

DIAGNOSA KERUSAKAN MESIN TIPE OM366A PADA MOBIL MERCEDES-BENZ MENGUNAKAN SISTEM PAKAR

Rizal*

Abstract

An expert system is a transfer of knowledge from an expert to a computer system, so the system can be used by people who are not experts in solving the problem as was done by experts. Many cases can be used as research in expert systems, one of which diagnoses the cause engine damage. This research will design an expert system that can diagnose the cause of damage to the engine type OM366A Mercedes-Benz. The method used in this study is the use of backward chaining method in the process of diagnosing it. This expert system is very useful to make it easier for technicians to perform diagnostics, so that will speed in getting a solution to the cause engine damage and repair actions in accordance with the selected type of damage and without the presence of an expert. This expert system application designed using Microsoft Visual Basic.Net 2008, Add-On Developer Express 2011 and Microsoft Access 2010 as a repository for the knowledge base.

Keywords : *expert systems, backward chaining, engine troubleshooting.*

Pendahuluan

Sistem pakar adalah pengalihan suatu pengetahuan dari seorang pakar ke sistem komputer, sehingga sistem tersebut dapat dipergunakan oleh orang yang tidak pakar dalam menyelesaikan masalah seperti yang biasa dilakukan oleh para pakar. Sistem pakar merupakan salah satu bidang teknik kecerdasan buatan yang cukup banyak diminati karena penerapannya di berbagai bidang, baik bidang ilmu pengetahuan maupun

^{1*} Dosen Teknik Informatika Universitas Malikussaleh

bisnis yang terbukti sangat membantu dalam mengambil keputusan dan sangat luas penerapannya.

Mobil Mercedes-Benz adalah merupakan kendaraan atau kereta tanpa kuda yang dapat berjalan sendiri yang diciptakan pada tahun 1886 oleh seorang berkebangsaan Jerman yang bernama Karl Benz. Sedangkan OM366A adalah tipe mesin bertenaga diesel yang dilengkapi dengan *Turbo Charger* .

Mobil Mercedes-Benz dengan mesin tipe OM366A, saat ini banyak dipergunakan sebagai alat transportasi umum seperti, bus penumpang antar lintas propinsi, truk pengangkut barang serta bus karyawan pada beberapa perusahaan vital yang ada di Aceh, terutama pada perusahaan migas PT. Arun NGL Lhokseumawe dimana tempat penulis melakukan penelitian akhir.

Pada suatu saat mesin tersebut dapat saja mengalami gangguan / kerusakan dan dalam keadaan rusak mobil tidak dapat dipergunakan untuk keperluan operasional, sehingga sangat dibutuhkan penanganan khusus dalam hal pendiagnosaan untuk menentukan penyebab kerusakan mesin tersebut, sehingga nantinya akan didapatkan suatu solusi terhadap penyebab kerusakan dan tindakan perbaikan yang harus dilakukan agar mesin tersebut kembali normal dan dapat dioperasikan kembali.

Namun dengan keterbatasan kemampuan dan pengetahuan para teknisi dalam usaha untuk menyelesaikan setiap permasalahan yang terjadi pada mesin OM366A dan pengambilan tindakan lanjutan untuk perbaikan, masih selalu membutuhkan hadirnya seorang pakar atau seorang yang ahli dibidang mesin tersebut. Sehingga hal ini sangat berpengaruh terhadap kinerja operasional perusahaan, terutama pada bidang pelayanan perbaikan mobil perusahaan.

Dengan latar belakang permasalahan tersebut, penulis yang juga salah satu siswa pada Mercedes-Benz *Training Section* dan *senior* teknisi pada perusahaan PT. Arun NGL, berusaha untuk merancang sebuah aplikasi sistem pakar berbasis komputer yang akan dapat membantu para teknisi dalam mendapatkan solusi untuk menentukan dimana penyebab kerusakan yang terjadi dan tindakan perbaikan apa yang harus dilakukan saat terjadinya gangguan atau kerusakan pada mesin OM366A tentunya dengan

mudah, cepat, tepat dan tanpa perlu hadirnya seorang pakar dalam setiap pemecahan masalah.

Berdasarkan latar belakang yang telah diuraikan diatas, maka dapat diketahui rumusan masalah. Rumusan-rumusan masalah tersebut adalah : Bagaimana menyelesaikan persoalan-persoalan yang timbul pada mesin tipe OM366A dan solusi-solusi yang biasa dilakukan terhadap persoalan tersebut. Bagaimana membuat mesin Inferensi yang akan digunakan untuk membangun sistem pakar terhadap permasalahan yang terjadi pada mesin tipe OM366A. Bagaimana membangun sistem pakar dalam penyelesaian persoalan yang timbul pada mesin tipe OM366A dengan menggunakan metode *Backward Chaining*

Agar pembahasan masalah tetap berada dalam batasan yang diinginkan, maka dengan ini dilakukan pembatasan masalah dengan ruang lingkup antara lain : Sistem pakar ini berbasis personal komputer, artinya tidak bekerja secara *Client-Server*. Hanya mendiagnosa penyebab kerusakan atau gangguan mesin tipe OM366A pada mobil Mercedes-Benz. Pendiagnosaan penyebab kerusakan hanya terbatas pada beberapa sistem bagian kerusakan antara lain, sistem udara, sistem gas buang, sistem bahan bakar, sistem pendingin, sistem pelumasan, sistem mekanik dan sistem kelistrikan. Metode yang digunakan disusun berdasarkan *Backward Chaining*.

KECERDASAN BUATAN

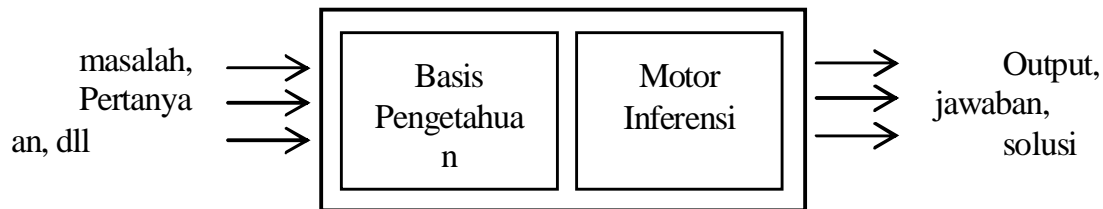
Kecerdasan buatan adalah suatu ilmu yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan oleh manusia. Definisi lain, kecerdasan buatan (*Artificial Intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

Cerdas = memiliki pengetahuan + pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan).

Agar mesin bisa cerdas (bertindak seperti dan sebaik manusia) maka harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar.

Dua bagian utama yg dibutuhkan untuk aplikasi kecerdasan buatan :

- a. Basis Pengetahuan (*knowledge base*) :
berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- b. Motor Inferensi (*inference engine*) :
kemampuan menarik kesimpulan berdasarkan pengetahuan



Gambar 1 Bagian Utama Aplikasi Kecerdasan Buatan

Beda kecerdasan buatan dan kecerdasan alami :

Kelebihan kecerdasan buatan :

1. Lebih bersifat permanen. Kecerdasan alami bisa berubah karena sifat manusia pelupa. Kecerdasan buatan tidak berubah selama sistem komputer dan program tidak mengubahnya.
2. Lebih mudah diduplikasi dan disebar. Mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama dan keahlian tidak akan pernah dapat diduplikasi dengan lengkap. Jadi jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.
3. Lebih murah. Menyediakan layanan komputer akan lebih mudah dan murah dibandingkan mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Bersifat konsisten dan teliti karena kecerdasan buatan adalah bagian dari teknologi komputer sedangkan kecerdasan alami senantiasa berubah-ubah
5. Dapat didokumentasi. Keputusan yang dibuat komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas

dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.

6. Dapat mengerjakan beberapa task lebih cepat dan lebih baik dibanding manusia.

Kelebihan kecerdasan alami :

1. Kreatif : manusia memiliki kemampuan untuk menambah pengetahuan, sedangkan pada kecerdasan buatan untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.
2. Memungkinkan orang untuk menggunakan pengalaman atau pembelajaran secara langsung. Sedangkan pada kecerdasan buatan harus mendapat masukan berupa input-input simbolik.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

SISTEM PAKAR

Sistem pakar adalah pengalihan suatu pengetahuan dari seorang pakar ke sistem komputer, sehingga sistem tersebut dapat dipergunakan oleh orang yang tidak pakar dalam menyelesaikan masalah seperti yang biasa dilakukan oleh para pakar. Sistem pakar yang baik dirancang untuk dapat menyelesaikan suatu permasalahan tertentu dengan meniru prinsip kerja dari para pakar. Sistem ini membantu orang awam dalam menyelesaikan masalah yang cukup rumit yang hanya dapat diselesaikan dengan bantuan para pakar.

Bagi para pakar sendiri, sistem ini akan membantu aktivitasnya sebagai seorang asisten yang sangat berpengalaman. Sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu. Beberapa pendapat mengenai definisi sistem pakar, antara lain :

1. Jackson (2002)

Sistem pakar adalah sebuah program komputer yang merepresentasikan dan mempertimbangkan dengan pengetahuan

dari beberapa subyek spesial dengan sebuah pandangan untuk menyelesaikan masalah-masalah atau memberikan nasehat.

2. Giarratano dan Riley (2005)

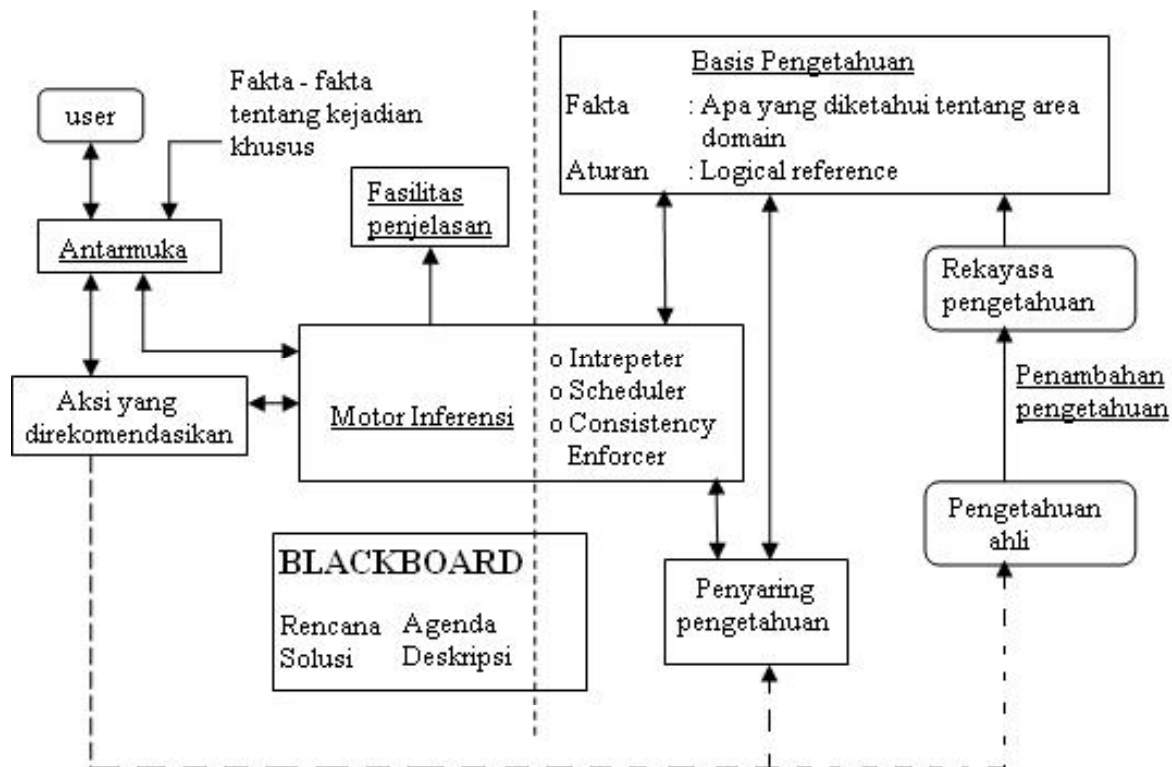
Sistem pakar adalah suatu sistem komputer yang mengemulasi kemampuan pembuatan keputusan dari seorang pakar.

3. Rich dan Knight (1991)

Sistem pakar adalah suatu sistem yang memiliki tujuan untuk menyelesaikan masalah yang biasa dilakukan oleh seorang pakar.

STRUKTUR SISTEM PAKAR

Menurut Turban dan Frenzel (1992), sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan pakar untuk memperoleh pengetahuan pakar.



Gambar 2 Struktur Sistem Pakar (Sri Kusumadewi, 2003)

Komponen – komponen (Turban dan Frenzel 1992) yang ada pada sistem pakar adalah :

1. Subsistem akuisisi pengetahuan

Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Pengetahuan itu bisa berasal dari pakar, buku, basis data, penelitian, dan gambar.

2. Basis pengetahuan

Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.

3. Mesin inferensi (*inference engine*)

Program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan dan *blackboard*, serta digunakan untuk memformulasikan konklusi.

4. Antarmuka (*User Interface*)

Digunakan untuk media komunikasi antara pengguna dan program.

5. *Blackboard*

Merupakan area dalam memori yang digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara.

Ada tiga tipe keputusan yang dapat direkam, yaitu :

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

6. Subsistem penjelasan

Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan:

- Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
- Bagaimana konklusi dicapai?
- Mengapa ada alternatif yang dibatalkan?
- Rencana apa yang digunakan untuk mendapatkan solusi?

7. Sistem penyaring pengetahuan

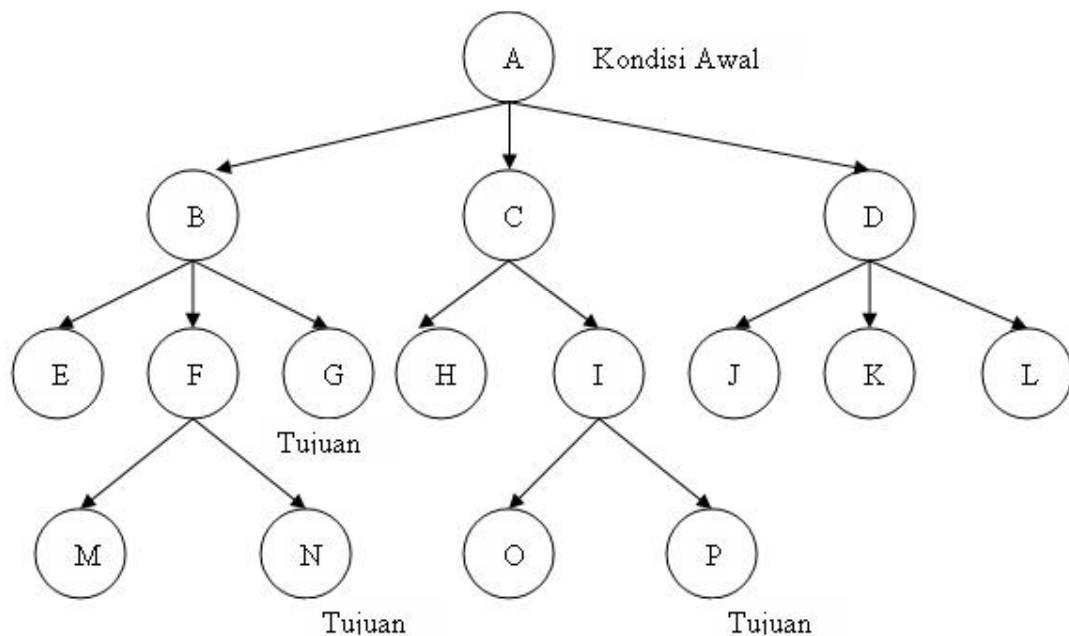
Sistem ini digunakan untuk mengevaluasi kinerja sistem pakar itu sendiri untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa mendatang.

METODE INFERENSI

Menurut Giarratano dan Riley (1998) metode inferensi untuk memecahkan suatu persoalan dalam sistem pakar dapat dilakukan dengan merangkai rantai produksi (*Chaining*). Jenis rantai produksi (*Chaining*) tersebut adalah :

1. *Forward Chaining*

Pemecahan masalah dari fakta - fakta kepada sebuah kesimpulan berdasarkan fakta - fakta yang ada. Dengan kata lain, pelacakan dimulai dari keadaan awal (informasi atau fakta yang ada) dan kemudian dicoba untuk mencocokkan dengan tujuan yang diharapkan.

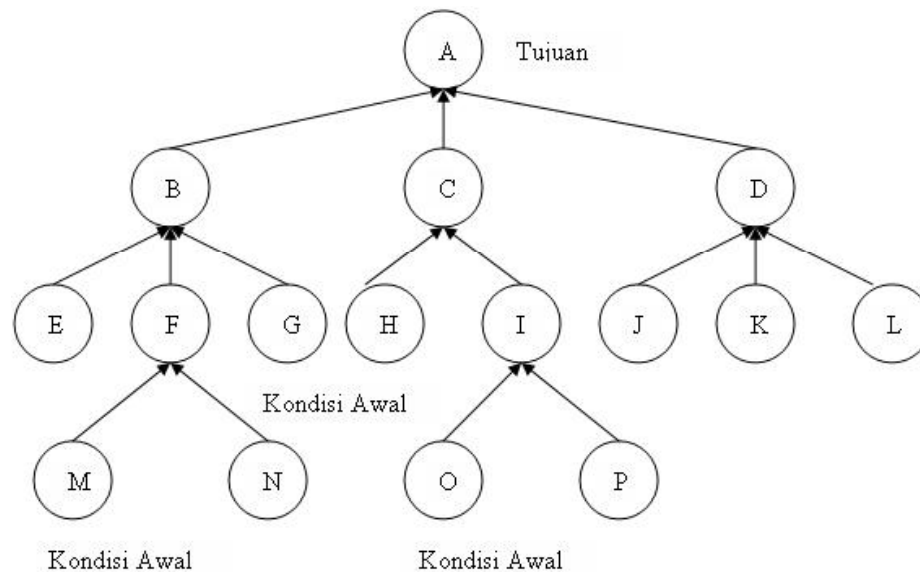


Gambar 3 *Forward Chaining*

2. *Backward Chaining*

Diagram dimana menghubungkan hipotesa berdasarkan fakta-fakta yang mendukung sebuah hipotesa. Dengan kata lain dapat dikatakan bahwa penalaran dimulai dari tujuan atau hipotesa,

kemudian dicocokkan dengan keadaan awal atau fakta-fakta yang ada.



Gambar 4 Backward Chaining

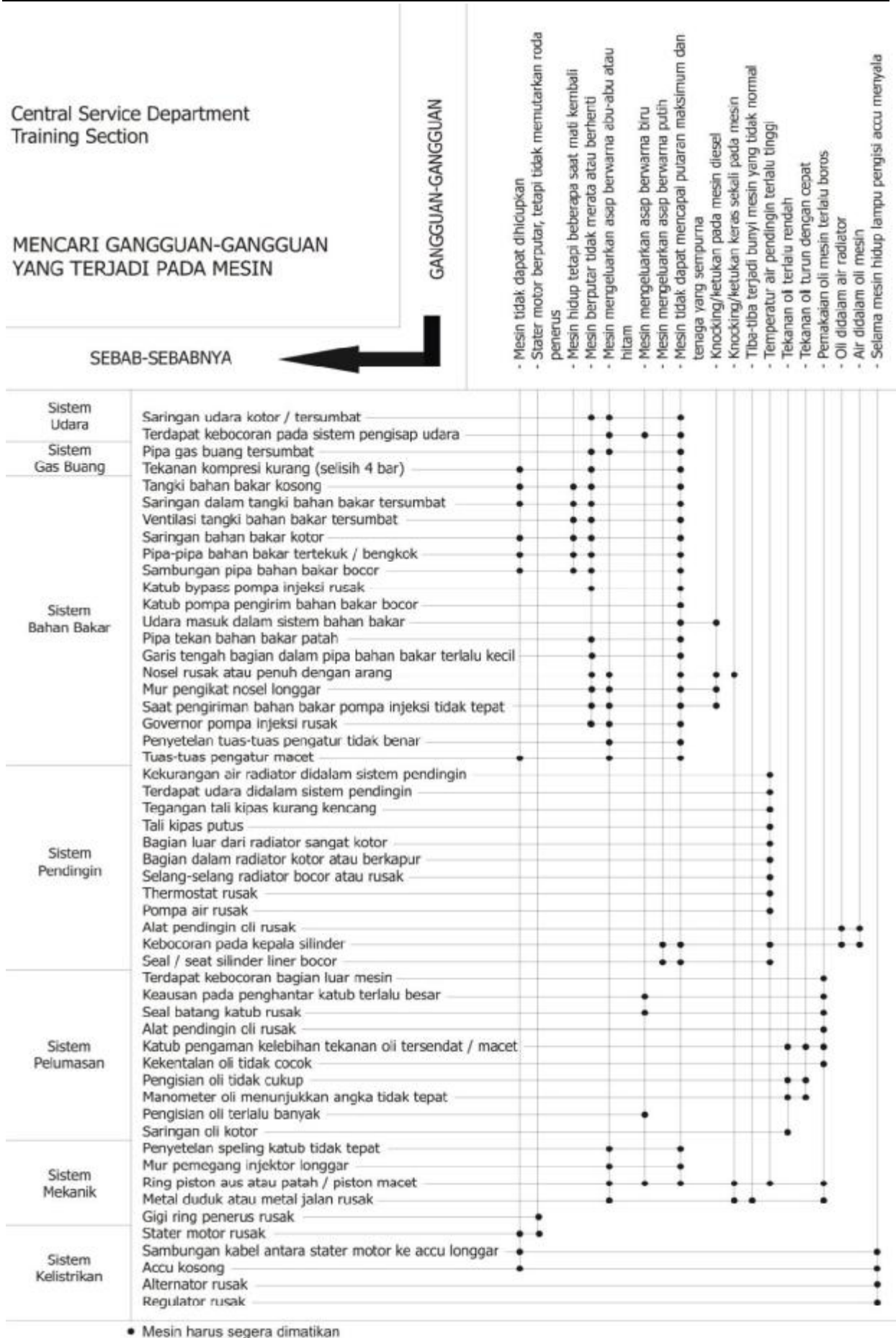
Ada beberapa faktor yang mempengaruhi pemilihan *backward* atau *forward* dalam memilih metode penalaran, antara lain:

1. Banyaknya keadaan awal dan tujuan. Jika jumlah keadaan awal lebih kecil daripada tujuan, maka digunakan penalaran *forward*. Sebaliknya, jika jumlah keadaan awal lebih banyak daripada tujuan, maka digunakan metode penalaran *backward*.
2. Rata-rata jumlah simpul (*node*) yang dapat diraih secara langsung dari suatu simpul. Lebih baik dipilih yang jumlah simpul tiap cabangnya lebih sedikit.
3. Apakah program butuh menanyai pengguna (*user*) untuk melakukan justifikasi terhadap proses penalaran? Jika ya, maka langkah baiknya jika dipilih arah yang lebih memudahkan pengguna.
4. Bentuk kejadian yang akan memicu penyelesaian masalah. Jika kejadian itu berupa fakta baru, maka lebih baik dipilih penalaran

forward . namun jika kejadian itu berupa *query*, maka lebih lebih baik dipilih penalaran *backward*.

DATA YANG DIGUNAKAN

Sumber data utama yang digunakan pada penelitian ini adalah berupa lembar data *Troubleshooting* yang berjudul “Mencari gangguan-gangguan yang terjadi pada mesin”, yang dikeluarkan oleh *Central Service Department Training Section Mercedes-Benz PT. Star Motor Indonesia (Dealer Mercedes-Benz for Indonesia)*. Data tersebut diperoleh saat penulis mengikuti pelatihan teknisi di *Mercedes-Benz Training Section* selama satu tahun. Lembar data tersebut dapat dilihat pada gambar dibawah ini.

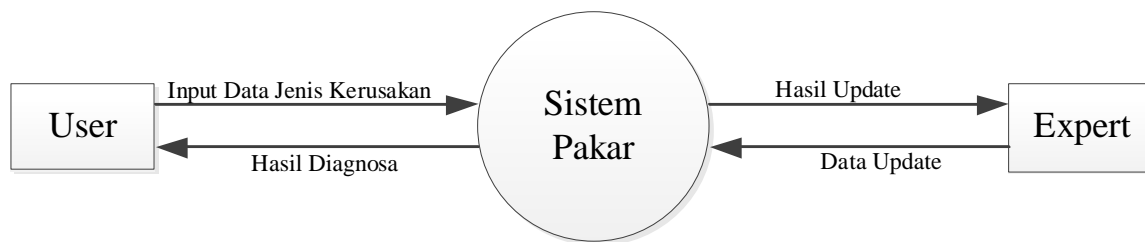


Gambar 5. Data perbaikan dari mercedez Benz

PERANCANGAN SISTEM

Diagram Konteks

Diagram Konteks dalam aplikasi ini bertujuan untuk menjelaskan aliran informasi dan transformasi data baik berupa pemasukan data oleh *knowledge engineer* maupun keluaran data yang dapat dilihat oleh *user*. Perancangan sistem ini dimulai dari diagram konteks DFD level 0, hingga DFD level 1.



Gambar 6 Diagram Konteks DFD Level 0

Pada gambar 4.1 menjelaskan tentang gambaran sistem pakar yang dirancang. *User* akan memasukan informasi berupa data-data jenis kerusakan yang terjadi pada mesin tipe OM366A. Berdasarkan informasi yang masuk dari *user*, sistem pakar akan memproses informasi tersebut dan akan mengeluarkan solusi pada *user* berupa penyebab kerusakan dan tindakan perbaikan. Sedangkan pakar berfungsi untuk meng-*update knowledge base* sistem pakar tersebut. Entitas-entitas yang berhubungan pada sistem pakar berupa masukan dan keluaran yang dapat digambarkan pada tabel berikut ini :

Tabel 1. Daftar *External Entity, Input dan Output*

Kesatuan Luar (<i>External Entity</i>)	Masukan (<i>Input</i>)	Keluaran (<i>Output</i>)
<i>User</i>	Data-data jenis kerusakan dari tiap-tiap <i>subsegment</i> yang dipilih pada sistem	Hasil Diagnosa (Solusi)
<i>Expert</i>	<i>Update</i>	Hasil <i>Update</i>

Tabel 1 menjelaskan suatu proses pengolahan sistem pakar yang berjalan, dan berikut adalah cara kerja dari tiap entitas :

1. *User* (Pengguna)

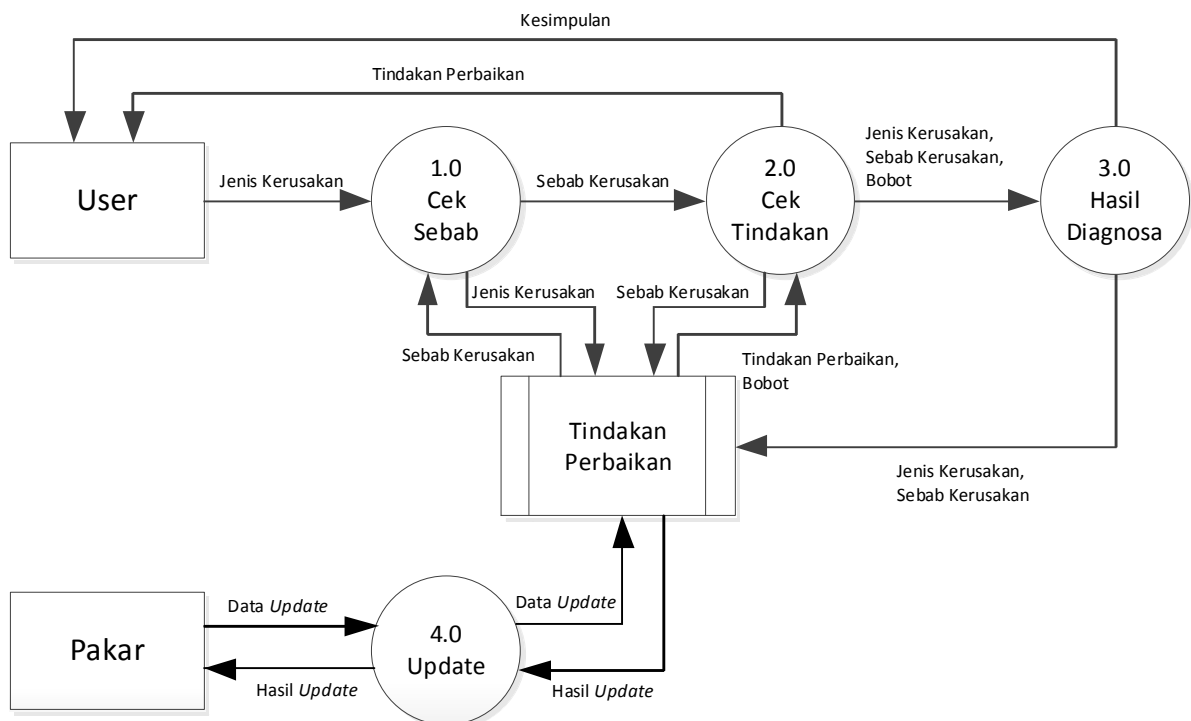
User mempunyai fungsi sebagai entitas pertama yang memberikan masukan mengenai macam-macam data jenis kerusakan dari tiap-tiap *sub segments* yang dipilih, ke dalam sistem pakar.

2. *Expert* (Pakar)

Entitas ini berfungsi meng-*update knowledge base* yang ada dalam sistem, hal ini dilakukan jika pengetahuan sang pakar bertambah, sehingga diperlukan penambahan dasar kepakaran juga pada sistem pakar tersebut.

DATA FLOW DIAGRAM (DFD)

Proses yang ada pada diagram konteks dapat dipecah lagi menjadi proses-proses yang lebih kecil dan lengkap dalam DFD level 1. Diagram untuk DFD level 1 dapat dilihat pada Gambar 7



Gambar 7. Diagram Rinci DFD Level 1

Penjelasan dari Diagram Aliran Data DFD Level 1 adalah sebagai berikut:

1. Proses 1.0

Pada proses ini sistem akan melakukan pengecekan sebab kerusakan terhadap jenis kerusakan yang dipilih oleh *user*. Pengecekan dilakukan dengan mencocokkan data dari sumber data yang ada pada tabel tindakan perbaikan. Pada saat proses ini berlangsung, peran sang *user* sangat mempengaruhi karena *user* yang akan memilih opsi-opsi yang ada dalam sistem pakar. Sehingga data jenis kerusakan yang dimasukkan oleh *user* akan mempengaruhi hasil akhir dari diagnosis yang akan dilakukan oleh sistem pakar.

2. Proses 2.0

Pada proses ini, setelah sebab kerusakan ditemukan maka sistem akan melakukan pengecekan tindakan perbaikan terhadap jenis kerusakan yang dipilih oleh *user*. Pengecekan dilakukan dengan mencocokkan data dari sumber data yang ada pada tabel tindakan perbaikan.

3. Proses 3.0

Pada proses ini, data yang telah direkam kemudian akan dicocokkan dengan sumber data pada tabel tindakan perbaikan. Setelah data tersebut didiagnosis, maka akan muncul hasil akhir yang berupa hasil diagnosa (solusi) yang akan digunakan oleh *user*.

4. Proses 4.0

Proses ini adalah proses *Update* yang hanya dilakukan oleh pakar yang telah diakui oleh pengguna sistem. Yang bertujuan untuk memodifikasi sumber data pada tabel tindakan perbaikan. Pada proses ini, pakar dapat memberikan masukan pada sistem berupa masukan *update knowledge base* jika hal tersebut dirasakan perlu.

PERANCANGAN TABEL DATA

Tabel 2. Tabel Login

Field Name	Data Type	Description
ID	AutoNumber	
username	Text	
password	Text	
level	Text	
photo	Ole Object	

Tabel *Login* digunakan untuk menyimpan data-data pengguna saat *login* sesuai *level* pengguna, baik sebagai *admin*/pakar maupun sebagai *user* biasa.

Tabel 3. Tabel Jenis Kerusakan

Field Name	Data Type	Description
kodejenis	Text	
uraianjeniskerusakan	Text	

Tabel Jenis Kerusakan digunakan untuk menyimpan data-data tentang jenis kerusakan mesin.

Tabel 4. Tabel Penyebab Kerusakan

Field Name	Data Type	Description
kodesebab	Text	
uraiansebabkerusakan	Text	

Tabel Penyebab Kerusakan digunakan untuk menyimpan data-data tentang penyebab kerusakan mesin.

Tabel 5. Tabel Tindakan Perbaikan

Field Name	Data Type	Description
ID	AutoNumber	
kodejenis	Text	
kodesebab	Text	

kodebobot	Number	
tindakanperbaikan	Text	

Tabel Tindakan Perbaikan digunakan untuk menyimpan data-data tentang tindakan perbaikan mesin.

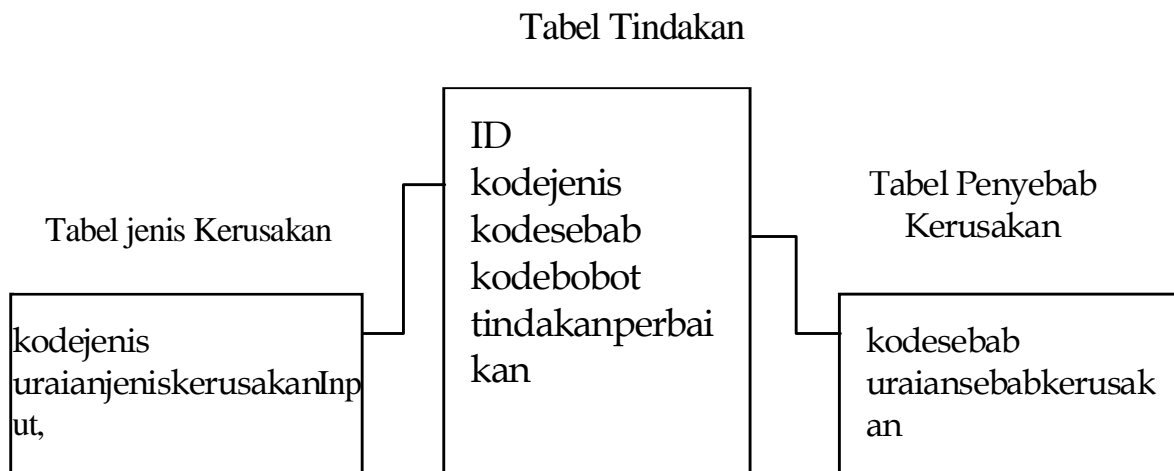
Tabel 6. Tabel Laporan

Field Name	Data Type	Description
ID	AutoNumber	
tanggal	Date/Time	
no_kendaraan	Text	
kesimpulan	Text	

Tabel Laporan digunakan untuk menyimpan data-data tentang laporan setelah semua proses diagnosis dilakukan.

ENTITY RELATIONSHIP DIAGRAM (ERD)

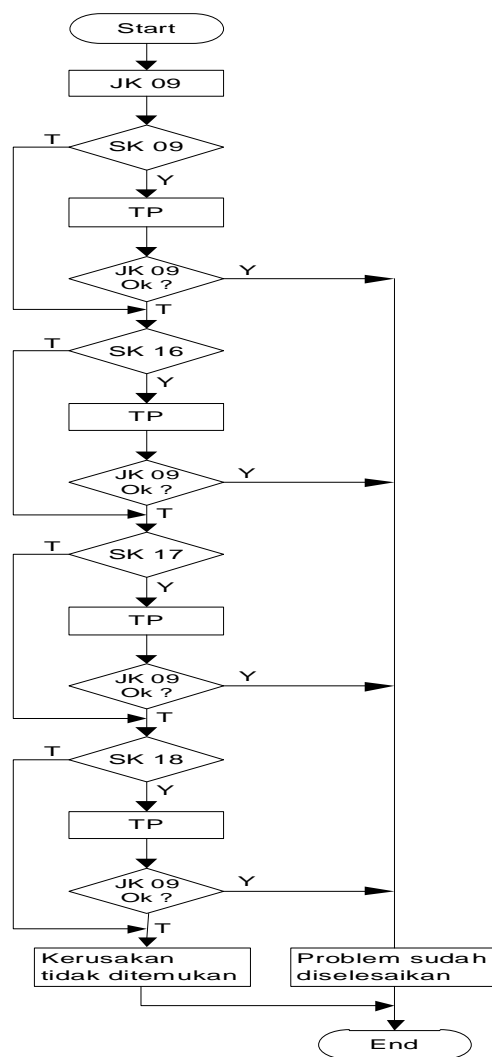
Entity Relationship Diagram pada aplikasi ini berguna untuk memudahkan dalam membuat konsep desain *database* yakni untuk menentukan berbagai entiti yang digunakan dan bagaimana data di dalam entiti dapat saling berhubungan. Untuk lebih jelasnya dapat dilihat pada Gambar 4.3



Gambar 8. Diagram Relasi Antar Tabel

PERANCANGAN MESIN INFERENSI

Pada perancangan sistem pakar untuk mendiagnosa penyebab kerusakan mesin tipe OM366A ini dilakukan dengan menggunakan mesin inferensi. Mesin inferensi *backward chaining* digunakan dalam sistem ini untuk mendiagnosa penyebab kerusakan mesin setelah *user* menjawab pertanyaan yang berhubungan dengan jenis kerusakan mesin. Dalam setiap menjawab pertanyaan, sistem telah menyediakan pilihan jawaban YA atau TIDAK yang akan dipilih oleh *user*. Setiap jawaban yang dipilih oleh *user* akan diproses oleh sistem dengan mencari kesesuaian aturan yang telah ditentukan *admin/pakar* dalam *knowledge-based*, sehingga nantinya akan diperoleh suatu solusi, berupa penyebab kerusakan dan tindakan perbaikannya. Penelusuran dalam metode *backward chaining* dapat dilihat pada *flowchart* dibawah ini.



Gambar 9. Flowchart Backward Chaining Untuk Jenis Kerusakan 09

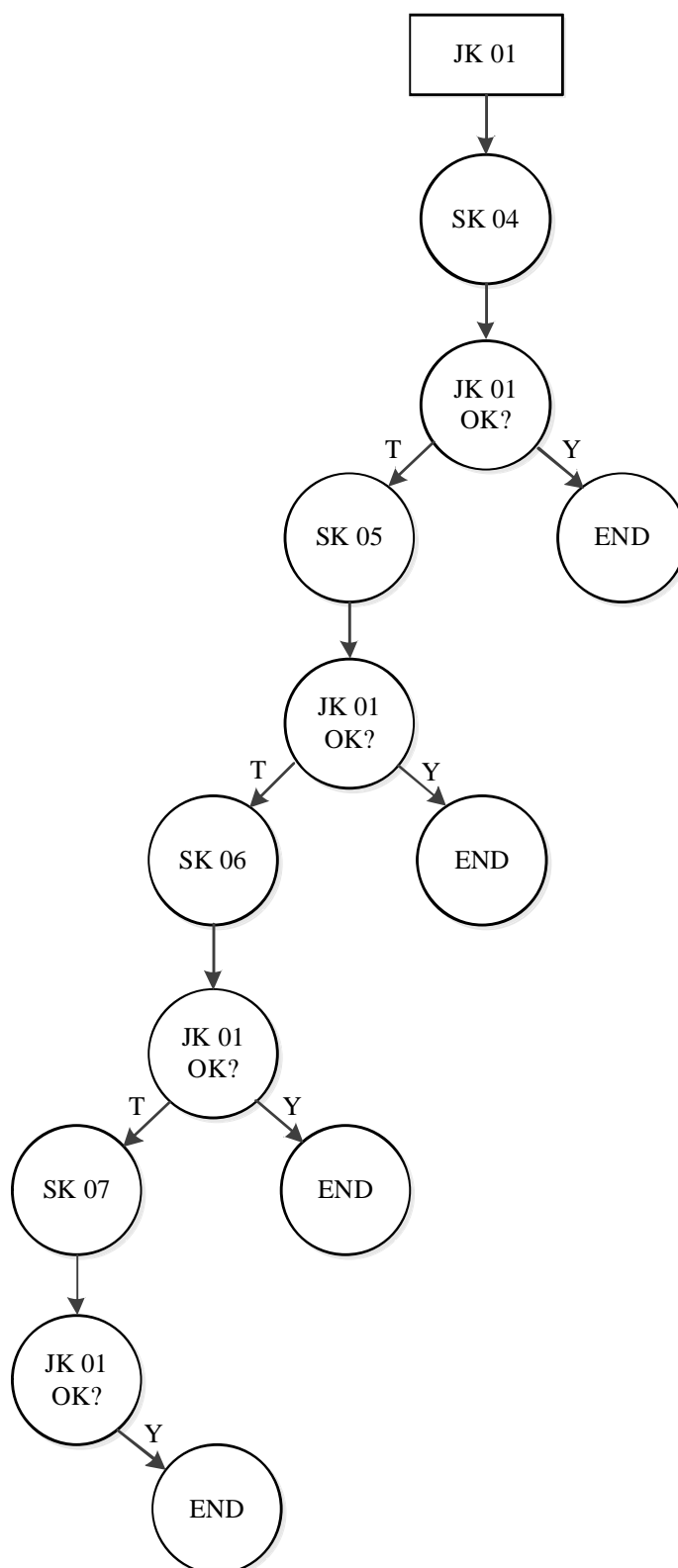
Tabel 7. Keterangan Kode *Flowchart Backward Chaining*

NO	KODE	KETERANGAN
1	JK09	Knocking atau ketukan pada mesin diesel
2	SK09	Pipa-pipa bahan bakar tertekuk atau bengkok
3	SK16	Nosel rusak atau penuh dengan arang
4	SK17	Mur pengikat nosel longgar
5	SK18	Saat pengiriman bahan bakar pompa injeksi tidak tepat
6	TP	Tindakan Perbaikan

Pada *flowchart* diatas, dapat dijelaskan bahwa data yang digunakan dalam sistem ini adalah data yang dimasukkan atau dipilih oleh *user*, yaitu jenis kerusakan mesin. Selanjutnya sistem akan menampilkan pertanyaan yang harus dijawab oleh *user* berdasarkan solusi penyebab kerusakan yang ditampilkan. Jika jawaban *user* TIDAK maka sistem akan menampilkan solusi penyebab kerusakan berikutnya, namun jika jawaban YA, maka sistem akan menampilkan solusi tindakan perbaikan, kemudian sistem akan melanjutkan ke pertanyaan selanjutnya, sistem akan memeriksa apakah permasalahan utama sudah dapat diselesaikan, jika jawaban YA maka sistem akan menampilkan diagnosa akhir yaitu problem sudah diselesaikan, jika jawaban TIDAK maka sistem akan menampilkan penyebab kerusakan yang lain. Jika kesemua penyebab kerusakan yang ditampilkan oleh sistem tidak ada yang benar, maka sistem akan menampilkan diagnosa akhir yaitu penyebab kerusakan tidak ada dalam database. Akhir dari proses diagnosa tersebut baik jenis kerusakan, penyebab kerusakan, tindakan perbaikan dan diagnosa akhir, semuanya akan dirangkum oleh sistem dalam info kesimpulan. Untuk memulai proses diagnosa yang baru cukup dengan menekan tombol reset, maka semua kolom penyebab kerusakan, tindakan perbaikan dan diagnosa akhir akan dikosongkan oleh sistem.

PERANCANGAN POHON KEPUTUSAN

Pembuatan pohon keputusan (*decision tree*) akan memudahkan sistem dalam melakukan penalaran hingga ditemukannya suatu kesimpulan yang tepat. Berikut ini pada Gambar 4.5 adalah perancangan pohon keputusan untuk sistem pakar dalam mendiagnosa penyebab kerusakan mesin tipe OM366A.



Gambar 10. Pohon Keputusan Untuk Jenis Kerusakan 01

Dalam penentuan parent/child dari pohon keputusan didasarkan pada nilai bobot untuk setiap sebab kerusakan pada tiap-tiap jenis kerusakan. Bobot yang paling tinggi urutannya akan menjadi parent

sedangkan bobot yang lebih rendah akan menjadi child dari pohon keputusan tersebut.

Tabel 8. Keterangan Kode Pohon Keputusan

NO	KODE	KETERANGAN
1	JK01	Mesin tidak dapat dihidupkan
2	SK04	Tekanan kompresi kurang (selisih 4 bar)
3	SK05	Tangki bahan bakar kosong
4	SK06	Saringan dalam tangki bahan bakar tersumbat
5	SK07	Ventilasi tangki bahan bakar tersumbat

IMPLEMENTASI SISTEM

Implementasi sistem merupakan suatu prosedur yang dilakukan untuk menampilkan hasil dari rancangan sistem, dimana program telah berhasil dijalankan. Berikut adalah tampilan serta pembahasan rancangan program Sistem Pakar Diagnosa Kerusakan Mesin Tipe OM366A Pada Mobil Mercedes-Benz.

Tampilan Menu Utama

Menu utama terdiri dari daftar pilihan yang dapat digunakan untuk mengakses berbagai macam fasilitas program. Menu utama secara keseluruhan dibangun dalam bentuk *pull down menu* yang dibuat menggunakan fasilitas *menu strip* dari *Visual Basic.Net 2008*. Menu dapat dipilih dengan cara mengklik pilihan-pilihan menu yang tersedia. Pada penelitian ini menu utama dibagi menjadi dua tampilan menu yaitu :

1. Tampilan Menu Utama *Admin/Pakar*

Pada tampilan form menu utama untuk *admin/pakar* ini terdapat pilihan menu *Admin* yang berfungsi untuk melakukan pengaturan terhadap perubahan *database system*, yaitu pengaturan terhadap perubahan *database User management*, Perawatan data daftar jenis

kerusakan, Perawatan data daftar penyebab kerusakan dan Perawatan data daftar tindakan perbaikan.



Gambar 11. Tampilan Menu Utama *Admin/Pakar*

2. Tampilan Menu Utama *User* biasa

Pada tampilan form menu utama untuk *user* biasa tidak terdapat pilihan menu *Admin*, sehingga pengguna sistem tidak akan dapat melakukan pengaturan atau perubahan terhadap *database system*.



Gambar 12. Tampilan Menu Utama *User* biasa

Tampilan Menu Analisis Kerusakan

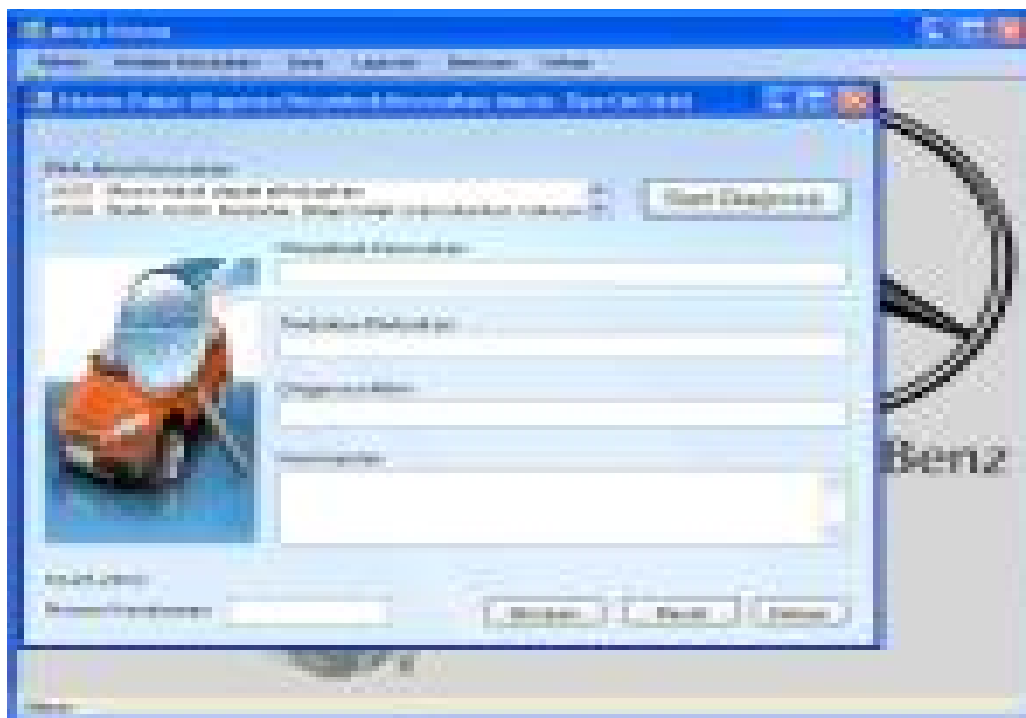
Menu Analisis Kerusakan adalah menu paling utama dalam perancangan sistem pakar ini, dimana pada menu inilah terjadinya proses pendiagnosaan dan solusi akhir yang dihasilkan oleh sistem, yaitu solusi terhadap penyebab kerusakan dan tindakan perbaikan mesin tipe OM366A yang terpasang pada kendaraan Mercedes-Benz.



Gambar 13. Tampilan Menu Analisis Kerusakan

Proses pendiagnosaan dan pembahasan program tersebut dapat dilihat pada tampilan program dibawah ini. Untuk mendapatkan hasil diagnosa penyebab kerusakan dan tindakan perbaikan akan terdapat beberapa tahap konsultasi antara pengguna dengan sistem itu sendiri.

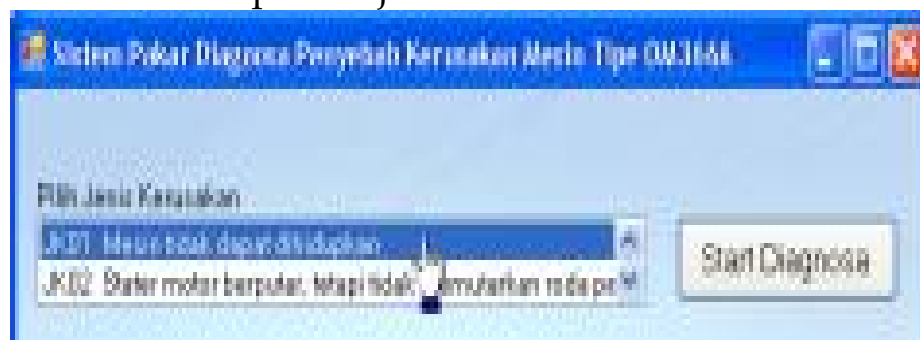
Untuk menampilkan antar muka proses pendiagnosaan tersebut adalah dengan cara klik **Menu Analisis**, pilih **Analisis Kerusakan Mesin Tipe OM366A** maka akan muncul tampilan antar muka proses pendiagnosaan seperti pada gambar dibawah ini.



Gambar 14. Tampilan Proses Diagnosa Penyebab Kerusakan Mesin

Adapun proses pendiagnosaan nya adalah sebagai berikut :

- Pilih salah satu jenis kerusakan yang terdapat dalam *Combo Box* pilihan jenis kerusakan.



Gambar 15. Tampilan Pilihan Jenis Kerusakan Mesin

- Tekan tombol *Start Diagnosa*

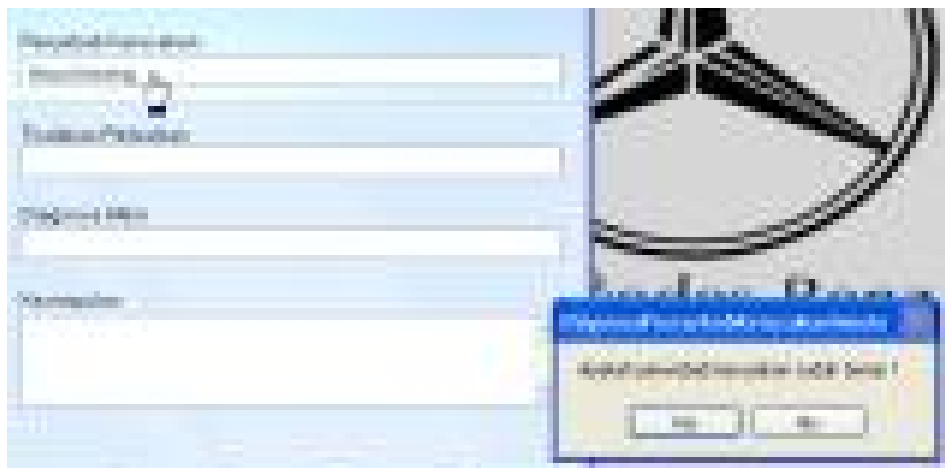
Pada saat inilah proses *inferensi engine* terjadi, yaitu kemampuan sistem untuk menalar serta menarik kesimpulan berdasarkan pengetahuan (*knowledge base*) dan aturan (*rule base*) yang telah tersimpan didalam sistem.



Gambar 16. Tampilan *Start Diagnosa*

Setelah tombol *start* diagnosa ditekan, sistem akan menampilkan solusi penyebab kerusakan pertama dari jenis kerusakan yang dipilih.

- Menjawab setiap pertanyaan yang diajukan oleh sistem
Sebelum tindakan perbaikan ditampilkan, sistem akan memberikan pertanyaan “apakah penyebab kerusakan sudah benar?”,



Gambar 17. Tampilan Solusi Penyebab Kerusakan Pertama

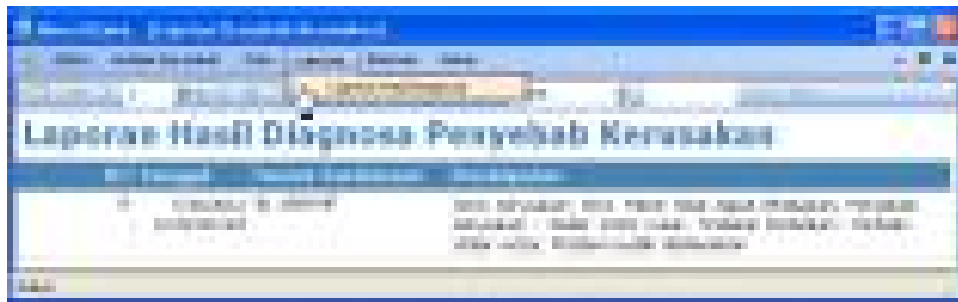
Informasi dari diagnosa akhir ini adalah menyatakan bahwa semua problem atau permasalahan terhadap penyebab kerusakan mesin tipe OM366A sudah selesai didiagnosa dan penyebab kerusakan telah ditemukan, sementara informasi yang terdapat dalam kesimpulan adalah berupa rangkuman informasi dari awal sampai akhir proses pendiagnosaan oleh sistem pakar ini. Hasil rangkuman informasi ini dapat dilihat pada menu laporan hasil diagnosa pada menu utama.

- Pada proses akhir diagnosa, pengguna dapat memasukkan nomor kendaraan kemudian klik tombol **Simpan**, ini berfungsi untuk menyimpan semua proses diagnosa dan pemberian nomor kendaraan adalah bertujuan untuk mempermudah pengguna atau teknisi dalam mendapatkan informasi terakhir penyebab kerusakan mesin yang terpasang pada kendaraan tersebut.

-

Tampilan Menu Laporan

Menu laporan ini berfungsi untuk memberikan informasi hasil diagnosa kepada pengguna sistem tentang prosespendiagnosaan yang pernah dilakukan pada mesin tersebut. Adapun informasi hasil laporan adalah terdiri dari waktupendiagnosaan dilakukan, nomor kendaraan yang didiagnosa serta kesimpulan hasil diagnosa.



Gambar 18. Tampilan Laporan Hasil Diagnosa

KESIMPULAN

Dari hasil pengujian sistem yang telah dilakukan, maka dapat diberikan beberapa kesimpulan sebagai berikut:

1. Dengan sistem ini *user* dapat dengan mudah mengetahui solusi penyebab kerusakan dan tindakan perbaikan yang harus dilakukan terhadap permasalahan kerusakan mesin tipe OM366A.
2. Penalaran dengan menggunakan mesin inferensi *backward chaining* dapat digunakan dalam perancangan sebuah sistem

pakar untuk mendiagnosa penyebab kerusakan mesin, karena telah teruji mendapatkan solusi yang tepat.

3. Penggunaan pohon keputusan juga sangat membantu proses mekanisme inferensi dalam melakukan penalaran menuju solusi yang diinginkan.
4. Secara teknis, sistem pakar ini sangat membantu dan menambah pengetahuan bagi pengguna sistem dalam pengambilan tindakan perbaikan terhadap kerusakan mesin.

DAFTAR PUSTAKA

Muhammad Arhami, *Konsep Dasar Sistem Pakar*, Andi, Yogyakarta, 2004

Rahmad Hakim S, *Visual Basic 2008 for Pemula Banget*, PT. Elex Media Komputindo, 2009

Sri Kusumadewi, *Teknik dan Aplikasi Artificial Intelligence*, Andi Yogyakarta, 2003

Team Penerbit ANDI, *Pengembangan Sistem Pakar Menggunakan Visual Basic*, Andi Yogyakarta, 2003

Turban, E, *Decision Support and Expert System ; Management Support System*, Prentice-Hall, Newyork, 2007

Mercedes-Benz Central Service Departement Training Section *Mencari gangguan-gangguan yang terjadi pada mesin*, 1992

PT. Star Motor Indonesia Central Service Departement, *Petunjuk Perbaikan Mesin Tipe OM366A*, 1987