

SISTEM PAKAR SEBAGAI ALAT BANTU MENGATASI MASALAH (STUDI KASUS KERUSAKAN SEPEDA MOTOR)

Adi Suwondo^a

^aProgram Studi Teknik Informatika Universitas Sains Al-Qur'an (UNSIQ) Wonosobo

^aEmail: adiunsiq@gmail.com

INFO ARTIKEL

Riwayat Artikel:

Diterima : 20 February 2014

Disetujui : 5 April 2014

Kata Kunci:

sistem pakar, basis pengetahuan, kerusakan

ABSTRAK

Mekanik sepeda motor dalam menganalisa kerusakannya sampai menentukan solusi yang tepat untuk diterapkan pada sepeda motor melewati langkah yang bertahap, pertama, apa keluhan pengguna akan keadaan sepeda motor, kemudian menganalisa masalah, mengembangkan pengetahuan, penyebab dari masalah yang timbul serta memberikan solusi yang dapat dikerjakan. Hal ini berturut – turut dilakukan oleh seorang mekanik sepeda motor, dengan jumlah kerusakan dan masalah yang timbul baik masalah yang telah hafal diketahuinya maupun masalah baru yang dapat sewaktu – waktu muncul dari permasalahan yang ada, seorang mekanik akan meruntut satu persatu hingga menemukan hasil akhir yang sesuai.

Pengetahuan dan daya ingat dari seorang mekanik dituntut untuk tidak menduplikasi masalah yang sebenarnya sudah dapat diselesaikan pada waktu terdahulu tetapi karena keterbatasan ingatan, masalah yang sebenarnya sudah memiliki dokumen tiba – tiba dapat hilang tanpa ada sedikitpun tersisa, sementara sebenarnya masalah yang ada pada sepeda motor sudah ditemukan solusinya dari kejadian yang pernah dilakukan pula.

Dari ulasan di atas pada penelitian ini akan dibuat sebuah sistem pakar yang memiliki basis pengetahuan dari mekanik yang telah direkam dalam database, sehingga solusi yang dicari dapat dengan cepat di bankitkan kembali melalui sistem yang akan dirancang.

ARTICLE INFO

Article History

Received : February 20, 2014

Accepted : April 5, 2014

Key Words :

Thermal comfort, PMV, SMK 1 Wonosobo

ABSTRACT

Motorcycle mechanic in analyzing the damage to determine the correct solution to be applied on a motorcycle passes gradual steps, first, what will be the state of the user complaints motorcycle, then analyze the problem, develop the knowledge, the cause of the problems that arise and provide a workable solution. It is straight - also done by a motorcycle mechanic, with the amount of damage and problems that arise either known problem that has been memorized and new problems that can be any - time emerge from the existing problems, a mechanic will meruntut one by one to find the final result appropriate.

Knowledge and memory of a mechanic is required to not duplicate the problem that actually can be completed on time earlier but due to memory limitations, the actual problem already has a document arrived - arrived can be lost without the slightest remaining, while the real problems that exist on the motorcycle has found the solution of the events that have been done anyway.

From the review above in this study will be made an expert system that has knowledge base of mechanics that have been recorded in the database, so that the solution can be quickly searched in bankitkan back through the system to be designed.

1. PENDAHULUAN

Pakar yang berbasis sistem komputer menjadi solusi tiap permasalahan yang ada, sifatnya yang mampu menyimpan dokumen lama dan membuka kembali dokumen tersebut menjadi sebuah inputan yang akurat tanpa ada kekurangan menjadikan sistem ini sebagai alternatif pakar pilihan guna penyelesaian masalah pada sepeda motor.

Sistem pakar berbasis komputer sama halnya seperti pakar yang memiliki fokus keilmuan pada bagian spesifik, sebagai contoh seorang pakar teknik mekanik ia mampu memberikan solusi pengetahuan tentang teknik mekanik yang digelutinya, begitu pula sistem pakar yang akan direncanakan, sistem ini akan mampu pula menjadi konsultan dari permasalahan sepeda motor, layaknya seorang pakar ahli dalam teknik permasalahan sepeda motor.

Sepeda motor sebagai alat transportasi minimalis yang memiliki fungsi lebih handal dan bebas hambatan dari pada kendaraan beroda empat, akhir – akhir ini diminati oleh pengguna jalan raya khususnya di kota – kota besar yang intensitas kemacetannya tinggi, sehingga mau tidak mau bagi para teknisi kendaraan roda dua ini dituntut untuk bekerja lebih cepat dan akurat.

Sehingga fungsi dari sebuah sistem yang mampu menganalisa permasalahan hingga memberikan solusinya pada sepeda motor akan sangat dibutuhkan guna melayani begitu banyaknya permasalahan yang ada dari setiap pengguna sepeda motor.

Permasalahan kerusakan sepeda motor menjadi tidak henti – hentinya dibahas dan ditemukan solusinya guna pemakaian yang tahan lama dan bernilai ekonomis bagi penggunaanya, berbagai macam teknik dan cara untuk dapat dengan cepat mengetahui kerusakan yang ada. Pakarpun dilibatkan dalam teknik penyelesaian dengan berbagai macam penelitian dan uji coba, hasilnya pakar dapat menemukan permasalahan – permasalahan baru teknik mekanik sepeda motor.

Penemuan – penemuan tersebut disimpan dan menjadi dokumen yang siap dibuka kembali apabila terjadi permasalahan yang telah ada pada index dokumen permasalahan. Tetapi semakin banyak dokumen tentang

permasalahan akan semakin berat pula mekanik mengingat kembali permasalahan – permasalahan yang sebenarnya sudah ada. Dari ulasan tersebut penulis merumuskan masalah yang ada yakni, kerusakan pada sepeda motor, sistem pakar dalam menganalisa kerusakan sepeda motor.

Adapun dalam penelitian ini akan dibatasi permasalahannya pada: kerusakan sepeda motor dapat dikatakan bahwa setiap munculnya teknologi baru akan menyertakan pula kemungkinan kerusakan baru yang ditemukan. Hadirnya motor – motor baru yang mengusung teknologi metic atau auto gear tidak dapat dikelompokkan pada permasalahan yang telah ditemukan pada motor – motor konvensional yang pernah ada.

Kerusakan sepeda motor memiliki bagian – bagian yang kompleks sehingga untuk memperjelas pembahasan supaya tidak panjang lebar penulis membatasi pada sistem kerusakan sepeda motor sesuai tingkat pengetahuan yang telah ada pada teknik perawatan dan analisa kerusakan sepeda motor konvensional.

2. TINJAUAN PUSTAKA

a. Sistem Pakar (*Expert System*)

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Martin dan Oxman, 1988).

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud antara lain: pembuatan keputusan (*decision making*), pemaduan pengetahuan (*knowledge using*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*) dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar (Martin dan Oxman, 1988).

Sistem pakar dibuat pada wilayah pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia

di salah satu bidang. Sistem pakar mencoba mencari solusi yang memuaskan sebagaimana yang dilakukan seorang pakar. Selain itu sistem pakar juga dapat memberikan penjelasan terhadap langkah yang diambil dan memberikan alasan atas saran atau kesimpulan yang ditemukannya.

Biasanya sistem pakar hanya digunakan untuk memecahkan masalah yang memang sulit untuk dipecahkan dengan pemrograman biasa, mengingat biaya yang diperlukan untuk membuat sistem pakar jauh lebih besar dari pembuatan sistem biasa

b. Arsitektur Sistem Pakar

Sistem pakar memiliki beberapa komponen utama, yaitu antarmuka pengguna (user interface), basis data sistem pakar (expert system database), fasilitas akuisi pengetahuan (knowledge acquisition facility), dan mekanisme inferensi (inference mechanism). Selain itu ada satu komponen yang hanya ada pada beberapa sistem pakar, yaitu fasilitas penjelasan (explanation facility) (Martin dan Oxman, 1988).

Antarmuka pengguna adalah perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem. Basis data sistem pakar berisi pengetahuan setingkat pakar pada subyek tertentu. Berisi pengetahuan yang dibutuhkan untuk memahami, merumuskan, dan menyelesaikan masalah.

Basis data ini terdiri dari 2 elemen dasar:

Fakta, situasi masalah dan teori yang terkait.

Heuristik khusus atau rules, yang langsung menggunakan pengetahuan untuk menyelesaikan masalah khusus.

Pengetahuan ini dapat berasal dari pakar, jurnal, majalah, dan sumber pengetahuan lain.

Fasilitas akuisisi pengetahuan merupakan perangkat lunak yang menyediakan fasilitas dialog antara pakar dengan sistem. Fasilitas akuisisi ini digunakan untuk memasukkan fakta – fakta dan kaidah – kaidah sesuai dengan perkembangan ilmu. Meliputi proses pengumpulan, pemindahan, dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi (buku, dan lain - lain) ke program komputer, yang bertujuan untuk memperbaiki dan atau mengembangkan basis pengetahuan (*knowledge base*).

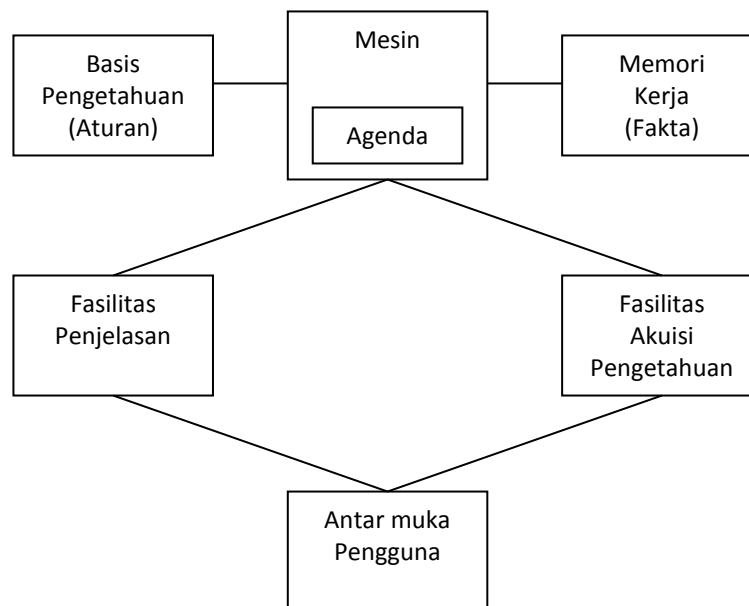
Mekanisme inferensi merupakan perangkat lunak yang melakukan penalaran dengan menggunakan pengetahuan yang ada untuk menghasilkan suatu kesimpulan atau hasil akhir. Dalam komponen ini dilakukan pemodelan proses berpikir manusia.

Fasilitas penjelasan berguna dalam memberikan penjelasan kepada pengguna mengapa komputer meminta suatu informasi tertentu dari pengguna dan dasar apa yang digunakan komputer sehingga dapat menyimpulkan suatu kondisi.

Ada 4 tipe penjelasan yang digunakan dalam sistem pakar, yaitu (Schnupp, 1989):

- Penjelasan mengenai jejak aturan yang menunjukkan status konsultasi.
- Penjelasan mengenai bagaimana sebuah keputusan diperoleh.
- Penjelasan mengapa sistem menanyakan suatu pertanyaan.
- Penjelasan mengapa sistem tidak memberikan keputusan seperti yang dikehendaki pengguna.

Arsitektur dasar dari sistem pakar dapat dilihat pada gambar 1 (Giarrantano dan Riley, 1994).



Gambar 1 Arsitektur Sistem Pakar

Memori kerja dalam arsitektur sistem pakar merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi, berisi fakta-fakta tentang suatu masalah yang ditemukan dalam proses konsultasi.

c. Kaidah Produksi

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

JIKA premis MAKA konklusi

JIKA masukan MAKA keluaran

JIKA kondisi MAKA tindakan

JIKA anteseden MAKA konsekuensi

JIKA data MAKA hasil

JIKA tindakan MAKA tujuan

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Anteseden mengacu pada situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Tindakan mengacu pada kegiatan

yang harus dilakukan sebelum hasil dapat diharapkan (Hanifah, 1998).

Kaidah dapat diklasifikasikan menjadi dua, yaitu kaidah derajat pertama (*first order rule*) dan kaidah meta (*meta rule*) (Giarrantano dan Riley, 1994). Kaidah derajat pertama adalah kaidah sederhana yang terdiri dari anteseden dan konsekuensi, misalnya:

JIKA bersin-bersin dan pusing

MAKA terserang penyakit flu

Kaidah meta adalah kaidah yang anteseden atau konsekuennya mengandung informasi tentang kaidah yang lain, misalnya:

Aturan 1:

JIKA mengalami kehilangan kesadaran yang berlangsung sangat singkat, sehingga aktivitas yang sedang berjalan terhenti.

DAN terkadang disertai dengan mata yang menatap kosong atau gerakan mioklonik darisekelompok otot mata atau wajah, otomatisme, kehilangan tonus otot (sehingga barang yang dipegang bisa terjatuh atau bila sedang berdiri bisa terjatuh)

DAN serangan berakhir dengan diikuti oleh pulihnya kesadaran

DAN berlangsung beberapa detik sampai setengah menit, dan

dapat berlangsung puluhan kali dalam sehari

Aturan 2:

JIKA	tipe sawan umum
ATAU	tipe sawan mioklonik
ATAU	tipe sawan lena
ATAU	tipe sawan tonik-klonik
DAN	EEG tidak menunjukkan adanya kelainan fokal
DAN	penyebab tidak diketahui
DAN	awitan berhubungan dengan usia
MAKA	terkena epilepsi idiopatik umum

Dalam contoh di atas, konsekuensi dari aturan 1, yaitu mengalami sawan lena, menjadi anteseden bagi aturan 2.

d. Inference Engine

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 (RP) 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. Ada dua metode inferensi yang penting dalam sistem pakar, yaitu runut maju (*forward chaining*) dan runut balik (*backward chaining*).

e. Runut Maju (Forward Chaining)

Runut maju berarti menggunakan himpunan aturan kondisi – aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses

menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil (Wilson, 1998). Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*) (Giarattano dan Riley, 1994).

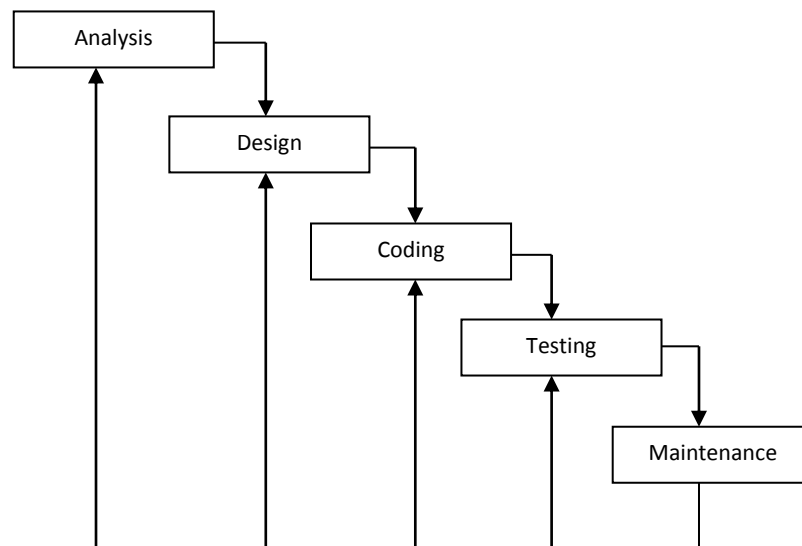
Dalam penalaran maju, aturan diuji satu demi satu dalam urutan tertentu. Urutan itu mungkin berupa urutan pemasukan aturan ke dalam basis aturan atau juga urutan lain yang ditentukan oleh pemakai. Saat tiap aturan diuji, sistem pakar akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisinya benar, maka aturan itu disimpan kemudian aturan berikutnya diuji. Sebaliknya bila kondisi salah, aturan itu tidak disimpan dan aturan berikutnya diuji. Proses ini akan berulang (*iterative*) sampai seluruh basis aturan teruji dengan berbagai kondisi.

Sebagai contoh penalaran maju adalah mengecek kerusakan mesin kendaraan bermotor akan dimulai dengan macam – macam kerusakan mesin yang akan ditelusuri kemudian dilanjutkan dengan jenis – jenis dari macam kerusakan yang dipilih, dan seterusnya sampai pada diagnosis kerusakan dan hasil akhir kesimpulan kerusakan tersebut.

Penalaran maju seperti yang digambarkan di atas sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, maka seluruh proses akan dikerjakan secara berurutan maju. Tetapi dalam masalah – masalah yang lain penalaran bisa saja dimulai dari hasil akhir yang berupa suatu hipotesis dan akan dicari pembuktiannya. Kasus semacam ini harus diselesaikan dengan penalaran mundur.

3. METODE PENELITIAN

Metode pada pengembangan sistem pakar ini menggunakan *waterfall* dimana runtutan proses digambarkan dalam gambar berikut:

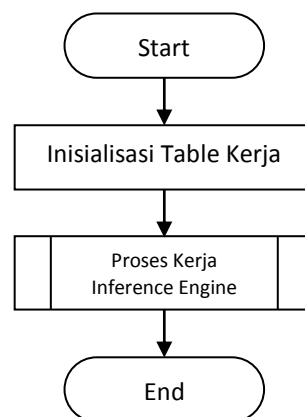


Gambar 2 Metode The Classic Life Cycle (Waterfall)

Tahapan-tahapan yang dilakukan melalui *waterfall* disusun sedemikian rupa dengan tujuan memudahkan mendokumentasi tiap-tiap prosesnya. Sedangkan dalam proses runutnya menggunakan metode *forward chaining*

3.1. Metode *Forward Chaining*

Penjelasan metode *forward chaining* akan dibagi menjadi dua bagian, yaitu: bagian inialisasi tabel kerja dan proses kerja inference engine. Metode *forward chaining* akan diterapkan pada bagian proses kerja *inference engine*. Diagram alur *inference engine* secara garis besar digambarkan seperti gambar 3 sebagai berikut:



Gambar 3 Diagram Alur Inference Engine

Pertama – tama program akan melakukan inialisasi dengan membuat tiga table kerja yaitu *working memory table*, *attribute queue table* dan *rule/ premise table*. *Rule* atau *premise table* akan berisi aturan atau dasar dari mana kesimpulan diambil, dimana datanya didapat dari tabel macam kerusakan. *Tabel attribute queue* akan berisi *queue* dari ciri – ciri kerusakan atau *rule* yang terdapat pada *rule* atau *premise tabel*, *rule* dari

attribute queue ini akan berfungsi sebagai bahan pertanyaan yang akan diajukan pada user. Isi dari *attribute queue* ini akan berubah terus sesuai macam *rule* yang masih terdapat pada tabel *rule* atau *premise*.

Inisialisasi tabel kerja dilaksanakan dengan membuat tabel *rule* atau premis, tabel *attribute queue*, dan tabel *working memory*. Pada kondisi awal seluruh tabel dikosongkan seperti dapat dilihat pada Tabel 1.

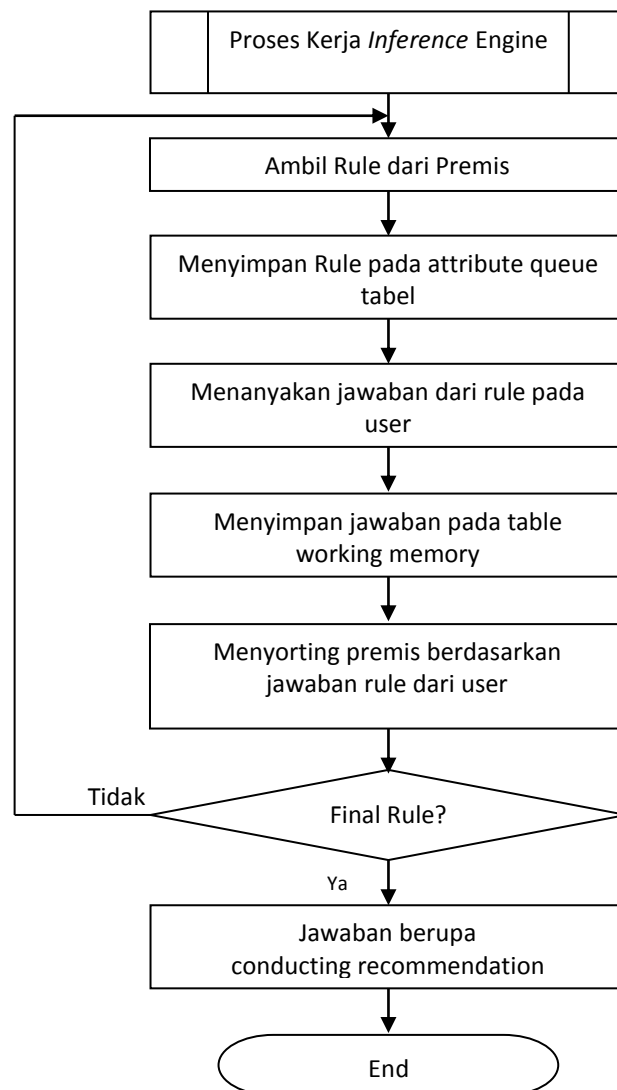
Tabel 1 Inisialisasi Tabel Kerja

Table Rule/ Premise		
PID	RID	PVALUE
Mesin tidak mau hidup atau susah dihidupkan	Busi	Baik
	Fuse (sekering)	Baik
	Saklar	Baik
Mesin mudah mati	Busi	Arus sedikit
	Fuse (sekering)	Baik
	Saklar	Baik
Mesin bunyi tidak normal	Busi	Rusak
	Fuse (sekering)	Baik
	Saklar	Rusak

Tabel Attribute Queue	Tabel Working Memory
Keadaan busi	busi

Tabel *rule* atau premis untuk memasukkan data atau status dari *rule* atau premis pada saat proses *inference* dijalankan. Tabel *attribute queue* untuk menampung pertanyaan

– pertanyaan yang dibutuhkan oleh *rule*. Tabel *working memory* untuk menyimpan jawaban dari pertanyaan – pertanyaan yang diajukan oleh *rule*.



Gambar 4 Diagram Alir Proses Kerja Inference Engine

Setelah terbentuk tabel kerja seperti di atas maka proses kerja inference engine mulai dilakukan. Keseluruhan sistem kerja *inference engine* dapat dilihat seperti pada gambar 4. Dimana pertama – tama ciri – ciri jenis kerusakan atau *rule* akan diambil dari data jenis kerusakan yang dijumpai dalam tabel rule atau premise kemudian ciri – ciri jenis kerusakan rule ini akan dimasukkan pada *attribute queue tabel* dan *tabel working memory* untuk ditanyakan harga dari *rule* tersebut pada user. Setelah mendapatkan jawaban dari user maka jawaban akan diinputkan pada tabel *working memory* kemudian tabel rule atau premise akan sorting sesuai dengan jawaban yang didapatkan dengan tidak mengikut sertakan data jenis

kerusakan yang memiliki *rule* yang bersesuaian tapi dengan harga rule yang tidak sama dengan jawaban yang didapat dari user. Hal ini dimaksudkan supaya semua data jenis kerusakan yang tidak memiliki ciri – ciri dengan jawaban user dapat langsung diabaikan sehingga proses pencarian data dapat dijalankan secara cepat dan efisien.

Seperti yang dapat dibandingkan perubahan yang terjadi pada tabel 2 dimana bila rule busi dijawab dengan baik, maka pada tabel *rule* atau *premise* data jenis kerusakan semua data yang memiliki *rule* busi yang tidak berharga baik akan dihapus dari tabel sehingga didapatkan bentuk tabel seperti pada tabel 2 yang hanya memuat data jenis kerusakan yang memiliki *rule* busi baik.

Tabel 2 Perubahan Isi Tabel Kerja

Table Rule/ Premise		
PID	RID	PVALUE
Mesin tidak mau hidup atau	Busi	Baik

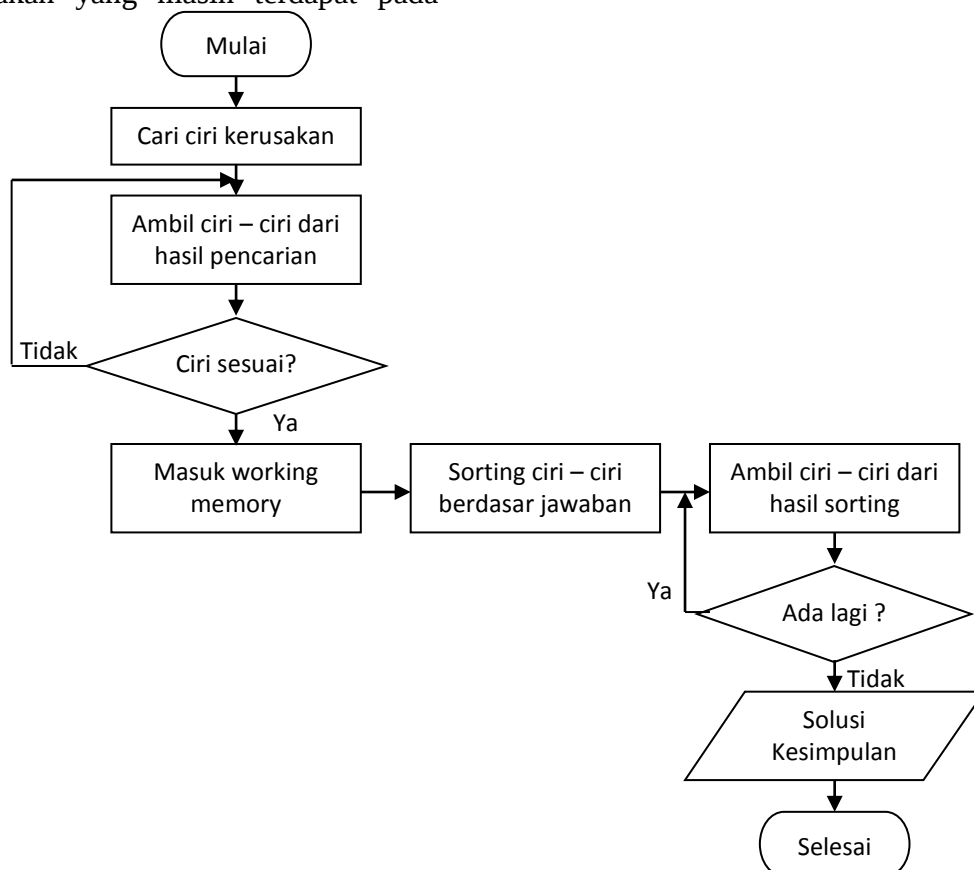
susah dihidupkan	Fuse (sekering)	Baik
	Saklar	Baik
Mesin mudah mati	Busi	Baik
	Fuse (sekering)	Baik
	Saklar	Baik

Tabel Attribute Queue	Tabel Working Memory
Keadaan Busi	Baik
Keadaan fuse (sekering)	?

Bila telah selesai untuk satu rule maka *attribute queue* untuk *rule* tersebut di-*disable*, kemudian diperiksa lagi apakah ada *rule* pada *premise* yang belum terjawab *attribute queue*. Bila masih ada maka proses di atas diulang kembali seperti bisa dilihat pada tabel 2 dimana diambil *rule* yang berikutnya yaitu *rule fuse* (sekering). Jika ternyata semua *rule* pada *tabel rule* atau *premise* telah terjawab dan tidak ada *rule* lagi yang tersisa, maka jenis kerusakan yang masih terdapat pada

tabel *rule* atau *premise* merupakan jawaban dari jenis kerusakan yang ditanyakan. Tidak menutup kemungkinan jawaban jenis kerusakan yang didapatkan akan lebih dari satu hal ini mengingat banyaknya kemiripan dari jenis kerusakan yang ada.

Program sistem pakar ini dirancang untuk memberikan beberapa alternatif jawaban dan ke semua kemungkinan itu akan ditampilkan pada suatu daftar list hasil.



Gambar 5 Flowchart Aplikasi Diagnosa Kerusakan Sepeda motor

Pertama – tama aplikasi akan memberikan pilihan kerusakan yang terjadi pada *user*, kemudian ciri – ciri kerusakan akan di ambil untuk diberikan kepada *user*, *user* memilih

sesuai kerusakan yang ada sebagai sebuah jawaban yang tersimpan pada memori kerja.

Pilihan dari *user* akan dijadikan kunci penyaringan ciri – ciri dari kerusakan yang

ada, kemudian pertanyaan dilanjutkan pada ciri – ciri yang telah terfilter tadi, pertanyaan tersebut akan terus diulang selama ciri – ciri pada hasil penyaringan belum habis.

Jika jumlah pertanyaan pada tabel yang telah tersaring telah habis maka user akan mendapatkan solusi dan kesimpulan dari permasalahan yang dicari.

4. HASIL DAN PEMBAHASAN

Form Basis Pengetahuan, form ini memiliki 2 tab halaman, yang pertama untuk mengisi jenis – jenis kerusakan dan halaman

kedua untuk mengisi penyebab kerusakan serta penanganannya.

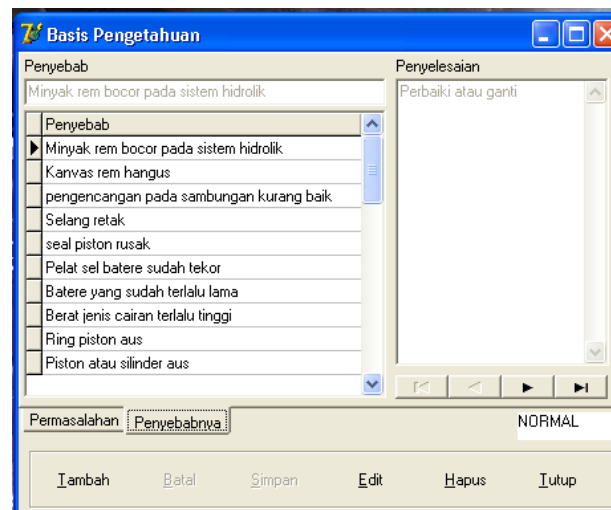
Tab pertama ini digunakan untuk mengisi basis pengetahuan jenis – jenis kerusakan yang ada, seperti contoh mesin susah hidup, mesin bunyi tidak normal, kopling selip dan lain sebagainya. Pada *form* ini terdapat satu buah *text box* yang digunakan untuk menuliskan jenis kerusakan serta beberapa tombol seperti navigasi untuk berpindah antar *record*, tambah untuk menambah jenis kerusakan baru, batal untuk membatalkan rencana inputan, simpan untuk menyimpan data baru, hapus digunakan untuk menghapus data serta tutup yang berfungsi untuk menutup *form*.

Gambar 6 Form Basis Pengetahuan Tab (1) Permasalahan

Tab kedua digunakan untuk mengisi basis pengetahuan penyebab kerusakan dari jenis kerusakan yang ada, seperti contoh mesin tidak mau hidup karena busi rusak, mesin bunyi tidak normal karena kerenggangan klep, kopling selip karena per (*pegas*) kopling lemah dan lain sebagainya.

Pada tab ini terdapat satu buah *text box* yang digunakan untuk menuliskan penyebab

kerusakan, memo untuk menuliskan langkah penanganannya yang dapat dikerjakan serta beberapa tombol seperti navigasi untuk berpindah antar *record*, tambah untuk menambah ciri kerusakan baru, batal untuk membatalkan rencana inputan, simpan untuk menyimpan data baru, hapus digunakan untuk menghapus data serta tutup yang berfungsi untuk menutup *form*.

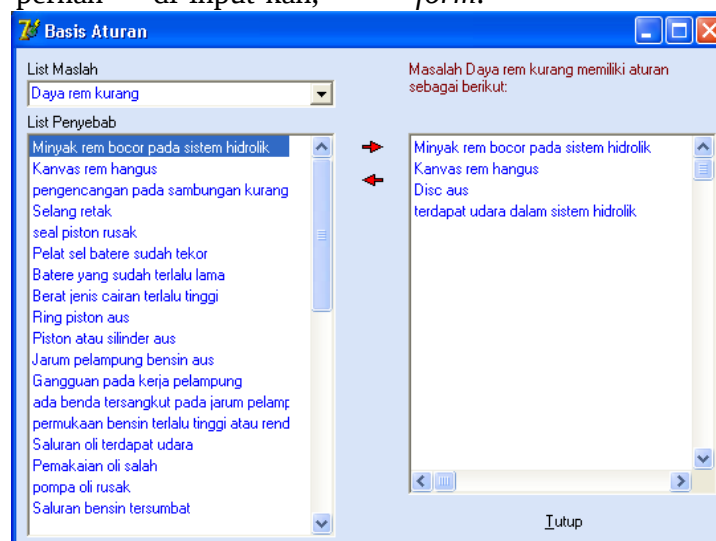


Gambar 7 Form Basis Pengetahuan Tab (2) Penyebab

Form Aturan, *form* aturan yang digunakan untuk membuat aturan – aturan yang mungkin timbul pada jenis kerusakan dengan mengintegrasikan ciri – ciri yang ada lengkap dengan diagnosa. *Form* ini hanya dapat diakses oleh seorang pakar untuk membuat basis aturan.

Pada *form* ini terdapat *combo* yang jika di klik akan menampilkan jenis – jenis kerusakan yang pernah di-input-kan,

kemudian *grid* yang menampilkan ciri – ciri kerusakan dan *grid* untuk mengintegrasikan antara jenis kerusakan dan ciri – ciri yang mungkin ada pada kerusakan yang ditampilkan pada *combo*. Juga terdapat 3 buah tombol yakni simpan untuk memasukkan ciri – ciri pada jenis kerusakan, hapus untuk menghapus ciri – ciri kerusakan pada jenis kerusakan dan tombol tutup untuk menutup *form*.



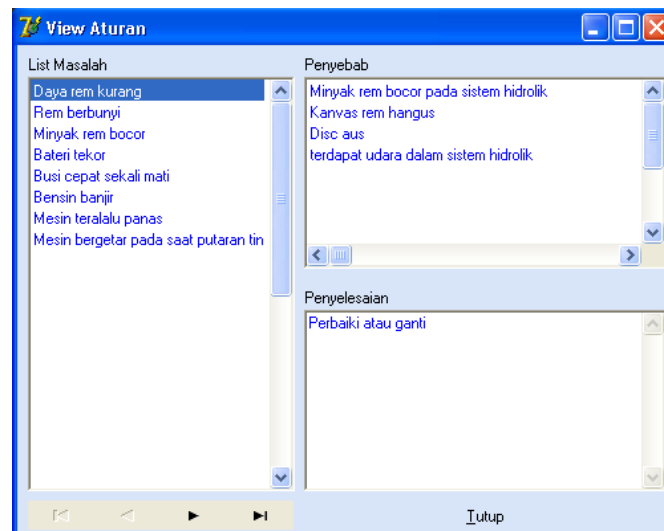
Gambar 8 Form Input Basis Aturan

4.1. Eksekusi Operasi

Form View, *form* view aturan yang digunakan untuk menampilkan semua aturan yang pernah dibuat yang berisi tentang jenis kerusakan dan ciri – ciri kerusakan dari jenis kerusakan yang ada pada *grid* jenis kerusakan.

Pada *form* ini terdapat *grid* yang menampilkan jenis kerusakan, memo yang

menampilkan gejala yang ada pada tiap jenis kerusakan, *grid* ciri – ciri kerusakan yang menampilkan ciri – ciri kerusakan dari *grid* jenis kerusakan dan memo yang menampilkan diagnosa dari ciri – ciri kerusakan. Juga terdapat tombol navigasi untuk berpindah antar data dan satu buah tombol tutup yang berfungsi untuk menutup *form*.



Gambar 9 Form View Aturan

Form Konsultasi, form konsultasi yang nantinya menjadi jembatan antara sistem dengan pengguna. Model konsultasi yang disediakan oleh sistem adalah dengan memberikan pilihan yang tersedia pada sistem, pengguna dapat mencari jenis kerusakan jika grid yang menampilkan jenis – jenis kerusakan terlalu banyak.

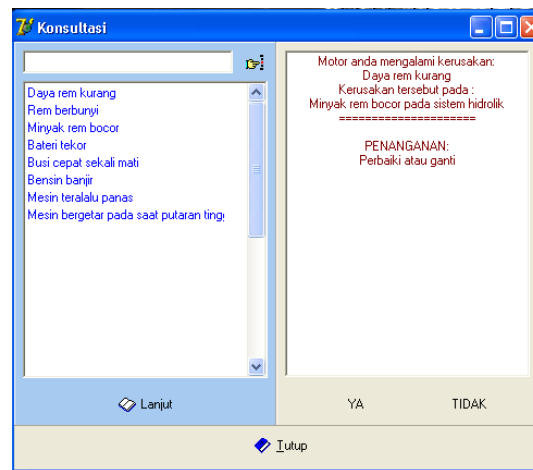
Berdasarkan pilihan pengguna inilah sistem akan menentukan diagnosa yang dapat dikerjakan oleh pengguna. Pada form konsultasi ini terdapat grid yang menampilkan jenis – jenis kerusakan yang dapat dipilih, kotak edit untuk melakukan pencarian berdasarkan kalimat yang di-input-kan pengguna dan tombol Lanjutkan untuk menuju pengetahuan selanjutnya.



Gambar 10 Form Konsultasi

Setelah user memilih jawaban TIDAK atas pertanyaan dari kondisi komponen yang tidak

dalam kondisi baik maka sistem akan memberikan solusinya pemecahannya.



Gambar 11 Solusi Penelusuran

5. KESIMPULAN

Sistem pakar dalam penggunaan sehari – hari dirasa sangat perlu untuk menjawab kebutuhan pengguna dalam menyelesaikan permasalahan, tentunya dengan masalah yang spesifik dan memiliki tingkat keakuratan yang tinggi, sehingga kebutuhan seorang pakar dalam setiap permasalahan dapat diwakilkan oleh sistem pakar berbasis komputer ini.

Setelah melakukan analisis, perancangan, implementasi dan evaluasi penanganan masalah kerusakan sepeda motor, maka penulis mengambil beberapa kesimpulan :

Sistem pakar ini diharapkan dapat menjadi alternatif pertama dan sebagai alat bantu bagi masyarakat yang membutuhkan.

- Sistem pakar ini juga diharapkan dapat membantu para mekanik dalam memecahkan permasalahan tentang kerusakan sepeda motor.
- User interface* terbukti mudah digunakan dan user merasa nyaman dalam menggunakannya.
- Rule yang dibuat memiliki tingkat akurasi yang cukup, sehingga dalam sesi konsultasi, kemungkinan seorang pengguna dengan masalahnya dapat didiagnosis dan dapat diberikan saran untuk penyelesaiannya.
- Inference engine bekerja dengan baik, sesuai dengan rule yang telah diprogramkan sebelumnya.

6. DAFTAR PUSTAKA

Kusrini, S.Kom, 2006, *Sistem Pakar Teori Dan Aplikasi*, Andi Yogyakarta,

Mahyunir, Tavid, 1989, *Analisa dan Perancangan Sistem Pengolah Data*, Elexmedia Computindo, Jakarta.

Sutanta Edhy, 2004, *Algoritma Teknik Penyelesaian Permasalahan Untuk Komputasi*, Graha Ilmu, Yogyakarta.

Tim Penerbit Andi, 2003, *Pengembangan Sistem Pakar Menggunakan Menggunakan Visual Basic*, Andi, Yogyakarta.

Widodo Nugroho, 2003, *Pengembangan Sistem Pakar Menggunakan Visual Basic*, Andi Yogyakarta.