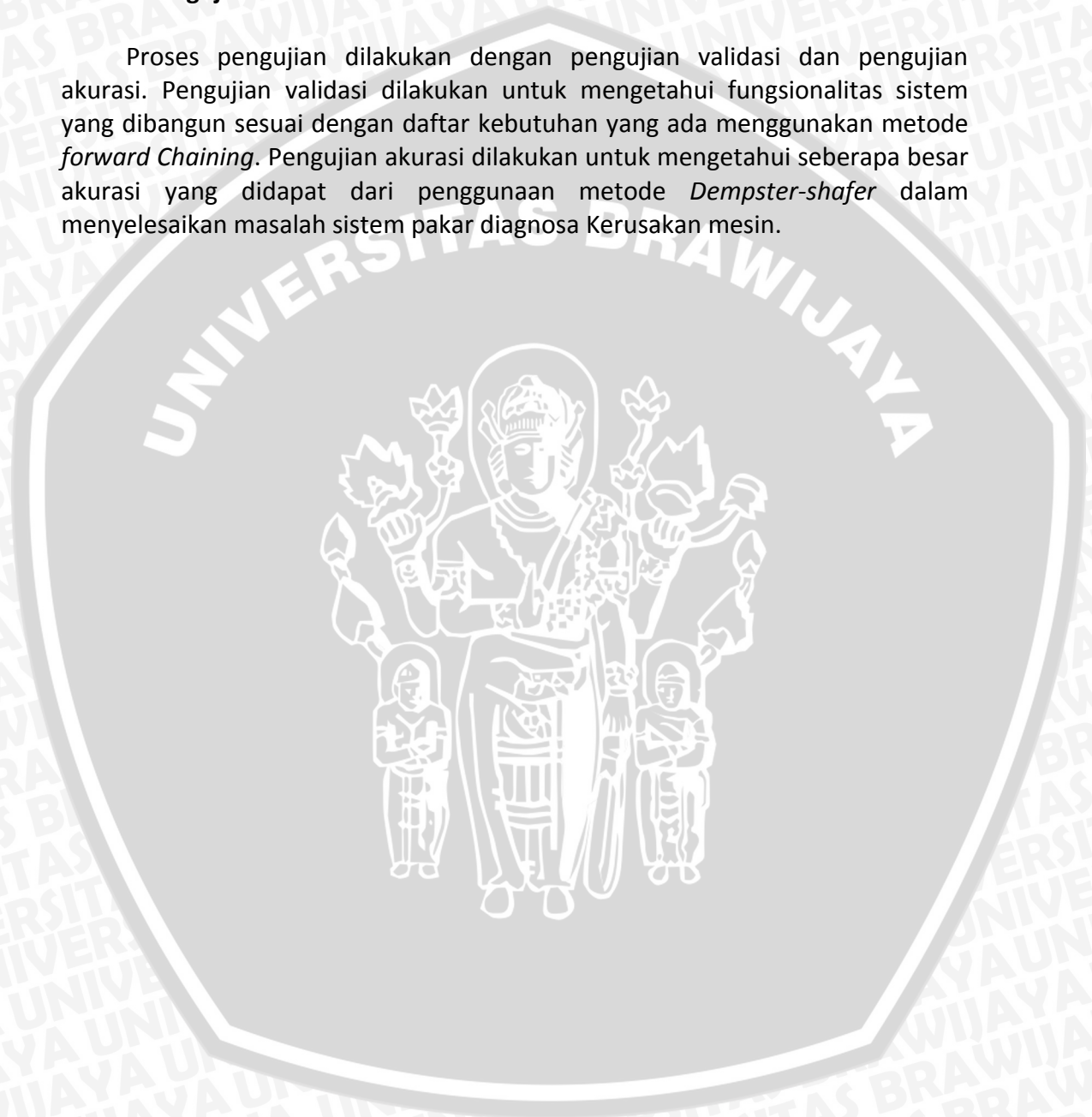
$m1(X)$ = mass function dari evidence (X)

$m2(Y)$ = mass function dari evidence (Y)

k= jumlah evidential conflict

2.10 Pengujian Sistem

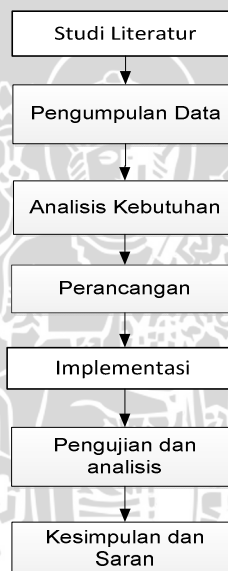
Proses pengujian dilakukan dengan pengujian validasi dan pengujian akurasi. Pengujian validasi dilakukan untuk mengetahui fungsionalitas sistem yang dibangun sesuai dengan daftar kebutuhan yang ada menggunakan metode *forward Chaining*. Pengujian akurasi dilakukan untuk mengetahui seberapa besar akurasi yang didapat dari penggunaan metode *Dempster-shafer* dalam menyelesaikan masalah sistem pakar diagnosa Kerusakan mesin.



BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan dibahas metode-metode yang akan digunakan dalam pembuatan sistem pakar untuk menganalisis kerusakan pada kendaraan roda 4 berdasarkan faktor resiko menggunakan metode *Forward chaining-dempster shafer*

Metodologi penelitian yang dilakukan dalam penelitian ini meliputi beberapa tahapan yaitu studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem dan pengambilan keputusan. Tahapan dalam penelitian tersebut dapat diilustrasikan dengan menggunakan blok diagram metodologi penelitian yang disajikan pada gambar 3.1.



Gambar 3. 1 Blok Diagram Metodologi Penelitian

3.1 Studi Literatur

Studi literatur merupakan tahap awal metode pengerjaan penelitian tugas akhir. Metode ini dilakukan untuk mendapatkan data literatur tambahan mengenai masalah yang diangkat dalam penelitian ini, yaitu definisi sistem pakar, Kerusakan mesin, penggunaan forward chaining dan metode *Dempster-shafer* yang bersumber dari buku, jurnal, karya ilmiah, dan situs-situs penunjang yang dapat membantu dalam penyelesaian penelitian tugas akhir.

3.2 Pengumpulan Data

Lokasi penelitian skripsi ini adalah di bengkel resmi Suzuki di UMC A. Yani, Surabaya. Variabel pada penelitian pada skripsi ini adalah kerusakan yang terjadi pada mobil dan bagaimana memberikan solusi berdasarkan nilai densitas gejala setiap kerusakan menggunakan metode *Dempster-Shafer*. Hipotesis dari penelitian ini adalah membuat aplikasi sistem untuk menentukan kerusakan apa yang terjadi pada mobil dan bagaimana solusi penanganannya.

Berdasarkan cara pengumpulan data untuk kegiatan penelitian terdapat dua jenis data yaitu data sekunder dan data primer. Data sekunder adalah data yang telah dikumpulkan oleh orang lain dan tidak dipersiapkan untuk kegiatan penelitian tetapi dapat digunakan untuk tujuan penelitian seperti buku literature. Sedangkan data primer adalah data yang didapatkan langsung dari responden penelitian. Metode pengumpulan data primer yang bersifat kuantitatif dapat menggunakan instrument kuisioner dan wawancara. Pada Tabel 3.1 dapat dilihat kebutuhan data pada penelitian ini.

Tabel 3. 1 Kebutuhan Data Penelitian

No	Kebutuhan Data	Sumber Data	Metode	Kegunaan Data
1	Data kerusakan mobil	Teknisi mobil	Wawancara pakar	Menentukan kerusakan mobil
2	Nilai densitas tiap gejala kerusakan mobil	Teknisi mobil	Wawancara pakar	Menentukan nilai densitas gejala tiap kerusakan mobil
3	Data kasus mobil yang terjadi kerusakan	Teknisi Mobil	Observasi	Data yang didapat akan digunakan sebagai contoh perhitungan dengan metode <i>Dempster-Shafer</i>

3.2.1 Analisa Kebutuhan Masukan

Pada tahap ini pakar memberikan masukan antara lain berupa:

- Data gejala kerusakan kendaraan roda 4 yang baru yang belum terdapat di dalam sistem. Data gejala meliputi id gejala dan nama gejala.
- Data pengguna berisi id pengguna.
- Data aturan ditambahkan sesuai dengan gejala dan jenis kerusakan yang ditimbulkan.

Berdasarkan masukan pakar, dapat digunakan sebagai basis pengetahuan dari sistem dalam mendiagnosa kerusakan kendaraan roda 4. Selain masukan dari pakar, juga terdapat daftar kebutuhan. Daftar kebutuhan ini meliputi sebuah kolom yang menguraikan kebutuhan sistem maupun *interface* yang harus

disediakan oleh masing-masing sistem. daftar kebutuhan fungsional keseluruhan sistem dapat dilihat pada tabel 3.2.

Tabel 3. 2 Daftar Kebutuhan Fungsional

ID	Requirement	Entitas	Keterangan
FO_01	Sistem mampu melakukan registrasi pengguna baru	UM	Registrasi
FO_02	Sistem mampu menerima inputan <i>login</i> (<i>user</i> dan admin)	UA, UM	<i>Login user</i> dan admin
FO_03	Sistem mampu mengelola data pakar	UA	Data pakar
FO_04	Sistem mampu menerima perubahan data kerusakan kendaraan roda 4	UA	CRUD data kerusakan kendaraan roda 4
FO_05	Sistem mampu menerima perubahan data gejala kerusakan	UA	CRUD data gejala
FO_06	Sistem mampu mengolah bobot gejala kerusakan dan relasi data	UA	Relasi dan bobot gejala kerusakan
FO_07	Sistem mampu mengelola data <i>user</i>	UM,UA	Data pengguna
FO_08	Sistem mampu menerima data gejala kerusakan yang diinputkan pengguna untuk diproses	UM,UA	Proses diagnosa
FO_09	Sistem mampu menampilkan hasil diagnosa berdasarkan gejala kerusakan yang dimasukkan oleh pengguna	UM,UA	Hasil Diagnosa
FO_10	Sistem mampu menyimpan data hasil diagnosa pengguna	UM,UA	Simpan Hasil
FO_11	Sistem mampu <i>logout</i>	UA, UM	<i>Logout</i>

3.2.2 Analisa Kebutuhan Proses

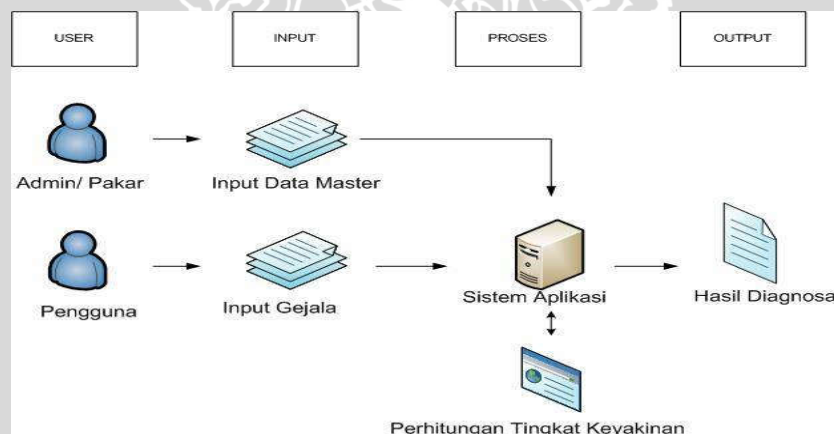
Inti proses dari sistem ini adalah analisis dengan penggabungan dua metode yaitu *forward chaining - dempster shafer*. Sistem akan melakukan perhitungan untuk menentukan jenis kerusakan kendaraan roda 4 berdasarkan gejala kerusakan yang dimasukkan oleh pengguna. pada sistem telah disediakan aturan basis pengetahuan untuk penelusuran jenis kerusakan yang ada pada kendaraan roda 4.

3.2.3 Analisa Kebutuhan Keluaran

Data keluaran dari sistem ini adalah hasil proses diagnosa menggunakan model perhitungan dari metode *forward chaining - dempster shafer*. Hasil diagnosa tersebut berdasarkan fakta gejala kerusakan yang muncul pada kerusakan kendaraan roda 4. Hasil *output* sistem terdiri dari jenis kerusakan, persentase, nama pengguna, asumsi kerusakan, deskripsi.

3.2.4 Perancangan Sistem Pakar

Tahap desain sistem dibutuhkan agar penulis dapat dengan mudah melakukan proses selanjutnya terhadap sistem. Pada Gambar 3.2 dapat dilihat rencana desain sistem yang akan digunakan oleh penulis.

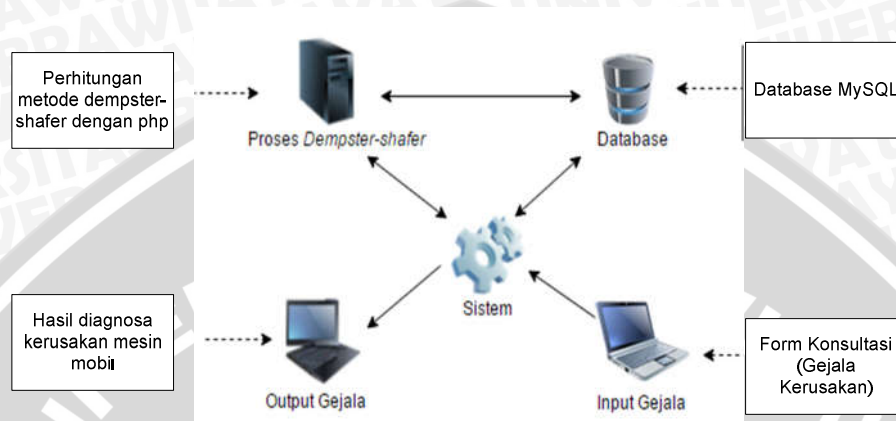


Gambar 3. 2 Desain Umum Perancangan Sistem

Pada Gambar 3.2 dijelaskan bagaimana cara aplikasi ini bekerja. Pertama, admin menginput data kerusakan, data gejala berikut dengan nilai densitasnya. Setelah data dari pakar disimpan ke dalam *database* maka data tersebut akan dijadikan acuan dari perhitungan identifikasi kerusakan menggunakan metode *Dempster-Shafer* pada sistem. Pengguna dapat melakukan identifikasi kerusakan dengan menginput gejala yang terjadi pada mobil ke dalam aplikasi. Sistem kemudian akan memproses nilai densitas berdasarkan gejala-gejala kerusakan yang dimasukkan oleh *user/pengguna* melalui proses perhitungan densitas *Dempster-Shafer* yang sudah ada. Hasil identifikasi kerusakan akan didapat melalui perhitungan nilai densitas tertinggi.

3.3 Implementasi Sistem

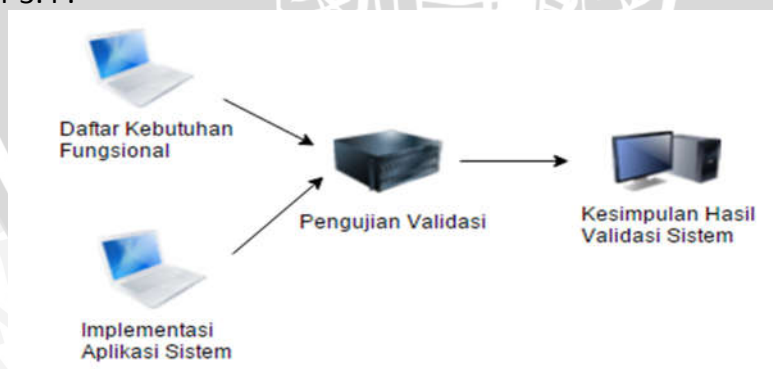
Implementasi perangkat lunak megacu kepada perancangan perangkat lunak. Implementasi diawali dengan penjabaran lingkungan perangkat lunak yang akan digunakan. Aplikasi yang dibangun menggunakan bahasa pemrograman PHP dengan database MySQL dan ditunjukkan pada gambar 3.3 .



Gambar 3. 3 Blok Diagram Implementasi Sistem

3.4 Pengujian dan Analisis

Pengujian dilakukan agar dapat menunjukkan bahwa perangkat lunak mampu bekerja sesuai dengan spesifikasi dan kebutuhan penggunaanya dan mengetahui kinerja dan performa aplikasi. Pengujian yang dilakukan menggunakan teknik *blackbox*. Pengujian *blackbox* digunakan untuk menguji fungsionalitas dari sistem. Langkah selanjutnya setelah melakukan pengujian melakukan analisa terhadap hasil pengujian sehingga didapatkan kesimpulan dari pengembangan perangkat lunak yang telah dilakukan. Ditunjukkan pada gambar 3.4 .



Gambar 3. 4 Blok Diagram Pengujian Validasi Sistem

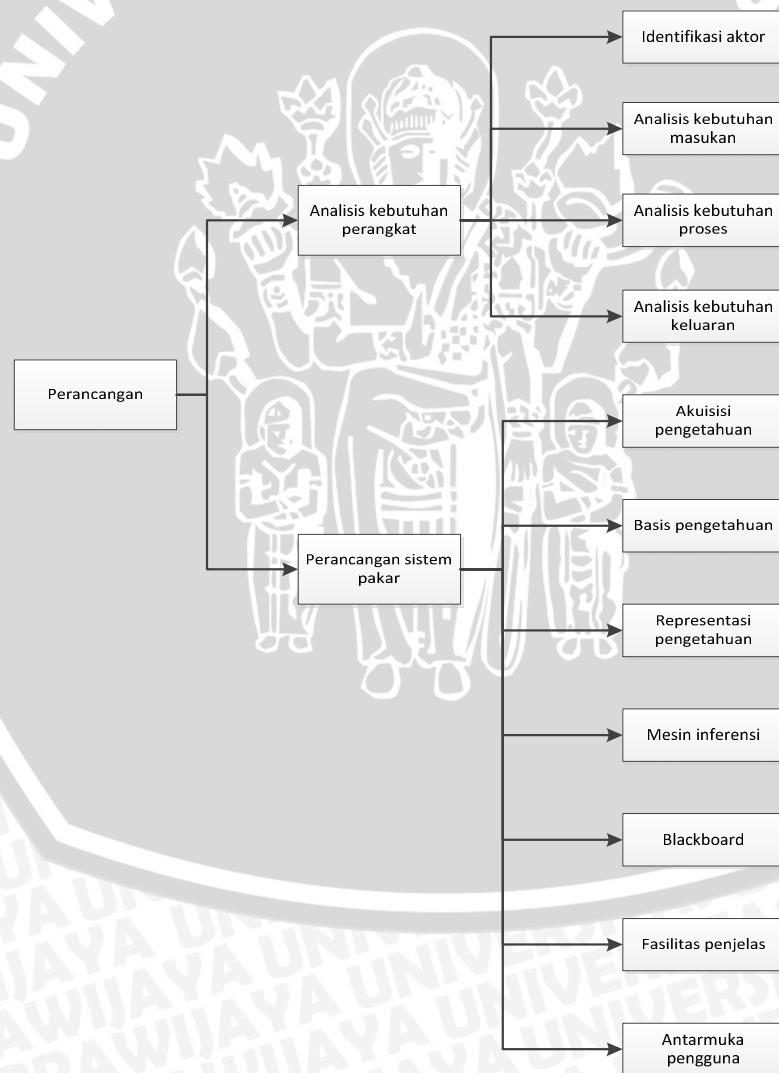
3.5 Penarikan Kesimpulan dan Saran

Tahap kesimpulan dan saran merupakan tahap akhir dari metodologi penelitian. Pada tahap ini, ditarik kesimpulan mengenai apa yang sudah dilakukan dan dicapai dalam pelaksanaan penelitian ini. Kesimpulan haruslah menjawab tujuan penelitian yang telah ditetapkan di Bab Pendahuluan. Tahap ini juga memberikan saran-saran yang mungkin dapat dijadikan sebagai masukan bagi yang ingin menindaklanjuti lebih jauh atau mengembangkan metode yang dipakai.



BAB 4 PERANCANGAN SISTEM

Bab ini membahas tentang perancangan pada aplikasi “Pemodelan Sistem Pakar Diagnosa Kerusakan Mesin Kendaraan Roda 4 Menggunakan Metode *Forward Chaining – Dempster Shafer*”. Pohon perancangan sistem pakar meliputi tiga tahapan yaitu analisa kebutuhan perangkat lunak, perancangan perangkat lunak, dan perancangan sistem pakar. Analisa kebutuhan perangkat lunak terdiri dari identifikasi aktor, analisa kebutuhan masukan, analisa kebutuhan proses, dan analisa kebutuhan keluaran. Perancangan perangkat lunak terdiri dari *entity relationship diagram* dan *data flow diagram*. Perancangan sistem pakar terdiri dari akuisisi pengetahuan, basis pengetahuan, mesin inferensi, *blackboard*, fasilitas penjelas dan antarmuka. Pohon perancangan sistem pakar dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Pohon Perancangan

4.1 Analisis Kebutuhan Perangkat

Analisis kebutuhan perangkat bertujuan untuk menganalisis dan mendapatkan semua kebutuhan yang diperlukan dalam pembuatan sistem pakar untuk menganalisis kerusakan pada kendaraan roda 4 menggunakan *Forward chaining-dempster shafer*. Analisis kebutuhan disesuaikan dengan lokasi dan variabel penelitian, menentukan kebutuhan daya yang akan digunakan, dan mempersiapkan alat dan bahan penelitian. Metode *forward chaining - dempster shafer* digunakan dalam penelitian ini untuk pengimplementasiannya.

Secara keseluruhan, kebutuhan yang digunakan dalam pembuatan sistem pakar :

- a. Kebutuhan *hardware* antara lain:
 - Laptop/PC Core i3 64bit
 - Mouse
- b. Kebutuhan *software* antara lain:
 - Microsoft windows 7 ultimate sebagai sistem operasi.
 - Microsoft office professional 2010 sebagai aplikasi untuk penyusunan laporan.
 - MySQL sebagai server *Database Manajement System*.
 - Bahasa pemrograman PHP.
 - Notepad++ sebagai alternatif pemrograman PHP
- c. Data yang dibutuhkan antara lain:
 - Data gejala kerusakan mobil .
 - Data nilai densitas tiap gejala kerusakan mesin mobil.
 - Deskripsi info dan solusi untuk mengurangi kerusakan yang dialami.

4.1.1 Analisis Kebutuhan Masukan

Pada tahapan ini pakar memberikan masukan antara lain berupa:

1. Data gejala baru yang belum terdapat dalam sistem. Data gejala meliputi kode gejala dan nama gejala.
2. Data Kerusakan berupa kode kerusakan, nama kerusakan, hal yang perlu dilakukan dan pencegahannya yang belum terdapat dalam sistem.
3. Data pengguna yang berisi nama, no. kendaraan, nomer handphone, *username* dan *password*.
4. Data aturan ditambahkan sesuai dengan gejala dan jenis kerusakan yang ditimbulkan. Pakar diminta memberikan nilai densitas dari masing-masing gejala.

Dari masukan pakar, dapat digunakan sebagai basis pengetahuan dari sistem dalam mendiagnosa kerusakan mesin mobil. Selain masukan dari pakar, juga terdapat daftar kebutuhan. Daftar kebutuhan ini meliputi sebuah kolom yang menguraikan kebutuhan sistem maupun *Interface* yang harus disediakan oleh sistem, dan pada kolom lain akan menunjukkan nama proses untuk fungsionalitas masing-masing kebutuhan.

4.1.2 Analisa Kebutuhan Proses

Inti proses dari sistem ini adalah analisis dengan penggabungan dua metode *forward chaining* - *dempster shafer*. Sistem akan melakukan perhitungan untuk menentukan jenis kerusakan kendaraan roda 4 berdasarkan gejala kerusakan yang dimasukkan oleh pengguna. pada sistem telah disediakan aturan basis pengetahuan untuk penelusuran jenis kerusakan yang ada pada mesin kendaraan roda 4.

4.1.3 Analisa Kebutuhan Keluaran

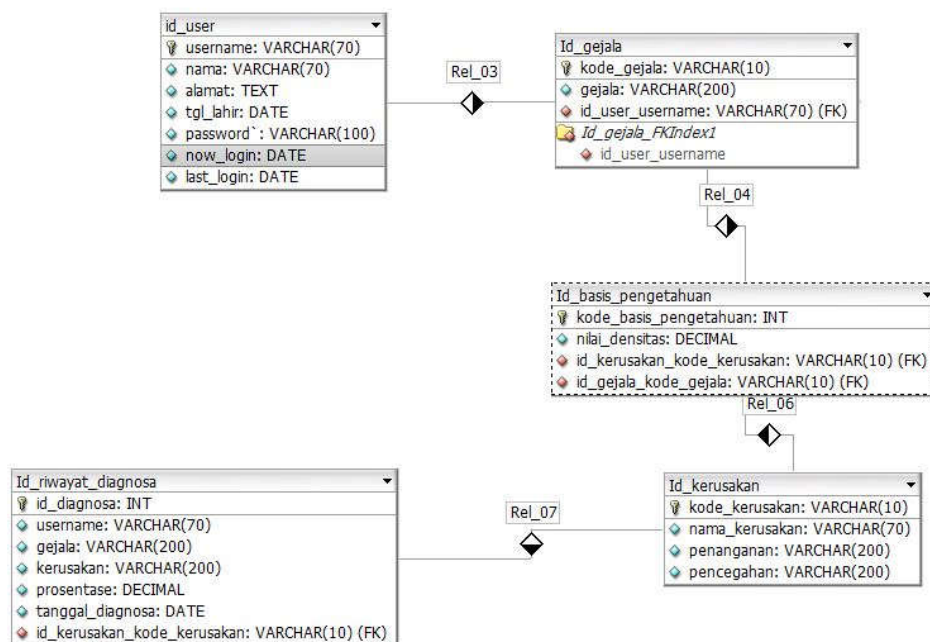
Data keluaran dari sistem ini adalah hasil proses diagnosa menggunakan model perhitungan dari metode *forward chaining* - *dempster shafer*. Hasil diagnosa tersebut berdasarkan fakta gejala kerusakan yang muncul pada kerusakan mesin kendaraan roda 4. Hasil *output* sistem terdiri dari jenis kerusakan, persentase kemungkinan, nama pengguna, no. Kendaraan, asumsi kerusakan, deskripsi dan penanganan.

4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak ini menjelaskan mengenai pola hubungan antar komponen-komponen detail sehingga mampu membentuk sebuah fungsi yang mampu memberikan pelayanan terhadap kebutuhan *user*. Perancangan perangkat lunak menggunakan *Entity Relationship Diagram* (ERD), *Data Flow Diagram* (DFD).

4.2.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang dipakai untuk mendokumentasikan data dengan pendeteksian jenis entitas dan hubungannya. ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang mempresentasikan seluruh fakta yang ditinjau dari keadaan yang nyata. Gambar 4.2 merupakan *Entity Relationship Diagram* Kerusakan Mesin Kendaraan Roda 4.



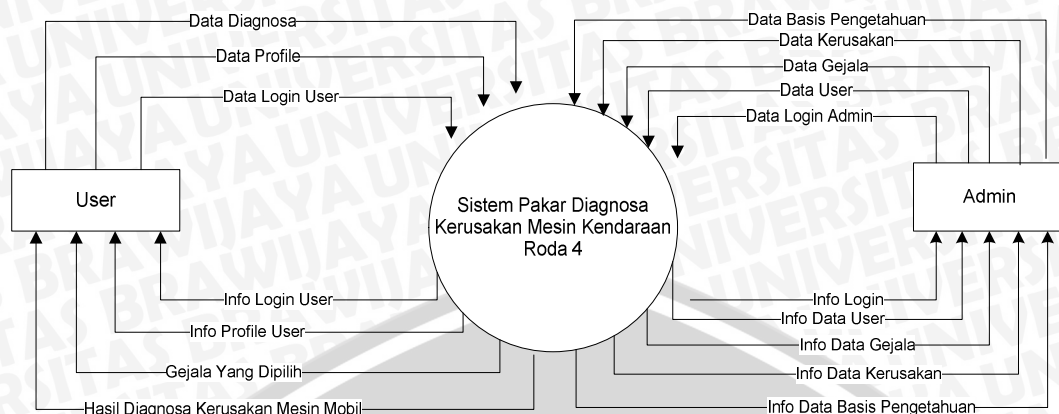
Gambar 4. 2 Entity Relationship Diagram

4.2.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan diagram alir yang dipresentasikan dalam bentuk lambang-lambang tertentu yang menunjukkan aliran data, proses, tempat penyimpanan data, dan entitas eksternal. Proses yang terjadi antara pengguna dengan sistem digambarkan dengan menggunakan diagram-diagram, yang dibagi menjadi beberapa bagian yaitu diagram konteks, DFD level 0 dan DFD level 1.

4.2.2.1 Diagram Konteks

Diagram konteks merupakan gambaran umum aliran data yang terdapat pada sistem pakar diagnosa kerusakan pada mesin mobil menggunakan metode *Dempster-shafer*. Proses aliran data berasal dari *user* dan *admin* seperti pada Gambar 4.3.



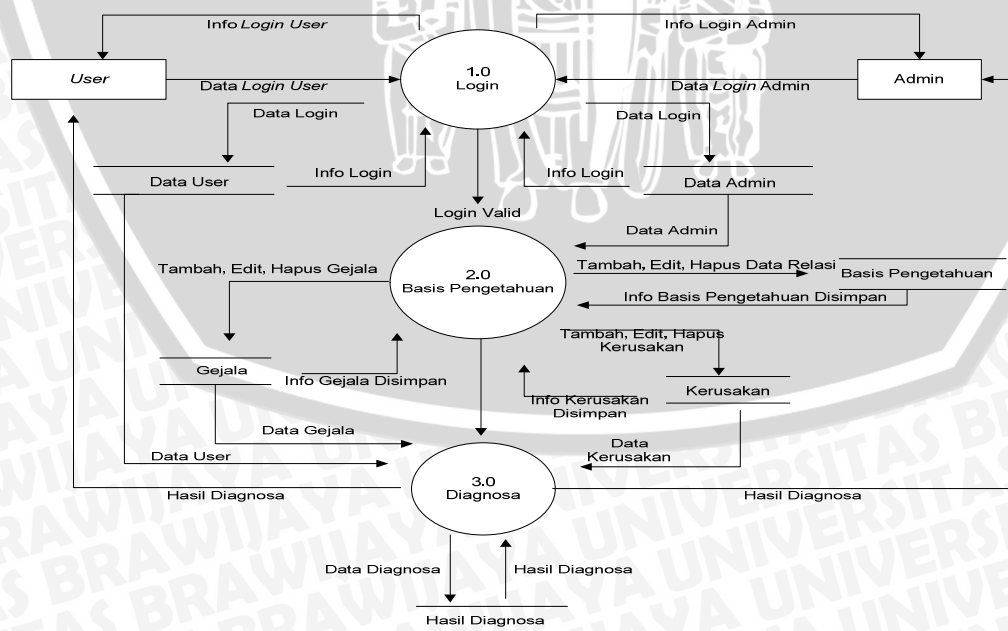
Gambar 4. 3 Context Diagram

Terdapat beberapa penggunaan paket data pada gambar 4.3 diagram konteks, diantaranya:

- Data login yaitu data yang digunakan pada proses *Login*, berisi data *username* dan data *password*.
- Data Kerusakan yaitu data yang digunakan untuk menyimpan data nama Kerusakan.
- Data gejala yaitu data yang digunakan untuk menyimpan data gejala-gejala dari Kerusakan mesin mobil.
- Data bobot gejala yaitu data yang digunakan untuk menyimpan data nama kerusakan dan nilai densitas.

4.2.2.2 DFD Level 0

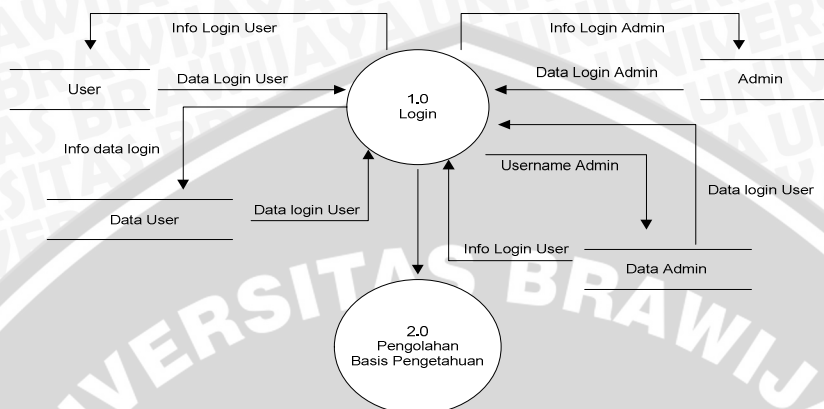
DFD Level 0 memiliki tiga proses utama dengan dua entitas yaitu *admin* dan *user* seperti pada Gambar 4.4.



Gambar 4. 4 DFD Level 0

4.2.2.3 DFD Level 1 Proses 1.0

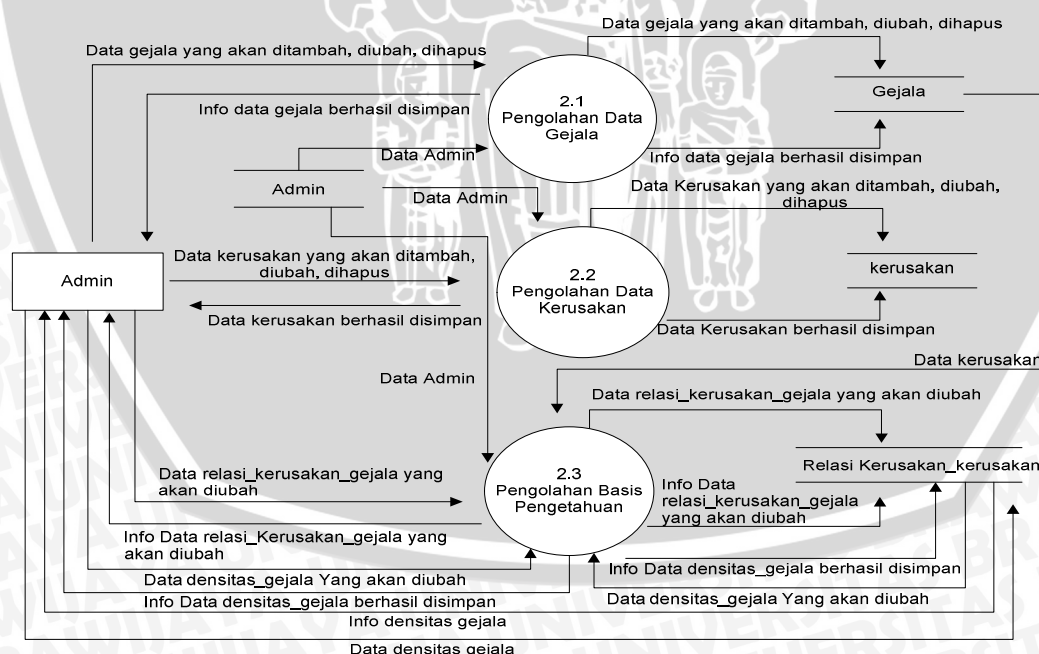
DFD Level 1 proses 1.0 menggambarkan proses *login*, dimana dibedakan menjadi dua pengguna yaitu *admin* dan *user*, seperti terlihat pada Gambar 4.5.



Gambar 4. 5 DFD Level 1 proses 1.0

4.2.2.4 DFD Level 1 Proses 2.0

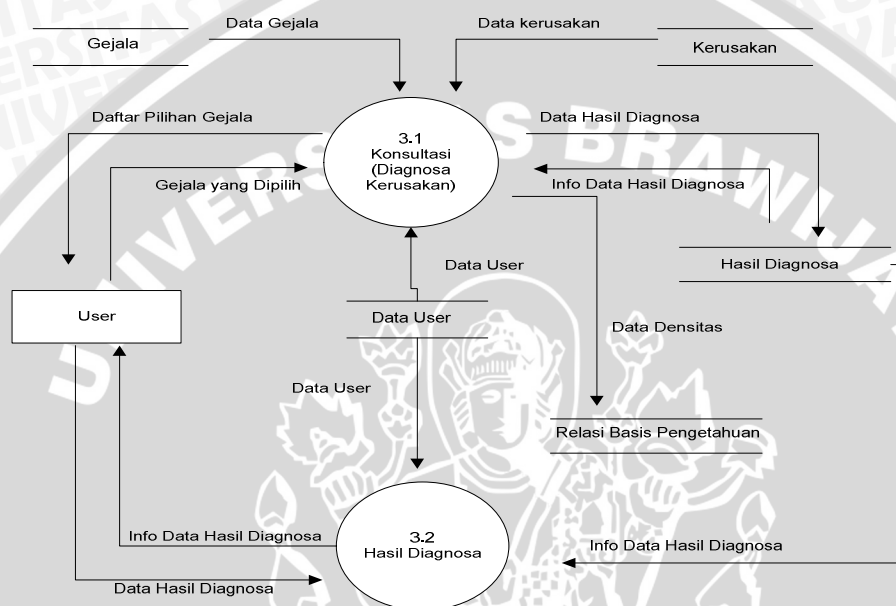
DFD Level 1 proses 2.0 menggambarkan proses pengolahan basis pengetahuan meliputi pengolahan data gejala, pengolahan data kerusakan, pengolahan data relasi kerusakan-gejala dan pengolahan data bobot gejala, seperti terlihat pada Gambar 4.6.



Gambar 4. 6 Proses DFD level 1 proses 2.0

4.2.2.5 DFD Level 1 Proses 3.0

DFD Level 1 proses 3.0 menggambarkan proses diagnosa kerusakan yang dilakukan oleh *user*. Dalam mendiagnosa kerusakan, *user* harus melakukan konsultasi dengan sistem dengan memilih gejala-gejala yang dirasakan sesuai daftar gejala yang ditampilkan oleh sistem. Setelah proses konsultasi selesai, sistem akan menampilkan hasil diagnosa. Untuk lebih jelas, dapat dilihat pada Gambar 4.7.

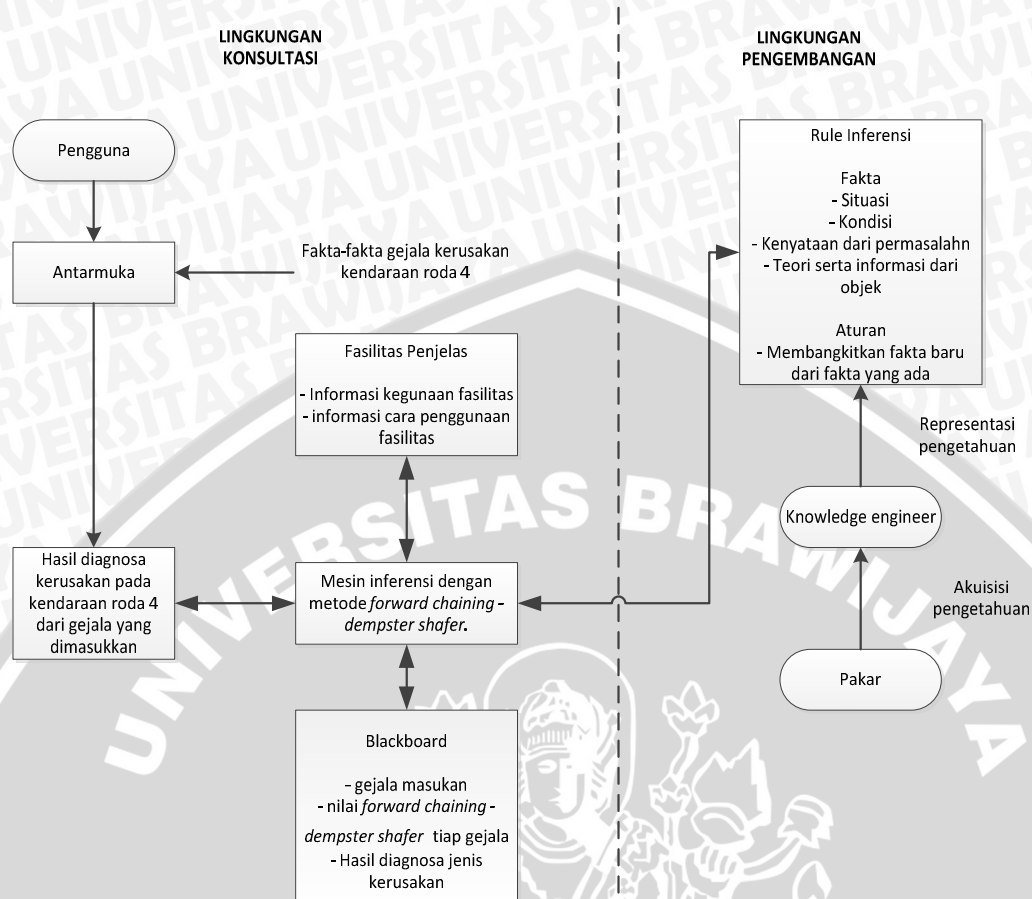


Gambar 4. 7 DFD Level 1 proses 3.0

4.3 Perancangan Sistem Pakar

Sistem pakar pada penelitian ini digunakan untuk mendiagnosa kerusakan kendaraan roda 4. Metode yang diterapkan yaitu *forward chaining - dempster shafer* yang digunakan untuk proses pengambilan kesimpulan.

Tahapan yang dilakukan yaitu sistem menerima masukan berupa keyakinan pengguna terhadap gejala yang dialami, semakin besar tingkat keyakinan yang dimasukkan dan semakin spesifik gejala yang dapat diamati maka diharapkan keputusan dapat mencapai presentase yang semakin tinggi. Hasil akhir dari sistem ini berupa keputusan jenis kerusakan apa yang dialami suatu kendaraan roda 4 dari gejala yang dimasukkan beserta persentase tingkat keyakinan *forward chaining - dempster shafer*. Gambaran konsep arsitektur sistem pakar deteksi kerusakan kendaraan roda 4 ditunjukkan pada gambar 4.8.



Gambar 4. 8 Kerangka Arsitektur Sitem Pakar Kerusakan Kendaraan Roda 4

4.3.1 Akuisisi Pengetahuan

Akuisisi pengetahuan adalah suatu proses untuk mengumpulkan data-data pengetahuan suatu masalah dari pakar. Bahan pengetahuan dapat diambil dari beberapa cara, sebagai contoh dari buku. Sumber pengetahuan tersebut diperoleh dengan kemampuan penulis agar data yang tersedia dapat diolah menjadi suatu solusi yang efisien, dan juga komunikasi yang baik. Pada penelitian ini metode yang dipakai adalah metode wawancara dengan pakar.

a. Wawancara

Wawancara bertujuan untuk memperoleh wawasan dari seorang pakar mengenai masalah pada penelitian ini. Pada wawancara ini dikumpulkan semua informasi tentang gejala kerusakan mesin kendaraan roda 4 yang terdiri dari beberapa gejala dan jenis kerusakan pada mesin kendaraan roda 4. Setiap gejala diberikan bobot densitas oleh pakar. Bobot yang telah diberikan pakar digunakan untuk perhitungan nilai persentase diagnosa kerusakan mesin kendaraan roda 4 dari masukan gejala oleh pengguna.

Pada wawancara ini, *Knowledge Engineer* mengumpulkan informasi tentang Kerusakan Mesin kendaraan roda 4. Informasi yang dikumpulkan berupa jenis-

jenis kerusakan dapat dilihat pada Tabel 4.1 dan Tabel 4.2 merupakan gejala-gejala Kerusakan Mesin Pada Kendaraan Roda 4.

Tabel 4. 1 Tabel Jenis-Jenis Kerusakan Mesin Kendaraan 4

Kode	Nama Kerusakan
K001	<i>Busi Mati</i>
K002	<i>Piston Rusak</i>
K003	<i>Sensor Tidak Berfungsi</i>
K004	Injector Buntu
K005	<i>Koil Rusak</i>
K006	Radiator Rusak

Tabel 4. 2 Gejala-Gejala Kerusakan Mesin Kendaraan 4

Kode	Gejala
G001	Mesin Injeksi brebet
G002	Mesin Tersendat saat di Gas
G003	Tarikan Berat
G004	Mesin Sulit dihidupkan
G005	Suara mesin berdecit
G006	Konsleting
G007	Overheating Radiator
G008	Oli bocor diruang mesin
G009	Selang radiator bocor
G010	Kipas radiator berputar terus
G011	Asap knalpot berwarna putih
G012	Tenaga mesin berkurang tiba-tiba
G013	Oli boros
G014	Bahan bakar boros
G015	Mesin tiba-tiba mati
G016	Lampu indikator nyala

b. Analisa Protokol (Aturan)

Pada analisa protokol, pakar akan diminta memberikan proses pemikirannya. Proses ini akan dijadikan sebagai rule inferensi tentang gejala yang ada dan pemberian bobot pada masing-masing gejala untuk perhitungannya. Diagram alir pembobotan pakar dimulai dengan mengumpulkan beberapa gejala yang menjadi acuan untuk kesimpulan. Kemudian pakar akan memberikan nilai bobot pada tiap gejala yang mengacu pada sumber.

Hasil akusisi pengetahuan gejala kerusakan mesin kendaraan roda 4 yang diperoleh dalam proses wawancara dapat dilihat pada Tabel 4.3 dan Nilai kepercayaan (densitas) dapat dilihat pada Tabel 4.4.

Tabel 4. 3 Tabel Akusisi Kerusakan Mesin Kendaraan Roda 4

Gejala		Jenis Kerusakan					
		K001	K002	K003	K004	K005	K006
G001	Mesin Injeksi brebet	✓			✓	✓	
G002	Mesin tersendat saat di gas	✓	✓	✓			
G003	Tarikan berat	✓			✓		✓
G004	Mesin sulit dihidupkan			✓	✓	✓	
G005	Suara mesin berdecit					✓	
G006	Konsleting			✓			
G007	Overheating radiator						✓
G008	Oli bocor diruang mesin		✓				
G009	Selang radiator bocor						✓
G010	Kipas radiator berputar terus						✓
G011	Asap knalpot berwarna putih		✓				
G012	Tenaga mesin berkurang tiba-tiba berkurang	✓	✓	✓	✓		
G013	Oli boros		✓				
G014	Bahan bakar boros	✓	✓	✓	✓	✓	
G015	Mesin tiba-tiba mati	✓			✓		
G016	Lampu indikator nyala			✓			✓

Tabel 4. 4 Tabel Nilai Densitas Kerusakan Mesin Kendaraan Roda 4

Gejala		Jenis Kerusakan					
		K001	K002	K003	K004	K005	K006
G001	Mesin Injeksi brebet	0,7			0,5	0,5	
G002	Mesin tersendat saat di gas	0,7	0,5	0,6			
G003	Tarikan berat	0,4			0,4		0,4
G004	Mesin sulit dihidupkan			0,3	0,7	0,7	
G005	Suara mesin berdecit					0,8	
G006	Konsleting			0,7			
G007	Overheating radiator						0,8
G008	Oli bocor diruang mesin		0,6				
G009	Selang radiator bocor						0,7
G010	Kipas radiator berputar terus						0,6
G011	Asap knalpot berwarna putih		0,8				
G012	Tenaga mesin berkurang tiba-tiba berkurang	0,5	0,3	0,3	0,5		
G013	Oli boros		0,4				

G014	Bahan bakar boros	0,4	0,3	0,2	0,6	0,8	
G015	Mesin tiba-tiba mati	0,6			0,4		
G016	Lampu indikator nyala			0,8			0,5

Keterangan :

K001	Busi Mati
K002	Piston Rusak
K003	Sensor Tidak Berfungsi
K004	Injector Buntu
K005	Koil Rusak
K006	Radiator Rusak

4.3.2 Basis Pengetahuan

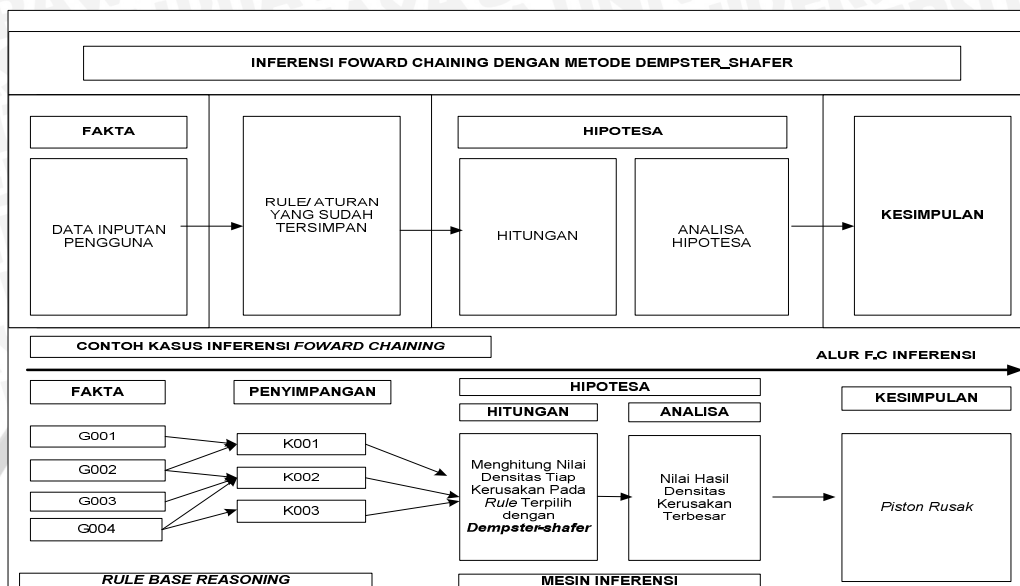
Basis pengetahuan berisi tentang pengetahuan yang relevan yang diperlukan untuk memahami, merumuskan, dan memecahkan persoalan. Basis pengetahuan tersebut terdiri dari dua pendekatan berbasis aturan yang direpresentasikan dalam bentuk fakta dan pendekatan berbasis kasus berisi tentang pencapaian solusi. Dalam penggunaan metode *Dempster-shafer* pengambilan data sebagai pengetahuan yang dibutuhkan terutama pada gejala-gejala dalam menentukan kerusakan mesin mobil, nilai densitas yang diberikan pakar akan dijadikan sebagai bahan perhitungan metode *Dempster-shafer*.

4.3.3 Mesin Inferensi

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. Mesin inferensi yang digunakan pada sistem ini adalah *Forward Chaining* dan *Dempster-shafer*. Pada penelitian ini penelusuran dimulai dari premis (gejala) untuk menentukan konklusi (kerusakan). Teknik seperti ini disebut teknik *Forward Chaining*. Hasil penelusuran didapat berdasarkan pada nilai kepastian tiap premis (gejala) yang dihitung menggunakan metode *Dempster-shafer*.

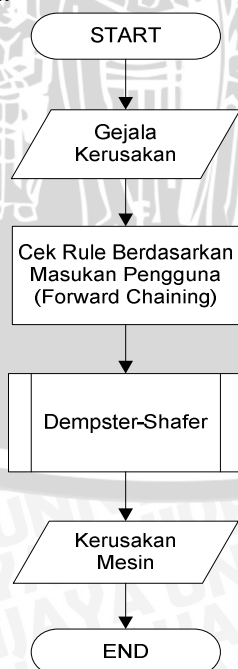
Metode *Forward chaining* yaitu dimulai dari sekumpulan fakta-fakta tentang suatu gejala yang diberikan oleh pengguna sebagai masukan pada sistem. Kemudian dilakukan proses pelacakan dari masukan gejala dengan gejala yang tersimpan pada basis data untuk kemudian diambil nilai kepercayaan tiap kerusakan yang sesuai. Setelah didapat nilai bobotnya maka dilakukan hipotesa

yang terdiri dari 2 bagian yaitu proses perhitungan dengan menggunakan *Dempster-shafer* dan analisa hipotesa hasil dari perhitungan akhir yang kemudian dijadikan sebagai kesimpulan. Hipotesa blok diagram alur proses metode inferensi *Foward chaining* dapat dilihat pada Gambar 4.10.



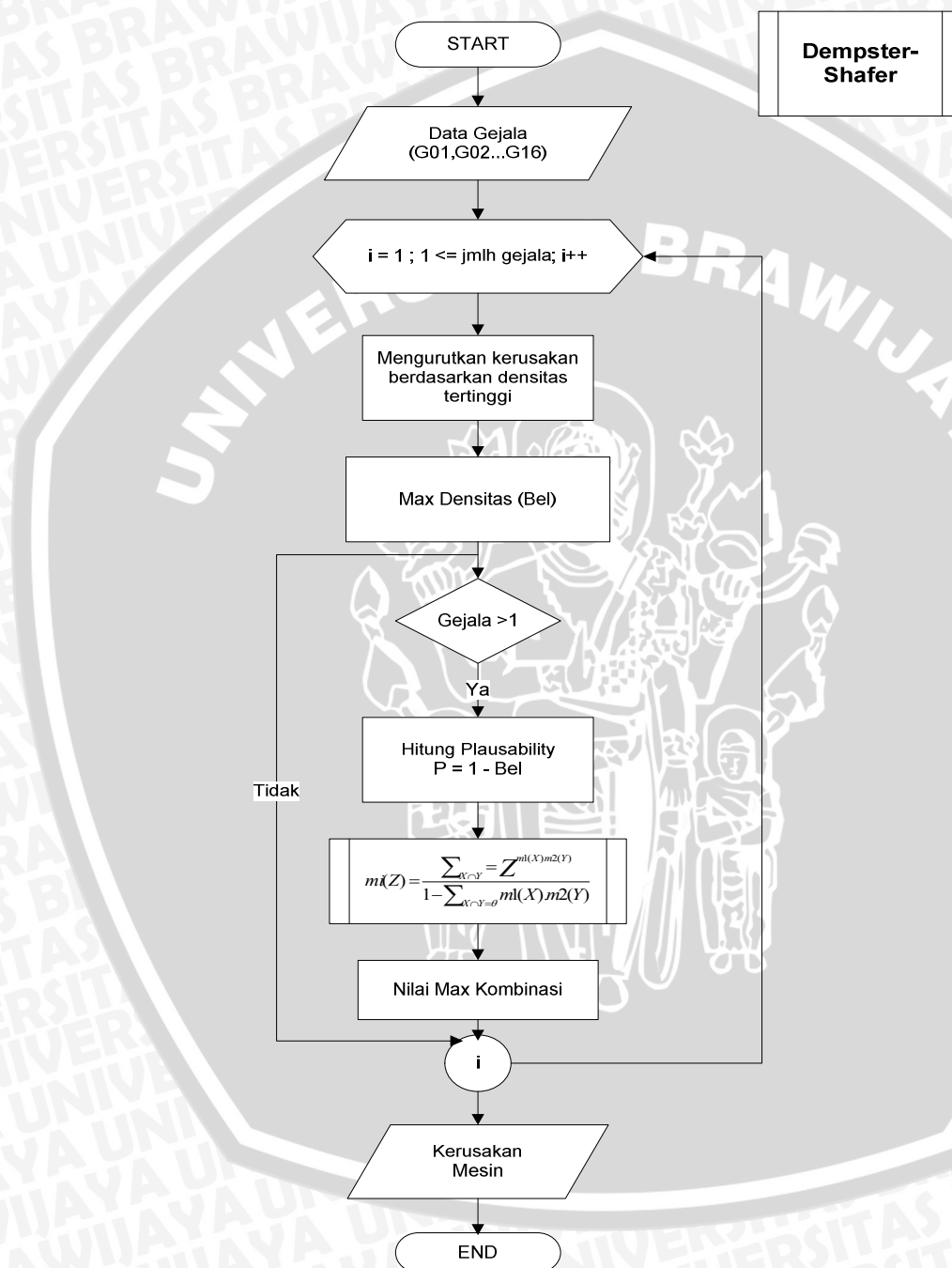
Gambar 4. 9 Mesin Inferensi Foward Chaining Dengan Metode Dempster-Shafer

Flowchart atau diagram alir gambaran pencarian solusi sistem pakar menggunakan metode *Forward Chaining* untuk mencari rule berdasarkan masukan pengguna dan Metode *Dempster-shafer* sebagai penarikan kesimpulan dapat dilihat pada gambar 4.11.



Gambar 4. 11 Flowchart Sistem

Untuk proses perhitungan inferensi metode *Dempster-shafer* sebagai penarikan kesimpulan dapat dilihat pada Gambar 4.12 yang merupakan gambaran pencarian solusi sistem pakar dengan menggunakan *flowchart* atau diagram alir.



Gambar 4. 12 Flowchart inferensi Dempster-shafer

Perhitungan dimulai dengan memasukkan nilai densitas tiap gejala oleh pakar yang kemudian disimpan dibasis data sebagai dasar perhitungan. Pengguna memasukkan gejala-gejala fakta kerusakan yang dirasakan oleh pemilik kendaraan ke dalam program. Didalam program akan dilakukan proses pencocokan gejala yang dimasukkan pengguna dengan gejala yang terdapat pada basis data sehingga didapatkan kemungkinan nama kerusakan dan nilai densitasnya untuk kemudian dihitung nilai *belief* dan *plausibility*-nya. Setelah didapatkan nilainya, kemudian dilihat jika banyaknya gejala adalah satu, maka dari hasil kemungkinan nama kerusakan yang sesuai dengan gejala tersebut dan memiliki nilai *belief* tertinggi yang akan dijadikan solusi.

Tetapi, jika gejala yang dimasukkan lebih dari satu, maka hasil dari nama kerusakan dan nilai *belief*, *plausibility* gejala ke-1 akan disimpan sementara pada *blackboard*. Untuk gejala ke-2 dilakukan tahapan yang sama dengan gejala ke-1 dan hasilnya juga disimpan sementara. Setelah didapatkan nilai dua gejala yang ada maka dapat dilakukan perhitungan untuk mencari nilai densitas gabungan atau nilai densitas ke-3 yang berasal dari nilai gejala satu dan dua serta kemungkinan nama kerusakan dihitung menggunakan persamaan 2.3. Dari hasil nilai densitas ke-3 akan didapatkan kemungkinan nama kerusakan dengan nilai densitas baru yang kemungkinan disimpan kedalam *blackboard*. Setelah mendapatkan nilai densitas ke-3 dan masih terdapat gejala lain yang dimasukkan, maka dilakukan perhitungan nilai densitas gabungan baru antara nilai densitas ke-3 dengan nilai dari gejala ke-3 seperti tahapan sebelumnya. Perhitungan tersebut terus berulang dilakukan selama gejala yang dimasukkan pengguna belum habis dihitung semua.

Jika sudah tidak ada gejala dari pengguna yang dihitung, maka solusinya akan didapatkan dari hasil nilai densitas gabungan yang paling terakhir dihitung. Jika nilai densitas yang didapat dari perhitungan terakhir lebih dari satu kemungkinan kerusakan, maka akan dipilih kemungkinan kerusakan yang memiliki nilai densitas tertinggi yang kemungkinan dijadikan solusinya.

4.3.4 Perhitungan Kasus Secara Manual

Perhitungan manual berfungsi untuk memberikan gambaran umum perancangan sistem yang akan dibangun. Contoh manualisasi akan dibagi menjadi 2 kasus, yaitu kasus 1 dengan perhitungan 1 gejala dimasukkan, kasus 2 dengan perhitungan 2 gejala dimasukkan. Perhitungan dimulai dengan proses *forward chaining* yaitu dimulai dengan pengumpulan fakta-fakta tentang gejala - gejala yang diberikan oleh pengguna sebagai masukan pada sistem. Kemudian dilakukan proses pelacakan dari masukan gejala dengan gejala yang tersimpan pada basis data untuk kemudian diambil nilai kepercayaan tiap kerusakan yang sesuai. Kemudian hasil penelusuran yang didapat berdasarkan pada nilai kepastian tiap premis (gejala) yang dihitung menggunakan metode *Dempster-shafer*. Tabel akusisi nilai densitas kerusakan yang akan dijadikan acuan perhitungan dapat dilihat pada Tabel 4.7.

4.3.4.1 Kasus 1 (Perhitungan 1 Gejala)

Dengan menganalisis gejala-gejala yang diberikan oleh *user* untuk mendapatkan kemungkinan nama kerusakannya, maka dilakukan perhitungan nilai densitas dari gejala kerusakan dengan menggunakan nilai kepercayaan dari pakar yang kemudian dimasukkan ke dalam persamaan 2.3 pada dasar teori.

Kasus 1 ini akan diberikan contoh dengan memasukkan 1 gejala yang terjadi pada salah satu kerusakan misalkan yaitu *mesin injeksi brebet*. Pada perhitungan ini diibaratkan ketika pengendara sedang melakukan perjalanan terasa mesin brebet saat sedang bersjalan dengan diagnosa mekanik, kerusakan mungkin yang dialami adalah *busi mati*, *Injector buntu* dan Koil rusak seperti pada tabel akusisi nilai densitas kerusakan pada Tabel 4.6.

➤ **Gejala 1: Mesin Injeksi Brevet**

Dilakukan observasi Mesin injeksi brebet sebagai gejala dari kerusakan dengan nilai: $m\{Busi\ Mati\}= 0,7$, $m\{Injektor\ buntu\}= 0,5$ dan $m\{Koil\ Rusak\}= 0,5$. Untuk m_1 nilai densitas yang dipilih adalah yang tertinggi, maka:

$$m_1\{K001, K004, K005\} = 0,7$$

Untuk mencari nilai *plausability* dari gejala yang dipilih menggunakan persamaan 2.2.

$$m_1\{\theta\} = 1 - 0,7 = 0,3$$

Diurutkan kerusakannya dari nilai densitas gejala tertinggi:

$$m\{Busi\ mati\} = 0,7$$

$$m\{Koil\ rusak\} = 0,5$$

$$m\{Injektor\ buntu\} = 0,5$$

Dari hasil perhitungan dengan metode Dempster-shafer karena gejala yang diketahui hanya 1 dan diagnosa kerusakannya lebih dari 1. Maka kesimpulannya dapat diurutkan dari nilai densitas tertinggi. Sehingga hasil diagnosanya dapat disimpulkan bahwa kendaraan tersebut mengalami kerusakan busi mati, Koil rusak dan Injektor buntu.

4.3.4.2 Kasus 2 (Perhitungan 2 Gejala)

Pada kasus 2 ini akan diberikan contoh dengan memasukkan 2 gejala. Pada perhitungan ini diibaratkan mobil terasa mesin injeksi brebet , suara mesin berdecit. Tabel akusisi nilai densitas kerusakan dapat dilihat pada Tabel 4.7.

➤ **Gejala 1: mesin injeksi brebet**

Dilakukan observasi kemampuan sosialisasi kurang sebagai gejala dari kerusakan dengan nilai: $m\{busi\ mati\} = 0,7$, $m\{injector\ buntu\} = 0,5$ dan $m\{Koil\ rusak\} = 0,5$. Untuk m_1 nilai densitas yang dipilih adalah yang tertinggi, maka:

$$m_1\{K001, K004, K005\} = 0,7$$

$$m_1\{\theta\} = 1 - 0,7 = 0,3$$

➤ **Gejala 2: suara mesin berdecit**

Kemudian diperoleh gejala selanjutnya setelah dilakukan observasi terhadap suara mesin berdecit gejala dari: $m\{Koil\ rusak\} = 0,8$. Untuk m_2 nilai densitas yang dipilih nilai tertinggi yaitu:

$$m_2\{K005\} = 0,8$$

$$m_2\{\theta\} = 1 - 0,8 = 0,2$$

Maka dihitung nilai densitas baru untuk beberapa kombinasi dengan fungsi densitas m_3 dapat dilihat pada Tabel 4.8.

Tabel 4. 5 Aturan Kombinasi Untuk m_3 Kasus 2

m1	m2	
	{K005}	θ
{K001, K004, K005} 0,7	{K005} 0,56	{K001, K004, K005} 0,14
θ 0,3	{ K005 } 0,24	θ 0,06

Sehingga dapat dihitung dengan persamaan 2.3

$$\circ m_3\{K005\} = \frac{0,56+0,24}{1-0} = 0,8$$

$$\circ m_3\{K001, K004, K005\} = \frac{0,14}{1-0} = 0,14$$

$$\circ m_3\{\theta\} = \frac{0,06}{1-0} = 0,06$$

Hasil diagnosa dari perhitungan *Dempster-shafer* diambil dari nilai densitas yang tertinggi. Maka dapat disimpulkan bahwa mobil tersebut mengalami kerusakan Busi mati dengan prosentase kemungkinan sebesar 14%.

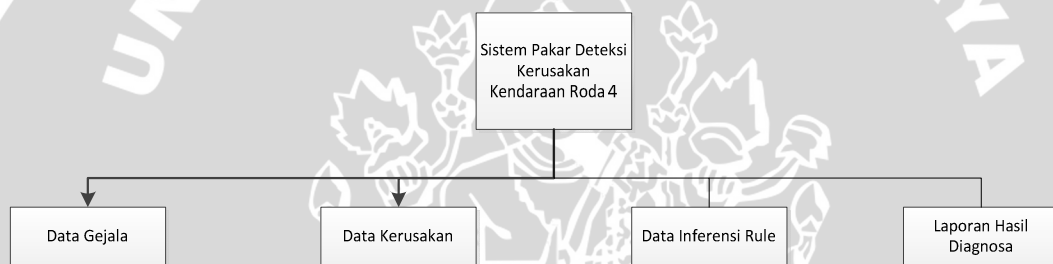
4.3.5 Daerah Kerja (Blackboard)

Blackboard merupakan area memori yang berfungsi sebagai basis data untuk merekam hasil sementara. *Blackboard* berisi rencana solusi yang berupa data yang digunakan sebagai bahan pertimbangan dalam memberikan kesimpulan akhir. Pada aplikasi diagnose kerusakan mesin kendaraan roda 4, data yang disimpan pada area ini adalah gejala dari pengguna, nilai perhitungan *belief* dan *plausibility* tiap gejala, hasil perhitungan densitas baru dan hasil akhirnya serta hasil diagnosa kerusakan.

4.3.6 Perancangan Antarmuka

- Perancangan Struktur Menu

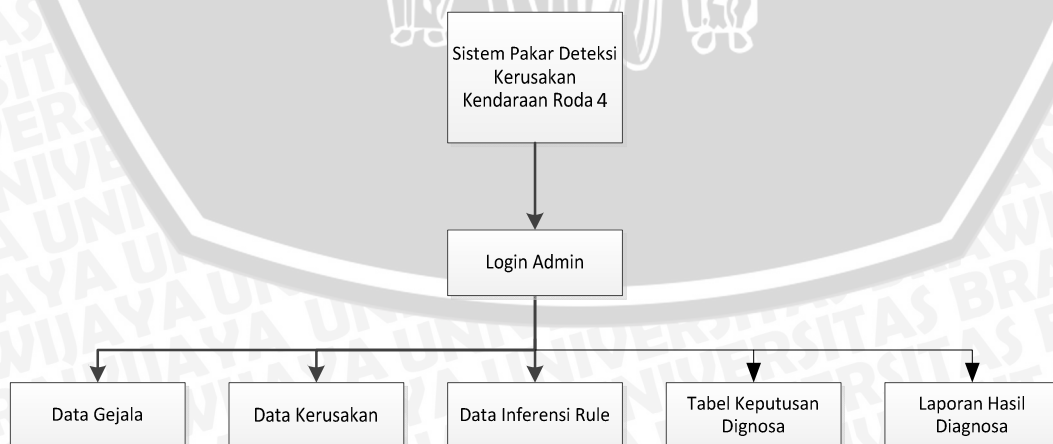
Perancangan struktur menu terdiri dari menu dan submenu yang berfungsi memudahkan pengguna dalam menggunakan sistem. Struktur pada sistem terdiri atas halaman utama, data gejala, data kerusakan mobil, data inferensi rule dan laporan hasil diagnosa kerusakan ditunjukkan pada gambar 4.13.



Gambar 4. 13 Struktur Menu

- Struktur Menu Pakar

Struktur menu pakar terdiri dari halaman utama, data gejala, data kerusakan mobil, data inferensi rule, tabel keputusan diagnosa dan laporan hasil seperti ditunjukkan pada gambar 4.14.



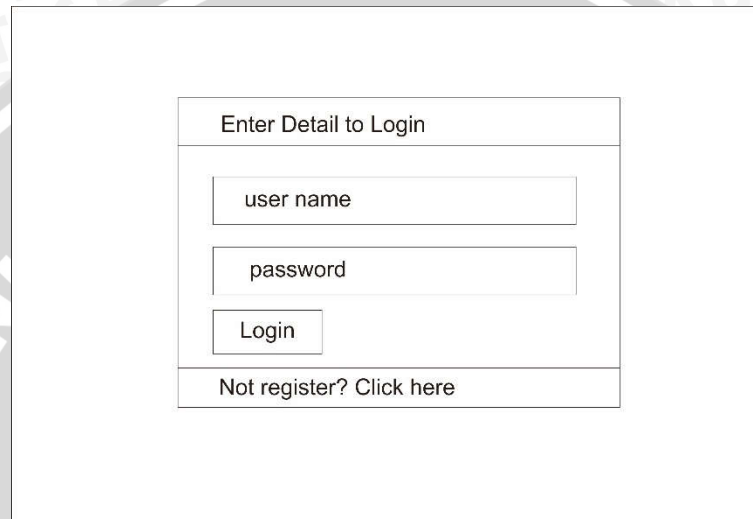
Gambar 4. 14 Struktur Menu Pakar

4.3.4 Perancangan Antarmuka Program

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang dibangun, sehingga dapat memudahkan dalam implementasi aplikasi.

4.3.4.1 Perancangan Antarmuka Halaman login

Rancangan antarmuka *login* yang terdiri dari *username* dan *password*. Rancangan ditunjukkan pada gambar 4.15.



Enter Detail to Login

user name

password

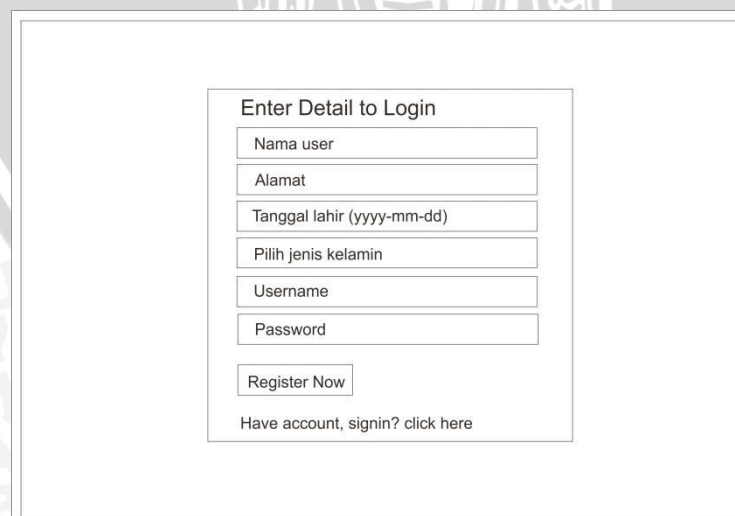
Login

Not register? Click here

Gambar 4. 15 Antarmuka Halaman Login

4.3.4.2 Antarmuka halaman register

Rancangan antarmuka halaman *Register*. Apabila pengguna belum memiliki akun untuk melakukan proses diagnosa, pengguna terlebih dahulu mendaftar dengan mengisi nama, alamat, jenis kelamin, tanggal lahir, *username* dan *password*. Rancangan ditunjukkan pada gambar 4.16.



Enter Detail to Login

Nama user

Alamat

Tanggal lahir (yyyy-mm-dd)

Pilih jenis kelamin

Username

Password

Register Now

Have account, signin? click here

Gambar 4. 16 Antarmuka Halaman Regis

4.3.4.3 Antarmuka halaman utama user

Rancangan antarmuka diagnose Kerusakan mesin. Halaman ini menampilkan pengguna yang melakukan diagnosa. Pengguna melakukan diagnosa dengan memberi tanda pada list gejalanya kemudian untuk mengetahui kesimpulannya dengan menekan tombol diagnosa. Rancangan ditunjukkan pada gambar 4.17.

HEADER	
Username	profile logout
Diagnosa	Diagnosa (pilih gejala-gejala yang dialami) ↕ ☐ Gejala 1 ☐ Gejala 2 ☐ Gejala 3 ☐ Gejala 4 ☐ Gejala 5 Diagnosa
Riwayat Diagnosa	
FOOTER	

Gambar 4. 17 Antarmuka Halaman Diagnosa

4.3.4.4 Antarmuka halaman riwayat diagnosa user

Rancangan antarmuka riwayat diagnosa. Halaman ini menampilkan rincian riwayat diagnosa yang pernah dilakukan oleh *user*. Rancangan ditunjukkan pada gambar 4.18.

HEADER																									
Username	profile logout																								
Diagnosa	Riwayat Diagnosa <table border="1"> <thead> <tr> <th>No</th> <th>Nama</th> <th>Kerusakan</th> <th>Prosentase</th> <th>Tanggal</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td> Edit Delete </td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	No	Nama	Kerusakan	Prosentase	Tanggal	Aksi						Edit Delete												
No	Nama	Kerusakan	Prosentase	Tanggal	Aksi																				
					Edit Delete																				
Riwayat Diagnosa																									
FOOTER																									

Gambar 4. 18 Antarmuka Riwayat Diagnosa

4.3.4.5 Antarmuka Halaman Master Gejala

Rancangan antarmuka gejala. Halaman ini berisi informasi tentang gejala-gejala kerusakan. Admin dapat menambah, mengubah dan menghapus gejala. Rancangan ditunjukkan pada gambar 4.19.

HEADER		profile		logout	
Username		Data Gejala			
		Tambah			
User Terdaftar	Kode	Gejala	Aksi		
Gejala			Edit Delete		
Kerusakan					
Basis Pengetahuan					
Riwayat sistem					
Diagnosa					
FOOTER					

Gambar 4. 19 Antarmuka Gejala

4.3.4.6 Antarmuka Halaman kerusakan

Rancangan antarmuka halaman kerusakan yang berisi informasi tentang kerusakan. Admin dapat menambah, melakukan perubahan dan menghapus data kerusakan. Rancangan ditunjukkan pada gambar 4.20.

HEADER		profile		logout	
Username		Data Kerusakan			
		Tambah			
User Terdaftar	Kode	Kerusakan	Penanganan	Pencegahan	Aksi
Gejala					Edit Delete
Kerusakan					
Master Basis Pengetahuan					
Riwayat Diagnosa					
Diagnosa					
FOOTER					

Gambar 4. 20 Antarmuka data kerusakan

4.3.4.7 Antarmuka Halaman Basis Pengetahuan

Rancangan antarmuka halaman basis pengetahuan. Admin dapat menambah nilai densitas dari setiap kerusakan dengan gejala tertentu. Admin juga dapat melakukan perubahan dan menghapus data basis pengetahuan. Rancangan ditunjukkan pada gambar 4.21.

HEADER		profile logout																								
Username		Data Basis Pengetahuan Tambah																								
User Terdaftar	<table border="1"> <thead> <tr> <th>No</th> <th>Kerusakan</th> <th>Gejala</th> <th>Densitas</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td> Edit Delete </td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>						No	Kerusakan	Gejala	Densitas	Aksi					Edit Delete										
No	Kerusakan	Gejala	Densitas	Aksi																						
				Edit Delete																						
Gejala																										
Kerusakan																										
Basis Pengetahuan																										
Riwayat Sistem																										
Diagnosa																										
FOOTER																										

Gambar 4. 21 Antarmuka Basis Pengetahuan

4.3.4.8 Antarmuka Halaman Diagnosa Admin

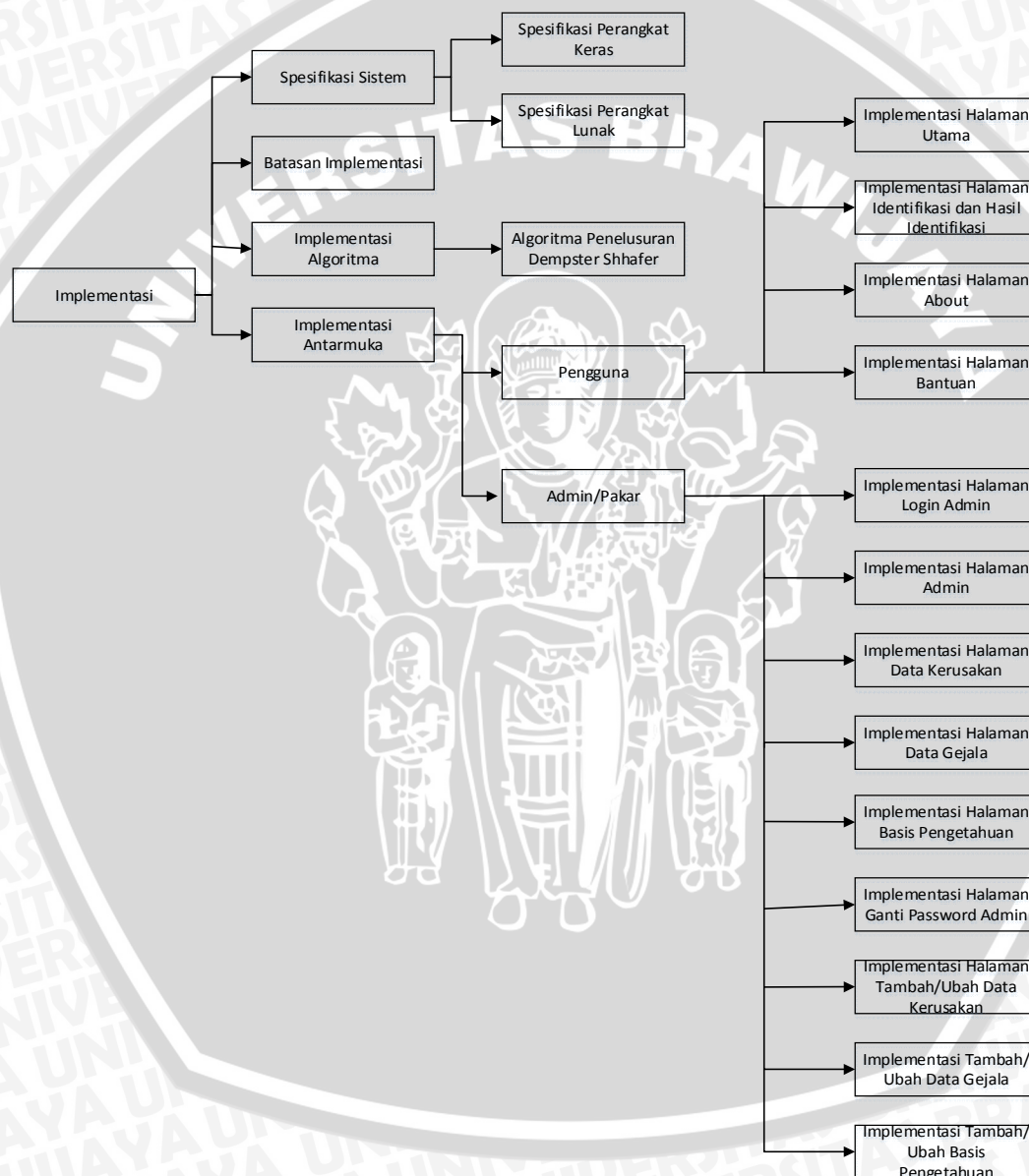
Rancangan antarmuka riwayat diagnosa. Halaman ini menampilkan rincian riwayat diagnosa yang pernah dilakukan oleh user terdaftar. Rancangan ditunjukkan pada gambar 4.22.

HEADER		profile logout																													
Username		Riwayat Diagnosa																													
User Terdaftar	<table border="1"> <thead> <tr> <th>No</th> <th>Nama</th> <th>Kerusakan</th> <th>Prosentase</th> <th>Tanggal</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td> Edit Delete </td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>							No	Nama	Kerusakan	Prosentase	Tanggal	Aksi						Edit Delete												
No	Nama	Kerusakan	Prosentase	Tanggal	Aksi																										
					Edit Delete																										
Gejala																															
Kerusakan																															
Basis Pengetahuan																															
Riwayat sistem																															
Diagnosa																															
FOOTER																															

Gambar 4. 22 Antarmuka Halaman Diagnosa Admin

BAB 5 IMPLEMENTASI

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari proses perancangan yang telah dibuat. Pembahasan terdiri dari penjelasan tentang batasan-batasan implementasi, implementasi basis pengetahuan, implementasi mesin inferensi dan implementasi antarmuka. Diagram implementasi identifikasi kerusakan Mesin Mobil menggunakan metode *Dempster-Shafer* dapat dilihat pada Gambar 5.1 .



Gambar 5. 1 Diagram Implementasi

5.1 Spesifikasi Sistem

Hasil kebutuhan dan perancangan perangkat lunak yang telah dibahas pada bab sebelumnya menjadi acuan untuk melakukan implementasi sebuah sistem yang dapat berfungsi sesuai kebutuhan. Spesifikasi sistem diimplementasi pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Pembuatan Sistem Pakar Diagnosa Kerusakan Mesin kendaraan roda 4 Dengan Metode *Forward chaining- Dempster Shafer* menggunakan sebuah laptop dengan spesifikasi pada Table 5.1.

Tabel 5. 1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel(R) Core(TM) i3-2410M CPU @ 2.50GHz ~2.5 GHz
Memori RAM	2.00 GB
Kartu Grafis	Intel(R) HD Graphics 3000
Hardisk	640 GB HDD

5.1.2 Spesifikasi Perangkat Lunak

Pembuatan Spemodelan Sstem Pakar Untuk Identifikasi Kerusakan Mesin Mobil Dengan Metode *Dempster Shafer* menggunakan sebuah aplikasi perangkat lunak dengan spesifikasi pada Table 5.2.

Tabel 5. 2 Spesifikasi Perangkat Lunak

Nama Aplikasi	Spesifikasi Perangkat Lunak
Sistem Operasi	Microsoft Windows 7 (64-bit)
Bahasa Pemrograman	HTML 5 dan PHP
Tools Pemrograman	Sublime Text 2
<i>Server Localhost</i>	XAMPP
DBMS	MySQL

5.2 Batasan – Batasan Implementasi

Batasan implementasi bertujuan untuk memberikan ruang lingkup yang jelas pada sistem. Beberapa batasan dalam mengimplementasikan Sistem Pakar Identifikasi Kerusakan Mesin Mobil adalah:

1. Masukan yang diterima oleh sistem adalah berupa inputan gejala-gejala yang dipilih tentang kondisi Mobil yang diinputkan oleh pengguna.
2. Keluaran yang diterima oleh pengguna berupa kerusakan dan nilai kepercayaannya.
3. Aplikasi yang digunakan berbasis web dengan basis data penyimpanan menggunakan MySQL.
4. Metode yang digunakan yaitu *Forward Chaining - Dempster-Shafer*.
5. Sistem pakar yang dibangun berguna untuk identifikasi kerusakan Mesin tipe Mesin Injeksi tahun produksi 2005 - 2010
6. Sistem pakar ini dapat mendeteksi 6 jenis kerusakan dengan 16 gejala.
7. Pengguna bisa langsung melakukan identifikasi kerusakan Mesin mobil.

5.3 Implementasi Algoritma Dempster Shafer

Proses identifikasi ini dengan cara menginputkan fakta-fakta gejala yang terjadi pada mobil pengguna. Kemudian dilakukan proses identifikasi, implementasi algoritma *Dempster-Shafer* dapat dilihat pada sourcecode 5.1.

```

1 function intersectArray($array1, $array2) {
2     $intersect = array();
3     if ($array1['kerusakan'] == 'theta' || implode(",", $array1['kerusakan']) ===
'theta') {
4         $intersect = $array2['kerusakan'];
5     } else if ($array2['kerusakan'] == 'theta' || implode(",", $array2['kerusakan'])
6 === 'theta') {
7         $intersect = $array1['kerusakan'];
8     } else {
9         if ($array2['densitas'] > $array1['densitas']) {
10             $reverse = TRUE;
11         } else {
12             $reverse = FALSE;
13         }
14         if ($reverse) {
15             for ($i = 0; $i < count($array2['kerusakan']); $i++) {
16                 for ($j = 0; $j < count($array1['kerusakan']); $j++) {

```

```

16         if ($array2['kerusakan'][$i] == $array1['kerusakan'][$j]) {
17             $intersect[] = $array2['kerusakan'][$i];
18             break;
19         }
20     }
21 }
22 } else {
23     for ($i = 0; $i < count($array1['kerusakan']); $i++) {
24         for ($j = 0; $j < count($array2['kerusakan']); $j++) {
25             if ($array1['kerusakan'][$i] == $array2['kerusakan'][$j]) {
26                 $intersect[] = $array1['kerusakan'][$i];
27                 break;
28             }
29         }
30     }
31 }
32 if (empty($intersect)) {
33     $intersect[] = 'kosong';
34 }
35 return $intersect;
36 }
37 function arraySort($array, $on, $order = SORT_ASC) {
38     $new_array = array();
39     $sortable_array = array();

```

```
40     if (count($array) > 0) {  
41         foreach ($array as $key1 => $value1) {  
42             if (is_array($value1)) {  
43                 foreach ($value1 as $key2 => $value2) {  
44                     if ($key2 == $on) {  
45                         $sortable_array[$key1] = $value2;  
46                     }  
47                 }  
48             } else {  
49                 $sortable_array[$key1] = $value1;  
50             }  
51         }  
52         switch ($order) {  
53             case SORT_ASC:  
54                 asort($sortable_array);  
55                 break;  
56             case SORT_DESC:  
57                 arsort($sortable_array);  
58                 break;  
59         }  
60         foreach ($sortable_array as $key1 => $value1) {  
61             $new_array[$key1] = $array[$key1];  
62         }  
63         return $new_array;  
64     }  
65 }
```

```

64 function maksDensitas($db, $kode_gejala) {
65     $query = "SELECT kode_kerusakan, nilai_densitas "
66         . "FROM mesin_basis_pengetahuan "
67         . "WHERE kode_gejala='$kode_gejala' "
68         . "ORDER BY nilai_densitas DESC";
69     $result = $db->query($query);
70     $kerusakan_densitas_by_gejala = array();
71     $daftar_kerusakan = array();
72     while ($row = $result->fetch_assoc()) {
73         $kerusakan_densitas_by_gejala[] = $row;
74         $daftar_kerusakan[] = $row['kode_kerusakan'];
75     }
76     // return kode kerusakan dengan densitas terbesar
77     return array($daftar_kerusakan,
78         $kerusakan_densitas_by_gejala[0]['nilai_densitas']);
79 }
80 function doDiagnosa($db, $gejala_array) {
81     $m_probabilitas = array();
82     $m = 0;
83     $jumlah_gejala = count($gejala_array);
84     for ($i = 0; $i < $jumlah_gejala; $i++) {
85         // hitung probabilitas fungsi densitas (m1) { m_kerusakan & m_theta}
86         if ($i == 0) {
87             $kerusakan_densitas = maksDensitas($db, $gejala_array[$i]);
88             $densitas_1 = $kerusakan_densitas[1];
89             $m_probabilitas[$m][0] = array(

```

```

88     "kerusakan" => $kerusakan_densitas[0],
89     "densitas" => $densitas_1);
90     $m_probabilitas[$m][] = array(
91         "kerusakan" => "theta",
92         "densitas" => 1 - $densitas_1);
93     if ($jumlah_gejala - 1 == $i) {
94         $m_output = $m_probabilitas[$m];
95     }
96     } else {
97         // hitung probabilitas fungsi densitas (m2) { m_kerusakan & m_theta}
98         $kerusakan_densitas = maksDensitas($db, $gejala_array[$i]);
99         $densitas_2 = $kerusakan_densitas[1];
100        $m_probabilitas[$m][] = array(
101            "kerusakan" => $kerusakan_densitas[0],
102            "densitas" => $densitas_2);
103        $m_probabilitas[$m][] = array(
104            "kerusakan" => "theta",
105            "densitas" => 1 - $densitas_2);
106        $m++;
107        $is_empty_set = FALSE;
108        $m_probabilitas_tmp_1 = array();
109        // hitung probabilitas fungsi densitas (m3) { m_kerusakan & m_theta}
110        for ($x = 0; $x < count($m_probabilitas[$m - 2]); $x++) {
111            for ($y = 0; $y < count($m_probabilitas[$m - 1]); $y++) {
                $intersected_array = intersectArray($m_probabilitas[$m - 2][$x],
                $m_probabilitas[$m - 1][$y]);
            }
        }
    }
}

```

```

112     $m_probabilitas_tmp_1[$m][] = array(
113         "kerusakan" => $intersected_array,
114         "densitas" => $m_probabilitas[$m - 2][$x]['densitas'] *
115         $m_probabilitas[$m - 1][$y]['densitas']);
116     if (is_array($intersected_array)) {
117         if (implode(",", $intersected_array) === 'kosong') {
118             $sis_empty_set = TRUE;
119         }
120     } else {
121         if ($intersected_array === 'kosong') {
122             $sis_empty_set = TRUE;
123         }
124     }
125 }
126 $m_probabilitas_tmp_2 = array();
127 /**
128  * Restructure Array
129  */
130 // change array structure & convert array to string
131 for ($tmp = 0; $tmp < count($m_probabilitas_tmp_1[$m]); $tmp++) {
132     if (is_array($m_probabilitas_tmp_1[$m][$tmp]['kerusakan'])) {
133         $imploded = implode(",",
134             $m_probabilitas_tmp_1[$m][$tmp]['kerusakan']);
135     } else {
136         $imploded = $m_probabilitas_tmp_1[$m][$tmp]['kerusakan'];
137     }

```

```

136     $m_probabilitas_tmp_2[$m][] = array(
137         "kerusakan" => $implode,
138         "densitas" => $m_probabilitas_tmp_1[$m][$tmp]['densitas'];
139     }
140     $result_m = array();
141     // remove duplicate "kerusakan" & sum densitas THEN reindex array
142     foreach ($m_probabilitas_tmp_2[$m] as $val) {
143         if (!isset($result_m[$val['kerusakan']])) {
144             $result_m[$val['kerusakan']] = $val;
145         } else {
146             $result_m[$val['kerusakan']]['densitas'] += $val['densitas'];
147         }
148     }
149     $result_m_reindex = array_values($result_m); // reindex array
150     $m_probabilitas_tmp_3 = array();
151     if ($is_empty_set) {
152         foreach ($result_m_reindex as $reindex) {
153             if ($reindex['kerusakan'] === 'kosong') {
154                 $him_kosong_val = $reindex['densitas'];
155                 break;
156             }
157         }
158         foreach ($result_m_reindex as $reindex) {
159             if ($reindex['kerusakan'] !== 'kosong') {
160                 $m_probabilitas_tmp_3[] = array(
161                     "kerusakan" => $reindex['kerusakan'],

```

```

160         "densitas" => $reindex['densitas'] / (1 - $him_kosong_val));
161     }
162 }
163 } else {
164     $m_probabilitas_tmp_3 = $result_m_reindex;
165 }
166 $m_probabilitas_tmp_4 = array();
167 /**
168  * Restructure Array - roolback
169  */
170 //explode array
171 foreach ($m_probabilitas_tmp_3 as $m_prob) {
172     if ($m_prob['kerusakan'] !== 'kosong' || $m_prob['kerusakan'] !==
173     'theta') {
174         $exploded = explode(",", $m_prob['kerusakan']);
175     } else {
176         $exploded = $m_prob['kerusakan'];
177     }
178     $m_probabilitas_tmp_4[] = array(
179         "kerusakan" => $exploded,
180         "densitas" => $m_prob['densitas']);
181 }
182 $m_probabilitas[$m] = $m_probabilitas_tmp_4;
183 if ($i == $jumlah_gejala - 1) {
184     $m_output = $m_probabilitas[$m];
185 }

```

```

184     }
185     $m++;
186 }
187 $m_output_temp = array();
188 /**
189  * Restructure Array
190  */
191 // change array structure & convert array to string
192 for ($tmp = 0; $tmp < count($m_output); $tmp++) {
193     if (is_array($m_output[$tmp]['kerusakan'])) {
194         $imploded = implode(",", $m_output[$tmp]['kerusakan']);
195     } else {
196         $imploded = $m_output[$tmp]['kerusakan'];
197     }
198     $m_output_temp[] = array(
199         "kerusakan" => $imploded,
200         "densitas" => $m_output[$tmp]['densitas']);
201 }
202 $sorted_m_probabilitas = arraySort($m_output_temp, 'densitas', SORT_DESC);
203 /**
204  * Restructure Array - roolback
205  */
206 $restructured = NULL;
207 foreach ($sorted_m_probabilitas as $m_prob) {
208     if ($m_prob['kerusakan'] !== 'kosong' || $m_prob['kerusakan'] !== 'theta') {
209         $exploded = explode(",", $m_prob['kerusakan']);

```

```

208     } else {
209         $exploded = $m_prob['kerusakan'];
210     }
211     $restructured[] = array(
212         "kerusakan" => $exploded,
213         "densitas" => $m_prob['densitas']);
214     }
215     $keputusan['kerusakan'] = $restructured[0]['kerusakan'][0];
216     $keputusan['densitas'] = $restructured[0]['densitas'];
217     if ($keputusan['kerusakan'] === 'theta') {
218         return NULL;
219     } else {
220         return array($keputusan['kerusakan'], $keputusan['densitas'] * 100);
221     }
222 }

```

Sourcecode 5.1 Implementasi Algoritma Proses Identifikasi
Penjelasan implementasi algoritma, yaitu :

1. Baris 64 – 78 : proses mencari nilai maksimal
2. Baris 79 - 85 : proses perhitungan 1 gejala
3. Baris 79 – 126 : proses perhitungan tabel kombinasi.
4. Baris 130 – 221 : proses pengubahan nilai array menjadi string

5.4 Implementasi Antarmuka

Antarmuka aplikasi Pemodelan Sistem Pakar Untuk Identifikasi Kerusakan Mesin Mobil ini digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak. Pada implementasi antarmuka perangkat lunak untuk pengguna tidak semua ditampilkan tetapi hanya tertentu saja yaitu halaman utama, halaman identifikasi dan halaman hasil identifikasi.

Sedangkan bagi admin/pakar bisa menampilkan semua halaman yaitu halaman *login* admin, halaman admin, halaman data kerusakan, halaman gejala, halaman basis pengetahuan, halaman ganti *password* admin, halaman

repository.ub.ac.id

tambah/ubah data kerusakan, halaman tambah/ubah data gejala, dan halaman tambah/ubah data pengetahuan.

5.4.1 Implementasi Halaman Login

Antarmuka halaman *login*. pengguna memasukkan *username* dan *password* untuk yang telah terdaftar sebelumnya, kemudian sistem akan mencocokkan dengan *database*. Implementasi dapat dilihat pada gambar 5.2.



Enter your username and password

Username

Password

Sign in

[Login](#) [Sign up](#)

Gambar 5. 2 Antarmuka Halaman Login Admin

5.4.2 Implementasi halaman Sign Up

Antarmuka halaman sign up atau daftar untuk user yang baru yang ingin melakukan diagnosa tetapi belum memiliki akun di sistem dapat dilihat pada gambar 5.3.



Please Fill Details To Register

Nama User

Alamat

Tanggal Lahir (yyyy-mm-dd)

Pilih Jenis Kelamin

Username

Password

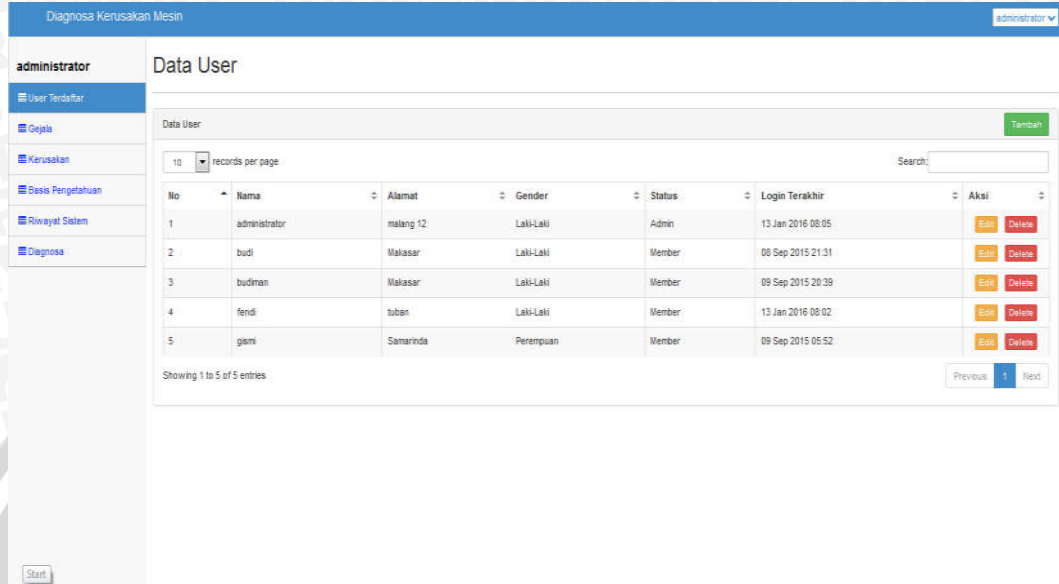
Register

[Login](#) [Sign up](#)

Gambar 5. 3 Implementasi halaman sign up

5.4.3 Implementasi Halaman Data User

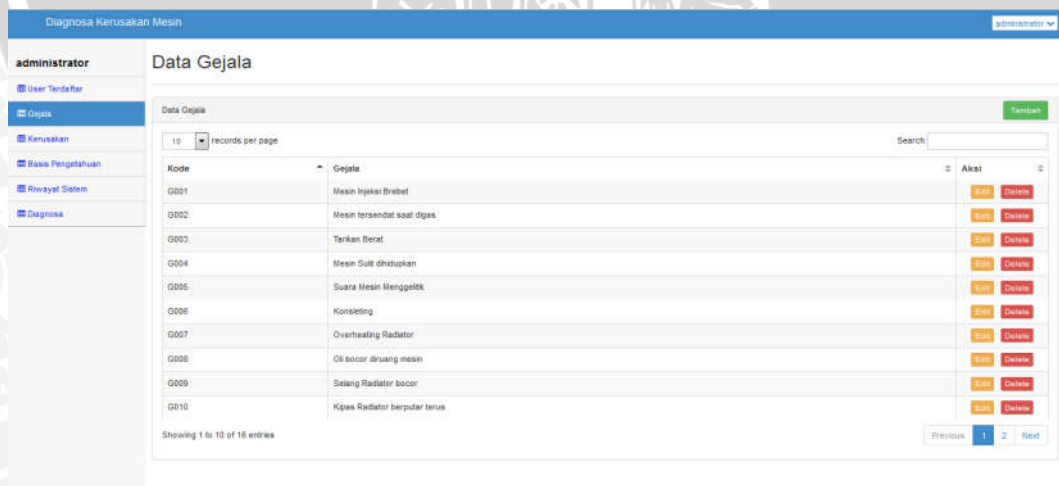
Antarmuka halaman data User, di halaman ini berisi tentang data user yang telah terdaftar pada database sistem. Implementasi ditunjukkan pada gambar 5.4 .



Gambar 5. 4 Antarmuka Halaman Data User

5.4.4 Implementasi Halaman Gejala Kerusakan

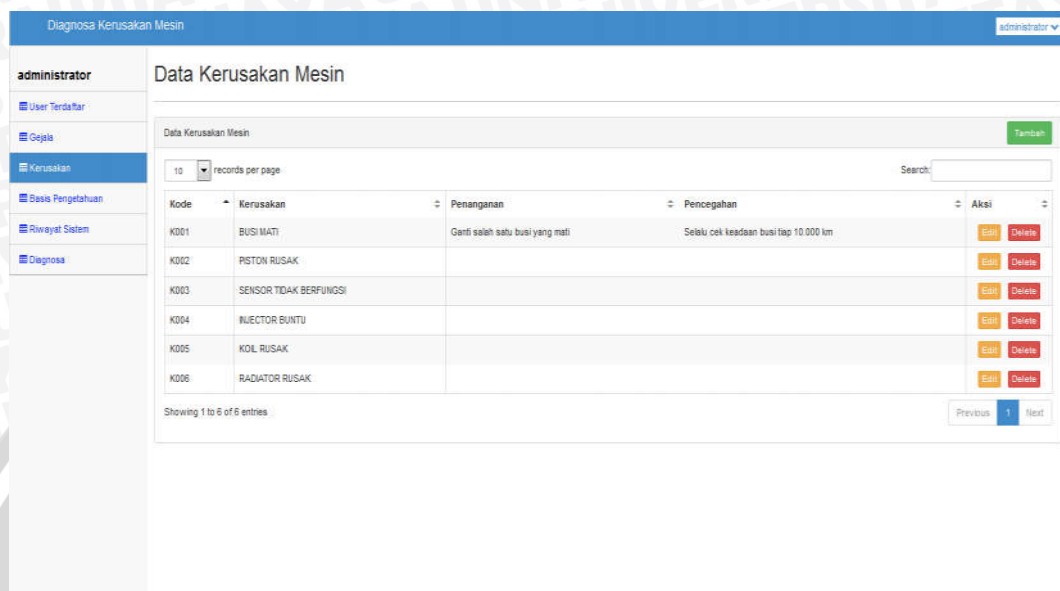
Antarmuka halaman data Gejala kerusakan, di halaman ini admin dapat menambah, mengedit dan menghapus data gejala. Implementasi ditunjukkan pada gambar 5.5 .



Gambar 5. 5 Antarmuka Halaman Data Gejala Kerusakan

5.4.5 Implementasi Halaman Data Kerusakan

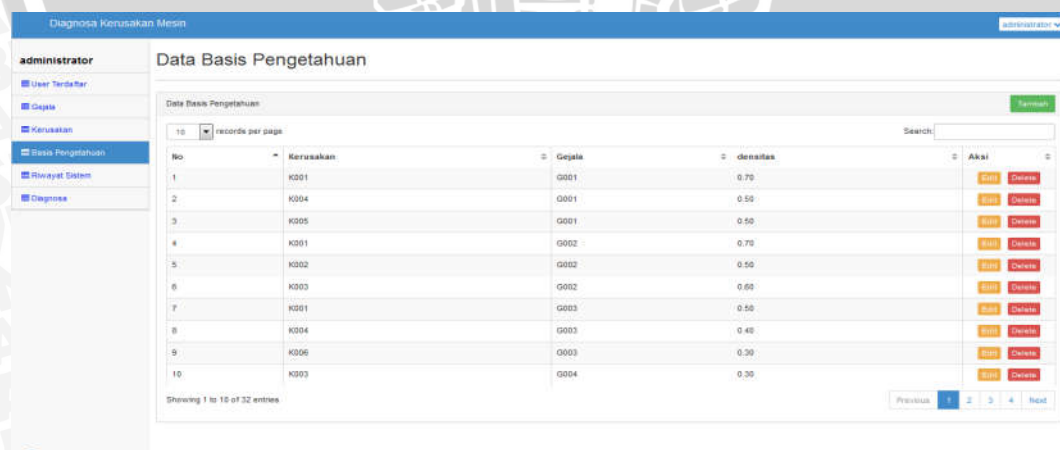
Antarmuka halaman data kerusakan, di halaman ini admin dapat menambah, mengedit dan menghapus data Kerusakan. Implementasi dapat ditunjukkan pada gambar 5.6 .



Gambar 5. 6 Antarmuka Data Kerusakan

5.4.6 Implementasi Halaman Basis Pengetahuan

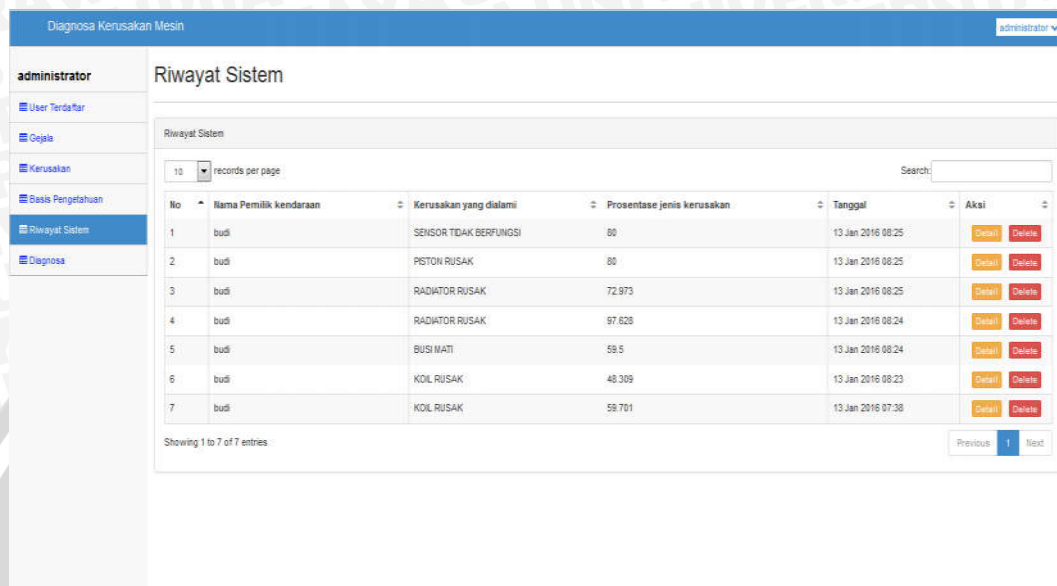
Halaman basis pengetahuan, dimana dalam halaman tersebut admin dapat melihat kerusakan, gejala, dan nilai densitas tiap gejala yang sudah dimasukkan sebelumnya. Dan admin dapat menambah, mengedit dan menghapus data. Implementasi ditunjukkan pada gambar 5.7 .



Gambar 5. 7 Antarmuka Halaman Basis Pengetahuan

5.4.7 Implementasi Halaman Riwayat Sistem

Halaman ini berisi informasi tentang hasil diagnosa yang berisi nama pengguna, kerusakan, presentase, tanggal serta admin dapat menghapus dan melihat detail laporan. Implementasi ditunjukkan pada gambar 5.8 .

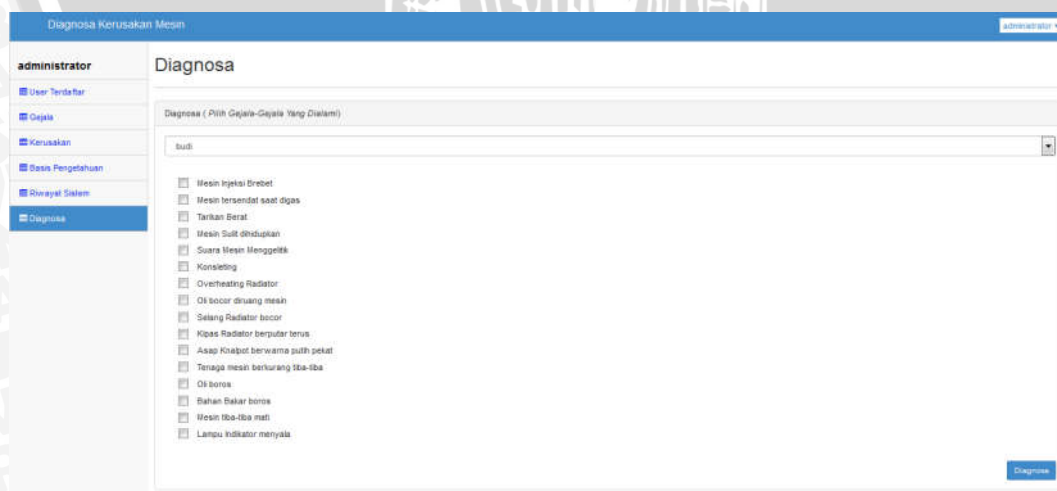


No	Nama Pemilik kendaraan	Kerusakan yang dialami	Presentase jenis kerusakan	Tanggal	Aksi
1	budi	SENSOR TIDAK BERFUNGSI	80	13 Jan 2016 08:25	Detail Delete
2	budi	PISTON RUSAK	80	13 Jan 2016 08:25	Detail Delete
3	budi	RADIATOR RUSAK	72.973	13 Jan 2016 08:25	Detail Delete
4	budi	RADIATOR RUSAK	97.628	13 Jan 2016 08:24	Detail Delete
5	budi	BUSI MATI	59.5	13 Jan 2016 08:24	Detail Delete
6	budi	KOIL RUSAK	48.309	13 Jan 2016 08:23	Detail Delete
7	budi	KOIL RUSAK	59.701	13 Jan 2016 07:38	Detail Delete

Gambar 5. 8 Antarmuka Halaman Riwayat Sistem

5.4.8 Implementasi Halaman Diagnosa

Di halaman tersebut dapat dilihat gejala-gejala yang dapat dipilih oleh admin atas dasar keluhan yang dialami oleh pengguna. Kemudian di gambar 5.6 adalah halaman hasil diagnosa, pengguna bisa melihat hasil dari diagnosa sesuai dengan gejala yang dimasukkan. Implementasi ditunjukkan pada gambar 5.9 .



Gambar 5. 9 Antarmuka Halaman Diagnosa

5.4.9 Implementasi Halaman Untuk User

Halaman untuk user, di halaman ini user hanya dapat mengakses halaman diagnosa untuk mendiagnosa awal dan halaman riwayat sistem saja dan tidak bisa mengakses ke halaman lain layaknya admin. Implementasi ditunjukkan pada gambar 5.10 .

The screenshot shows a web application titled "Diagnosa Kerusakan Mesin". At the top right, there is a user profile icon labeled "fendi". On the left side, there is a sidebar menu with two items: "Diagnosa" (highlighted with a blue bar) and "Riwayat Diagnosis". The main content area is titled "Diagnosa" and contains a sub-header "Diagnosa (Pilih Gejala-Gejala Yang Dialami)". Below this, there is a list of 15 engine symptoms, each preceded by an unchecked checkbox:

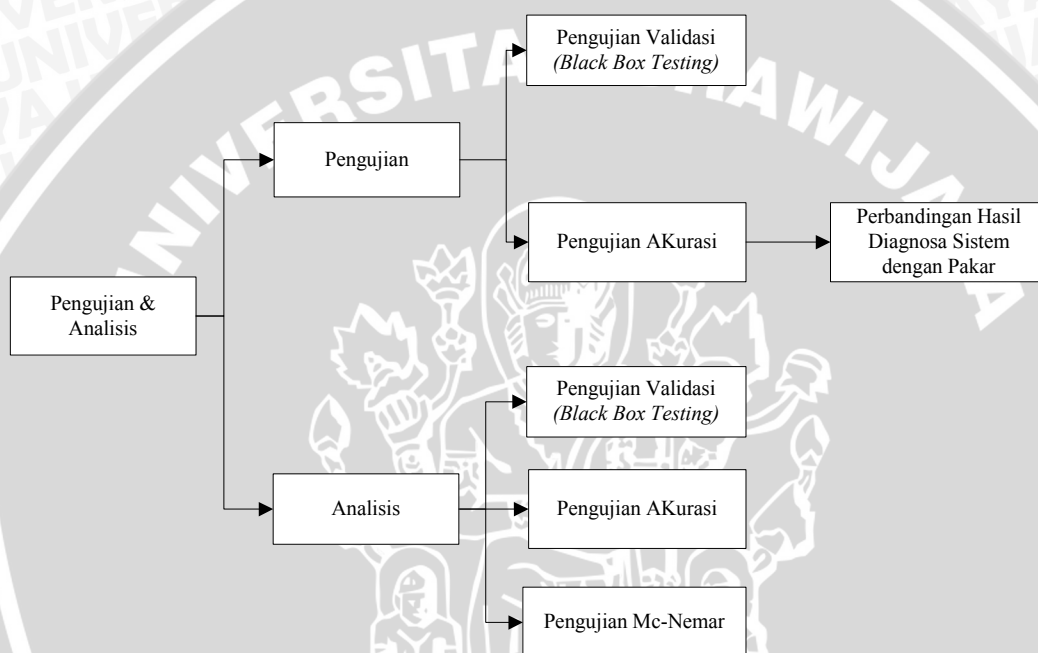
- ☐ Mesin injeksi Brebet
- ☐ Mesin tersendat saat diges
- ☐ Tarikan Berat
- ☐ Mesin Sulk dihidupkan
- ☐ Suara Mesin Menggelitik
- ☐ Konektling
- ☐ Overheating Radiator
- ☐ Oli bocor druang mesin
- ☐ Selang Radiator bocor
- ☐ Kipas Radiator berputar terus
- ☐ Asap Knalpot berwarna putih pekat
- ☐ Tenaga mesin berkurang tiba-tiba
- ☐ Oli boros
- ☐ Bahan Bakar boros
- ☐ Mesin tiba-tiba mati
- ☐ Lampu Indikator menyala

At the bottom right of the list, there is a blue button labeled "Diagnosa".

Gambar 5. 10 Antar muka halaman untuk user

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini membahas mengenai pengujian dan analisis sistem terhadap sistem pakar diagnosa Kerusakan Mesin kendaraan roda 4 menggunakan metode *forward chaining – dampster shafer*. Proses pengujian dilakukan melalui dua cara yaitu pengujian akurasi dan pengujian validasi. Pada pengujian validasi akan digunakan teknik pengujian *Black Box (Black Box Testing)*. Pengujian akurasi kasus yang telah diimplementasikan menjadi sistem pakar. Analisa hasil pengujian dilakukan untuk menganalisa hasil pengujian yang telah dilakukan. Pohon pengujian dan analisis dapat dilihat pada Gambar 6.1.



Gambar 6. 1 Pohon Pengujian dan Analisis

6.1 Pengujian Sistem

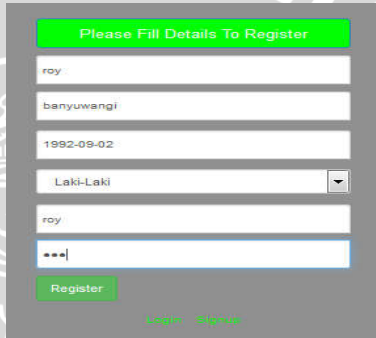
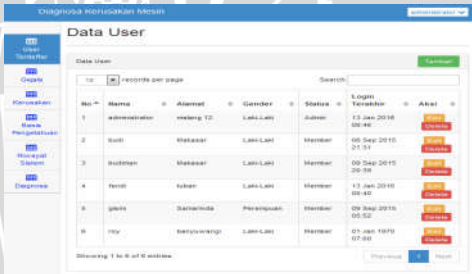
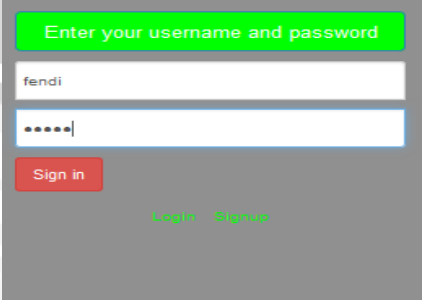
Proses pengujian dilakukan dengan pengujian validasi dan pengujian akurasi. Pengujian validasi dilakukan untuk mengetahui fungsionalitas sistem yang dibangun sesuai dengan daftar kebutuhan yang ada. Pengujian akurasi dilakukan untuk mengetahui seberapa besar akurasi yang didapat dari penggunaan metode *Forward Chaining – Dempster shafer* dalam menyelesaikan masalah sistem pakar diagnosa.

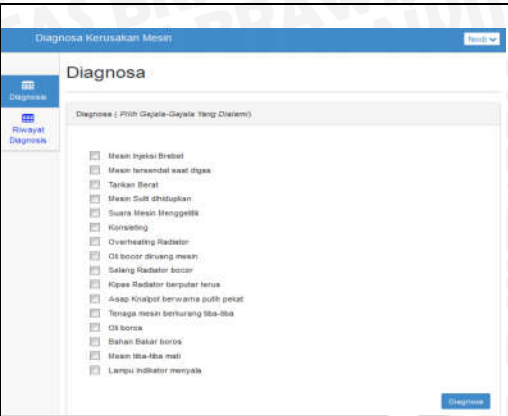
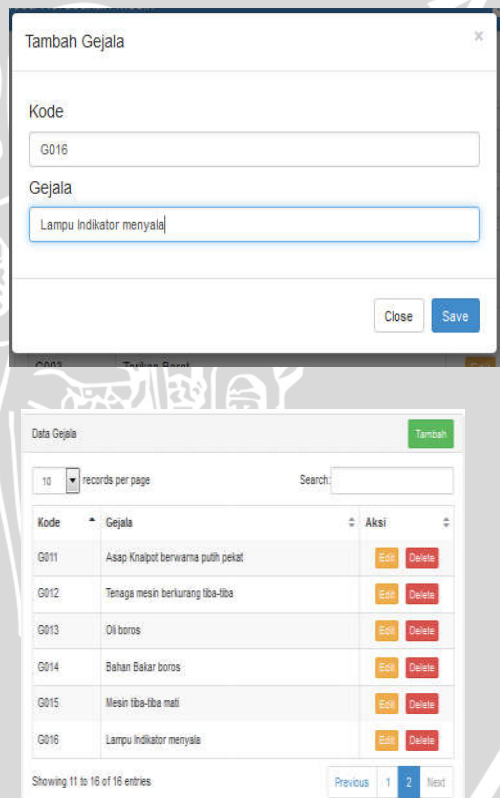
6.1.1 Pengujian Validasi

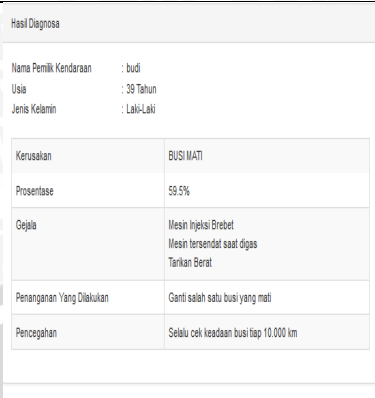
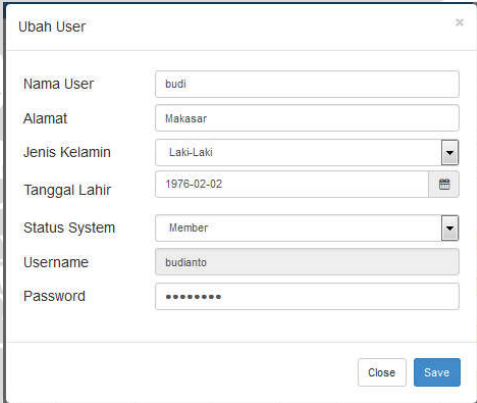
Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dalam daftar kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *black box*,

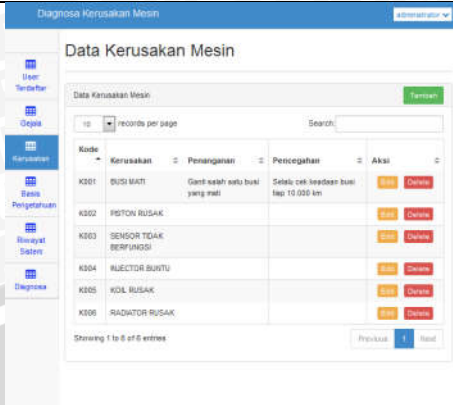
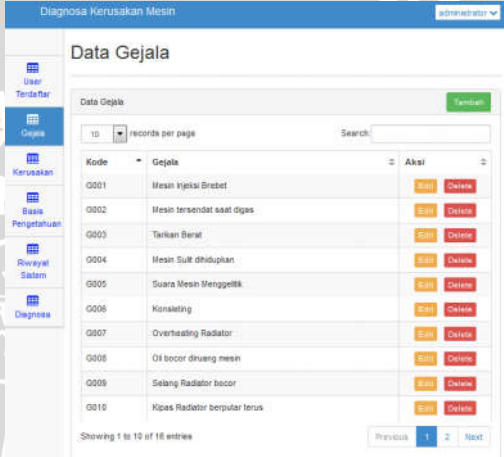
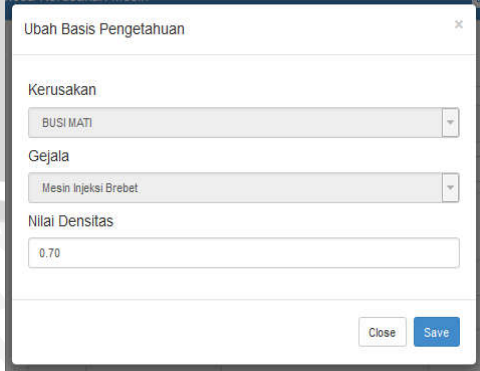
Karena tidak difokuskan terhadap alurnya jalan algoritma program namun lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan. Pada setiap kebutuhan dilakukan proses pengujian dengan kasus uji masing-masing untuk mengetahui kesesuaian antar kebutuhan dengan kinerja sistem. Pada tabel 6.1 ditunjukkan hasil pengujian validasi aplikasi pemodelan sistem pakar diagnose Kerusakan Mesin kendaraan roda 4.

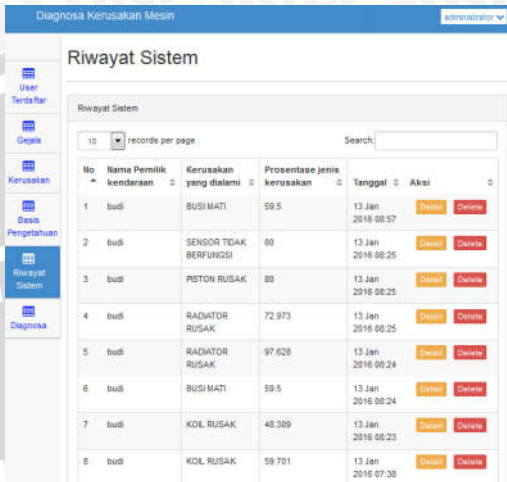
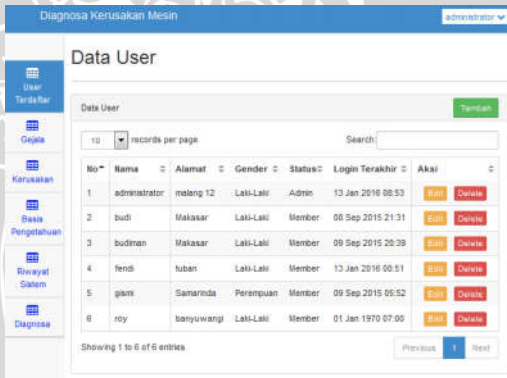
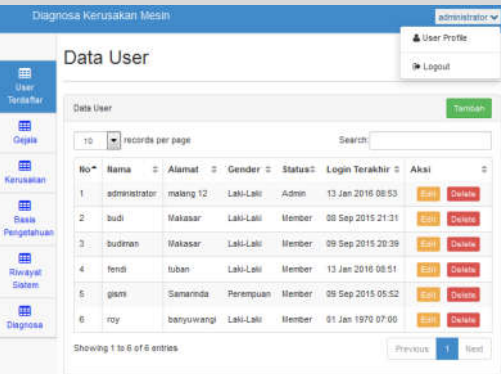
Tabel 6. 1 Hasil Pengujian Validasi Fungsional Sistem

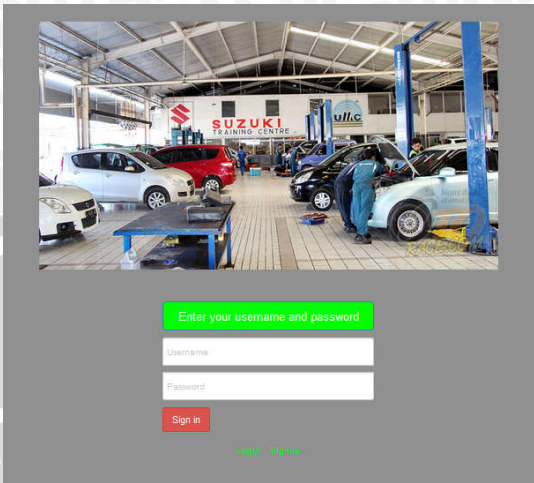
ID	Nama Kasus	Hasil yang diharapkan	Hasil yang didapatkan	Status Validasi
KF_01	Registrasi Pengguna Baru	Sistem mampu melakukan registrasi pengguna baru	Sistem dapat melakukan registrasi pengguna baru dan disimpan ke dalam <i>database</i>.  	Valid
KF_02	Login	Sistem mampu menerima inputan Login	Sistem dapat menerima inputan Login sehingga pengguna dapat mengakses menu lengkap sesuai tingkat levelnya. 	

				Valid
KF_03	Input Data Gejala	Sistem mampu menerima masukan data gejala	Sistem dapat menerima masukan dan menyimpan data gejala setiap kerusakan yang dilakukan oleh pakar. 	Valid
KF_04	Proses Diagnosa	Sistem mampu menyimpan hasil diagnosa	Sistem dapat menampilkan hasil proses diagnosa dan kesimpulan nama kerusakan masukan gejala dari pengguna.	

				Valid
KF_05	Biodata Pengguna	Sistem mampu melakukan perubahan pada data pengguna.	Sistem mampu menampilkan profil dari <i>user</i> yang melakukan <i>login</i> serta memperbarui profil saat pengguna melakukan <i>edit</i> profil. 	Valid
KF_06	Data Penyimpanan	Sistem mampu menambah, mengubah data Kerusakan	Sistem mampu menerima perubahan data Kerusakan kendaraan.	Valid

				
KF_07	Data Gejala	Sistem mampu melakukan perubahan data gejala	Sistem mampu menerima perubahan data terhadap gejala dari kerusakan mesin kendaraan. 	Valid
KF_08	Data Basis Pengetahuan	Sistem mampu menambah nilai densitas gejala	Sistem mampu menambah nilai data densitas gejala yang dilakukan oleh pakar 	Valid

KF_09	Riwayat Diagnosa	Sistem mampu menampilkan riwayat diagnosa	Sistem dapat menampilkan riwayat diagnosa yang dilakukan oleh pengguna. 	Valid
KF_10	Manajemen User	Sistem mampu mengolah data user.	Sistem dapat mengelola data user. 	Valid
KF_11	Logout	Sistem mampu logout	Sistem dapat melakukan proses <i>logout</i>. 	Valid

				
--	--	--	--	--

Berdasarkan pengujian validasi terhadap 11 tindakan dalam daftar kebutuhan dengan metode *Black Box Testing* menunjukkan bahwa sistem pakar diagnosa kerusakan mesin kendaraan roda 4 memiliki nilai validasi yang dapat dihitung

$$\begin{aligned}\text{Validasi} &= \frac{11}{11} \times 100\% \\ &= 100\%\end{aligned}$$

Dari 11 kasus uji yang telah dilakukan pengujian *Black Box* menunjukkan nilai valid sebesar 100% yang menandakan bahwa fungsionalitas sistem dapat berjalan dengan baik sesuai dengan daftar kebutuhan.

6.1.2 Pengujian Akurasi

Pengujian akurasi dilakukan untuk mengetahui performa keakurasian dari sistem pakar untuk memberikan hasil diagnosa kerusakan mesin kendaraan yang dialami oleh user. Data yang diuji berjumlah 25 sampel data kasus kerusakan mesin kendaraan yang didapat dari pakar (mekanik) sebagai dasar perbandingan pada pengujian. Pengujian yang dilakukan yaitu hasil yang diperoleh dari perhitungan sistem akan dibandingkan dengan hasil analisa dari pakar. Perhitungan aplikasi sistem dapat dilihat pada Sub Bab 4.3.4 pada kasus 2 yang menjelaskan analisa perhitungan secara manual dari salah satu sampel data dari pakar. Hasil pengujian akurasi sistem pakar dari 30 sampel yang telah diuji ditunjukkan pada tabel 6.2.

Tabel 6. 2 Tabel Pengujian Akurasi Hasil Diagnosa Sistem dengan Pakar

Mobil ke-	Gejala yang dialami	Hasil Identifikasi pakar	Hasil Identifikasi sistem	Akurasi
1	Lampu indikator menyala, tenaga mesin tiba-tiba berkurang	Sensor tidak berfungsi	Sensor tidak berfungsi	1
2	Suara mesin berdecit, overheating radiator	radiator rusak	Radiator Rusak	1
3	Mesin tersendat saat digas, tarikan berat	busi mati	busi mati	1
4	Mesin sulit dihidupkan, mesin tersendat saat digas	Sensor rusak, injector buntu	Sensor tidak berfungsi	1
5	Tenaga mesin berkurang tiba-tiba, bahan bakar boros, asap knalpot berwarna putih	piston rusak	Piston rusak	1
6	Mesin sulit dihidupkan, Overheating radiator	Radiator rusak	Radiator rusak	1
7	Tenaga mesin berkurang tiba-tiba, bahan bakar boros	koil rusak	koil rusak	1
8	Oli boros, bahan	Piston rusak	Piston Rusak	1

	bakar boros			
9	Mesin sulit dihidupkan, tenaga mesin berkurang tiba-tiba	Injektor buntu	Injektor buntu	1
10	Konsleting, lampu indikator nyala	Sensor tidak berfungsi	Sensor tidak berfungsi	1
11	Asap kenalpot warna putih, oli boros	Piston rusak	Piston rusak	1
12	Tarikan berat, overheating radiator	Radiator rusak	Radiator rusak	1
13	Overheating radiator, tenaga mesin berkurang	Radiator Rusak	Radiator rusak	1
14	Mesin injeksi brebet, mesin sulit dihidupkan	Koil rusak	Injektor buntu	0
15	Mesin tersendat di gas, bahan bakar boros, mesin sulit dihidupkan	Piston rusak	Piston rusak	1
16	Mesin sulit dihidupkan, mesin injeksi brebet	Injector mati/buntu	Busi mati	0
17	Mesin sulit dihidupkan, tenaga mesin berkurang tiba-tiba, bahan bakar boros	Injector Buntu	Injector buntu	1

18	Mesin tersendat di gas, mesin sulit dihidupkan, konsleting	Sensor tidak berfungsi	Sensor tidak berfungsi	1
19	Tenaga mesin berkurang, oli boros, bahan bakar boros	Piston rusak	Piston rusak	1
20	Tarikan berat, overheating radiator, lampu indicator menyala	Radiator rusak	Radiator rusak	1
21	Mesin injeksi brebet, mesin mobi berdecit, mesin sulit dihidupkan	Koil rusak	Koil rusak	1
22	Tenaga mesin berkurang, bahan bakar boros, mesin tiba-tiba mati	Busi mati	Busi mati	1
23	Mesin tersendat saat digas, oli bocor, konsleting	Piston rusak , sensor rusak	Sensor tidak berfungsi	1
24	Tarikan berat, oli boros, bahan bakar boros	Piston rusak	Piston rusak	1
25	Tenaga mesin berkurang, bahan bakar boros, mesin tiba-tiba mati	Busi Mati/ kotor	Busi mati	1
26	Lampu indikator nyala, mesin tiba-	Sensor tidak	Sensor tidak	1

	tiba mati	berfungsi	berfungsi / rusak	
27	Mesin mobil berdecit, konsleting	Koil rusak	Koil rusak	1
28	Overheating radiator, oli bocor diruang mesin, tenaga mesin berkurang tiba-tiba	Radiator rusak	Radiator rusak	1
29	Bahan bakar boros, asap knalpot berwarna putih	Piston rusak	Piston rusak	1
30	Asap knalpot berwarna putih, kipas radiator berputar terus, oli boros	Piston rusak, Radiator buntu	Piston rusak	1

Hasil akurasi bernilai 1 artinya diagnosa sistem sama dengan diagnosa pakar. Sebaliknya, hasil akurasi bernilai 0 artinya keluaran dari diagnosa sistem tidak sama dengan keluaran diagnosa pakar. Berdasarkan Tabel 6.2 telah dilakukan pengujian akurasi dengan 30 sampel data kerusakan mesin dan menghasilkan nilai akurasi sesuai perhitungan akurasi menggunakan persamaan 2.4.

$$\text{Nilai Akurasi} = \frac{28}{30} \times 100\% = 93\%$$

Dapat disimpulkan bahwa akurasi sistem pakar menggunakan metode *Forward Chaining - Dempster shafer* berdasarkan 30 data diagnosa gejala Kerusakan mesin yang telah diuji mempunyai tingkat keberhasilan yang cukup baik sesuai dengan diagnosa pakar yaitu sebesar 93%.

6.1.3 Pengujian pengaruh perubahan nilai Densitas *Dempster Shafer*

Tujuan penambahan dari perubahan nilai Densitas ini adalah untuk mengetahui pengaruh nilai densitas jika diubah-ubah dalam mendapatkan akurasi yang terbaik. Data pengujian pengaruh perubahan nilai densitas ini dilakukan dengan melakukan uji coba 2 kali terhadap sistem. Pengujian pertama dilakukan dengan cara menurunkan semua nilai densitas pada setiap gejala

sebanyak 0,1. Pengujian kedua dilakukan dengan menaikkan semua nilai densitas pada setiap gejala sebanyak 0,1. Hasil yang diperoleh dengan 2 kali pengujian (menaikkan dan menurunkan) menunjukkan hasil diagnosis kerusakan mesin kendaraan roda 4 tetap sama dengan hasil diagnosa pakar, namun untuk hasil nilai sedikit berbeda dikarenakan menyesuaikan nilai densitas yang baru.

Berikut ini merupakan hasil pengujian perubahan pada sistem setelah nilai densitas diturunkan 0,1 dan dinaikkan 0,1 dari 30 sampel yang ditunjukkan pada tabel 6.3

Tabel 6.3 Hasil pengujian perubahan nilai densitas

Mobil ke -	Gejala yang dialami	Hasil identifikasi pakar	Hasil identifikasi sistem (Nilai -0,1)	Hasil identifikasi sistem (Nilai +0,1)	Akurasi
1	Lampu indikator menyala, tenaga mesin tiba-tiba berkurang	Sensor tidak berfungsi	Sensor tidak berfungsi	Sensor tidak berfungsi	akurat
2	Suara mesin berdecit, overheating radiator	radiator rusak	Radiator Rusak	Radiator Rusak	akurat
3	Mesin tersendat saat digas, tarikan berat	busi mati	busi mati	busi mati	akurat
4	Mesin sulit dihidupkan, mesin tersendat saat digas	Sensor rusak, injector buntu	Sensor tidak berfungsi	Sensor tidak berfungsi	akurat
5	Tenaga mesin berkurang tiba-tiba, bahan bakar boros, asap knalpot berwarna putih	piston rusak	Piston rusak	Piston rusak	Akurat
6	Mesin sulit dihidupkan,	Radiator	Radiator	Radiator	akurat

	Overheating radiator	rusak	rusak	rusak	
7	Tenaga mesin berkurang tiba-tiba ,bahan bakar boros	koil rusak	koil rusak	koil rusak	Akurat
8	Oli boros, bahan bakar boros	Piston rusak	Piston Rusak	Piston Rusak	Akurat
9	Mesin sulit dihidupkan, tenaga mesin berkurang tiba-tiba	Injektor buntu	Injektor buntu	Injektor buntu	Akurat
10	Konsleting, lampu indikator nyala	Sensor tidak berfungsi	Sensor tidak berfungsi	Sensor tidak berfungsi	Akurat
11	Asap kenalpot warna putih, oli boros	Piston rusak	Piston rusak	Piston rusak	Akurat
12	Tarikan berat, overheating radiator	Radiator rusak	Radiator rusak	Radiator rusak	Akurat
13	Overheating radiator, tenaga mesin berkurang	Radiator Rusak	Radiator rusak	Radiator rusak	Akurat
14	Mesin injeksi brebet, mesin sulit dihidupkan	Koil rusak	Injektor buntu	Koil rusak	Akurat (+0,1) Tidak akurat (-0,1)
15	Mesin tersendat di gas, bahan bakar boros, mesin sulit dihidupkan	Piston rusak	Piston rusak	Piston rusak	Akurat

16	Mesin sulit dihidupkan, mesin injeksi brebet	Injector mati/buntu	Busi mati	Injector buntu	Akurat (+0,1) Tidak akurat (-0,1)
17	Mesin sulit dihidupkan, tenaga mesin berkurang tiba-tiba, bahan bakar boros	Injector Buntu	Injector buntu	Injector buntu	Akurat
18	Mesin tersendat di gas, mesin sulit dihidupkan, konsleting	Sensor tidak berfungsi	Busi mati	Injector buntu	Tidak akurat
19	Tenaga mesin berkurang, oli boros, bahan bakar boros	Piston rusak	Piston rusak	Piston rusak	Akurat
20	Tarikan berat, overheating radiator, lampu indicator menyala	Radiator rusak	Radiator rusak	Radiator rusak	Akurat
21	Mesin injeksi brebet, mesin mobi berdecit, mesin sulit dihidupkan	Koil rusak	Injector buntu	Injector buntu	Tidak akurat
22	Tenaga mesin berkurang, bahan bakar boros, mesin tiba-tiba mati	Busi mati	Busi mati	Busi mati	Akurat
23	Mesin tersendat saat digas, oli	Piston rusak , sensor rusak	Sensor tidak berfungsi	Sensor tidak	Akurat

	bocor, konsleting			berfungsi	
24	Tarikan berat, oli boros, bahan bakar boros	Piston rusak	Piston rusak	Piston rusak	akurat
25	Tenaga mesin berkurang, bahan bakar boros, mesin tiba-tiba mati	Busi Mati/ kotor	Piston rusak	Busi mati	Akurat (+01) Tidak akurat (-0,1)
26	Lampu indikator nyala, mesin tiba- tiba mati	Sensor tidak berfungsi	Sensor tidak berfungsi	Sensor tidak berfungsi	Akurat
27	Mesin mobil berdecit, konsleting	Koil rusak	Koil rusak	Koil rusak	Akurat
28	Overheating radiator, oli bocor diruang mesin, tenaga mesin berkurang tiba-tiba	Radiator rusak	Radiator rusak	Radiator rusak	Akurat
29	Bahan bakar boros, asap knalpot berwarna putih	Piston rusak	Piston rusak	Piston rusak	Akurat
30	Asap knalpot berwarna putih, kipas radiator berputar terus, oli boros	Piston rusak, Radiator buntu	Busi mati	Busi mati	Tidak akurat

Berdasarkan Tabel 6.3 telah dilakukan pengujian perubahan 30 sampel data gejala kerusakan mesin kendaraan roda 4 dengan merubah nilai densitas menghasilkan nilai akurasi yang berbeda.

- Nilai densitas diturunkan 0,1

$$\text{Nilai akurasi} = \frac{\text{Jumlah data akurat}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{27}{30} \times 100\% = 90\%$$

- Nilai densitas dinaikkan 0,1

$$\text{Nilai akurasi} = \frac{\text{Jumlah data akurat}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{24}{30} \times 100\% = 80\%$$

Dari analisa pengujian perubahan nilai densitas, keduanya mengalami perubahan akurasi saat nilai densitas diturunkan maupun dinaikkan 0,1. Akurasi saat nilai densitas diturunkan menjadi 80%, sedangkan saat nilai densitas dinaikkan menjadi 90%.

6.2 Analisa pengujian

Proses analisis yang dilakukan memiliki tujuan untuk mendapatkan kesimpulan dari hasil pengujian sistem pakar diagnosa kerusakan mesin kendaraan dengan metode *Forward Chaining - Dempster shafer*. Analisis dihasilkan dari setiap pengujian yang telah dilakukan baik pengujian validasi maupun pengujian akurasi.

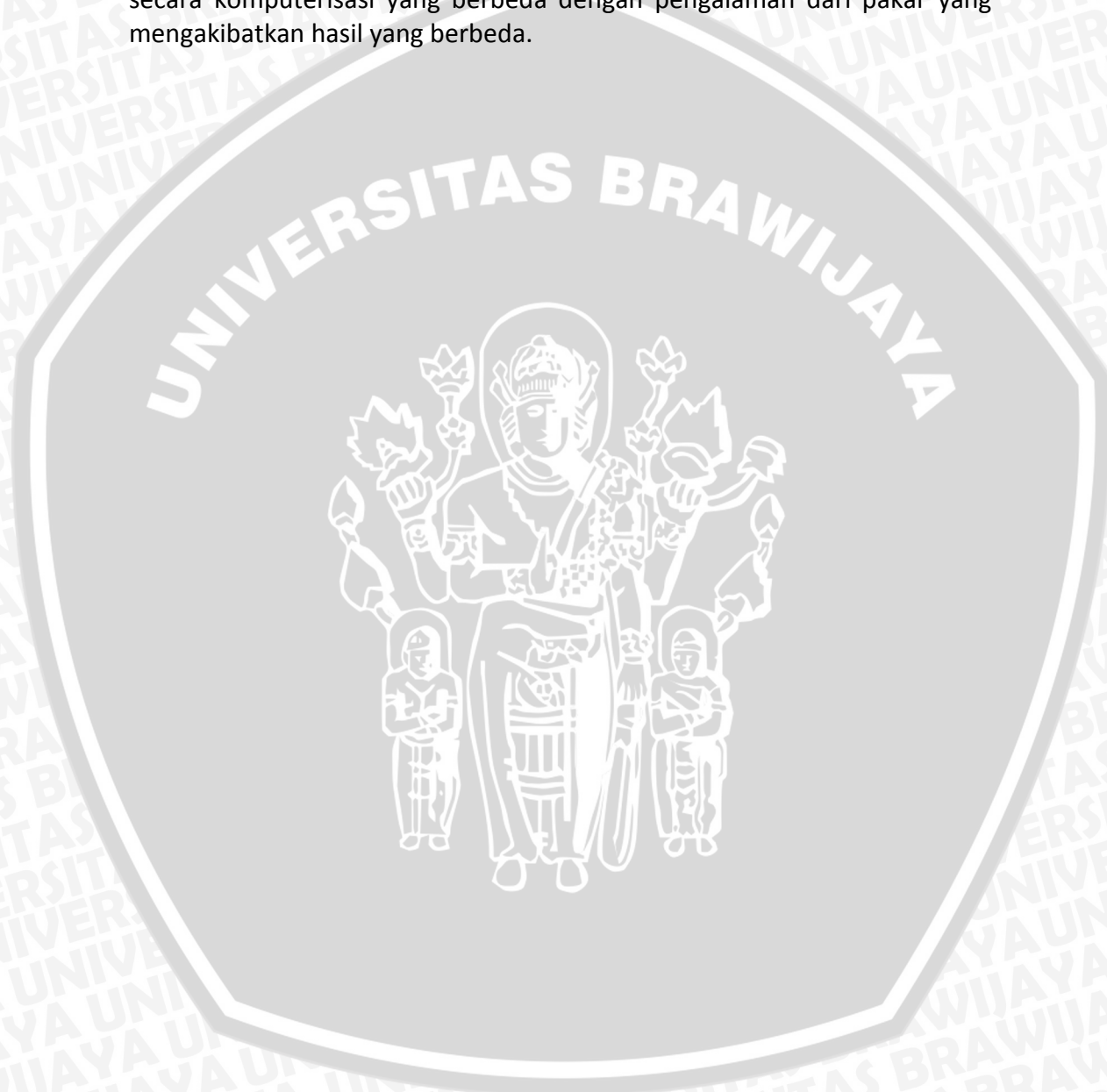
6.2.1 Analisis Validasi

Berdasarkan hasil perbandingan fungsionalitas sistem dengan daftar kebutuhan fungsional dihasilkan nilai validasi sebesar 100% sesuai Tabel 6.1. Nilai prosentase 100% diperoleh dari pembagian data yang valid sebanyak 11 dari 11 daftar kebutuhan fungsional. Sehingga fungsional sistem sudah memenuhi kebutuhan yang diperlukan pada sistem pakar diagnosa kerusakan mesin kendaraan dan sesuai dengan metode yang digunakan untuk mendiagnosa hasilnya.

6.2.2 Analisis Akurasi

Berdasarkan data observasi yang diberikan oleh pakar mengenai kasus-kasus kerusakan mesin yang pernah terjadi maka dihasilkan nilai akurasi sebesar 93% dari penggunaan perhitungan metode *Dempster-shafer* yang terdapat pada tabel 6.2. Nilai persentase 93% diperoleh dari pembagian data benar sebanyak 28 dari 30 data kasus uji. Hasil perbedaan antara diagnosa sistem dengan diagnosa pakar disebabkan beberapa hal, yaitu:

1. Ketidakakurasian sistem pakar ini sebesar 8% yang dapat disebabkan oleh beberapa kemungkinan, yaitu subyektifitas pakar dalam pemberian nilai kepercayaan gejala kerusakan dikarenakan keyakinan tiap pakar berbeda dan masukan data gejala fakta yang komplikasi dengan kerusakan yang belum terdapat pada sistem.
2. Ketidakakurasian sistem juga dapat dipengaruhi dari hasil perhitungan secara komputerisasi yang berbeda dengan pengalaman dari pakar yang mengakibatkan hasil yang berbeda.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil perancangan dan pengujian yang dilakukan, dapat diambil kesimpulan :

1. Metode *Forward Chaining* - *Dempster shafer* dapat diterapkan pada Pemodelan Sistem Pakar Untuk Mendiagnosa Kerusakan Mesin Kendaraan Roda 4. Sistem diimplementasikan menggunakan bahasa pemrograman PHP dan database MySQL. Sistem dapat mendiagnosa Kerusakan Mesin Kendaraan Roda 4 dengan memberikan informasi mengenai jenis Kerusakan Mesin, gejala-gejala dan solusi pencegahan sehingga dapat membantu masyarakat dalam mendeteksi dini kerusakan mesin yang terjadi pada kendaraan mereka. Kriteria yang digunakan 16 gejala berdasarkan hasil wawancara dengan pakar dengan 6 jenis Kerusakan Mesin.
2. Berdasarkan hasil pengujian didapatkan hasil sebagai berikut:
 - a. Pengujian validasi fungsionalitas sistem menunjukkan bahwa fungsionalitas sistem sebesar 100%. Nilai persentase 100% diperoleh dari pembagian data yang valid sebanyak 11 dari 11 daftar kebutuhan. Sehingga fungsional sistem sudah memenuhi kebutuhan yang diperlukan pada sistem ini.
 - b. Hasil pengujian akurasi dari 30 kasus uji menggunakan densitas gejala yang berasal dari pakar menghasilkan akurasi sebesar 93%. Dimana nilai 93% adalah nilai akurasi yang telah dihitung untuk menentukan tingkat keakuratan sistem

7.2 Saran

Sistem identifikasi kerusakan mobil dengan menggunakan metode *Dempster-Shafer* ini masih memiliki beberapa kekurangan. Saran yang dapat diberikan untuk pengembangan penelitian dimasa yang akan datang adalah:

1. Pada sistem ini masih memiliki banyak kekurangan salah satunya masih terdapat perbedaan hasil antara sistem dan pakar, nantinya diharapkan dapat dilakukan penelitian lebih lanjut dalam menentukan nilai densitas dari tiap gejala sehingga akurasi sistem menjadi lebih optimum.
2. Dikarenakan pada sistem yang sekarang masih belum mencakup semua jenis kerusakan dan gejala, diharapkan nantinya Sistem dapat dikembangkan dengan menambahkan gejala baru dan kerusakan mobil jika ditemukannya gejala dan kerusakan baru pada mobil.

DAFTAR PUSTAKA

- [DEW-14] Dewi Mustika. 2014. Aplikasi Sistem Pakar Untuk Pendeteksian Dan Penanganan Dini Pada Penyakit Sapi Dengan Metode Dempster-Shafer Berbasis Web. Malang: Universitas Brawijaya.
- [FAH-14] Fahraini Bacharuddin. 14. Pemodelan Dan Simulasi. Jakarta: Universitas Mercu Buana.
- [HEN-10] Hendra Wijaya, 2010. Definisi, Karakteristik dan Prinsip-Prinsip Pemodelan Sistem. Tersedia di: <<http://www.slideserve.com/molimo/analisis-dan-perancangan-perangkat-lunak>> [Diakses 17 Oktober 2015]
- [HID-10] Hidayati, Iswari Nur, 2010. Pemanfaatan Teori Bukti Dempster-Shafer Untuk Optimalisasi Penggunaan Lahan Berdasarkan Data Spasial dan Citra Multisumber. Yogyakarta: Universitas Gajah Mada.
- [ARH-04] Arhami, Muhammad. 2004. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi. Yogyakarta.
- [KUS-03] Sri Kusumadewi, 2003. Artificial Intelligence (Teknik dan Aplikasinya). Yogyakarta: Graha Ilmu.
- [PRI-11] Prihatini, PutuManik, 2011. Metode Ketidakpastian Dan Kesamaran Dalam Sistem Pakar. Bali: Politeknik Negeri Bali.
- [SUL-08] Sulistyohati, Aprilia, 2008. Aplikasi Sistem Pakar Diagnosa Penyakit Ginjal dengan Metode Dempster-Shafer. Yogyakarta: Universitas Islam Indonesia.
- [GIN-09] Ginintasi Rahayu, 2009. Proses Pembelajaran Anak Berkebutuhan Khusus. Bandung: Universitas Pendidikan Indonesia.
- [GAO-13] Gaol, S, D, Lumban. 2013. *Sistem Pakar Mendeteksi Gizi Buruk Pada Balita Berbasis Web Dengan Menggunakan Metode Certainty Factor*. Pelita Informatika Budi Darma, Volume : V, Nomor: 1, November 2013.
- [KUR-14] Kurniawati, D, Pratama. 2014. *Implementasi Metode Dempster Shafer Pada Sistem Pakar Untuk Diagnosa Jenis-jenis Penyakit Diabetes Melitus*. Jurusan Teknik Informatika. Fakultas Ilmu Komputer Universitas Dian Nuswantoro, Semarang.
- [LIS-08] Listiyono, H. 2008. *Merancang dan Membuat Sistem Pakar*. Jurnal Teknologi Informasi DINAMIK, Vol. XIII, No. 2, juli 2008, pp: 115-124.

- [PAT-09] Pathirana, S., Kawabata, Masato dan Goonatilake, Rohitha. 2009. *Study Of Potential Risk Of Dengue Disease Outbreak in Sri Lanka Using GIS and Statistical Modelling*. Jrural Trop Public Health 2009, VOL 8, pp: 8-17.
- [PRA-07] Pradana, C. P. dan S. Kusumadewi. 2007. *Aplikasi Diagnosis Penyakit Hepatitis untuk Mobile Devices Menggunakan J2ME*. Media Informatika, VI. 5, No. 2, Desember 2007, hal: 87-98.
- [PUT-11] Putri, P, Amanda dan Mustafidah, H. 2011. *Sistem Pakar untuk Mendiagnosa Penyakit Hati Menggunakan Metode Forward Chaining*. JUITA ISSN: 2086-9398 Vol. I Nomor 4, Nopember 2011.
- [RAM-14] Ramdhan, W. dan Y. Siagian. 2014. *Sistem Pakar Tes Kepribadian (Papikostik) dengan Metode Forward Chaining*. Jurnal Teknologi dan Sistem Informasi, Vol. 1, No. 1, Desember 2015, Hal: 20-25.
- [WIC-12] Wicaksono, P. 2012. *Rancang Bangun Expert System Diagnosa Penyakit Anak Menggunakan Metode Forward Chaining dan Backward Chaining*. Universitas Siliwangi. Bandung.

