

**PERANCANGAN SISTEM PAKAR
UNTUK TROUBLESHOOTING SEPEDA MOTOR
MENGUNAKAN VP-EXPERT**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar sarjana teknik



Disusun oleh :

Prana Fistianduta

Nim 0110623057-62

DEPARTEMEN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN MESIN

MALANG

2007

KATA PENGANTAR

Segala puji syukur kepada Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul **Perancangan Sistem Pakar Untuk Troubleshooting Sepeda Motor dengan Menggunakan VP-Expert** ini dengan baik.

Pada kesempatan ini kami selaku penulis mengucapkan ucapan terima kasih kami kepada semua pihak yang telah memberikan dukungan moral maupun spiritual, diantaranya :

1. Bapak Ir. Bambang Indrayadi, MT selaku Ketua Jurusan Mesin Fakultas Teknik Universitas Brawijaya.
2. Bapak Ir. Djoko Sutikno, M.Eng selaku Sekretaris Jurusan Mesin Fakultas Teknik Universitas Brawijaya.
3. Bapak Ir. Masduki, MM selaku Ketua Kelompok Konsentrasi Teknik Industri Jurusan Mesin Fakultas Teknik Universitas Brawijaya.
4. Bapak Ir. Purnomo Budi Santoso, MSc., Ph.D, selaku Dosen Pembimbing Pertama Jurusan Mesin Fakultas Teknik Universitas Brawijaya.
5. Bapak Nasir Widha Setyanto, ST.,MT, selaku Dosen Pembimbing Kedua Jurusan Mesin Fakultas Teknik Universitas Brawijaya.
6. Seluruh Staf Pengajar Jurusan Teknik Mesin.
7. Seluruh Staf Administrasi Jurusan Mesin Fakultas Teknik Universitas Brawijaya.
8. Semua pihak yang telah mendukung penyelesaian skripsi ini.

Penulis menyadari skripsi ini jauh dari sempurna, oleh karena itu kami selaku penulis mengharapkan saran dan kritik yang membangun.

Malang, Februari 2007

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
ABSTRAK	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	2
1.4 Perumusan Masalah	3
1.5 Tujuan Penelitian	3
1.6 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	4
2.9 Pengantar	4
2.10 Kecerdasan	4
2.2.1 Definisi Kecerdasan Buatan	5
2.2.2 Sejarah Kecerdasan Buatan	5
2.2.3 Perbandingan AI dengan Progam komputer konvensional	6
2.2.4 Bidang Aplikasi Kecerdasan Buatan	7
2.11 Sistem Pakar	10
2.3.1 Definisi Sistem Pakar	10
2.3.2 Keuntungan Sistem Pakar	15
2.3.3 Keterbatasan Sistem Pakar	16
2.3.4 Konsep Umum Sistem Pakar	17
2.12 Struktur Sistem Pakar	18
2.4.1 Basis Pengetahuan (<i>Knowledge Base</i>)	20
2.4.2 <i>Working Memory</i> (Aktif Memori)	20
2.13 Bagaimana Sistem Pakar Melakukan Inferensi ?	20
2.5.1 Sistem Perantain Maju (<i>Forward Chaining Systems</i>)	21

2.5.2	Strategi Penyelesaian Konflik (<i>Conflict Resolution Strategy</i>)	23
2.5.3	Sistem Perantaraan Balik (<i>Backward Chaining Systems</i>)	24
2.5.4	Pemilihan Sistem Inferensi	25
2.5.5	Representasi Pengetahuan	27
2.5.6	Kaidah Produksi, Pengetahuan dan Kaidah Inferensi	30
2.5.7	Keuntungan Sistem Produksi	33
2.6	Akuisisi Pengetahuan (<i>Knowledge Acquisition</i>)	34
2.7	Pengetahuan Formal (<i>Formal Knowledge</i>)	34
2.7.1	Engine Berputar Secara Kasar	35
2.7.2	Karburator	36
2.8	Pengetahuan Pakar (<i>Expert Knowledge</i>)	39
2.8.1	Engine	40
2.8.2	Memeriksa engine Akan Suara-suara Abnormal dan Cara-cara Memperbaikinya	42
2.9	Alat Pengembangan Sistem Pakar	43
2.10	Pengantar <i>VP-Expert</i>	43
2.11	Fasilitas <i>VP-Expert</i>	44
2.12	Penggunaan <i>VP-Expert</i>	46
2.13	Struktur Utama <i>VP-Expert</i>	47
BAB III METODOLOGI PENELITIAN		52
3.1	Pendahuluan	52
3.2	Studi Lapangan	52
3.3	Identifikasi Masalah	52
3.4	Tujuan Perancangan	52
3.5	Studi Literatur	53
3.6	Metode Perancangan Expert System	53
3.7	Kesimpulan dan Saran	53
BAB IV PERANCANGAN SISTEM PAKAR		56
4.1	Pengantar	56
4.2	Perancangan Sistem Pakar	56
4.2.1	Perancangan Blok Diagram Domain Pengetahuan	56

4.2.2	Pembentukan <i>Dependency Diagram</i>	56
4.2.3	Perancangan <i>Decision Table</i>	56
4.2.4	Perancangan Aturan (<i>Rules</i>)	57
4.2.5	Perancangan <i>User Interface (UI)</i>	85
4.2.6	Pemasukkan Rancangan ke <i>VP-Expert</i>	91
4.3	Pengujian Sistem	94
4.3.1	Verifikasi	95
4.3.2	Validasi	98
BAB V	101
5.1	Kesimpulan	101
5.2	Saran	101

DAFTAR PUSTAKA

LAMPIRAN



ABSTRAK

Prana Fistianduta, 2001. Perancangan Sistem Pakar Untuk Troubleshooting Sepeda Motor Menggunakan VP-Expert. Skripsi Jurusan Mesin Fakultas Teknik Universitas Brawijaya Malang.

Dosen Pembimbing : I. Ir. Purnomo Budi Santoso, MSc., Ph.D

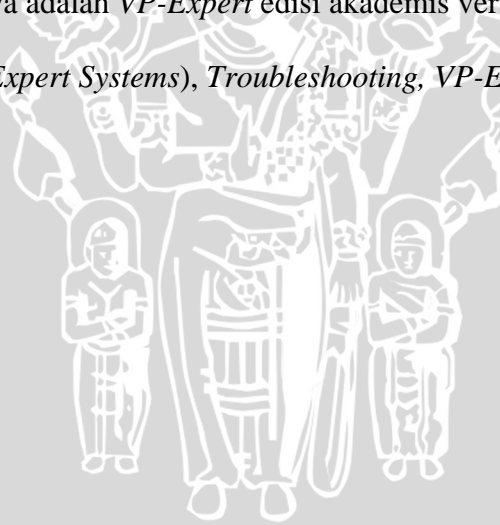
II. Nasir Widha Setyanto, ST.,MT

Sepeda motor adalah kendaraan yang paling diminati saat ini, namun secara umum para pemilik sepeda motor masih awam akan pengetahuan atau kepakaran dibidang kerusakan kendaraan mereka. Disamping itu, pakar kerusakan sepeda motor masih sedikit dan biaya untuk mengupah para pakar ini masih relatif tinggi.

Salah satu bidang terapan AI yang tertua adalah sistem pakar (SP, *Expert System*). SP adalah program aplikasi yang bertindak seperti layaknya seorang pakar (*Expert*). Dengan menggunakan pengetahuan yang disimpan dalam basis pengetahuan (*Knowledge Base*) dan mesin penalar (*Inference Engine*), seorang pemakai yang tidak berpengalaman sekalipun dapat memecahkan suatu masalah sulit dan bisa mengambil keputusan yang tepat dan akurat seperti yang dilakukan oleh seorang pakar. SP adalah alat untuk memproduksi kepakaran di suatu bidang secara masal.

Dalam skripsi ini dirancang dan dibuat SP untuk *troubleshooting* (pelacakan penyebab kerusakan sepeda motor dan cara mengatasinya) untuk membantu dan mempermudah para pemilik sepeda motor dalam mengetahui kerusakan sepeda motor mereka. Alat pengembangnya adalah *VP-Expert* edisi akademis versi 2.0.

Kata Kunci: Sistem Pakar (*Expert Systems*), *Troubleshooting*, *VP-Expert*



BAB I PENDAHULUAN

1.1 Latar Belakang

Sepeda motor merupakan jenis kendaraan bermotor yang paling digemari saat ini. Apalagi dengan dicabutnya subsidi BBM, maka banyak orang yang gajinya pas-pasan lalu beralih dari mobil ke sepeda motor. Selain itu, ada sementara orang yang membeli sepeda motor untuk mencari uang, misalnya bekerja sebagai tukang ojek. Walau pemilik sepeda motor begitu menjamur, namun sayang sekali masih banyak pemilik kendaraan kurang memahami atau bahkan tidak mengerti permasalahan yang berhubungan dengan kerusakan sepeda motor.

Merk sepeda motor sangat banyak, misalnya Honda, Yamaha, Suzuki, dan seterusnya. Setiap merk mempunyai “Penyakit Khas” yang berbeda-beda. Penyakit atau kerusakan sepeda motor antara lain disebabkan oleh sistem listrik, sistem pengapian, sistem transmisi, maupun mesinnya sendiri. Dari pengamatan sepintas, nampak bahwa pakar dibidang kerusakan sepeda motor jumlahnya kurang memadai dengan jumlah pemilik sepeda motor dan biaya perawatan serta perbaikan kerusakan sepeda motor masih termasuk tinggi bagi sebagian besar masyarakat Indonesia.

Untunglah saat ini manusia hidup dalam jaman digital yang antara lain ditandai dengan banyaknya program-program aplikasi yang sangat canggih yang mampu membantu manusia dalam menyelesaikan pekerjaan. Salah satu bidang teknologi informasi yang sedang populer dan berkembang pesat sekali adalah bidang *Artificial Intelligence* (AI, kecerdasan buatan). AI merupakan sub bidang pengetahuan komputer yang berusaha agar program komputer atau mesin dapat menirukan kecerdasan otak manusia, dengan demikian diharapkan komputer tersebut dapat membantu manusia dalam memecahkan berbagai masalah. Salah satu bidang terapan AI yang tertua adalah sistem pakar (SP, *Expert System*). SP adalah program aplikasi yang bertindak seperti layaknya seorang pakar (*Expert*). Dengan menggunakan pengetahuan yang disimpan dalam basis pengetahuan (*Knowledge Base*) dan mesin penalar (*Inference Engine*), seorang pemakai yang tidak berpengalaman sekalipun dapat memecahkan suatu masalah sulit dan bisa mengambil keputusan yang tepat dan akurat seperti yang dilakukan oleh seorang pakar. Berbagai aplikasi berbasis SP telah dikembangkan terutama untuk mengatasi keterbatasan dan mahalannya kepakaran alami

(manusia). Jargon, seorang ahli dibidang komputer mengatakan bahwa SP merupakan produksi massal akan suatu kepakaran.

Kelebihan dari SP adalah keandalannya yang dapat bekerja secara terus menerus (24 jam sehari), dibandingkan dengan seorang ahli yang bekerja dengan jangka waktu yang terbatas. Disamping itu untuk menghasilkan kepakaran dibidang tertentu akan jauh lebih mudah dan cepat sebab hanya perlu mengcopy program yang sama dan dijalankan dikomputer lain, sedangkan untuk menghasilkan seorang ahli atau pakar dibutuhkan waktu yang relatif panjang dan biaya yang tidak sedikit. Ada berbagai alat yang dapat digunakan untuk mengembangkan SP, misalnya : dengan bahasa konvensional seperti *Pascal*, *Delphi*, *VB*, dan sebagainya, dengan bahasa AI seperti *Prolog* dan *LISP*, atau dengan *shell* seperti *VP-Expert*.

Mengingat semua faktor diatas maka dirasa perlu untuk merancang dan mengembangkan suatu SP yang dapat mengadopsi kemampuan para pakar dalam melakukan *Troubleshooting* (pelacakan kerusakan) sepeda motor untuk membantu dan mempermudah para pemilik sepeda motor dalam mengetahui kerusakan komponen sepeda motor mereka.

1.2 Identifikasi Masalah

Atas dasar latar belakang diatas maka dapat diidentifikasi masalah sebagai berikut:

1. Sepeda motor adalah kendaraan yang paling diminati saat ini, namun secara umum para pemilik sepeda motor masih awam akan pengetahuan atau kepakaran dibidang kerusakan kendaraan mereka.
2. Merk sepeda motor bermacam-macam, demikian pula jenis kerusakan komponen dari sepeda motor juga sangat bervariasi.
3. Pakar kerusakan sepeda motor masih sedikit dan biaya untuk mengupah para pakar ini masih relatif tinggi.

1.3 Batasan Masalah

Masalah perlu dibatasi agar kedalaman, keluasan dan ruang lingkup masalah sesuai dengan waktu yang tersedia dan tingkat riset yang dituju, dalam hal ini adalah tingkat riset untuk tugas akhir level S1 dengan waktu penyelesaian satu semester. Batasan masalah yang perlu dilakukan dalam skripsi ini adalah mencakup hal-hal sebagai berikut :

1. Merk sepeda motor yang diteliti adalah *YAMAHA FI-ZR*. Tahun pembuatan 2002.
2. Jenis kerusakan difokuskan pada karburator, engine, & suara-suara abnormal.

3. Perancangan SP tidak menggunakan *Certainty Factors* (Faktor Kepastian).
4. Perancangan SP sebatas *Prototype* dan menggunakan alat pengembang aplikasi yang disebut *VP-Expert* versi akademik.

1.4 Perumusan Masalah

Atas dasar pembatasan masalah diatas, maka perlu dirumuskan masalah yang sudah manageable (mampu ditangani) dan sudah lebih bersifat spesifik sebagai berikut :

“Bagaimana mengembangkan SP dengan menggunakan *VP-Expert* untuk membantu para pemilik sepeda motor Yamaha dalam melakukan *Troubleshooting* kerusakan sepeda motor mereka ?”

1.5 Tujuan Penelitian

Tujuan bertujuan untuk menjawab rumusan masalah, untuk mencapai tujuan seringkali harus melalui “tujuan antara” atau langkah-langkah yang harus dilalui demi mencapai tujuan utama / akhir. Mengingat hal ini, tujuan dari Tugas Akhir ini disusun atas dasar langkah-langkah sebagai berikut :

1. Memperoleh pengetahuan (*Knowledge Acquisition*) tentang kerusakan sepeda Yamaha.
2. Menyusun aturan-aturan (*Rules*) dalam bidang *troubleshooting* kerusakan sepeda motor yang akan dimasukkan kedalam SP
3. Merancang SP dalam level *prototype* dengan menggunakan *VP-Expert* dalam 2 komponen utama, yaitu :
 - *Rules* (aturan)
 - *User Interface*
4. Membuat manual kecil untuk menjalankan *prototype*.

1.6 Manfaat Penelitian

Manfaat yang diharapkan dan diperoleh dari perancangan Tugas Akhir ini adalah:

1. Mempelajari masalah kerusakan sepeda motor (*Otomotive*).
2. Belajar mengembangkan program *ES* walaupun dalam bentuk *prototype*.
3. *ES* yang dikembangkan dapat membantu para pemilik sepeda motor untuk *troubleshooting* kerusakan kendaraan mereka.
4. Membudayakan rekayasa *software* dalam negeri agar terbebas dari ketergantungan (“Penjajahan”) *software* luar negeri yang sangat mahal.

BAB II

TINJAUAN PUSTAKA

2.1 Pengantar

Di masa kini, Kecerdasan Buatan (*Artificial Intelligence/AI*) telah menjadi wacana umum yang sangat penting dan jamak dijumpai. Namun masih banyak menyisakan pertanyaan skeptis tentang “Mesin Berfikir”: “Betulkah sebuah mesin dapat benar-benar berfikir dengan dirinya sendiri?”, atau “Jika benar-benar dapat berfikir sendiri, apakah proses berfikirnya sama dengan kita?”, dan “Seberapa handal?”. Sebagai salah satu cabang aplikasi AI, Sistem Pakar (SP) adalah cabang yang tertua dan banyak dimanfaatkan untuk meringankan tugas manusia.

Landasan teori disusun untuk memberikan dasar pengetahuan dalam perancangan sistem pakar (SP). Adapun topik-topik yang akan dibahas dalam landasan teori ini adalah sebagai berikut :

1. *Artificial Intelligence* (AI) sebagai induk dari SP.
2. Sistem pakar (*Expert System*).
3. Metode Pengembangan SP.
4. Alat Pengembangan SP yaitu *VP-Expert*.

2.2 Kecerdasan

Dari kamus, arti kecerdasan adalah : kemampuan untuk mengerti atau memahami (*The Faculty Of Understanding*). Perilaku cerdas dapat ditandai dengan:

1. Belajar atau mengerti dari pengalaman.
2. Memecahkan hal yang bersifat mendua (*Kontradiktif*).
3. Merespon situasi baru dengan cepat (*Fleksibel*).
4. Menggunakan alasan untuk memecahkan problem secara efektif.
5. Berurusan dengan situasi yang membingungkan.
6. Memahami dengan cara biasa (*Rasional*).
7. Menerapkan pengetahuan untuk memanipulasi lingkungan.
8. Mengenali elemen penting pada suatu situasi.

Sebuah ujian yang dapat dilakukan untuk menentukan apakah sebuah Komputer atau mesin menunjukkan perilaku cerdas didesain oleh Alan Turing. Tes Turing menyatakan sebuah mesin dikatakan pintar hanya apabila seorang pewawancara (manusia) yang berbicara dengan orang lain dan mesin yang keduanya tidak terlihat olehnya, tidak mampu menentukan mana yang manusia dan

mana yang mesin, meskipun dia telah berulang-ulang melontarkan pertanyaan yang sama.

2.2.1 Definisi Kecerdasan Buatan

Tidak ada kesepakatan mengenai definisi Kecerdasan Buatan, beberapa definisi KB antara lain adalah:

1. Sebuah studi tentang bagaimana membuat komputer mengerjakan sesuatu yang dapat dikerjakan manusia (Rich, 1991).
2. Cabang ilmu komputer yang mempelajari otomatisasi tingkah laku cerdas (Setiawan, 1993).
3. Suatu perilaku sebuah mesin yang jika dikerjakan oleh manusia akan disebut cerdas (Turing, et. al, 1996).

Kebanyakan ahli setuju bahwa Kecerdasan Buatan berhubungan dengan 2 ide dasar. Pertama, menyangkut studi proses berfikir manusia, dan kedua, berhubungan dengan merepresentasikan proses tersebut melalui mesin (komputer, robot, dll)

Kemampuan untuk problem solving adalah salah satu cara untuk mengukur kecerdasan dalam berbagai konteks. Terlihat di sini bahwa mesin cerdas akan diragukan untuk dapat melayani keperluan khusus jika tidak mampu menangani permasalahan remeh atau kecil yang biasa dikerjakan orang secara rutin. Terdapat beberapa alasan untuk memodelkan performa manusia dalam hal ini:

1. Untuk menguji teori *psikologis* dari performa manusia.
2. Untuk membuat komputer dapat memahami penalaran (*Reasoning*) manusia.
3. Untuk membuat manusia dapat memahami penalaran komputer.
4. Untuk mengeksplorasi pengetahuan apa yang dapat diambil dari manusia.

Menurut Winston dan Prendergast (1984), tujuan dari Kecerdasan Buatan adalah :

1. Membuat mesin menjadi lebih pintar.
2. Memahami apakah kecerdasan (*Intelligence*) itu.
3. Membuat mesin menjadi lebih berguna.

2.2.2 Sejarah Kecerdasan Buatan

Di awal abad 20, seorang penemu Spanyol, Torres Y Quevedo, membuat sebuah mesin yang dapat men'skak-mat' raja lawannya dengan sebuah ratu dan raja. Perkembangan secara sistematis kemudian dimulai segera setelah diketemukannya komputer digital. Artikel ilmiah pertama tentang Kecerdasan Buatan ditulis oleh Alan Turing pada tahun 1950, dan kelompok riset pertama dibentuk tahun 1954 di Carnegie Mellon University oleh Allen Newell dan Herbert Simon. Namun bidang Kecerdasan

Buatan baru dianggap sebagai bidang tersendiri di konferensi Dartmouth tahun 1956, di mana 10 peneliti muda memimpikan mempergunakan komputer untuk memodelkan bagaimana cara berfikir manusia. Hipotesis mereka adalah: “Mekanisme berfikir manusia dapat secara tepat dimodelkan dan disimulasikan pada komputer digital”, dan ini yang menjadi landasan dasar Kecerdasan Buatan.

2.2.3 Perbandingan AI dengan Program Komputer Konvensional

Program komputer konvensional prosesnya berbasis algoritma, yakni formula matematis atau prosedur sekuensial yang mengarah kepada suatu solusi. Algoritma tersebut dikonversi ke program komputer yang memberitahu komputer secara pasti instruksi apa yang harus dikerjakan. Algoritma yang dipakai kemudian menggunakan data seperti angka, huruf, atau kata untuk menyelesaikan masalah.

Perangkat lunak AI berbasis representasi serta manipulasi simbolik. Di sini simbol tersebut berupa huruf, kata, atau angka yang merepresentasikan obyek, proses dan hubungan keduanya. Sebuah obyek bisa jadi seorang manusia, benda, pikiran, konsep, kejadian, atau pernyataan suatu fakta. Menggunakan simbol, kita dapat menciptakan basis pengetahuan yang berisi fakta, konsep, dan hubungan di antara keduanya. Kemudian beberapa proses dapat digunakan untuk memanipulasi simbol tersebut untuk menghasilkan nasehat atau rekomendasi untuk penyelesaian suatu masalah. Perbedaan dasar antara AI dengan program komputer konvensional diberikan dalam Tabel 2.1.



Tabel 2.1 Perbandingan Sistem Konvensional dan Sistem Pakar
 Sumber : Muhammad Arhami. *Konsep Dasar Sistem Pakar*. Yogyakarta. Penerbit ANDI, 2005, halaman 8.

Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesan umumnya digabung dalam satu program sekuensial	Basis pengetahuan dari mekanisme pemrosesan (<i>Inferensi</i>)
Program tidak pernah salah (kecuali pemrogramnya yang salah)	Program bisa saja melakukan kesalahan
Tidak menjelaskan mengapa input dibutuhkan atau bagaimana hasil yang diperoleh	Penjelasan (<i>Explanation</i>) merupakan bagian dari sistem pakar
Mebutuhkan semua input data	Tidak harus membutuhkan semua input data atau fakta
Perubahan pada program merepotkan	Perubahan pada kaidah dapat dilakukan dengan mudah
Sistem bekerja jika sudah lengkap	Sistem dapat bekerja hanya dengan kaidah yang sedikit
Eksekusi secara <i>Algoritmik (Step by step)</i>	Eksekusi dilakukan secara <i>heuristik dan logis</i>
Manipulasi efektif pada database yang besar	Manipulasi efektif pada basis pengetahuan yang besar
Efisiensi adalah tujuan utama	Efektivitas adalah tujuan utama
Data kuantitatif	Data kualitatif
Representasi dalam numeric	Representasi pengetahuan dalam simbolik
Menangkap, menambah dan mendistribusi data <i>Numeric</i> atau informasi	Menangkap, menambah dan mendistribusi pertimbangan (<i>Judgment</i>) dan pengetahuan

2.2.4 Bidang Aplikasi Kecerdasan Buatan

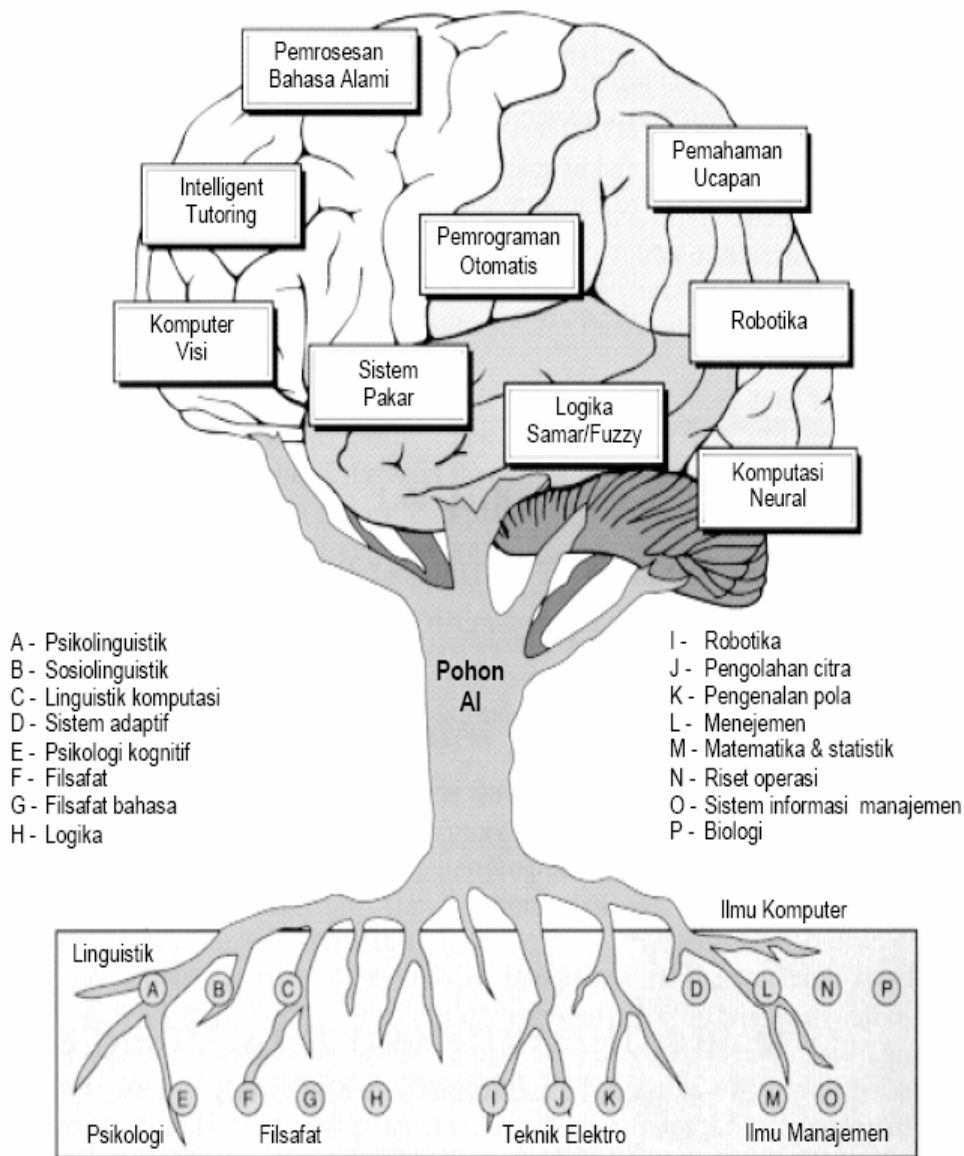
Penerapan Kecerdasan Buatan meliputi berbagai bidang seperti ditunjukkan pada bagian akar pohon AI dalam Gambar 2.1 antara lain : Bahasa (*Linguistik*), Psikologi, Filsafat, Teknik Elektro, Ilmu Komputer, dan Ilmu Manajemen. Sedangkan sistem cerdas yang banyak dikembangkan saat ini adalah:

Sistem Pakar (*Expert System*), yaitu program konsultasi (*Advisory*) yang mencoba menirukan proses penalaran seorang pakar/ahli dalam memecahkan masalah yang rumit. Sistem Pakar merupakan aplikasi AI yang paling banyak. Lebih detail tentang Sistem Pakar akan diberikan dalam bab berikutnya.

Pemrosesan Bahasa Alami (*Natural Language Processing*), yang memberi kemampuan pengguna komputer untuk berkomunikasi dengan komputer dalam bahasa mereka sendiri (bahasa manusia). Sehingga komunikasi dapat dilakukan

dengan cara percakapan alih-alih menggunakan perintah yang biasa digunakan dalam bahasa komputer biasa. Bidang ini dibagi 2 lagi :

- a. Pemahaman bahasa alami, yang mempelajari metode yang memungkinkan komputer mengerti perintah yang diberikan dalam bahasa manusia biasa. Dengan kata lain, komputer dapat memahami manusia.
- b. Pembangkitan bahasa alami, sering disebut juga sintesa suara, yang membuat komputer dapat membangkitkan bahasa manusia biasa sehingga manusia dapat memahami komputer secara mudah.



Gambar 2.1 Pohon Kecerdasan Buatan dan Aplikasi Utamanya
 Sumber : <http://balzach.staff.ugm.ac.id/AI/Diktat%20Kecerdasan%20Buatan.pdf>

Pemahaman Ucapan/Suara (*Speech/Voice Understanding*), adalah teknik agar komputer dapat mengenali dan memahami bahasa ucapan. Proses ini memungkinkan seseorang berkomunikasi dengan komputer dengan cara berbicara kepadanya. Istilah “Pengenalan Suara” mengandung arti bahwa tujuan utamanya adalah mengenai kata yang diucapkan tanpa harus tahu artinya, di mana bagian itu merupakan tugas “Pemahaman Suara”. Secara umum prosesnya adalah usaha untuk menerjemahkan apa yang diucapkan seorang manusia menjadi kata-kata atau kalimat yang dapat dimengerti oleh komputer.

Sistem Sensor dan Robotika. Sistem sensor, seperti sistem visi dan pencitraan, serta sistem pengolahan sinyal, merupakan bagian dari robotika. Sebuah robot, yaitu perangkat elektromekanik yang diprogram untuk melakukan tugas manual, tidak semuanya merupakan bagian dari AI. Robot yang hanya melakukan aksi yang telah diprogramkan dikatakan sebagai robot bodoh yang tidak lebih pintar dari lift. Robot yang cerdas biasanya mempunyai perangkat sensor, seperti kamera, yang mengumpulkan informasi mengenai operasi dan lingkungannya. Kemudian bagian AI robot tersebut menerjemahkan informasi tadi dan merespon serta beradaptasi jika terjadi perubahan lingkungan.

Komputer Visi, merupakan kombinasi dari pencitraan, pengolahan citra, pengenalan pola serta proses pengambilan keputusan. Tujuan utama dari komputer visi adalah untuk menerjemahkan suatu pemandangan. Komputer visi banyak dipakai dalam kendali kualitas produk industri.

Intelligent Tutoring/Intelligent Computer Aided Instruction, adalah komputer yang mengajari manusia. Belajar melalui komputer sudah lama digunakan, namun dengan menambahkan aspek kecerdasan di dalamnya, dapat tercipta komputer “Guru” yang dapat mengatur teknik pengajarannya untuk menyesuaikan dengan kebutuhan “Murid” secara individual. Sistem ini juga mendukung pembelajaran bagi orang yang mempunyai kekurangan fisik atau kelemahan belajar.

Mesin Belajar (*Machine Learning*), yang berhubungan dengan sekumpulan metode untuk mencoba mengajari atau melatih komputer untuk memecahkan masalah atau mendukung usaha pemecahan masalah dengan menganalisa kasus-kasus yang telah terjadi. Contoh dua metode mesin belajar yang paling populer adalah komputasi neural dan logika samar.

Aplikasi lain dari AI misalnya untuk merangkum berita, pemrograman komputer secara otomatis, atau menerjemahkan dari suatu bahasa ke bahasa yang lain,

serta aplikasi dalam permainan (Ingat pertandingan catur antara Grand Master Anatoly Karpov dengan komputer Deep Thought dari IBM).

2.3 Sistem Pakar

2.3.1 Definisi Sistem Pakar

Ketika hendak membuat suatu keputusan yang kompleks atau memecahkan masalah, seringkali kita meminta nasehat atau berkonsultasi dengan seorang pakar atau ahli. Seorang pakar adalah seseorang yang mempunyai pengetahuan dan pengalaman spesifik dalam suatu bidang; misalnya pakar komputer, pakar uji tak merusak, pakar politik dan lain-lain. Semakin tidak terstruktur situasinya, semakin mengkhusus (dan mahal) konsultasi yang dibutuhkan.

Sistem Pakar (*Expert System*) adalah usaha untuk menirukan seorang pakar. Biasanya Sistem Pakar berupa perangkat lunak pengambil keputusan yang mampu mencapai tingkat performa yang sebanding seorang pakar dalam bidang problem yang khusus dan sempit. Ide dasarnya adalah : kepakaran ditransfer dari seorang pakar (atau sumber kepakaran yang lain) ke komputer, pengetahuan yang ada disimpan dalam komputer, dan pengguna dapat berkonsultasi pada komputer itu untuk suatu nasehat, lalu komputer dapat mengambil inferensi (menyimpulkan, mendeduksi, dll.) seperti layaknya seorang pakar, kemudian menjelaskannya ke pengguna tersebut, bila perlu dengan alasan-alasannya. Sistem Pakar malahan terkadang lebih baik unjuk kerjanya daripada seorang pakar manusia.

Professor Edward Feigenbaum dari Universitas Stanford yang merupakan seorang pelopor awal dari teknologi sistem pakar, yang mendefinisikan sistem pakar sebagai "... suatu program komputer cerdas yang menggunakan *knowledge* (Pengetahuan) dan prosedur inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang yang ahli untuk menyelesaikannya. (Feigenbaum:1982). Suatu sistem adalah suatu sistem komputer yang menyamai (*Emulates*) kemampuan pengambilan keputusan dari seorang pakar. Istilah *Emulates* berarti bahwa sistem pakar diharapkan dapat bekerja dalam semua hal seperti seorang pakar. Suatu emulasi jauh lebih kuat daripada suatu emulasi jauh lebih kuat daripada suatu simulasi yang hanya membutuhkan sesuatu yang bersifat nyata dalam beberapa bidang atau hal.

Definisi sistem pakar menurut John Durkin (1992:7) adalah sebagai berikut :
“ *Expert system is a computer program designed to model the problem solving ability of a human expert* ”. Jadi Sistem Pakar adalah suatu program komputer yang

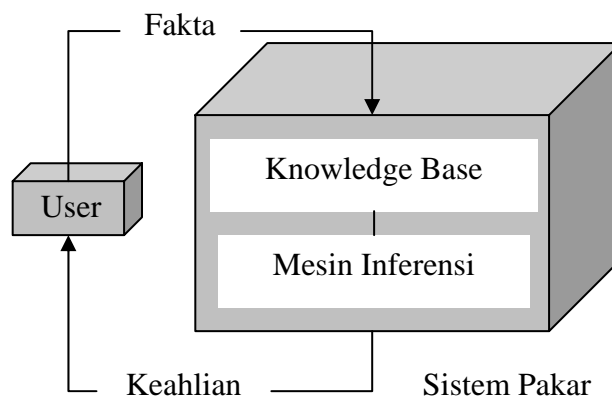
dirancang untuk memodelkan kemampuan menyelesaikan masalah dari seorang pakar.

Sedangkan definisi Sistem Pakar menurut Dr. Suyoto (2004:181) adalah sebagai berikut : “ Sistem Pakar adalah system yang didesain dan diimplementasikan dengan bantuan bahasa pemrograman tertentu untuk dapat menyelesaikan masalah seperti yang dilakukan oleh para ahli ”.

Sistem pakar adalah salah satu cabang AI yang membuat penggunaan secara luas *Knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *Knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya. Ketika sistem pakar dikembangkan pertama kali sekitar tahun 70-an sistem pakar hanya berisi *knowledge* yang *eksklusif*. Namun demikian sekarang ini istilah sistem pakar sudah digunakan untuk berbagai macam sistem yang menggunakan teknologi sistem pakar, program dan perangkat keras yang dirancang untuk membantu pengembangan dan pembuatan sistem pakar.

Knowledge dalam sistem pakar mungkin saja seorang ahli, atau *knowledge* yang umumnya terdapat dalam buku, majalah dan orang yang mempunyai pengetahuan tentang suatu bidang. Istilah sistem pakar, sistem *knowledge base*, atau sistem pakar *knowledge base*, sering digunakan dengan arti yang sama. Kebanyakan orang menggunakan istilah sistem pakar karena lebih singkat, bahkan waktu belum benar-benar pakar, hanya menggunakan *knowledge* secara umum.

Gambar 2.2 menggambarkan konsep dasar suatu sistem pakar *knowledge base*. Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Bagian dalam sistem pakar terdiri dari 2 komponen utama, yaitu *knowledge base* yang berisi *knowledge* dan mesin inferensi yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan *respons* dari sistem pakar atas permintaan pengguna.



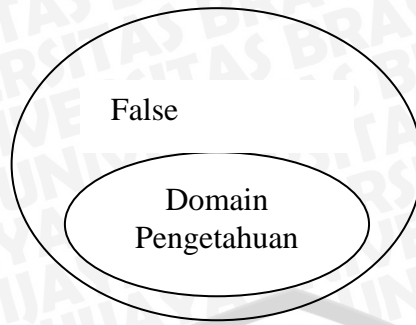
Gambar 2.2 Konsep Dasar Fungsi Sistem Pakar

Sumber : Muhammad Arhami. *Konsep Dasar Sistem Pakar*. Yogyakarta. Penerbit ANDI, 2005.halaman 4.

Penggunaan sistem *knowledge base* (Basis Pengetahuan) juga dirancang untuk aksi pemandu cerdas seorang ahli. Pemandu cerdas dirancang dengan teknologi sistem pakar karena memberikan banyak keuntungan terhadap pengembangannya. Semakin banyak *knowledge* yang ditambahkan untuk pemandu cerdas maka sistem tersebut akan semakin baik dalam bertindak sehingga semakin menyerupai pakar sebenarnya. Pengembangan suatu pemandu cerdas merupakan permulaan bagi pengembangan sistem pakar yang lebih rangkap lagi.

Suatu *knowledge* dari sistem pakar bersifat khusus untuk suatu domain masalah saja. Domain masalah adalah bidang atau ruang lingkup yang khusus, seperti kedokteran, keuangan, bisnis, ilmu pengetahuan atau teknik. Sistem pakar menyerupai kepakaran manusia yang secara umum dirancang untuk menjadi pakar dalam suatu domain masalah saja.

Knowledge dari sistem pakar tentang penyelesaian masalah yang khusus disebut dengan *domain knowledge* dari suatu pakar. Sebagai contoh, sistem pakar kedokteran yang dirancang untuk mendiagnosis infeksi penyakit akan mempunyai suatu uraian *knowledge* tentang gejala-gejala penyakit yang disebabkan oleh infeksi penyakit. Dalam kasus ini domain *knowledge*-nya adalah bidang kedokteran yang terdiri dari *knowledge* tentang penyakit, gejala, dan cara pengobatan. Gambar 2.3 menggambarkan hubungan antara domain masalah dan domain *knowledge*. Dapat dilihat juga bahwa *domain knowledge* secara keseluruhan merupakan bagian dari domain masalah.



Gambar 2.3 Hubungan Antara Domain Knowledge dan Domain Masalah
 Sumber : Muhammad Arhami. *Konsep Dasar Sistem Pakar*. Yogyakarta. Penerbit ANDI, 2005.halaman 5.

Sistem pakar sebagai suatu teknologi baru masih menyimpan hal-hal baru yang dapat dipelajari. Tabel 2.2 meringkaskan gambaran perbedaan orang-orang yang terlibat di dalam suatu teknologi. Pada tabel ini *technologist* mungkin seorang insinyur atau perancang perangkat lunak sementara teknologi mungkin berupa perangkat keras dan perangkat lunak. Dalam menyelesaikan suatu permasalahan, ada beberapa pertanyaan yang perlu dijawab atau teknologi tidak akan berhasil digunakan. Seperti juga *tool* lainnya, sistem pakar mempunyai aplikasi yang sesuai dan juga tidak sesuai untuk digunakan.

Tabel 2.2 Perbandingan Kemampuan Seorang Pakar dengan Sistem Pakar
 Sumber : Muhammad Arhami. *Konsep Dasar Sistem Pakar*. Yogyakarta. Penerbit ANDI, 2005, halaman 6.

Factor	Human Expert	Expert System
Time Availability	Hari kerja	Setiap saat
Geografis	Local / tertentu	Dimana saja
Keamanan	Tidak tergantikan	Dapat diganti
Dapat habis (<i>Perishable</i>)	Ya	Tidak
Performansi	Variable	Konsisten
Kecepatan	Variable	Konsisten
Biaya	Tinggi	Terjangkau

Dari tabel diatas dapat dikembangkan penjelasan lebih lanjut tentang keunggulan sistem pakar dibanding seorang pakar, yaitu :

1. Sistem pakar bisa digunakan setiap hari menyerupai sebuah mesin sedangkan seorang pakar tidak mungkin bekerja terus menerus setiap hari tanpa istirahat.
2. Sistem pakar merupakan suatu software yang dapat diperbanyak dan kemudian dibagikan ke berbagai lokasi maupun tempat yang berbeda-beda untuk digunakan, sedangkan seorang pakar hanya bekerja pada satu tempat dan pada saat yang bersamaan.

3. Suatu sistem pakar dapat diberi pengamanan untuk menentukan siapa saja yang mempunyai hak akses untuk mengguankannya dan jawaban yang diberikan oleh sistem terbebas dari proses *Intimidasi* (Ancaman), sedangkan seorang pakar 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. Walaupun pengetahuan yang dimilinya dalam waktu yang singkat tidak akan hilang, akan tetapi bisa pindah tugas atau dipecah dari pekerjaannya sehingga organisasi yang mempekerjakannya akan kehilangan seorang pakar yang berbakat.
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 dapat dipengaruhi oleh faktor-faktor luar seperti yang disebutkan ketika sedang menyelesaikan 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 udah 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).

Ada beberapa alasan mendasar mengapa sistem pakar dikembangkan untuk menggantikan seorang pakar, diantaranya :

1. Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Seorang pakar adalah mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat (*Hostile Environment*).

Kepakaran (*Expertise*) adalah pengetahuan yang meluas (*Ekstensif*) dan spesifik yang diperoleh melalui rangkaian pelatihan, membaca, dan pengalaman. Pengetahuan membuat pakar dapat mengambil keputusan secara lebih baik dan lebih

cepat daripada non-pakar dalam memecahkan problem yang kompleks. Kepakaran mempunyai sifat berjenjang, pakar top memiliki pengetahuan lebih banyak daripada pakar junior.

Tujuan Sistem Pakar adalah untuk mentransfer kepakaran dari seorang pakar ke komputer, kemudian ke orang lain (yang bukan pakar). Proses ini tercakup dalam rekayasa pengetahuan (*Knowledge Engineering*) yang akan dibahas kemudian. Aktivitas yang dilakukan untuk memindahkan kepakaran adalah :

1. *Knowledge Acquisition* (dari pakar atau sumber lainnya)
2. *Knowledge Representation* (ke dalam komputer)
3. *Knowledge Inferencing*
4. *Knowledge Transferring*

2.3.2 Keuntungan Sistem Pakar

Sistem pakar (*Expert System*) merupakan paket perangkat lunak atau paket program komputer yang ditujkan sebagai penyedia nasihat dan sarana bantu dalam memecahkan masalah di bidang-bidang spesialisasi tertentu seperti sains, rekayasa, matematika, kedokteran, pendidikan dan sebagainya. Sistem pakar merupakan subset dari *Artificial Intelligence*.

Ada beberapa keunggulan sistem pakar, diantaranya dapat :

1. Menghimpun data dalam jumlah yang sangat besar.
2. Menyimpan data tersebut untuk jangka waktu yang panjang dalam suatu bentuk tertentu.
3. mengerjakan perhitungan secara cepat dan tepat dan tanpa jemu mencari kembali data yang tersimpan dengan kecepatan tinggi.

Sementara kemampuan sistem pakar diantaranya adalah :

1. Menjawab berbagai pertanyaan yang menyangkut bidang keahliannya.
2. Bila diperlukan dapat menyajikan asumsi dan alur penalaran yang digunakan untuk sampai kejawaban yang dikehendaki.
3. Menambah fakta kaidah dan alur penalaran sah yang baru ke dalam otaknya.

Selanjutnya keuntungan bila menggunakan sistem pakar, diantaranya adalah :

1. Menjadikan pengetahuan dan nasihat lebih mudah didapat.
2. Meningkatkan *output* dan *produktivitas*.
3. Menyimpan kemampuan dan keahlian pakar.
4. Meningkatkan penyelesaian masalah menerusi paduan pakar, penerangan, sistem pakar khas.

5. Meningkatkan *reliabilitas*.
6. Memberikan jawaban (*respons*) yang cepat.
7. Merupakan panduan yang cerdas (*Intelligence*).
8. Dapat bekerja dengan informasi yang kurang lengkap dan mengandung ketidakpastian.
9. Basis Data Cerdas (*Intelligence Database*), bahwa sistem dapat digunakan untuk mengakses basis data dengan cara cerdas (Kerschberg : 86, Schur : 88).

Selain keuntungan-keuntungan diatas, sistem pakar seperti halnya sistem lainnya, juga memiliki kelemahan, diantaranya adalah :

1. Masalah dalam mendapatkan pengetahuan di mana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, dan walaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda.
2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharannya.
3. Boleh jadi sistem tak dapat membuat keputusan.
4. Sistem pakar tidaklah 100% menguntungkan, walaupun seorang tetap tidak sempurna atau tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan.

Dalam hal ini peran manusia tetap merupakan faktor dominan.

Kelemahan-kelemahan atau kekurangan dari sistem pakar tersebut bukanlah sama sekali tidak bisa diatasi, tetapi dengan terus melakukan perbaikan dan pengolahan berdasarkan pengalaman yang telah ada maka hal itu diyakini akan dapat diatasi, walaupun dalam waktu yang panjang dan terus menerus.

2.3.3 Keterbatasan Sistem Pakar

Metodologi Sistem Pakar yang ada tidak selalu mudah, sederhana dan efektif.

Berikut adalah keterbatasan yang menghambat perkembangan Sistem Pakar:

1. Pengetahuan yang hendak diambil tidak selalu tersedia.
2. Kepakaran sangat sulit diekstrak dari manusia.
3. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.
4. Adalah sangat sulit bagi seorang pakar untuk mengabstraksi atau menjelaskan langkah mereka dalam menangani masalah

5. Pengguna Sistem Pakar mempunyai batas kognitif alami, sehingga mungkin tidak bisa memanfaatkan sistem secara maksimal.
6. Sistem Pakar bekerja baik untuk suatu bidang yang sempit.
7. Banyak pakar yang tidak mempunyai jalan untuk mengecek apakah kesimpulan mereka benar dan masuk akal.
8. Istilah dan jargon yang dipakai oleh pakar dalam mengekspresikan fakta seringkali terbatas dan tidak mudah dimengerti oleh orang lain.
9. Pengembangan Sistem Pakar seringkali membutuhkan perekayasa pengetahuan (*Knowledge Engineer*) yang langka dan mahal.
10. Kurangnya rasa percaya pengguna menghalangi pemakaian Sistem Pakar.
11. Transfer pengetahuan dapat bersifat subyektif dan bias.

2.3.4 Konsep Umum Sistem Pakar

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*) jika...maka (*IF...THEN*).

Walaupun cara diatas sangat sederhana, namun banyak hal yang berarti dalam membangun system pakar dengan mengekspresikan pengetahuan pakar dalam bentuk aturan diatas.

Turban (1995) menyatakan bahwa 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 di bidang tertentu yang didapatkan dari pelatihan, membaca atau pengalaman. Contoh bentuk pengetahuan yang merupakan keahlian adalah :

1. Fakta-fakta pada lingkup permasalahan tertentu.
2. Teori-teori pada lingkup permasalahan tertentu.
3. Prosedur-prosedur dan aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
4. Strategi-strategi global untuk menyelesaikan masalah. pengetahuan tentang pengetahuan (*Meta-Knowledge*).

Bentuk-bentuk tersebut memungkinkan para ahli untuk dapat mengambil keputusan lebih cepat dan lebih baik dari seorang yang bukan ahli.

Seorang ahli adalah seorang yang mempunyai pengetahuan tertentu dan mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik

permasalahan (*Domain*), menyusun kembali pengetahuan jika dipandang perlu, memilah aturan jika dibutuhkan, 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 4 aktivitas, yaitu tambahan pengetahuan (dari para ahli atau sumber-sumber lainnya), representasi pengetahuan (ke komputer), inferensi pengetahuan dan pengalihan pengetahuan ke pengguna. Pengetahuan yang disimpan di computer dinamakan dengan nama basis pengetahuan (*Knowledge Base*). Ada dua tipe pengetahuan, yaitu fakta dan prosedur.

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 harus dapat diprogram untuk membuat inferensi (*Inference Engine*)

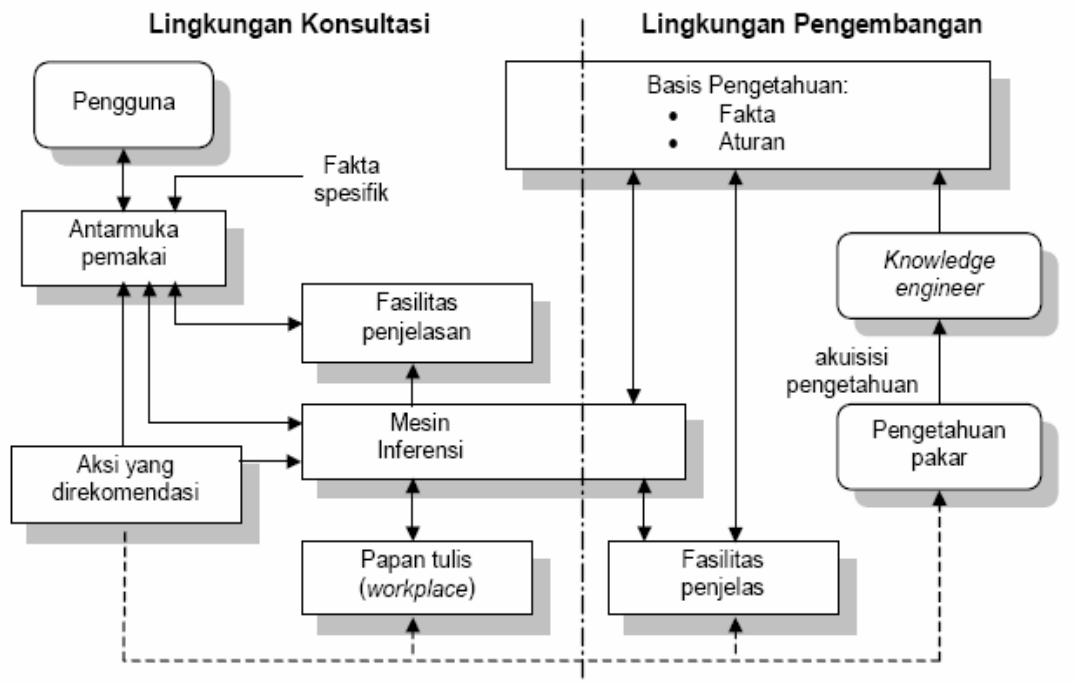
Menurut Turban (1995) terdapat tiga orang yang terlibat dalam lingkungan system pakar, yaitu :

1. Pakar, Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.
2. Perakayasa System (*Knowledge Engineer*), Knowledge engineer adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan *counter example* dan menerangkan kesulitan-kesulitan konseptual.
3. Pemakai, sistem pakar memiliki beberapa pemakai, yaitu : pemakai bukan pakar, pelajar, pembangun sistem pakar yang ingin meningkatkan dan menambah basis pengetahuan, dan pakar.

2.4 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*Development Environment*) dan lingkungan konsultasi (*Consultation Environment*) dan lingkungan konsultasi (*Consultation Environment*) (Turban, 1995). Lingkungan pengembangan system pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan system 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 2.4 berikut ini :



Gambar 2.4 Arsitektur Sistem Pakar

Sumber : <http://balzach.staff.ugm.ac.id/AI/Diktat%20Kecerdasan%20Buatan.pdf>

Komponen-komponen yang terdapat dalam sistem pakar adalah seperti pada Gambar 2.4, yaitu antarmuka pengguna (*User Interface*), Basis Pengetahuan, Akuisisi Pengetahuan, Mesin Inferensi, fasilitas Penjelasan, Perbaikan Pengetahuan.

Keputusan kompleks melibatkan kombinasi yang rumit dari berdasar fakta dan heuristik pengetahuan. Dalam urutan untuk komputer untuk mampu mendapat kembali dan secara efektif menggunakan pengetahuan heuristik, pengetahuan harus diorganisir pada suatu format yang mudah diakses yang mencirikan antar data, pengetahuan, dan struktur kendali. Karena alasan ini, sistem pakar dibagi menjadi tiga tingkatan :

- Dasar Pengetahuan terdiri dari pemecahan masalah aturan, prosedur, dan data hakiki relevan kepada daerah masalah.
- Memori Aktip mengacu pada tugas data spesifik untuk masalah dalam pembahasan.
- Mesin Kesimpulan adalah suatu mekanisme kendali umum yang menerapkan secara aksioma pengetahuan di dalam pengetahuan mendasarkan kepada data yang task-specific untuk menerima beberapa solusi atau kesimpulan.

2.4.1 Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

Dalam studi kasus pada sistem berbasis pengetahuan terdapat beberapa karakteristik yang dibangun untuk membantu kita di dalam membentuk serangkaian prinsip-prinsip arsitekturnya. Prinsip tersebut meliputi :

- a. Pengetahuan merupakan kunci kekuatan sistem pakar
- b. Pengetahuan sering tidak pasti dan tidak lengkap.
- c. Pengetahuan sering miskin spesifikasi.
- d. Amatir menjadi ahli secara bertahap.
- e. Sistem pakar harus fleksibel.
- f. Sistem pakar harus transparan.

Sejarah penelitian di bidang AI telah menunjukkan berulang kali bahwa pengetahuan adalah kunci untuk setiap sistem cerdas (*Intelligence System*).

Basis pengetahuan berisi pengetahuan yang dibutuhkan untuk memahami, memformulasi, dan memecahkan masalah. Basis pengetahuan tersusun atas 2 elemen dasar:

- Fakta, misalnya: situasi, kondisi, dan kenyataan dari permasalahan yang ada, serta teori dalam bidang itu
- Aturan, yang mengarahkan penggunaan pengetahuan untuk memecahkan masalah yang spesifik dalam bidang yang khusus. Atau bisa juga diartikan lain

Basis pengetahuan (*knowledge base*) adalah fasilitas untuk menyimpan pengetahuan pakar. Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan, dan menyelesaikan masalah, dimana tersusun atas fakta yang berupa informasi tentang bagaimana cara membangkitkan fakta baru dari fakta yang sudah diketahui. Basis pengetahuan merupakan inti dari sistem pakar.

2.4.2 Working memory (*Aktip Memori*)

Mengacu pada tugas data spesifik untuk masalah dalam pembahasan.

2.5 Bagaimana Sistem Pakar Melakukan Inferensi?

Teknik inferensi, Menurut George F. Luger (1990:94), definisi inferensi adalah : *"Inference is the process used in an expert system of deriving new*

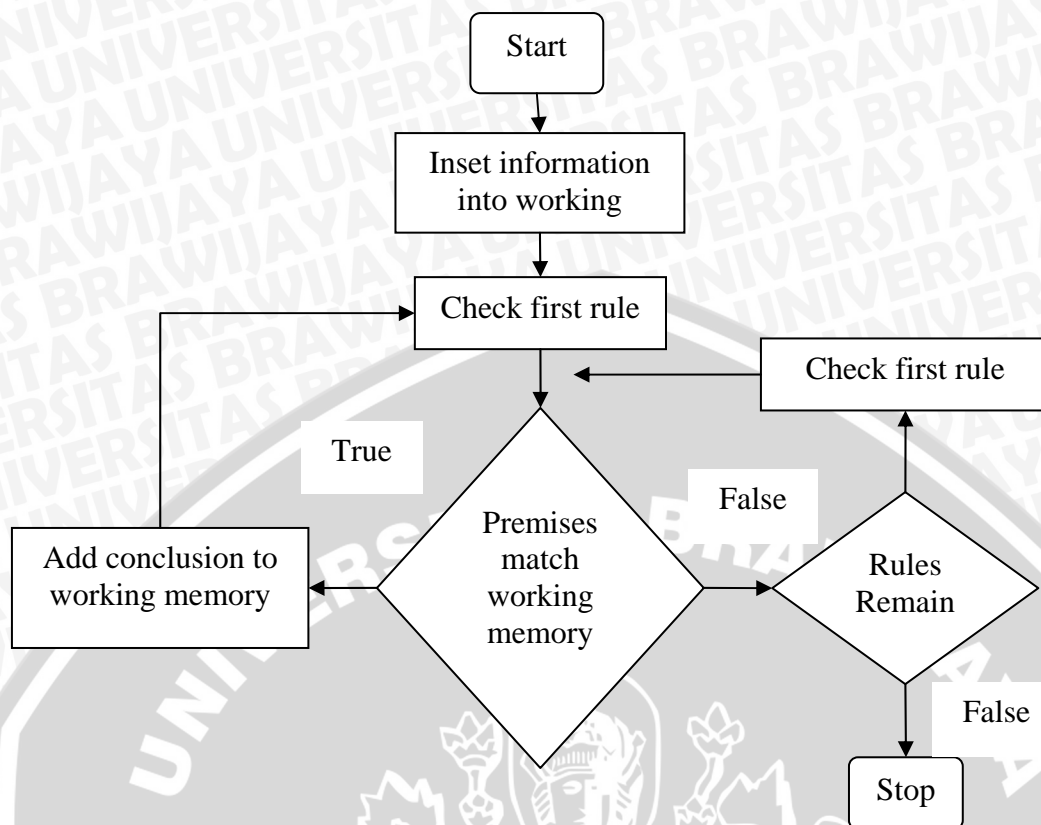
information from known information”. Yang dapat diartikan bahwa inferensi adalah proses yang digunakan dalam sebuah sistem pakar untuk menghasilkan informasi baru dari informasi yang telah diketahui. Mesin inferensi adalah salah satu bagian dari sistem pakar dimana menerapkan teknik inferensi untuk mengolah basis pengetahuan sehingga sistem sampai pada suatu kesimpulan.

Mesin Inferensi (*Inference Engine*) merupakan otak dari Sistem Pakar. Juga dikenal sebagai penerjemah aturan (*Rule Interpreter*). Komponen ini berupa program komputer yang menyediakan suatu metodologi untuk memikirkan (*Reasoning*) dan memformulasi kesimpulan. Kerja mesin inferensi meliputi:

1. Menentukan aturan mana akan dipakai
2. Menyajikan pertanyaan kepada pemakai, ketika diperlukan.
3. Menambahkan jawaban ke dalam memori Sistem Pakar.
4. Menyimpulkan fakta baru dari sebuah aturan
5. Menambahkan fakta tadi ke dalam memori.

2.5.1 Sistem Perantaraan Maju (*Forward Chaining Systems*)

Forward chaining atau disebut juga pelacakan ke depan adalah suatu metode pelacakan yang dimulai dengan satu atau beberapa fakta awal sehingga didapatkan hasil atau goal yang belum diketahui sebelumnya. Proses bergerak maju dengan cara mencocokkan fakta tersebut dengan premis atau bagian *IF* yang ada pada aturan-aturan dalam basis pengetahuan. Jika saat dieksekusi semua premis bernilai benar (*True*), maka bagian *THEN* akan diambil sebagai fakta baru. Proses pelacakan ini akan dilakukan terus menerus secara berantai samai ditemukan hasil atau goal yang diinginkan atau sampai tidak terdapat aturan lagi. Proses *Forward Chaining* data ditunjukkan pada Gambar 2.5



Gambar 2.5 Flowchart Proses Inferensi Forward Chaining

Gambar tersebut dapat dijelaskan sebagai berikut :

- Masukkan informasi (*Fakta*) ke dalam memori kerja. Untuk mencari hasil dari tujuan yang diinginkan.
- Cek informasi tersebut pada aturan yang pertama. Aturan tersebut berisikan premis (terdapat pada bagian *IF*) dan konklusi atau kesimpulan (terdapat pada bagian *THEN*).
- Jika fakta tersebut terdapat pada premis. Maka konklusi ditambahkan dalam memori kerja. Jika tidak terdapat dalam premis, maka cek aturan berikutnya.
- Jika masih terdapat aturan, maka cek aturan selanjutnya, tetapi jika aturan tidak terdapat lagi. Maka proses selesai.

Berikut contoh dari *forward chaining* :

- Fakta (*Facts*) : A,B,C
- Aturan (*Rules*) :
 1. If A & B then X
 2. If X & C then Y
 3. If Y & D then Z
- Goal?
-

Langkah	Fakta	keterangan
0	ABC	Langkah awal, terdapat 3 fakta
1	ABCX	Fakta baru (X) berasal dari aturan 1
2	ABCXY	Fakta baru (Y) berasal dari aturan 2
3	ABCXYZ	Goal (Z) didapat dari aturan 3

Pada sistem perantaraan maju, fakta-fakta dalam dalam sistem disimpan dalam memori kerja dan secara kontinyu diperbarui. Aturan dalam sistem merepresentasikan aksi-aksi yang harus diambil apabila terdapat suatu kondisi khusus pada item-item dalam memori kerja, sering disebut aturan kondisi-aksi. Kondisi biasanya berupa pola yang cocok dengan item yang ada di dalam memori kerja, sementara aksi biasanya berupa penambahan atau penghapusan item dalam memori kerja.

2.5.2 Strategi Penyelesaian Konflik (*Conflict Resolution Strategy*)

Strategi penyelesaian konflik dilakukan untuk memilih aturan yang akan diterapkan apabila terdapat lebih dari 1 aturan yang cocok dengan fakta yang terdapat dalam memori kerja. Di antaranya adalah:

- a. *No duplication*. Jangan memicu sebuah aturan dua kali menggunakan fakta/data yang sama, agar tidak ada fakta yang ditambahkan ke memori kerja lebih dari sekali.
- b. *Recency*. Pilih aturan yang menggunakan fakta yang paling baru dalam memori kerja. Hal ini akan membuat sistem dapat melakukan penalaran dengan mengikuti rantai tunggal ketimbang selalu menarik kesimpulan baru menggunakan fakta lama.
- c. *Specificity*. Picu aturan dengan fakta prakondisi yang lebih khusus (*spesific*) sebelum aturan yang menggunakan prakondisi lebih umum. Contohnya : jika kita mempunyai aturan “JIKA (burung X) MAKA TAMBAH (dapat_terbang X)” dan “JIKA (burung X) DAN (penguin X) MAKA TAMBAH (dapat_berenang X)” serta fakta bahwa tweety adalah seekor penguin, maka lebih baik memicu aturan kedua dan menarik kesimpulan bahwa tweety dapat berenang.

d. *Operation priority*. Pilih aturan dengan prioritas yang lebih tinggi. Misalnya ada fakta (bertemu kambing), (ternak kambing), (bertemu macan), dan (binatang_buas macan), serta dua aturan: “JIKA (bertemu X) DAN (ternak X) MAKA TAMBAH (memberi_makan X)” dan “JIKA (bertemu X) DAN (binatang_buas X) MAKA TAMBAH (melarikan_diri)”, maka kita akan memilih aturan kedua karena lebih tinggi prioritasnya.

2.5.3 Sistem Perantaraan Balik (*Backward Chaining Systems*)

Backward chaining atau pelacakan ke belakang adalah suatu metode penalaran yang dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. *Backward chaining* berusaha untuk membuktikan kebenaran suatu hipotesis dengan cara mengumpulkan informasi (fakta-fakta tambahan)

Proses *backward chaining* dimulai dengan suatu goal yang akan dibuktikan kebenarannya. prosesnya dimulai dengan mencari pada memori kerja. Pertanyaannya apakah goal tersebut sebelumnya sudah ada. Jika goal tidak terdapat pada memori kerja, maka goal ini dicocokkan dengan konklusi atau bagian THEN yang ada pada aturan-aturan dalam basis pengetahuan. Jika terdapat konklusi yang tepat, maka bagian premis dari aturan tersebut dicari pada fakta-fakta pendukung. Jika tidak terdapat pada fakta pendukung. Maka premis tersebut akan dijadikan goal baru atau disebut juga subgoal. Subgoal ini akan dicocokkan dengan konklusi yang ada pada aturan-aturan lainnya. proses pencocokkan akan dilakukan berulang-ulang sampai ditemukan premis yang bukan merupakan konklusi dari suatu aturan (*primitive rule*).

Berikut ini adalah contoh dari proses *backward chaining*

- Hipotesa : Z
- Fakta : A, B, C
- Aturan :
 1. If A & B then X
 2. If X & C then Y
 3. If Y or D then Z

Langkah	Subgoal	Fakta	Keterangan
0	Z	ABC	
1	Y	ABC	Aturan 3, memberi subgoal Y
2	X(&C)	ABC	Aturan 2, memberi subgoal X
3	(A&B)	ABCX	Aturan 1, memberi fakta X
4		ABCXY	Aturan 2, memberi fakta Y
5		ABCXYZ	Aturan 3, membuktikan hipotesa Z benar

2.5.4 Pemilihan Sistem Inferensi

Secara umum kita dapat memakai panduan berikut untuk menentukan apakah kita hendak memilih perantaraan maju atau balik untuk Sistem Pakar yang kita bangun. Panduan tersebut tercantum dalam Tabel 2.3 berikut ini.

Tabel 2.3 Panduan untuk memilih Sistem Inferensi

Sumber : <http://balzach.staff.ugm.ac.id/AI/Diktat%20Kecerdasan%20Buatan.pdf>

Perantaraan Maju	Perantaraan Balik
Ada banyak hal yang hendak dibuktikan	Hanya akan membuktikan fakta (hipotesis) tunggal
Hanya sedikit fakta awal yang dipunyai	Terdapat banyak fakta awal
Ada banyak aturan yang berbeda yang dapat memberikan kesimpulan yang sama	Jika terdapat banyak aturan yang memenuhi syarat untuk dipicu (fire) pada suatu siklus

Tabel 2.4 Beberapa Karakteristik FWC dan BWC

Sumber : Muhammad Arhami. *Konsep Dasar Sistem Pakar*. Yogyakarta. Penerbit ANDI, 2005.halaman 114.

Forward chaining	Backward chaining
Perencanaan, monitoring, kontrol	Diagnosis
Disajikan untuk masa depan	Disajikan untuk masa lalu
Antecedent ke konsekuen	Konsekuen ke antecedent
Data memandu, penalaran dari bawah ke atas	Tujuan memandu, penalaran dari atas ke bawah
Bekerja ke depan untuk mendapatkan solusi apa yang mengikuti fakta	Bekerja ke belakang untuk mendaaptkan fakta yang mendukung hipotesis
Breadth first search dimudahkan	Depth first search dimudahkan

Papan Tulis (*Blackboard / Workplace*), adalah memori / lokasi untuk bekerja dan menyimpan hasil sementara. Biasanya berupa sebuah basis data.

Antarmuka Pemakai (*User Interface*). Sistem Pakar mengatur komunikasi antara pengguna dan komputer. Komunikasi ini paling baik berupa bahasa alami, biasanya disajikan dalam bentuk tanya-jawab dan kadang ditampilkan dalam bentuk gambar/grafik. Antarmuka yang lebih canggih dilengkapi dengan percakapan (*Voice Communication*). Atau bisa juga diartikan, antarmuka pemakai (*User Interface*) adalah bagian yang menghubungkan antara pemakai dengan system pakar. Sistem pakar akan mengajukan pertanyaan-pertanyaan berbentuk “Ya” atau “Tidak” (*Yes Or No Question*) atau berbentuk menu pilihan. Sistem pakar akan mengambil kesimpulan berdasarkan jawaban-jawaban dari pemakai tadi. Pengembangan Awal dari suatu sistem pakar dilakukan oleh tenaga ahli dan insinyur pengetahuan. Tidak sama dengan program yang paling konvensional, di mana saja para programmer dapat

membuat keputusan disain program dari sebagian besar para sistem pakar. sistem diterapkan melalui suatu usaha regu. Suatu pertimbangan kebutuhan tentang pemakai akhir adalah sangat penting merancang isi dan alat penghubung pemakai tentang sistem pakar.

Subsistem Penjelasan (*Explanation Facility*). Kemampuan untuk menjejak (*Tracing*) bagaimana suatu kesimpulan dapat diambil merupakan hal yang sangat penting untuk transfer pengetahuan dan pemecahan masalah. Komponen subsistem penjelasan harus dapat menyediakannya yang secara interaktif menjawab pertanyaan pengguna, misalnya:

1. “Mengapa pertanyaan tersebut anda tanyakan ?”
2. “Seberapa yakin kesimpulan tersebut diambil ?”
3. “Mengapa alternatif tersebut ditolak ?”
4. “Apa yang akan dilakukan untuk mengambil suatu kesimpulan ?”
5. “Fakta apalagi yang diperlukan untuk mengambil kesimpulan akhir ?”

Atau bisa juga diartikan Subsistem Penjelasan (*Explanation Facility*) adalah fasilitas yang digunakan untuk menjelaskan proses pengambilan keputusan atau penalaran kepada pemakai (menjawab pertanyaan “bagaimana” dan “mengapa”). Salah satu dari karakteristik kunci dari suatu sistem pakar adalah fasilitas penjelasan. Dengan kemampuan ini , suatu tenaga sistem pakar dapat menjelaskan bagaimanamendatangkan kesimpulannya.Pemakai dapat menanyakan pertanyaan berhadapan dengan apa, bagaimana, dan mengapa aspek suatu masalah. Sistem yang ahli akan kemudian menyediakan pemakai itu dengan jejak konsultasi memproses, menunjukkan alur pemikiran kunci itu mengikuti sepanjang konsultasi itu. Kadang-Kadang suatu sistem tenaga ahli diperlukan untuk memecahkan permasalahan lain, yang mungkin tidak dapat secara langsung berhubungan dengan masalah yang spesifik pada tangan, tetapi solusi siapa akan mempunyai suatu dampak pada total pemecahan masalah memproses Fasilitas Penjelasan membantu sistem pakar untuk memperjelas dan membenarkan mengapa penyimpangan seperti itu boleh jadi diperlukan.

Sistem Penghalusan Pengetahuan (*Knowledge Refining System*). Seorang pakar mempunyai sistem penghalusan pengetahuan, artinya, mereka bisa menganalisa sendiri performa mereka, belajar dari pengalaman, serta meningkatkan pengetahuannya untuk konsultasi berikutnya. Pada Sistem Pakar, *swa-evaluasi* ini

penting sehingga dapat menganalisa alasan keberhasilan atau kegagalan pengambilan kesimpulan, serta memperbaiki basis pengetahuannya.

2.5.5 Representasi Pengetahuan

1. Aturan atau Kaidah

Di sini pengetahuan disajikan dalam aturan-aturan yang berbentuk pasangan keadaan-aksi (*condition-action*): “JIKA keadaan terpenuhi atau terjadi MAKA suatu aksi akan terjadi”. Sistem Pakar yang basis pengetahuannya melulu disajikan dalam bentuk aturan produk disebut sistem berbasis-aturan (*rule-based system*). Kondisi dapat terdiri atas banyak bagian, demikian pula dengan aksi. Urutan keduanya juga dapat dipertukarkan letaknya. Contohnya:

A. JIKA suhu berada di bawah 20°C MAKA udara terasa dingin.

B. Udara terasa dingin JIKA suhu berada di bawah 20°C.

C. JIKA suhu berada di bawah 20°C ATAU suhu berada di antara 20-25°C DAN angin bertiup cukup kencang MAKA udara terasa dingin.

Contoh dari MYCIN, Sistem Pakar untuk mendiagnosis dan merekomendasikan perlakuan yang tepat untuk infeksi darah tertentu:

IF the infection is primary-bacteremia

AND the site of the culture is one of the sterile sites

AND the suspected portal of entry is the gastrointestinal tract

THEN there is suggestive evidence (0.7) that infection is bacteroid.

2. Definisi Sistem Produksi

Pengetahuan dalam system produksi boleh juga direpresentasikan oleh himpunan kaidah dalam bentuk :

IF [kondisi] THEN [aksi]

Dengan sebuah control system dan basis data. *Control system* memberikan aturan penerjemahan dan pengurutan. Basis data aberaksi sebagai konteks cadangan untuk record yang kondisinya dievaluasi oleh kaidah dan informasi dimana kaidah akan beraksi. Berikutnya untuk sintaks IF THEN, kaidah produksi juga sering digambarkan sebagai pasangan-pasangan berikut kondisi aksi, *antecedent consequent*, pola aksi, situasi response.

Contoh-contoh produksi dari pengalaman sehari-hari yaitu :

IF [mengendarai 15 mil perjam melewati batas AND melihat melalui kaca spion ada lampu merah yang berkedip-kedip]

THEN [pinggirkan AND berhenti]
 IF [sakit kepala AND sakit tenggorokan AND hidung tersumbat]
 THEN [ambil aspirin AND istirahat]
 IF [barometer turun AND petir di barat laut]
 THEN [arahkan ke pelabuhan yang aman]

Contoh-contoh lain yang lebih mendekati secara analitik untuk mengklasifikasikan suatu objek dimulai dengan kaidah-kaidah berikut ini :

IF [kategori adalah sebuah form OR sebuah pewarnaan OR sebuah teksture]
 THEN [objek mempunyai sebuah permukaan]
 IF [kategori adalah sebuah permukaan OR jelek OR kualitas *tactile*]
 THEN [objek mempunyai kulaitas luar]
 IF [kategori adalah sebuah ukuran OR kualitas luar OR massa Or suatu zat]
 THEN [objek mempunyai kualitas fisik]

Struktur *intuitif* secara alami untuk representasi pengetahuan yang telah dihasilkan oleh system produksi merupakan metode yang lebih disukai oleh banyak sistem pakar, termasuk perintis program *DENDRAL* dan *MYCIN*. Sebagai contoh, kaidah produksi dari sistem tersebut seperti yang ditunjukkan di bawah ini.

DENDRAL

IF [the spectrum of the molecule has two peaks at masses X_1 and X_2 such that

1. $X_1 + X_2 = M + 28$ AND
2. $X_1 - 28$ is a high peak AND
3. $X_2 - 28$ is a high peak AND
4. At least one of X_1 or X_2 is high]

THEN [the molecule contains a ketone group]

MYCIN

IF [a) the stain of the organism is gramneg AND

b) the morphology of the organism is rod AND

c) the patient is a compromised host]

THEN [there is suggestive evidence (0,6) that the identity of the organism is *pseudomonas*]

Sistem produksi memainkan sejumlah peran yang penting yang akan kita periksa kelebihan dan kekurangannya lebih luas dalam bagian ini, yaitu :

1. Kekuatan skema representasi pengetahuan. Sistem produksi tidak hanya direpresentasikan dalam pengetahuan tetapi juga kegiatan. Arsitektur sistem produksi pada kenyataannya ekuivalen dengan bentuk umum dari mesin turing dan diidentifikasi sebagai “Mesin Newell”.
2. Jembatan penghubung penelitian AI dengan system pakar. Sistem produksi memberikan suatu bahasa yang merupakan representasi dari pengetahuan pakar yang sangat alami.
3. Model *heuristic* untuk perilaku manusia. Studi mengenai perilaku manusia merupakan protokol asli yang membawa Newell untuk merumuskan konsep produksi. Paradigma dari perilaku manusia menyajikan suatu fungsi *heuristic* yang bernilai dari pemicu penelitian lebih lanjut. Pada penutup bagian ini kita akan memeriksa beberapa implikasi filosofi sistem produksi manusia.

Dari kesamaan perumusan kedua kaidah produksi itu, kita catat suatu gambaran atraktif pertama dari system produksi secara formal : pemisahan secara rapi dari struktur program dari data. Ada tiga elemen utama dari semua system produksi, yaitu:

1. Database global

Database global merupakan struktur data utama dari system produksi. Database mungkin mempunyai jangkauan dari sebuah daftar sederhana atau matriks kecil hingga kekompleksan, relasi dan struktur indeks. Hal ini merupakan struktur dasar dimana kaidah produksi dapat beroperasi. Hal ini merupakan struktur dinamis dan berubah-ubah secara kontinyu sebagai hasil dari operasi kaidah produksi. Database global juga menjadi acuan untuk konteks, memori cadangan jangka pendek, atau memori kerja.

2. Kaidah Produksi

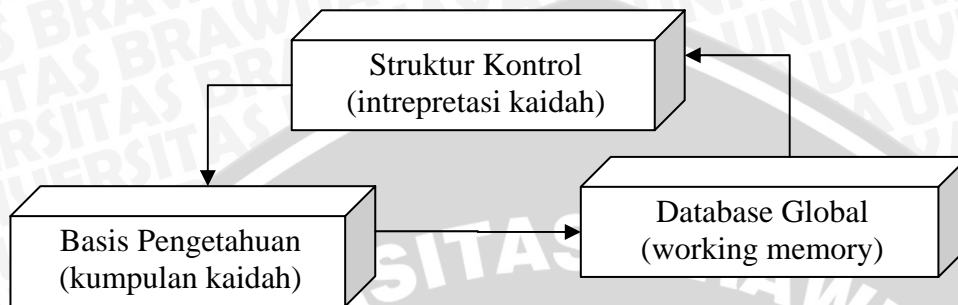
Sebagaimana yang diindikasikan oleh contoh diatas, kaidah produksi mempunyai bagian kondisi (*IF*) yang disebut bagian kanan dan aksi (*THEN*) disebut bagian kiri. Jika sisi kiri kadang-kadang dinamakan kondisi atau premis yang dipenuhi oleh database, maka kaidah-kaidah dapat diterapkan dan subjek menjadi pemicu bagi *system control*.

3. Sistem Kontrol

System control merupakan program penterjemah yang esensial untuk mengontrol urutan di mana kaidah-kaidah produksi dipicu dan menyelesaikan konflik jika lebih dari satu kaidah yang diaplikasikan. *System control* secara berulang-ulang

mengaplikasikan kaidah-kaidah untuk database hingga sebuah gambaran dari tujuan yang dihasilkan. Kemudian mendeteksi kejadian seperti tujuan dan record kaidah yang telah diaplikasikan untuk mencapainya bagi referensi sebelumnya.

Hubungan antara ketiga elemen tersebut dan iterasi alami dari operasi system produksi diilustrasikan pada Gambar 2.6 berikut ini :



Gambar 2.6 Komponen Sistem Produksi

Sumber : Muhammad Arhami. *Konsep Dasar Sistem Pakar*. Yogyakarta. Penerbit ANDI, 2005.halaman 78.

Masalah yang dihasilkan oleh sistem nyata untuk tiga elemen ini dinamakan masalah representasi (*representasion problem*). Hal ini merupakan masalah utama dalam rancangan sistem produksi sembarang.

2.5.6 Kaidah Produksi, Pengetahuan dan Kaidah Inferensi

Kaidah produksi dituliskan dalam bentuk pernyataan Jika Maka (IF THEN). Pernyataan ini menghubungkan bagian premis (IF) dan bagian kesimpulan (THEN) yang dituliskan dalam bentuk :

IF [premis] THEN [konklusi]

Apabila bagian premis dipenuhi maka bagian konklusi akan bernilai benar. Sebuah kaidah terdiri dari klausa-klausa. Sebuah klausa mirip dengan sebuah kalimat dengan subyek, kata kerja dan obyek yang menyatakan suatu fakta. Suatu aturan juga dapat terdiri dari beberapa premis dan lebih dari satu konklusi. Untuk merumuskan beberapa domain pengetahuan secara akurat diperlukan banyak kaidah produksi. aturan-aturan ini menyediakan rincian obyek, karakteristik, dan tindakan-tindakan yang harus diambil. Untuk dapat meliputi suatu obyek diperlukan banyak rincian aturan. Aturan-aturan ini membentuk pangkalan pengetahuan yang kemudian menjadi bagian sistem produksi.

Biasanya pengetahuan diturunkan dalam bentuk pernyataan *linguistic* dari pakar (Negoita, 1985). Sebagai contoh:

Bercak pada daun is sangat banyak

Ujung daun kering is banyak

Dalam contoh pertama, *variable linguistic* “bercak pada daun” memiliki nilai *linguistic* “sangat banyak”, sedangkan pada contoh kedua, *variable linguistic* “ujung daun kering” memiliki nilai *linguistic* “banyak”.

Bagian premis dalam aturan produksi dapat memiliki lebih dari satu proposisi. Proposisi-proposisi tersebut dihubungkan dengan menggunakan operator logika AND atau OR. Sebagai contoh :

Bercak pada daun is sangat banyak

AND ujung daun kering is banyak

Aturan produksi digunakan untuk menyatakan hubungan antara dua bentuk pernyataan *linguistic* yang masing-masing merupakan bagian premis dan konklusi/kesimpulan. Sebagai contoh :

IF bercak pada daun is sangat banyak

AND ujung daun kering is banyak

THEN penyakit bercak ungu is puso

Kaidah menurut Turban (2001) dapat dibuat dalam beberapa bentuk yang berbeda, yaitu :

1. IF premis THEN konklusi. Contohnya, IF pendapatan anda tinggi THEN kesempatan anda untuk diaudit oleh IRS adalah tinggi.
2. Konklusi, IF premis, kesempatan anda diaudit adalah tinggi IF pendapatan anda tinggi.
3. Inklusi dari ELSE, IF pendapatan anda tinggi OR deduksi anda luar biasa THEN kesempatan anda diaudit tinggi OR ELSE kesempatan anda diaudit rendah.
4. Kaidah yang lebih kompleks, IF tingkat kredit adalah tinggi AND gaji lebih dari \$10,000 dandaftar pinjamannya termasuk dalam kategori B.

Ada dua tipe kaidah yang umum dalam AI, yaitu pengetahuan dan inferensi. Kaidah pengetahuan atau kaidah deklaratif menyatakan semua fakta dan hubungannya tentang suatu permasalahan. Kaidah inferensi atau kaidah *procedural* pada sisi lain merupakan nasihat atau saran tentang bagaimana menyelesaikan suatu masalah yang diberikan dengan fakta tertentu yang diketahui.

Sebagai contoh asumsikan bahwa anda seorang pebisnis dari penjualan dan pembelian emas. Kaidah pengetahuannya seperti terlihat berikut ini :

1. Kaidah 1 IF konflik internasional dimulai THEN harga emas naik

2. Kaidah 2 IF tingkat inflasi mengalami kemunduran THEN harga emas turun.
3. Kaidah 3 IF konflik internasional berlangsung selama lebih dari 7 hari AND IF konflik ini terjadi di timur tengah THEN beli emas.

Kaidah inferensi (procedural) seperti yang diberikan berikut ini

1. Kaidah 1 IF data dibutuhkan tidak dalam system THEN permintaanya dari pengguna.
2. Kaidah 2 IF lebih dari satu kaidah digunakan THEN nonaktifkan kaidah sembarang yang tidak menambah data baru.

Kaidah inferensi berisi kaidah tentang kaidah. Tipe ini dimana kaidah pengetahuan menjadi basis pengetahuan sedangkan kaidah inferensi bagian dari mesin inferensi.

Representasi kaidah terutama dapat diaplikasikan bila dibutuhkan untuk merekomendasi suatu bagian aksi berdasarkan kejadian yang dapat diobservasi. Ada beberapa keuntungan penggunaan kaidah, yaitu :

1. Kaidah mudah dimengerti. Mereka mudah disampaikan karena merupakan bentuk alami dari pengetahuan.
2. Inferensi dan penjelasan mudah diperoleh atau diturunkan.
3. Modifikasi dan perawatan relative lebih mudah.
4. Ketidakpastian lebih mudah dikombinasikan dengan kaidah.
5. Setiap kaidah sering saling independent dari semua kaidah.

Selain itu terdapat pula keterbatasan dari representasi kaidah ini, yaitu :

1. Pengetahuan yang kompleks membutuhkan beribu-ribu kaidah, yang mungkin agak sukar membuatnya, baik untuk menggunakan system maupun untuk perawatannya.
2. Pembangun menyukai kaidah; oleh karenanya mereka mencoba kekuatan semua pengetahuan kedalam kaidah dibandingkan mencapai representasi yang lebih sesuai.
3. System dengan banyak kaidah mungkin mempunyai batasan pencarian dalam control program. Beberapa program mempunyai kesulitan dalam mengevaluasi system berbasis kaidah dan membuat inferensi.

Representasi kaidah mempunyai karakteristik-karakteristik seperti yang ditunjukkan dalam table berikut :

Tabel 2.5 Karakteristik dan Representasi Kaidah
 Sumber : Muhammad Arhami. *Konsep Dasar Sistem Pakar*. Yogyakarta. Penerbit ANDI, 2005.halaman 82.

	Bagian Pertama	Bagian Kedua
Nama	Premis Antecedent Situasi IF	Konklusi Konsekuen Aksi THEN
Alami	Kondisi, sama dengan pengetahuan deklaratif	Resolusi, sama dengan pengetahuan Prsedural
Ukuran	Dapat mempunyai banyak IF	Biasanya hanya mempunyai satu konklusi
Pernyataan	Pernyataan AND Pernyataan OR	Semua kondisi harus benar untuk konklusi benar Jika ada kondisi pernyataan OR benar maka konklusinya benar

2.5.7 Keuntungan Sistem Produksi

Ada beberapa ciri-ciri system produksi yang menjadikannya berbentuk lebih khusus dan menarik dari representasi pengetahuan untuk sistem pakar, yaitu :

1. *Exspressiveness dan intuitiveness*

Pengalaman dalam pekerjaan keahlian manusia mengindikasikan pengulangan berat dari tema “*well, in the case of so on and so I usually do such and such*”. Tema ini memetakan keadaan alami dalam bentuk IF...THEN dari kaidah produksi. Kaidah-kaidah produksi secara esensial menyatakan kepada kita bahwa apa yang dilakukan dalam situasi tertentu. Karena banyak sistem pakar yang diorganisasikan berkaitan dengan nasihat / saran apa yang harus dilakukan, maka sifat dari system produksi ini adlah merupakan suatu hal khusus secara alami untuk representasi pengetahuan.

2. *Simplicity*

Struktur seragam dari sintaks IF...THEN dalam system berbasis kaidah memberikan suatu kesederhanaan yang menarik untk representasi pengetahuan. Ciri ini meningkatkan suatu keadaan yang dapat dibaca dari kaidah produksi dan komunikasi antara berbagai bagian dari program tunggal. Kaidah produksi dengan sintaksnya yang tepat mengarah kepada pendokumentasian diri.

3. *Modularity dan modifiability*

Dengan kealamiannya, kaidah produksi mengkodekan bentuk diskret informasi yang secara umum tidak berhubungan dengan kaidah produksi yang lain, kecuali jika ada suatu kaidah produksi eksplisit yang menghubungkan mereka. Informasi dapat

diperlakukan sebagai suatu kumpulan fakta yang independent yangmana boleh ditambahkan atau dihapus dari sistem dengan secara esensial tidak mengakibatkan “*Side Effect*” (efek sampingan) yang mengganggu. Ciri modular sistem produksi menunjukkan kenaikan perbaikan dan setelanyang bagus dari sistem produksi dengan tidak mengalami pengurangan kinerja.

4. *Knowledge intensive (pengetahuan intensive)*

Catatan bahwa tiga bagian struktur dari intepretasi kaidah, basis pengetahuan, dan memori kerja (*Working Memory*) memberikan suatu pemisahan yang efektif dari basis pengetahuan yang berasal dari intepretasi kaidah atau mesin inferensi. Jadi mesin inferensi menjadi tujuan umum dan kerjasama yang efektif, secara prinsip, pada berbagai macam basis pengetahuan. Basis pengetahuan disusun dari kaidah produksi, yang pada gilirannya yang sangat utama adalah “pengetahuan murni” (*Pure Knowledge*) karena membutuhkan isi, bukan kontrol atau informasi pemrograman. Karena setiap kaidah produksi ekuivalen untuk suatu ringkasan dan kejelasan kalimatnya maka masalah *semantic* diselesaikan dengan struktur representasi.

2.6 Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi Pengetahuan (*Knowledge Acquisition*) adalah proses yang kompleks. Dalam arti yang sempit *Knowledge Acquisition* adalah suatu proses untuk mempelajari, memperoleh, dan mengorganisir pengetahuan dari seorang pakar. Dalam arti yang lebih luas *Knowledge Acquisition* melibatkan semua proses *Knowledge Engineering* termasuk menginterview pakar, menguji dan memperbaiki pengetahuan bersama pakar dan pemakai (*End User*) sampai sistem siap digunakan. Sebagai objek dari pengembangan sistem pakar ini adalah troubleshooting sepeda motor Yamaha F1-ZR, tahun 2002.

Sumber pengetahuan yang diperlukan untuk menghasilkan suatu *Expert System* diperoleh dari dua hal penting, yaitu :

- Pengetahuan Formal (*Formal Knowledge*)
- Pengetahuan Pakar (*Expert Knowledge*)

Pembahasan dimulai dari Pengetahuan Formal.

2.7 Pengetahuan Formal (*Formal Knowledge*)

Pengetahuan Formal adalah sumber pengetahuan yang kita peroleh dari text book, katalog serta artikel. Mengingat objek penelitian adalah motor Yamaha F1-ZR maka sumber utama pengetahuan formal penelitian ini antara lain adalah buku-buku panduan *Automotive Troubleshooting* dan *Manual Guide*.

Untuk mengenal objek penelitian, perlu diberikan spesifikasi sepeda motor Yamaha F1-ZR seperti terlihat pada Tabel 2.6.

Untuk lebih memperjelas diagram, maka berikut ini akan diberikan beberapa contoh penjelasan mendetil mengenai diagnosa kerusakan pada engine dan karburator.

2.7.1 Engine Berputar Secara Kasar

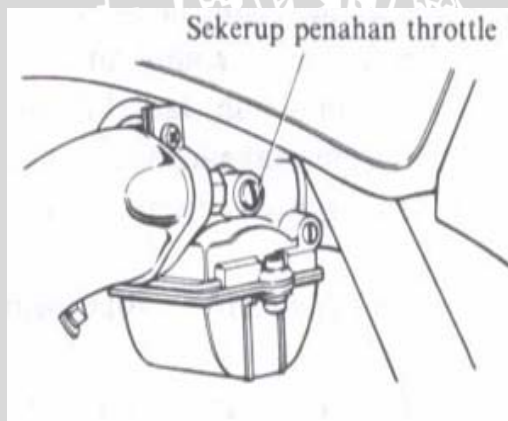
• "Idle" Secara Kasar

Gejala

Bila gas diputar, engine mati atau "Idle" secara kasar.

Kemungkinan-kemungkinan penyebab kerusakan :

- 1) Penyetelan putaran rendah yang tidak tepat di karburator.
- 2) Hubungan kabel tegangan tinggi yang longgar.
- 3) Kebocoran udara ke dalam manifold.
- 4) Aliran "pilot" yang tersumbat.
- 5) Penyetelan ignition yang tidak tepat.
- 6) Penyetelan celah katup "rocker arm" yang tidak tepat.



Gambar 2.7 Penyetel Kecepatan Idle

Diagnosa kerusakan :

- (1) Periksalah apakah penyetelan putaran rendah telah dilakukan dengan baik. pada kecepatan tinggi, engine berputar dengan halus tetapi bila gas ditutup, engine mati. Dalam hal ini, kemungkinan adalah bahwa penyetelan putaran sudah tidak tepat. Pertama-tama, pastikan bahwa engine "Idle" pada putaran yang telah ditetapkan. Panaskan engine tersebut, dan periksalah putaran "Idle". Bila sepeda motor tidak dilengkapi dengan tachometer, carilah di pasaran dan pasanglah pada sepeda motor anda. Kecepatan "Idle" bervariasi tergantung model, tetapi biasanya adalah diantara 1200 rpm sampai 1400 rpm untuk sepeda motor 100 cc. Bila pembacaan tachometer

di luar spesifikasi, penyetelan kecepatan "Idle" dilakukan dengan menyetel sekerup penahan pada gas. Dengan mengeluarkan sekerup kecepatan akan turun.

Untuk membuat penyetelan kecepatan rendah, penyetelan sekerup penahan gas dan sekerup "Pilot" kedua-duanya diperlukan, dan beberapa pengalaman diperlukan.

(2) Periksa hubungan dari kabel tegangan tinggi ke busi, dan bila longgar, perbaikilah sambungannya.

(3) Periksa kebocoran udara pada manifold pemasukan.

Baut-baut atau ban pengencang manifold yang longgar menyebabkan udara masuk ke dalam manifold dan membuat campuran bensin-udara terlalu miskin, jadi menyebabkan engine mati. Periksa sambungan manifold, dan keraskan baut-baut atau ban pengencang tersebut.

(4) Periksa aliran "Pilot" apakah tersumbat.

Bila aliran "Pilot" atau "Jet" di dalam karburator tersumbat dengan debu atau kotoran lainnya, bensin tidak cukup masuk sehingga menghasilkan performance kecepatan rendah yang tidak baik.

(5) Penyetelan "Ignition Timing" tidak tepat.

Bila "Ignition Timing" tidak tepat, bunga api tidak terjadi pada waktu yang tepat, sehingga menyebabkan engine mati.

(6) Celah katup tidak tepat.

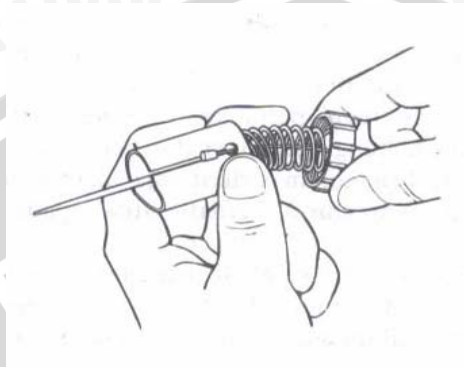
Periksa agar celah antara katup dan rocker arm berada di antara spesifikasi yang telah ditentukan manufacturer engine. Bila di luar spesifikasi, celah tersebut harus disetel kembali.

2.7.2 Karburator

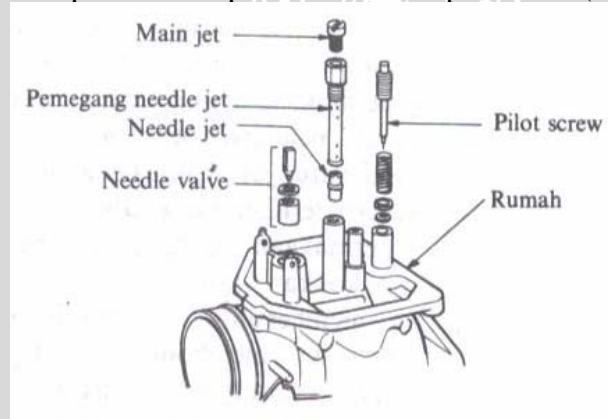
(1) Membersihkan Karburator

Karburator membuat butir-butir bensin mengapung, dan bercampur dengan udara sebelum diberikan ke silinder. Karenanya, karburator mempunyai banyak saluran bensin dan udara, dan bila filter bensin atau penyaring udara rusak, saluran-saluran ini akan tersumbat debu kotoran lainnya. Bila "Performance" kecepatan rendah menjadi berkurang tiba-tiba atau engine tidak mempunyai daya, periksa penyaring udara dan filter bensin apakah tersumbat. Karburator tersebut harus dibersihkan sebagai berikut :

- i. Pertama-tama tutuplah katup bensin, dan lepaskanlah selang (*Hose*) bensin. Lepaskanlah penutup (*Plug*) aliran keluar, ruang apung (*Float Chamber*), dan buanglah bensin di dalamnya. Hindarilah api terbuka.
- ii. Berikutnya, lepaskanlah penjepit (*Clamps*) di kedua sisi dari penyaring udara dan silinder, lepaskanlah kabel starter atau kabel "Choke", dan seperti terlihat pada Gambar 2.8, lepaskanlah tutup atas karburator.



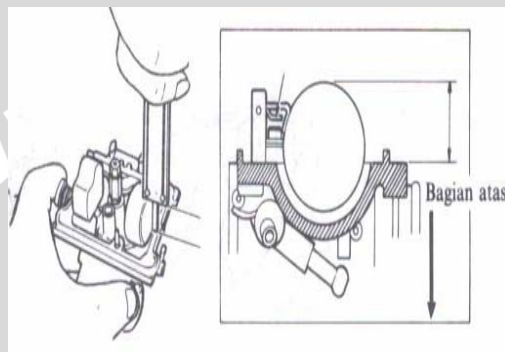
Gambar 2.8 Melepaskan Tutup Atas Dari Katup Throttle (*Throttle Valve*)



Gambar 2.9 *Jet dan Nozzle* Karburator

- iii. Lepaskanlah ruang apung, dan seperti terlihat pada Gambar 2.9, *Nozzle* dapat dilihat, lepaskanlah secara hati-hati dan periksalah apakah tersumbat atau luka bergaris-garis, kemudian semprotkanlah tiap-tiap *Nozzle* dengan udara kompresi. Bila udara kompresi tidak ada, masukkanlah ke dalam mulut anda, dan tiuplah keluar. Janganlah pernah memakai kawat sebab akan menyebabkan luka garis pada *Nozzle*. Bila melepaskan "Pilot Screw" atau "Air Screw" dipasang, harus diundurkan sesuai spesifikasi manual pemilik. Jumlah putaran keluar dispesifikasikan dalam tangga $\frac{1}{4}$ putaran.
- iv. Lepaskanlah katup jarum (*Needle Valve* atau *Float Valve*) secara hati-hati, dan periksalah apakah tersumbat atau aus. bila jarum terasa aus dengan ujung jari anda, maka jarum tersebut harus diganti dengan yang baru.

v. Bila mengecek tinggi permukaan apung, taruhlah karburator pada posisi terbalik (*Inverted*) seperti terlihat pada Gambar 2.10 dan sambil meletakkannya secara horizontal, ukurlah jarak "A" dengan memakai jangka sorong (*Vernier Calipers*). Tinggi permukaan apung telah dispesifikasikan di dalam manual servis dan karenanya, minta pendapat bengkel servis. Bila tinggi permukaan apung tidak tepat, peganglah katup jarum dan bengkokkanlah "Tang" ke dalam atau keluar secara hati-hati seperti terlihat pada Gambar 2.11 pembengkokkan sedikit saja akan mengubah tinggi permukaan apung secara besar. Karenanya berhati-hatilah pada waktu membengkokkan "Tang" tersebut.



Gambar 2.10 Memeriksa Permukaan Apung

vi. Setelah membersihkan karburator, pasanglah kembali dengan membalikkan prosedur waktu membongkar. Pastikan untuk menarik keluar "Pilot Screw" dan "Air Screw" secara tepat. Kedua bagian kecil ini dapat mudah hilang atau tergores. Karenanya perhatikanlah secara khusus.

vii. Pada karburator dengan pipa udara keluar (*Air Vent Pipe*) seperti terlihat pada Gambar 2.12 periksalah apakah pipa tersumbat atau retak.

(2) Penyetelan kecepatan "Idle"

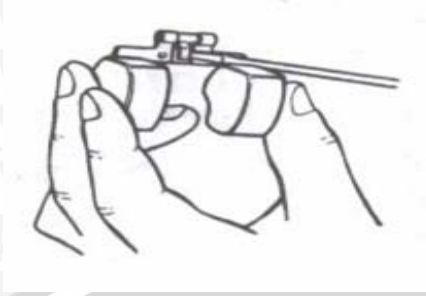
Bila kecepatan "*Idle*" ditetapkan terlalu tinggi, maka pemakaian bahan bakar akan berlebih, dan bila terlalu rendah, engine mengarah menjadi sering mati (*Stall*).

Penyetelan kecepatan "*Idle*" harus dibuat dengan engine yang telah panas. Untuk penyetelannya, lakukanlah sebagai berikut :

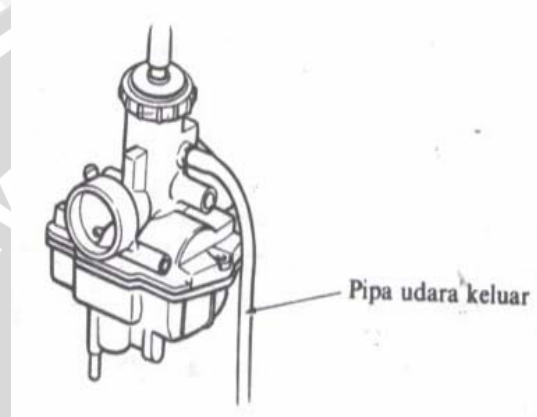
i. Putarlah sekerup penahan "*Throttle*" seperti terlihat pada Gambar 2.13 sehingga "*Idle*" lebih cepat.

ii. Berikutnya, putarlah masuk "*Air Screw*" atau "*Pilot Screw*" seperti terlihat pada Gambar 2.13 dengan obeng sehingga kecepatan engine menurun. Memutar keluar akan menaikkan kecepatan engine, dan lebih memasukkannya akan mengurangi kecepatan. Dengan perkataan lain, kecepatan engine dapat dinaikkan dengan memutar

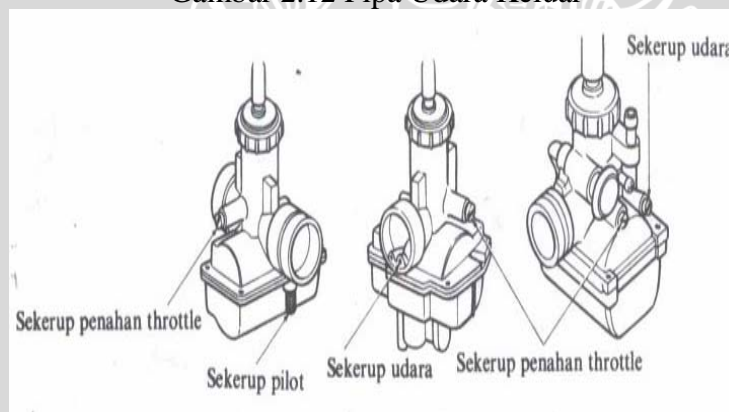
screw masuk atau keluar. Karenanya, bila engine berputar lebih cepat, berhentilah, memutar screw tersebut.



Gambar 2.11 Menyetel Permukaan Apung



Gambar 2.12 Pipa Udara Keluar



Gambar 2.13 Penyetel Kecepatan "Idle"
(Prosedur Berbeda Tergantung Tipe Karburator)

iii. Dengan screw pada posisi ini, kecepatan engine agak tinggi. Karenanya sekerup penahan "Throttle" harus dikendorkan. Bila engine "Idle" pada kecepatan spesifikasi, berhentilah memutar sekerup ini. Pada waktu penyetelan kecepatan "Idle", pakailah *Tachometer* untuk membuat penyetelan yang benar.

2.8 Pengetahuan Pakar (*Expert Knowledge*)

Pengetahuan pakar adalah sumber pengetahuan dari seorang pakar atau beberapa orang pakar sebagai hasil pengalaman kerja bertahun-tahun dalam bidangnya. Pengetahuan ini sering diistilahkan sebagai *Heuristik*. Pengalaman

pemecahan masalah ini dapat dijadikan sumber pengetahuan dari sistim pakar yang dibangun.

Untuk riset ini, *Expert Knowledge* diperoleh dari pakar di Bengkel Mulia Motor, Jalan Raya Bandulan No.1 Malang, Nama mekanik : Yuli, dengan pengalaman kerja selama kurang lebih lima tahun. Beberapa contoh *Expert Knowledge* untuk otomotif troubleshooting dibahas dibawah ini :

2.8.1 Engine

▪ Penyetelan "Clearance" Dari Katup (Engine Empat Langkah)

Seperti terlihat pada Gambar 2.14, (*Clearance*) katup adalah celah di antara sekrup penyetel "Rocker Arm" dan tangkai katup. Bila "Clearance" berlebihan, katup akan memberikan bunyi klik klik sebagai akibat sekrup penyetel "Rocker Arm" memukul tangkai katup. Bila terlalu sedikit, timing dari penutupan katup akan lambat, sehingga mengakibatkan kehilangan kompresi dan pengurangan daya. Karena itu dianjurkan, untuk menyetel "Clearance" katup sesuai spesifikasi manufacturer.

Ada dua macam katup, katup masuk dan katup buang, dengan "Clearance" katup yang berbeda. Kedua "Clearance" itu memang membingungkan, sehingga perhatian khusus harus benar-benar diberikan bila melakukan penyetelan katup. Untuk membuat penyetelan yang benar diperlukan satu set pengukur celah atau "Feeler Gauge". Bila tidak ada, mintalah bengkel servis untuk melakukannya.

- i. "Clearance" katup harus disetel pada saat engine dingin dan dengan kedua katup masuk dan katup buang sama sekali tertutup.
- ii. Pertama-tama, matikanlah "Switch Ignition", dan kemudian masukkan gigi ke posisi netral. Agar mempermudah pekerjaan berikutnya, lepaskanlah tangki bensin (dengan beberapa pengecualian). Pada beberapa model, penyetelan "Clearance" katup tidak mungkin tanpa melepaskan tangki bensin. Dalam hal ini, tutuplah katup bensin yang dipasang pada tangki bensin agar bensin tidak keluar.
- iii. Lepaskanlah busi dan tutup penyetel katup dari kepala silinder. Pada model yang mempunyai tutup lubang "Timing", lepaskanlah tutup tersebut bersama-sama tutup generator.

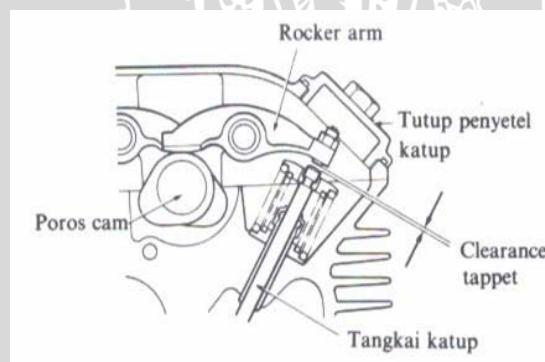
Dengan memakai kunci "Pas", putarlah poros engkol sesuai arah operasi normal sampai tanda "T" pada rotor generator tepat sesuai dengan tanda pada rumah engkol (*Crankcase*), seperti terlihat pada Gambar 2.15. Bila kedua tanda ini sudah tepat, berarti torak (*Piston*) berada di titik mati atas (*Top Dead Center = TDC*) pada langkah kompresi, dan katup masuk dan katup buang kedua-keduanya tertutup sama sekali.

Dengan torak pada posisi ini, gerak utamanya "Rocker Arm" dapat diperiksa dengan menggerak-gerakkan "Rocker Arm" dengan tangan. Perhatikan bahwa rotor dari generator mempunyai tanda lain (untuk penyetelan "Timing" dari "Ignition") di samping tanda "T" tersebut.

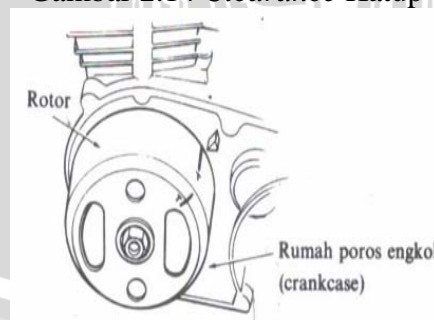
iv. Setelah "Clearance" katup dengan memakai voeler atau "Feeler Gauge" agar sesuai spesifikasi manufacturer umpamanya, bila "Clearance" katup adalah 0,08 mm, pakailah secara rangkap dua lembar dari "Feeler Gauge" masing-masing setebal 0,05 mm dan 0,03 mm.

Seperti terlihat pada Gambar 2.16, masukkanlah "Feeler Gauge" tersebut di antara "Clearance" dari sekerup penyetel "Rocker Arm" dan tangkai katup. Bila "Feeler Gauge" tersebut dapat masuk celah dengan sedikit tahanan, berarti "Clearance" tersebut adalah tepat. Bila bergerak terlalu leluasa atau tidak dapat masuk sama sekali, berarti "Clearance" perlu disetel kembali.

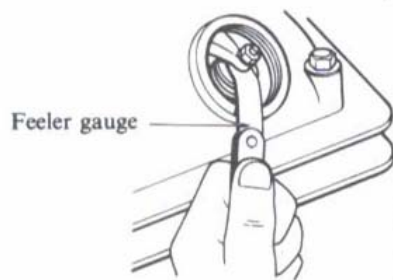
Karena "Feeler Gauge" sangat tipis, pakailah dengan hati-hati agar tidak tertekuk sewaktu memasukkannya ke dalam celah.



Gambar 2.14 Clearance Katup



Gambar 2.15 Tanda-Tanda Timing Katup



Gambar 2.16 Memasukkan Sebuah *Feeler Gauge*

2.8.2 Memeriksa Engine Akan Suara-Suara Abnormal dan Cara-Cara Memperbaikinya

Seperti diterangkan terlebih dahulu, sebuah suara klik klik yang terdengar dari kepala silinder disebabkan "Clearance" katup yang berlebihan.

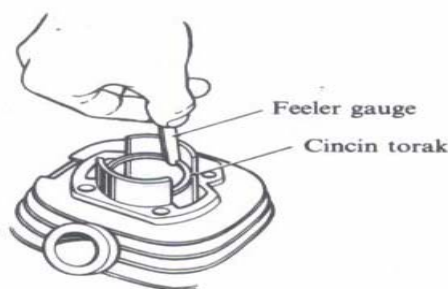
Makin dingin engine, makin keras suara tersebut. Makin tinggi putaran engine suara juga makin keras. Karena itu, adalah relatif mudah untuk menentukan penyebab dari suara. Untuk menyatel "Clearance" katup, ikutilah paragraf terdahulu.

Setiap engine dapat menghasilkan suara abnormal secara tiba-tiba atau secara perlahan-lahan, apakah engine itu dua-langkah ataupun empat langkah. Dalam hal ini, menentukan sebab dari suara akan sangat sukar dengan hanya mendengarkan suara tersebut. Hentikanlah engine segera dan bawalah sepeda motor anda ke bengkel servis terdekat secepatnya agar diperbaiki. Bila anda terus memakai sepeda motor anda tanpa diperbaiki, suatu kerusakan berat dari engine dapat terjadi.

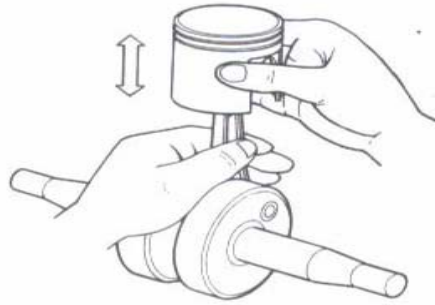
Bila anda ingin memeriksa engine sendiri, ikutilah tindakan-tindakan sebagai :

Seperti terlihat pada Gambar 2.17, pasanglah cincin torak di dalam silinder dan ukurlah celah diantara kedua ujung cincin tersebut dengan "*Feeler Gauge*". Bila celah itu di luar daerah 0,2 sampai 0,4 mm gantilah dengan cincin yang baru.

Bila memeriksa keausan dari ujung kecil batang tengah penggerak (*Connecting Rod*) seperti terlihat pada Gambar 2.18, peganglah bagian tengah penggerak tersebut dan gerakkanlah naik turun.



Gambar 2.17 Memeriksa Cincin Torak (*Piston Ring*)



Gambar 2.18 Memeriksa Gerak Main dari Torak

2.9 Alat pengembangan sistem pakar

Pada pengembangan sistem pakar terdapat beberapa metode yaitu :

1. *Language*

language bisa berarti sebuah penerjemah perintah-perintah yang ditulis dalam aturan tertentu. Bahasa sistem pakar menyediakan mesin inferensi untuk mengeksekusi perintah-perintah dalam bahasa tersebut. Tergantung pada bentuk implementasinya, mesin inferensi bisa digunakan baik pada metode *forward chaining* atau *backward chaining* ataupun gabungan keduanya.

Language dibedakan menjadi dua macam, yaitu :

- *Conventional Language*, contoh : Pascal, Visual Basic, Java, C++
- *Artificial Intelligence Language*, contoh : LISP, Prolog, Small Talk

2. *Tool*

Tool merupakan bahasa pemrograman dengan tambahan fasilitas program (*utility*) untuk mendukung pengembangan, proses pencarian kesalahan program, dan pengiriman data ke program aplikasi. Fasilitas program tersebut termasuk teks, editor grafik, *debugger*, *file management*, dan juga *code generator*. Dalam kasus tertentu, sebuah *tool* dengan berbagai fasilitasnya dapat digabungkan dalam sebuah sistem untuk menghasilkan sebuah tampilan yang umum bagi pengguna. Contoh : *Runtime Application Language* (RAL).

3. *Shell*

Shell merupakan sebuah *tool* dengan tujuan khusus, yaitu untuk tipe aplikasi tertentu dimana pengguna hanya memberikan pengetahuan pada basis pengetahuan. Contoh: *EMYCIN* (*empty MYCIN*) *shell*, *VP-Expert*. Pada bab berikutnya akan dikaji lebih mendalam tentang pengembangan sistem pakar dengan metode shell, yaitu menggunakan *Software VP-Expert*.

2.10 Pengantar VP-Expert

Artificial Intelligence (AI) atau disebut juga kecerdasan buatan merupakan sebuah konsep luas yang meliputi sejumlah bidang dan aplikasi yang berbeda, antara lain *pattern matching*, *machine learning*, *robotics*, *natural language processing*, dan teknologi *expert system*. Teknologi *expert system* atau sistem pakar, dimana merupakan bidang yang difokuskan pada laporan tugas khusus ini, adalah suatu perangkat lunak komputer yang dapat menandingi atau meniru kemampuan manusia dalam memecahkan suatu masalah. Seperti layaknya seorang pakar. Sistem pakar memberikan rekomendasi, dengan cara menarik kesimpulan berdasarkan pengetahuan (*knowledge*) yang dimiliki oleh seorang pakar. Juga dengan cara pengumpulan informasi yang dibutuhkan berkaitan dengan permasalahan yang dihadapi. Pengetahuan tersebut disimpan didalam aturan (*rules*) IF-THEN, sedangkan informasi yang lain yang dibutuhkan diperoleh dari pihak yang membutuhkan rekomendasi, yaitu dengan cara menanyakan pertanyaan-pertanyaan yang berkaitan dengan permasalahan yang dihadapi selama konsultasi tersebut berlangsung. kecerdasan (*Intelligence*) yang dimiliki oleh sistem pakar, yaitu suatu mekanisme yang menggabungkan atau mencocokkan antara pengetahuan dan informasi tersebut untuk kemudian menarik kesimpulan, disebut dengan " *Inference Engine* ".

VP-Expert memungkinkan setiap orang untuk membangun sebuah sistem pakar yang sesungguhnya. Tidak seperti bahasa-bahasa kompleks pemrograman AI yang lainnya, yang mana menuntut penggunanya untuk menguasai keahlian pemrograman-pemrograman rumit untuk dapat membangun sistem pakar dari dasar, *VP-Expert* sangat mudah dan sederhana untuk digunakan.

VP-Expert adalah sebuah alat (*tool*) pengembangan sistem pakar. Hal ini berarti *VP-Expert* menyediakan Mesin Inferensi (*Inference Engine*), antar muka pemakai (*user interface*), dan *commands*, yaitu perintah-perintah yang diperlukan untuk membuat sistem pakar bekerja. Satu-satunya hal yang harus dilakukan saat menggunakan *VP-Expert* adalah mengkodekan pengetahuan dari subyek tertentu kedalam basis pengetahuan (*knowledge base*). Seperti terlihat, penggunaan *VP-Expert*, contohnya dalam hal konstruksi aturan (*rule*) dan *build-in Editor*, menjadikan pembangunan suatu sistem pakar lebih mudah untuk dilaksanakan daripada yang diperkirakan.

2.11 Fasilitas *VP-Expert*

Terdapat banyak alat pengembangan sistem pakar (*expert system tools*) yang dijual dipasaran, tetapi *VP-Expert* menawarkan gabungan dari fasilitas-fasilitas

unggul sehingga menjadikan pilihan yang tepat untuk mengembangkan sistem pakar.

Fasilitas-fasilitas khusus yang disediakan *VP-Expert* tersebut meliputi :

1. Kemampuan untuk bertukar (*sharing*) data dengan VP-Info atau basis data (*File Database*) *dBASE*, *VP – Planner*, *VP – Planner plus*, 1-2-3, atau lembar kerja (*File worksheet*) *Symphony*, dan file teks ASCII.
2. Perintah Induksi (*Induce command*), secara otomatis menghasilkan basis pengetahuan dari tabel yang dibuat dalam file teks, basis data, atau lembar kerja.
3. Mesin Inferensi (*Inference engine*), menggunakan metode *backward* dan *forward chaining* untuk memecahkan permasalahan.
4. Jendela pengembangan (*optional development windows*), mengizinkan pengguna untuk mengamati proses dibalik layar tentang bagaimana cara kerja mesin penyimpul dalam menjalankan basis pengetahuan untuk memecahkan masalah selama konsultasi berlangsung.
5. Faktor kepastian atau keyakinan (*confidence factors*) yang memungkinkan pengguna untuk memperkirakan informasi yang tidak pasti didalam basis pengetahuan.
6. Penggunaan bahasa Inggris sederhana, seperti dalam konstruksi aturannya.
7. Perintah-perintah (*Commands*), mengujinkan *VP-Expert* untuk menjelaskan aksi-aksinya selama kegiatan konsultasi.
8. *Knowledge base chaining*, memungkinkan pengguna untuk membuat basis pengetahuan yang berukuran melebihi ukuran memori.
9. *Built-in* text editor.
10. Pembuat pertanyaan otomatis (*automatic question generation*).
11. Kemampuan untuk mencatat dan menampilkan secara grafis pola pencarian aturan demi aturan yang digunakan dibalik layar selama kegiatan konsultasi.
12. Fungsi matematis *floating point*.
13. Kemampuan untuk mengeksekusi atau menjalankan basis pengetahuan dengan cepat.
14. Kemampuan untuk mengeksekusi atau menjalankan program-program eksternal DOS.
15. *Hypertext*.
16. Gambar-gambar yang dinamis (*Dynamic images*).
17. *Smart forms*.
18. *Graphics primitive*.

19. Dapat menggunakan mouse (*Mouse Support*).

20. Dan masih banyak lagi.

2.12 Penggunaan *VP-Expert*

Sampai saat ini, sistem pakar telah banyak diaplikasikan dalam berbagai bidang kehidupan untuk membantu memecahkan permasalahan tertentu. Bidang kedokteran, mesin, bisnis, geologi, perpajakan, perbankan, dan hukum merupakan sebagian contoh bidang-bidang yang telah menerapkan penggunaan sistem pakar secara nyata. Pada bidang-bidang tersebut, sistem pakar telah sukses digunakan untuk mendiagnosa suatu penyakit, menentukan kelayakan kredit calon debitur, menganalisa struktur suatu bangunan, memberikan pelatihan pada karyawan, dan merokendasikan strategi didalam bisnis. Dari kondisi diatas, dapat disimpulkan bahwa pengetahuan dari berbagai bidang dapat diterapkan dalam suatu sistem pakar.

Kriteria untuk menentukan pantas atau tidaknya suatu sistem pakar diterapkan pada suatu subjek adalah berdasarkan pertanyaan berikut : apakah pengetahuan dari suatu subyek tersebut dapat diterjemahkan dalam aturan IF – THEN ?

RULE 1

IF pengetahuan = dapat_diterjemahkan

THEN subyek = pantas;

Sehingga dapat dikatakan bahwa sistem pakar memiliki kemungkinan tanpa batas untuk diterapkan diberbagai bidang apabila memenuhi kriteria tersebut.

Berhubungan dengan hal tersebut, maka *VP-Expert* tidak hanya dapat digunakan pada instansi-instansi, tetapi juga dapat pula digunakan di rumah untuk membangun suatu sistem pakar yang memberikan rekomendasi atau nasehat dalam hal-hal, seperti mengurus kebun, memasak, merawat anak, merawat tubuh, memperbaiki mobil, dan sebagainya.

Walaupun *VP-Expert* merupakan salah satu alat yang digunakan dengan serius untuk mengembangkan sistem pakar, namun tidak berarti bahwa penggunaan *VP-Expert* dibatasi hanya untuk sistem pakar, terdapat banyak aplikasi lain yang menggunakan aturan pada *VP-Expert* untuk menyimpan pengetahuan Pakar (*Expert Knowledge*). Beberapa aplikasi *VP-Expert* juga menggunakan aturan untuk menghubungkan masukan (*input*) dari pengguna dengan informasi yang disimpan secara eksternal didalam file teks, basis data, atau lembar kerja.

Sebagai tambahan, untuk membangun suatu sistem pakar yang tepat, *VP-Expert* dapat digunakan untuk :

1. Membuat *custom front ends* untuk file lembar kerja dan basis data yang dapat digabungkan.
2. Membuat file teks berindeks untuk akses yang cepat.
3. Mendesain sistem panduan atau pedoman untuk tes atau percobaan dan cara kerja.

2.13 Struktur Utama VP-Expert

Secara garis besar, struktur basis pengetahuan VP-Expert terdiri dari tiga bagian utama, yaitu *actions block*, *rules block*, dan *Questions block*. Tiap bagian struktur utama VP-Expert tersebut dapat dijelaskan sebagai berikut :

1. Blok Aksi (*Actions block*)

Kata kunci *ACTIONS* merupakan tanda awal dari blok aksi. Kata *ACTIONS* berfungsi untuk menentukan dan mengatur agenda untuk konsultasi. Blok aksi ini akan menampilkan pesan pembuka diawal konsultasi, menetapkan *goal variable* (berisi rekomendasi akhir) yang nilainya dicari selama konsultasi berlangsung, dan menampilkan kesimpulan akhir yang diperoleh kepada pengguna melalui sebuah pesan penutup di akhir konsultasi. Umumnya, blok aksi mengandung paling sedikit satu buah kata kunci *FIND*, dimana kata kunci akan mengakibatkan *control* ditransfer ke blok aturan selam konsultasi.

2. Blok Aturan (*Rules Block*)

Blok aturan merupakan “Otak” dari basis pengetahuan yang akan digunakan untuk memecahkan permasalahan yang sedang dikonsultasikan. Blok ini mengandung aturan IF-THEN.

Berikut adalah aturan untuk sebuah VP-Expert:

Struktur Rule untuk VP-Expert

- RULE** label kata kunci RULE diikuti oleh sebuah label.
IF menandai awal dari satu atau lebih kondisi RULE
THEN menandai awal dari sebuah kesimpulan RULE
ELSE kata kunci ELSE ini boleh digunakan, boleh juga tidak, yaitu untuk menandai awal dari kesimpulan alternatif yang mungkin terjadi dari sebuah RULE.
BECAUSE kata kunci BECAUSE ini boleh digunakan, boleh juga tidak, yaitu untuk memberikan penjelasan tentang maksud dari suatu premis, kesimpulan, atau kesimpulan alternative dari RULE (sebagai jawaban dari pertanyaan “mengapa” dan “bagaimana” yang diajukan oleh pengguna kepada sistem).

Sebuah RULE harus diakhiri dengan tanda titik koma (;)

Dalam RULE, kondisi-kondisi dapat dikombinasikan dengan menggunakan operator logika sebagai berikut :

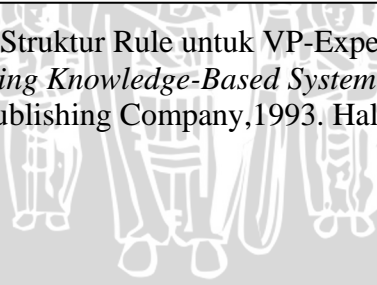
AND kedua kondisi harus bernilai benar (true) untuk dapat menghasilkan sebuah kesimpulan. Dalam sebuah aturan, lebih dari 20 kondisi dapat dikombinasikan dengan menggunakan operator AND.

OR kesimpulan akan diperoleh jika salah satu atau kedua kondisi bernilai benar(true).

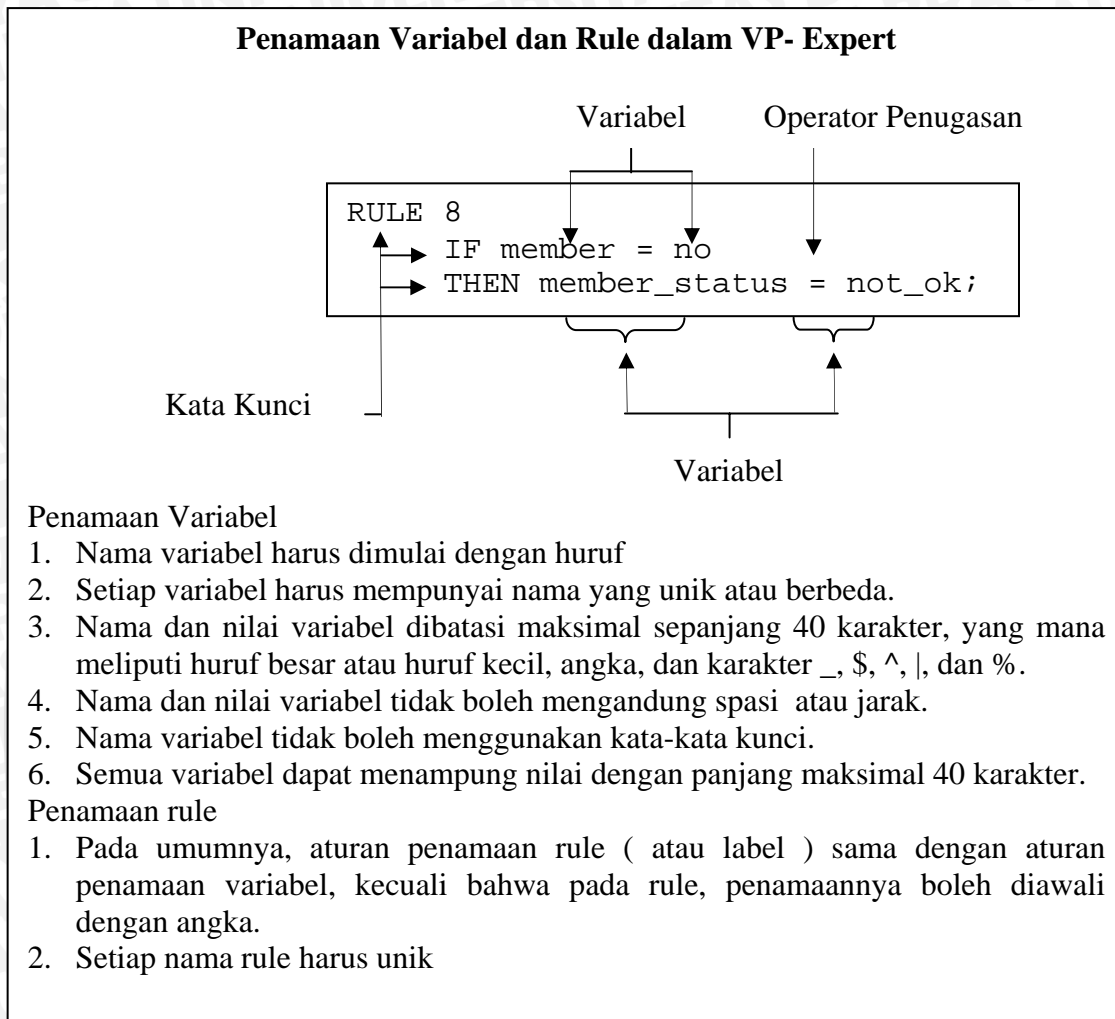
Operator AND dan OR juga dapat digunakan secara bersamaan dalam sebuah aturan. Tetapi, akibatnya hanya 10 kondisi yang dapat dikombinasikan.

Gambar 2.7 Struktur Rule untuk VP-Expert

Sumber : D.G. Dologite. *Developing Knowledge-Based System Using VP-Expert*. New York: Macmillan Publishing Company, 1993. Halaman 30



Berikut akan dijelaskan tentang aturan penamaan variabel dan *rule* dalam VP-Expert:



Gambar 2.8 Penamaan Variabel dan Rule dalam VP-Expert

Sumber : D.G. Dologite. *Developing Knowledge-Based System Using VP-Expert*. New York: Macmillan Publishing Company, 1993. Halaman 31.

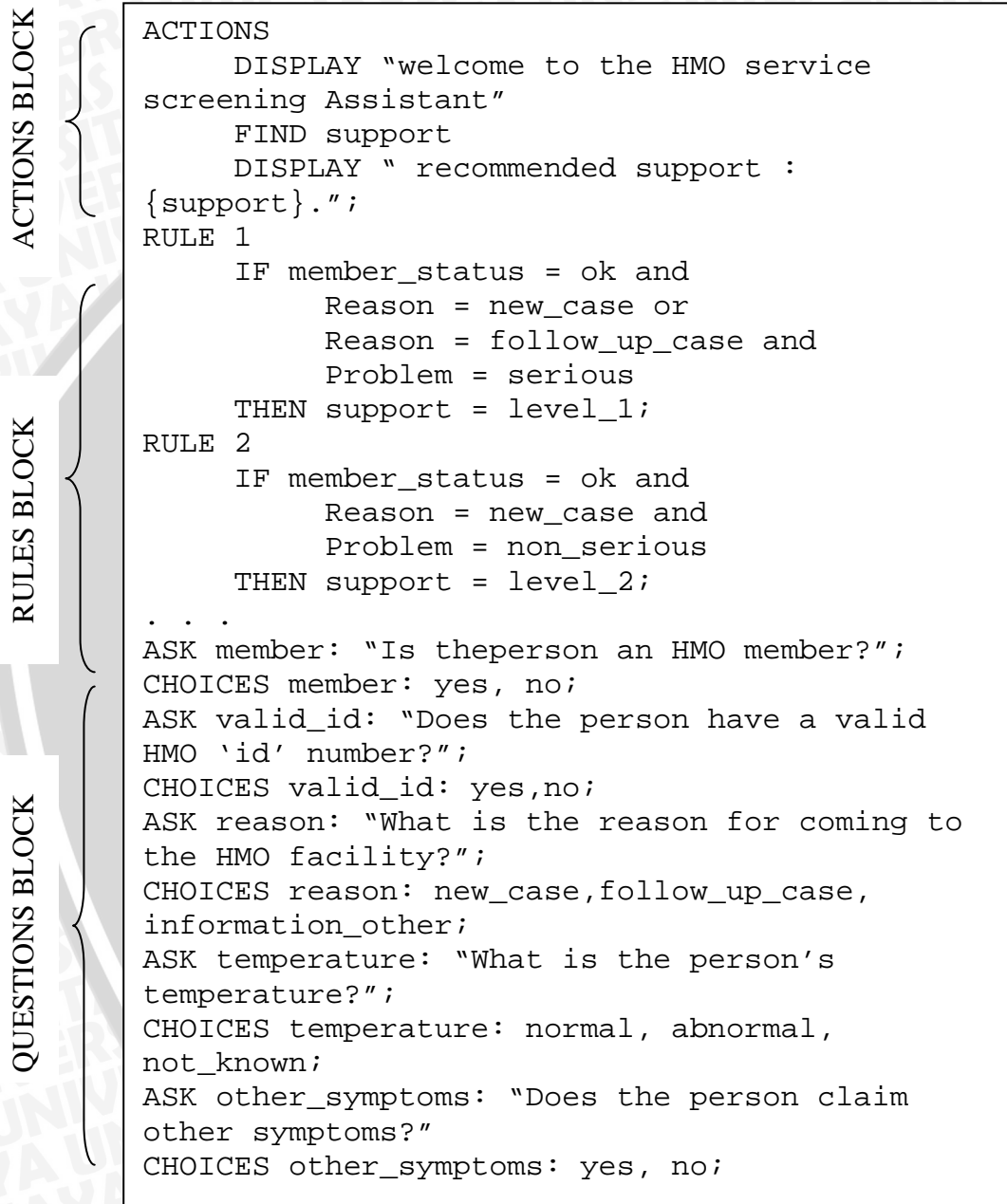
3. Blok Pertanyaan (*Questions Block*)

Seperti halnya blok aturan, blok pertanyaan juga merupakan “otak” dari basis pengetahuan yang akan digunakan untuk memecahkan permasalahan yang sedang dikonsultasikan. Perbedaannya, blok pertanyaan berisi tentang pertanyaan-pertanyaan yang akan diajukan kepada pengguna selama konsultasi berlangsung, disertai dengan pilihan jawabannya untuk mendapatkan informasi yang dibutuhkan sistem dalam usaha memecahkan permasalahan yang dihadapi.

Untuk menkodekan pertanyaan-pertanyaan dalam basis pengetahuan, diperlukan dua buah kata kunci sebagai berikut : ASK, digunakan untuk mengajukan pertanyaan kepada pengguna yang mana jawaban dari pertanyaan tersebut akan

disimpan dalam variable yang belum memiliki nilai dari suatu aturan. CHOICES, digunakan sebagai kata penghubung dari kata kunci ASK, yaitu untuk memberikan batasan nilai atau pilihan jawaban yang diperbolehkan kepada pengguna, yang mana nilai tersebut selanjutnya ditugaskan ke variable yang memintanya.

Berikut merupakan contoh struktur basis pengetahuan VP-Expert :



Gambar 2.9 Struktur Basis Pengetahuan VP-Expert

Sumber : D.G. Dologite. *Developing Knowledge-Based System Using VP-Expert*. New york: Macmillan Publishing Company, 1993. Halaman 34.

Antarmuka Pengguna (*User Interface*) yang digunakan pada saat konsultasi dengan sebuah sistem pakar terdiri dari tiga elemen, yaitu : Pesan Pembuka (*Opening Message*), Pertanyaan-pertanyaan dengan pilihan jawaban, dan Pesan Penutup (*Closing Message*).



BAB III

METODOLOGI PENELITIAN

3.1 Pendahuluan

Metodologi penelitian dan pengembangan merupakan tahap-tahap penelitian yang harus ditetapkan terlebih dahulu sebelum melakukan pemecahan masalah, sehingga penelitian dapat dilakukan secara terarah dan memudahkan didalam menganalisa permasalahan yang ada.

Metodologi penelitian dan pengembangan yang digunakan dalam penyusunan Tugas Akhir ini, dapat diurutkan sebagai berikut:

1. Studi Lapangan
2. Identifikasi Masalah
3. Tujuan Perancangan
4. Studi Literatur
5. Metode Perancangan SP
6. Kesimpulan dan Saran

3.2 Studi Lapangan

Studi lapangan dimaksudkan untuk mencari dan menentukan topik Tugas Akhir yang sesuai konsentrasi, bakat dan minat seorang mahasiswa. Studi lapangan dilaksanakan Untuk riset ini, *Expert Knowledge* diperoleh dari pakar di Bengkel Mulia Motor, Jalan Raya Bandulan No.1 Malang, Nama mekanik : Yuli, dengan pengalaman kerja selama kurang lebih lima tahun. Dari studi lapangan ini, penulis tertarik pada bidang Sistem Pakar (SP).

3.3 Identifikasi Masalah

Setelah topik Tugas Akhir ditentukan (mengenai SP) maka dicari permasalahan yang cocok untuk diexpertsystemkan. Salah satu permasalahan yang didapat adalah tentang troubleshooting sepeda motor, karena mengingat langkanya para pakar kerusakan sepeda motor dan banyaknya para pemakai sepeda motor.

3.4 Tujuan Perancangan

Berdasarkan permasalahan yang ada maka dapat dibuat tujuan perancangan, yaitu hasil yang diharapkan dari perancangan. Tujuan perancangan perlu ditentukan agar penulisan serta perancangan Tugas Akhir ini tidak menyimpang dari yang diinginkan dan dapat mencapai serta mengukur hasil yang optimal. Jadi maksud dari penetapan tujuan perancangan antara lain adalah :

- Agar perancangan lebih sistematis (tahap demi tahap)
- Untuk mengevaluasi keberhasilan rancangan

3.5 Studi Literatur

Langkah selanjutnya adalah melakukan studi literatur. Langkah ini dilakukan dengan mengumpulkan dan mempelajari buku, literatur, jurnal, *website* dan sumber-sumber lain mengenai expert system, serta materi-materi lain yang terkait, seperti : *knowledge representation*, *knowledge base* dan alat bantu yaitu *VP-Expert*.

3.6 Metode Perancangan Expert System

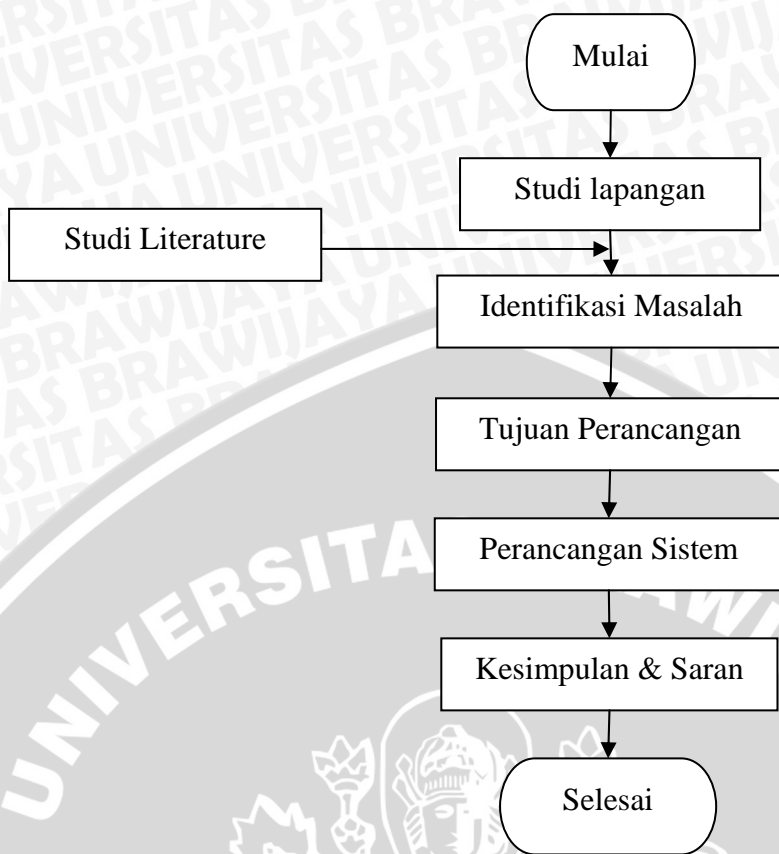
Knowledge base merupakan elemen utama dari ES. Langkah-langkah pengembangan *knowledge base* yang digunakan dalam perancangan ini adalah mengikuti metode pengembangan ES menurut D.G. Dologite (1993:20). Terdapat beberapa langkah dalam pengembangan knowledge base, antara lain :

1. Membuat blok diagram tentang domain pengetahuan.
2. Membuat blok diagram target keputusan (faktor-faktor kritis).
3. Mengubah diagram akhir pada langkah dua (ii) ke bentuk diagram ketergantungan (*Dependency Diagram*).
4. Membuat tabel keputusan (*Decision Table*) sesuai dengan dependency diagram.
5. Mengubah *decision table* menjadi aturan dalam bentuk *IF-THEN*.
6. Memasukkan aturan kedalam program *expert system*.

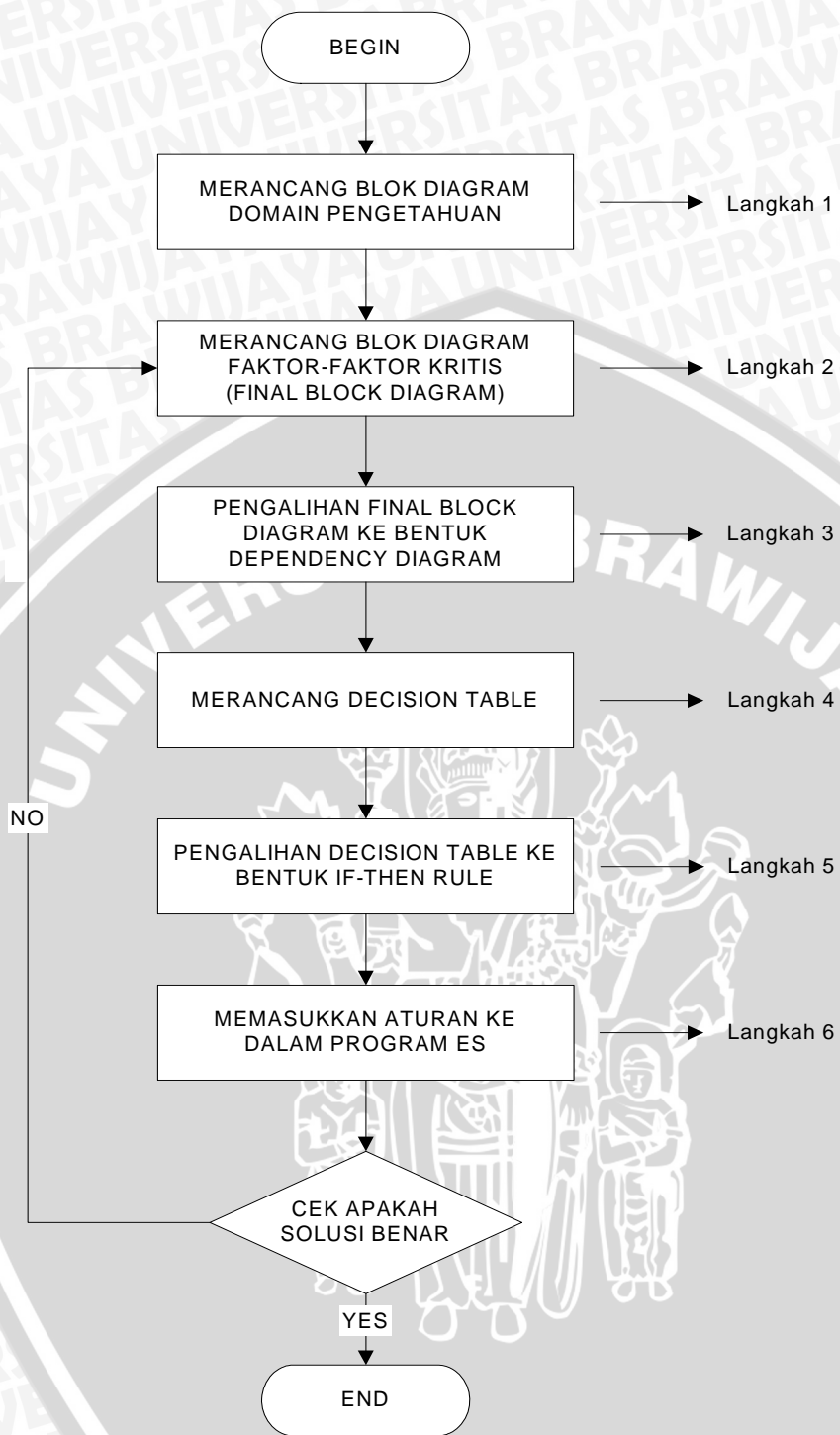
3.7 Kesimpulan dan Saran

Tahap ini merupakan hasil penelitian dan pengembangan secara keseluruhan. Beberapa kesimpulan akan ditarik dan beberapa saran akan diberikan demi pengembangan sistem.

Flowchart metodologi penelitian dan pengembangan dalam Tugas Akhir ini dapat dilihat pada gambar 3.1 sedangkan flowchart perancangan *knowledge base* dapat dilihat pada gambar 3.2.



Gambar 3.1. Flowchart Metodologi Penelitian



Gambar 3.2 Flowchart Perancangan Sistem Pakar

BAB IV

PERANCANGAN SISTEM PAKAR

4.1 Pengantar

Setelah menentukan metodologi penelitian pada BAB III, maka pada bab ini akan difokuskan pada perancangan sistem pakar (SP) dan pengujian program. Perancangan SP yang akan dibahas meliputi perancangan blok diagram domain pengetahuan, perancangan blok diagram faktor-faktor kritis, pengalihan final blok diagram ke bentuk *Dependency Diagram*, membuat decision table dari dependency diagram, membuat aturan dalam bentuk *If-Then*, merancang user interface dan memasukkan aturan kedalam program SP. Sesuai dengan batasan masalah maka perancangan SP ini diperuntukan bagi *troubleshooting* kerusakan sepeda motor Yamaha, setelah perancangan selesai maka dilakukan uji verifikasi dan validasi.

4.2 Perancangan Sistem Pakar

4.2.1 Perancangan Blok Diagram Domain Pengetahuan

Pada langkah ini akan dibuat blok diagram secara umum dari domain pengetahuan yang dipilih. Domain pengetahuan yang dipilih adalah mengenai *troubleshooting*. Pada langkah ini ditentukan area dari domain pengetahuan yang akan diprototipekan. Gambar 4.1 berikut adalah bentuk blok diagram untuk domain yang dimaksud. Dari gambar nampak bahwa fokus pengembangan SP pada masalah mesin dan non mesin. Block diagram pada gambar 4.2 menjelaskan faktor-faktor kritis yang mempengaruhi rekomendasi yang akan diberikan oleh SP.

4.2.2 Pembentukan Dependency Diagram

Dependency diagram merupakan diagram yang mengindikasikan hubungan antara pertanyaan, aturan, nilai dan rekomendasi dari suatu *knowledge base*. Bentuk segitiga menunjukkan himpunan rule (rule set) dan nomor dari himpunan tersebut. Bentuk kotak menunjukkan hasil dari rule baik berupa kesimpulan awal, fakta baru maupun keputusan atau hasil akhir. Sedangkan tanda tanya menunjukkan kondisi yang mempengaruhi suatu rule set. Gambar 4.3 adalah bentuk *dependency diagram* *troubleshooting* sepeda motor berdasarkan block diagram sebelumnya.

4.2.3 Perancangan Decision Table

Perancangan *Decision Table* (DT) dilakukan berdasarkan bentuk segitiga pada dependency diagram (Gambar 4.3). DT diperlukan untuk menunjukkan hubungan

antara nilai-nilai dengan hasil atau keluaran yang didapat dari tahap menengah atau dari rekomendasi akhir. Gambar 4.4 adalah bentuk *Decision Table* (DT) troubleshooting sepeda motor berdasarkan bentuk segitiga pada dependency diagram (Gambar 4.3) sebelumnya.

4.2.4 Perancangan Aturan (*Rules*)

Setelah DT terbentuk, maka dilakukan perubahan dari DT menjadi aturan berbentuk IF-THEN. Pada dasarnya, struktur penulisan sebuah rule dimulai dengan kata kunci IF yang diikuti oleh kondisi yang akan dievaluasi. Serangkaian kondisi (lebih dari satu kondisi) dapat dihubungkan dengan menggunakan operator logika AND dan OR. Jika semua kondisi dalam rule terpenuhi, maka klausa THEN, atau kesimpulan, akan diputuskan. Hal ini mengakibatkan variable yang terletak disebelah kiri dari tanda sama dengan dalam klausa THEN menjadi menyimpan nilai dari variable yang terletak disebelah kanan tanda sama dengan.

Berikut ini adalah contoh beberapa rules yang digunakan dalam sistem pakar untuk troubleshooting sepeda motor.

```
! ===== RULES BLOCK =====
! ===== RULES 1 =====

RULE 1_1
IF   BLOCKA = OK AND
     A = YA
THEN SUPPORT = BLOCKA;

RULE 1_2
IF   A = YA
THEN BLOCKA = OK;

RULE 1_3
IF   A = YA
THEN BLOCKA = NOT_OK;

RULE 1_4
IF   A = TIDAK
THEN BLOCKA = NOT_OK;

RULE MESSAGE_BLOCKA
IF   SUPPORT = BLOCKA
THEN MESSAGE = TROUBLESHOOTING_SEPEDA_MOTOR
      DISPLAY "
1 = TROUBLSHOOTING DI MESIN
2 = TROUBLESHOTING DI NON MESIN
{MESSAGE}.";

! ===== RULES 2 =====
RULE 2_1
IF   BLOCKB = OK AND
     B = YA
THEN SUPPORT = BLOCKB;
```

```
RULE 2_2
IF B = YA
THEN BLOCKB = OK;

RULE 2_3
IF B = YA
THEN BLOCKB = NOT_OK;

RULE 2_4
IF B = TIDAK
THEN BLOCKB = NOT_OK;

RULE MESSAGE_BLOCKB
IF SUPPORT = BLOCKB
THEN MESSAGE = KERUSAKAN_PADA_MESIN
DISPLAY "
1 = KUNCI
2 = BENJIN
3 = BUSI
4 = KABEL BUSI
5 = FILTER UDARA
6 = KNALPOT
7 = KARBURATOR
8 = ACCU
9 = MOTOR
10 = OLI {MESSAGE}.";
```

```
! ===== RULES 3 =====
```

```
RULE 3_1
IF BLOCKB1 = OK AND
B1 = YA
THEN SUPPORT = BLOCKB1;

RULE 3_2
IF B1 = YA
THEN BLOCKB1 = OK;

RULE 3_3
IF B1 = YA
THEN BLOCKB1 = NOT_OK;

RULE 3_4
IF B1 = TIDAK
THEN BLOCKB1 = NOT_OK;

RULE MESSAGE_BLOCKB1
IF SUPPORT = BLOCKB1
THEN MESSAGE = SEPULUH_KALI
DISPLAY " COBA TERUS {MESSAGE}.";
```

```
! ===== RULES 4 =====
```

```
RULE 4_1
IF BLOCKB1A = OK AND
B1A = YA
THEN SUPPORT = BLOCKB1A;

RULE 4_2
IF B1A = YA
THEN BLOCKB1A = OK;
```




```

RULE 4_3
IF B1A = YA
THEN BLOCKB1A = NOT_OK;

RULE 4_4
IF B1A = TIDAK
THEN BLOCKB1A = NOT_OK;

RULE MESSAGE_BLOCKB1A
IF SUPPORT = BLOCKB1A
THEN MESSAGE = TIGA_KALI
    DISPLAY " COBA SEKALI LAGI {MESSAGE}.";

! ===== RULES 5 =====

RULE 5_1
IF BLOCKB1B = OK AND
    B1B = YA
THEN SUPPORT = BLOCKB1B;

RULE 5_2
IF B1B = YA
THEN BLOCKB1B = OK;

RULE 5_3
IF B1B = YA
THEN BLOCKB1B = NOT_OK;

RULE 5_4
IF B1B = TIDAK
THEN BLOCKB1B = NOT_OK;

RULE MESSAGE_BLOCKB1B
IF SUPPORT = BLOCKB1B
THEN MESSAGE = DENGAN_TELITI
    DISPLAY " PERIKSA DAN CEK {MESSAGE}.";

! ===== RULES 6 =====

RULE 6_1
IF BLOCKB1C = OK AND
    B1C = YA
THEN SUPPORT = BLOCKB1C;

RULE 6_2
IF B1C = YA
THEN BLOCKB1C = OK;

RULE 6_3
IF B1C = YA
THEN BLOCKB1C = NOT_OK;

RULE 6_4
IF B1C = TIDAK
THEN BLOCKB1C = NOT_OK;

RULE MESSAGE_BLOCKB1C
IF SUPPORT = BLOCKB1C
THEN MESSAGE = PERIKSA_DAN_CEK_DENGAN_TELITI
    DISPLAY "
    1 = AKI DALAM KEADAAN TEGANGAN RENDAH 'KURANG TERISI'
    2 = TERMINAL AKI LONGGAR ATAU DALAM KONTAK YANG BURUK

```



3 = SWITCH STARTER ATAU RELAY STARTER RUSAK
 4 = MOTOR STARTER RUSAK {MESSAGE}.";

! ===== RULES 7 =====

RULE 7_1
 IF BLOCKB1D = OK AND
 B1D = YA
 THEN SUPPORT = BLOCKB1D;

RULE 7_2
 IF B1D = YA
 THEN BLOCKB1D = OK;

RULE 7_3
 IF B1D = YA
 THEN BLOCKB1D = NOT_OK;

RULE 7_4
 IF B1D = TIDAK
 THEN BLOCKB1D = NOT_OK;

RULE MESSAGE_BLOCKB1D
 IF SUPPORT = BLOCKB1D
 THEN MESSAGE = PERIKSA_DAN_CEK_DENGAN_TELITI
 DISPLAY "
 1 = TANGKI BENSIN KOSONG
 2 = KARBURATOR BANJIR
 3 = BUSI BASAH
 4 = SISTIM IGNITION RUSAK
 5 = TERLALU PANAS
 6 = KATUP TIDAK DISETEL DENGAN BAIK
 'HANYA ENGINE EMPAT LANGKAH' {MESSAGE}.";

! ===== RULES 8 =====

RULE 8_1
 IF BLOCKB1E = OK AND
 B1E = YA
 THEN SUPPORT = BLOCKB1E;

RULE 8_2
 IF B1E = YA
 THEN BLOCKB1E = OK;

RULE 8_3
 IF B1E = YA
 THEN BLOCKB1E = NOT_OK;

RULE 8_4
 IF B1E = TIDAK
 THEN BLOCKB1E = NOT_OK;

RULE MESSAGE_BLOCKB1E
 IF SUPPORT = BLOCKB1E
 THEN MESSAGE = PERIKSA_DAN_CEK_DENGAN_TELITI
 DISPLAY "
 1 = KOPLING STARTER SLIP
 2 = RODA GIGI STARTER PATAH {MESSAGE}.";

```
! ===== RULES 9 =====
```

```
RULE 9_1
IF   BLOCKB2 = OK AND
     B2 = YA
THEN SUPPORT = BLOCKB2;
```

```
RULE 9_2
IF   B2 = YA
THEN BLOCKB2 = OK;
```

```
RULE 9_3
IF   B2 = YA
THEN BLOCKB2 = NOT_OK;
```

```
RULE 9_4
IF   B2 = TIDAK
THEN BLOCKB2 = NOT_OK;
```

```
RULE MESSAGE_BLOCKB2
IF   SUPPORT = BLOCKB2
THEN MESSAGE = SEPULUH_KALI
      DISPLAY " COBA TERUS {MESSAGE}.";
```

```
! ===== RULES 10 =====
```

```
RULE 10_1
IF   BLOCKB2A = OK AND
     B2A = YA
THEN SUPPORT = BLOCKB2A;
```

```
RULE 10_2
IF   B2A = YA
THEN BLOCKB2A = OK;
```

```
RULE 10_3
IF   B2A = YA
THEN BLOCKB2A = NOT_OK;
```

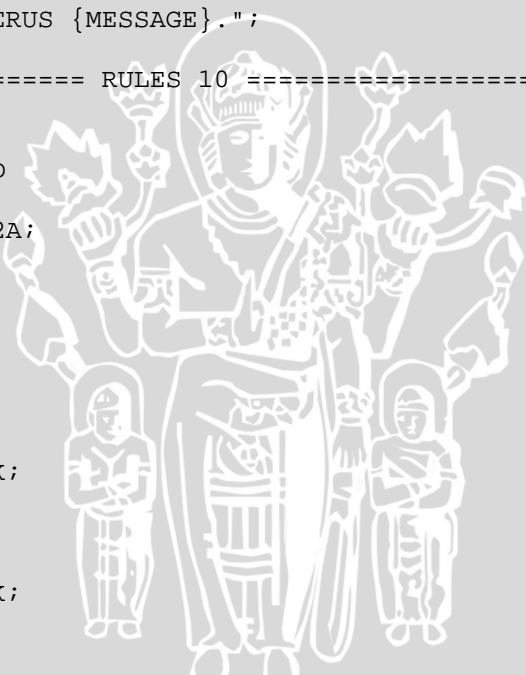
```
RULE 10_4
IF   B2A = TIDAK
THEN BLOCKB2A = NOT_OK;
```

```
RULE MESSAGE_BLOCKB2A
IF   SUPPORT = BLOCKB2A
THEN MESSAGE = TIGA_KALI
      DISPLAY " COBA SEKALI LAGI {MESSAGE}.";
```

```
! ===== RULES 11 =====
```

```
RULE 11_1
IF   BLOCKB2B = OK AND
     B2B = YA
THEN SUPPORT = BLOCKB2B;
```

```
RULE 11_2
IF   B2B = YA
THEN BLOCKB2B = OK;
```



```
RULE 11_3
IF B2B = YA
THEN BLOCKB2B = NOT_OK;
```

```
RULE 11_4
IF B2B = TIDAK
THEN BLOCKB2B = NOT_OK;
```

```
RULE MESSAGE_BLOCKB2B
IF SUPPORT = BLOCKB2B
THEN MESSAGE = DENGAN_TELITI
DISPLAY " PERIKSA DAN CEK {MESSAGE}.";
```

! ===== RULES 12 =====

```
RULE 12_1
IF BLOCKB2C = OK AND
B2C = YA
THEN SUPPORT = BLOCKB2C;
```

```
RULE 12_2
IF B2C = YA
THEN BLOCKB2C = OK;
```

```
RULE 12_3
IF B2C = YA
THEN BLOCKB2C = NOT_OK;
```

```
RULE 12_4
IF B2C = TIDAK
THEN BLOCKB2C = NOT_OK;
```

```
RULE MESSAGE_BLOCKB2C
IF SUPPORT = BLOCKB2C
THEN MESSAGE = PERIKSA_DAN_CEK_DENGAN_TELITI
DISPLAY "
1 = TORAK MACET
2 = BANTALAN POROS ENKGOL MACET {MESSAGE}.";
```

! ===== RULES 13 =====

```
RULE 13_1
IF BLOCKB2D = OK AND
B2D = YA
THEN SUPPORT = BLOCKB2D;
```

```
RULE 13_2
IF B2D = YA
THEN BLOCKB2D = OK;
```

```
RULE 13_3
IF B2D = YA
THEN BLOCKB2D = NOT_OK;
```

```
RULE 13_4
IF B2D = TIDAK
THEN BLOCKB2D = NOT_OK;
```

```
RULE MESSAGE_BLOCKB2D
IF SUPPORT = BLOCKB2D
THEN MESSAGE = PERIKSA_DAN_CEK_DENGAN_TELITI
```



```

DISPLAY "
1 = AKI DALAM KEADAAN TEGANGAN RENDAH 'IGNITION DENGAN AKI'
2 = BENSIN HABIS
3 = KARBURATOR BANJIR
4 = BUSI BASAH
5 = SISTIM IGNITION RUSAK
6 = TERLALU PANAS
7 = KATUP TIDAK DISETEL DENGAN BAIK
'HANYA ENGINE EMPAT LANGKAH' {MESSAGE}.";

```

```
! ===== RULES 14 =====
```

```

RULE 14_1
IF   BLOCKB2E = OK AND
     B2E = YA
THEN SUPPORT = BLOCKB2E;

```

```

RULE 14_2
IF   B2E = YA
THEN BLOCKB2E = OK;

```

```

RULE 14_3
IF   B2E = YA
THEN BLOCKB2E = NOT_OK;

```

```

RULE 14_4
IF   B2E = TIDAK
THEN BLOCKB2E = NOT_OK;

```

```

RULE MESSAGE_BLOCKB2E
IF   SUPPORT = BLOCKB2E
THEN MESSAGE = PERIKSA_DAN_CEK_DENGAN_TELITI
DISPLAY "
1 = RODA GIGI KICKSTARTER PATAH
2 = KOPLING KICKSTARTER RUSAK {MESSAGE}.";

```

```
! ===== RULES 15 =====
```

```

RULE 15_1
IF   BLOCKB3 = OK AND
     B3 = YA
THEN SUPPORT = BLOCKB3;

```

```

RULE 15_2
IF   B3 = YA
THEN BLOCKB3 = OK;

```

```

RULE 15_3
IF   B3 = YA
THEN BLOCKB3 = NOT_OK;

```

```

RULE 15_4
IF   B3 = TIDAK
THEN BLOCKB3 = NOT_OK;

```

```

RULE MESSAGE_BLOCKB3
IF   SUPPORT = BLOCKB3
THEN MESSAGE = DENGAN_TELITI
DISPLAY " CEK DAN PERIKSALAH {MESSAGE}.";

```

! ===== RULES 16 =====

```

RULE 16_1
IF   BLOCKB3A = OK AND
     B3A = YA
THEN SUPPORT = BLOCKB3A;

RULE 16_2
IF   B3A = YA
THEN BLOCKB3A = OK;

RULE 16_3
IF   B3A = YA
THEN BLOCKB3A = NOT_OK;

RULE 16_4
IF   B3A = TIDAK
THEN BLOCKB3A = NOT_OK;

RULE MESSAGE_BLOCKB3A
IF   SUPPORT = BLOCKB3A
THEN MESSAGE = DENGAN_TELITI
     DISPLAY " CEK DAN PERIKSALAH {MESSAGE}.";

```

! ===== RULES 17 =====

```

RULE 17_1
IF   BLOCKB3B = OK AND
     B3B = YA
THEN SUPPORT = BLOCKB3B;

RULE 17_2
IF   B3B = YA
THEN BLOCKB3B = OK;

RULE 17_3
IF   B3B = YA
THEN BLOCKB3B = NOT_OK;

RULE 17_4
IF   B3B = TIDAK
THEN BLOCKB3B = NOT_OK;

RULE MESSAGE_BLOCKB3B
IF   SUPPORT = BLOCKB3B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
     DISPLAY "
1 = SARINGAN UDARA TERSUMBAT
2 = TIMING IGNITION TERLAMBAT
3 = KOPLING SLIP
4 = REM MENAHAN
5 = KARBURATOR RUSAK
6 = ALIRAN BAHAN BAKAR TERSUMBAT
7 = ENGINE TERLALU PANAS
8 = CINCIN TORAK PATAH
9 = TORAK ATAU SILINDER AUS {MESSAGE}.";

```

! ===== RULES 18 =====

```

RULE 18_1
IF   BLOCKB3C = OK AND
     B3C = YA

```

```

THEN SUPPORT = BLOCKB3C;

RULE 18_2
IF B3C = YA
THEN BLOCKB3C = OK;

RULE 18_3
IF B3C = YA
THEN BLOCKB3C = NOT_OK;

RULE 18_4
IF B3C = TIDAK
THEN BLOCKB3C = NOT_OK;

RULE MESSAGE_BLOCKB3C
IF SUPPORT = BLOCKB3C
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = KECEPATAN IDLE TIDAK DISETEL SECARA TEPAT
2 = SISTIM IGNITION RUSAK 'MISFIRING'
'TIMING IGNITION YANG SALAH' DAN LAIN LAIN
3 = ADA UDARA SEKUNDER YANG TERHISAP
4 = CLEARANCE KATUP TIDAK DISETEL DENGAN TEPAT {MESSAGE}.";

! ===== RULES 19 =====

RULE 19_1
IF BLOCKB4 = OK AND
B4 = YA
THEN SUPPORT = BLOCKB4;

RULE 19_2
IF B4 = YA
THEN BLOCKB4 = OK;

RULE 19_3
IF B4 = YA
THEN BLOCKB4 = NOT_OK;

RULE 19_4
IF B4 = TIDAK
THEN BLOCKB4 = NOT_OK;

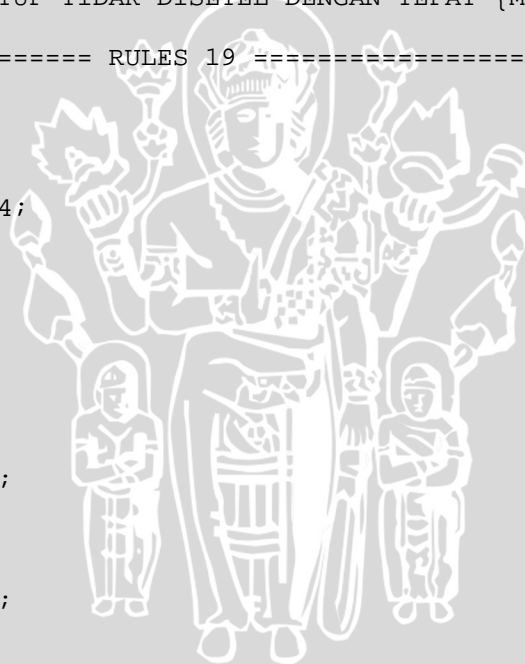
RULE MESSAGE_BLOCKB4
IF SUPPORT = BLOCKB4
THEN MESSAGE = DENGAN_TELITI
DISPLAY " CEK DAN PERIKSALAH {MESSAGE}.";

! ===== RULES 20 =====

RULE 20_1
IF BLOCKB4A = OK AND
B4A = YA
THEN SUPPORT = BLOCKB4A;

RULE 20_2
IF B4A = YA
THEN BLOCKB4A = OK;

```



```
RULE 20_3
IF B4A = YA
THEN BLOCKB4A = NOT_OK;
```

```
RULE 20_4
IF B4A = TIDAK
THEN BLOCKB4A = NOT_OK;
```

```
RULE MESSAGE_BLOCKB4A
IF SUPPORT = BLOCKB4A
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = KEBOCORAN OLI
2 = OLI NAIK DAN TERBAKAR
3 = OLI TURUN DAN TERBAKAR 'EMPAT LANGKAH'
4 = POMPA OLI TIDAK DISETEL DENGAN TEPAT {MESSAGE}.";
```

```
! ===== RULES 21 =====
```

```
RULE 21_1
IF BLOCKB4B = OK AND
B4B = YA
THEN SUPPORT = BLOCKB4B;
```

```
RULE 21_2
IF B4B = YA
THEN BLOCKB4B = OK;
```

```
RULE 21_3
IF B4B = YA
THEN BLOCKB4B = NOT_OK;
```

```
RULE 21_4
IF B4B = TIDAK
THEN BLOCKB4B = NOT_OK;
```

```
RULE MESSAGE_BLOCKB4B
IF SUPPORT = BLOCKB4B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = OLI BERCAMPUR DENGAN GAS
2 = OLI BERCAMPUR DENGAN AIR PENDINGIN
'ENGINE DENGAN PENDINGIN AIR' {MESSAGE}.";
```

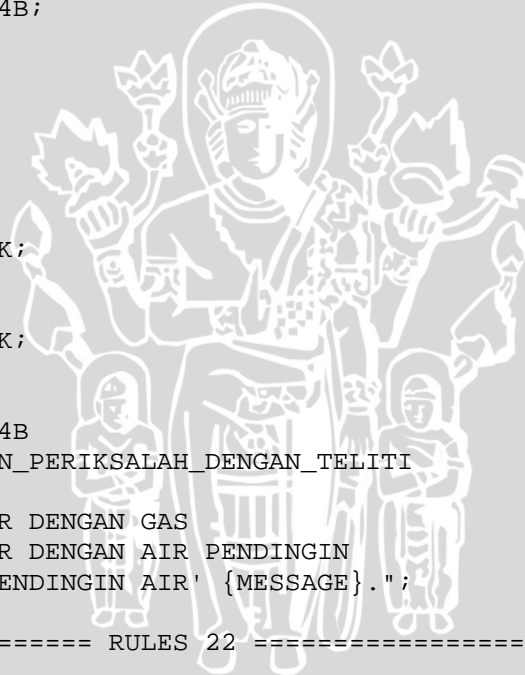
```
! ===== RULES 22 =====
```

```
RULE 22_1
IF BLOCKB4C = OK AND
B4C = YA
THEN SUPPORT = BLOCKB4C;
```

```
RULE 22_2
IF B4C = YA
THEN BLOCKB4C = OK;
```

```
RULE 22_3
IF B4C = YA
THEN BLOCKB4C = NOT_OK;
```

```
RULE 22_4
IF B4C = TIDAK
THEN BLOCKB4C = NOT_OK;
```




```

RULE MESSAGE_BLOCKB4C
IF SUPPORT = BLOCKB4C
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = PENGARUH PEMBERSIHAN OLI 'NORMAL'
2 = OLI BERISI KOTORAN KOTORAN
3 = OLI MEROSOT KUALITASNYA {MESSAGE}.";

```

! ===== RULES 23 =====

```

RULE 23_1
IF BLOCKB4D = OK AND
B4D = YA
THEN SUPPORT = BLOCKB4D;

```

```

RULE 23_2
IF B4D = YA
THEN BLOCKB4D = OK;

```

```

RULE 23_3
IF B4D = YA
THEN BLOCKB4D = NOT_OK;

```

```

RULE 23_4
IF B4D = TIDAK
THEN BLOCKB4D = NOT_OK;

```

```

RULE MESSAGE_BLOCKB4D
IF SUPPORT = BLOCKB4D
THEN MESSAGE = DENGAN_TELITI
DISPLAY " CEK DAN PERIKSALAH {MESSAGE}.";

```

! ===== RULES 24 =====

```

RULE 24_1
IF BLOCKB4E = OK AND
B4E = YA
THEN SUPPORT = BLOCKB4E;

```

```

RULE 24_2
IF B4E = YA
THEN BLOCKB4E = OK;

```

```

RULE 24_3
IF B4E = YA
THEN BLOCKB4E = NOT_OK;

```

```

RULE 24_4
IF B4E = TIDAK
THEN BLOCKB4E = NOT_OK;

```

```

RULE MESSAGE_BLOCKB4E
IF SUPPORT = BLOCKB4E
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = BENSIN KOTOR
2 = KARBURATOR RUSAK
3 = SARINGAN UDARA TERSUMBAT
4 = SISTIM IGNITION RUSAK
5 = KOMPRESI RENDAH

```

6 = MUFFLER TERSUMBAT
 7 = KOPLING SLIP {MESSAGE}."

! ===== RULES 25 =====

RULE 25_1
 IF BLOCKB4F = OK AND
 B4F = YA
 THEN SUPPORT = BLOCKB4F;

RULE 25_2
 IF B4F = YA
 THEN BLOCKB4F = OK;

RULE 25_3
 IF B4F = YA
 THEN BLOCKB4F = NOT_OK;

RULE 25_4
 IF B4F = TIDAK
 THEN BLOCKB4F = NOT_OK;

RULE MESSAGE_BLOCKB4F
 IF SUPPORT = BLOCKB4F
 THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
 DISPLAY "
 1 = REM MENAHAN
 2 = TEKANAN BAN RENDAH {MESSAGE}."

! ===== RULES 26 =====

RULE 26_1
 IF BLOCKB4G = OK AND
 B4G = YA
 THEN SUPPORT = BLOCKB4G;

RULE 26_2
 IF B4G = YA
 THEN BLOCKB4G = OK;

RULE 26_3
 IF B4G = YA
 THEN BLOCKB4G = NOT_OK;

RULE 26_4
 IF B4G = TIDAK
 THEN BLOCKB4G = NOT_OK;

RULE MESSAGE_BLOCKB4G
 IF SUPPORT = BLOCKB4G
 THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
 DISPLAY "
 1 = RODA GIGI TIDAK DIPAKAI SECARA BENAR
 2 = ENGINE SERING BERPUTAR TERLALU CEPAT
 3 = KEBIASAAN MEMAKAI PERCEPATAN YANG BESAR
 4 = BEBAN TERLALU BESAR {MESSAGE}."



! ===== RULES 27 =====

```
RULE 27_1
IF   BLOCKB5 = OK AND
     B5 = YA
THEN SUPPORT = BLOCKB5;
```

```
RULE 27_2
IF   B5 = YA
THEN BLOCKB5 = OK;
```

```
RULE 27_3
IF   B5 = YA
THEN BLOCKB5 = NOT_OK;
```

```
RULE 27_4
IF   B5 = TIDAK
THEN BLOCKB5 = NOT_OK;
```

```
RULE MESSAGE_BLOCKB5
IF   SUPPORT = BLOCKB5
THEN MESSAGE = DENGAN_TELITI
      DISPLAY " CEK DAN PERIKSALAH {MESSAGE}.";
```

! ===== RULES 28 =====

```
RULE 28_1
IF   BLOCKB5A = OK AND
     B5A = YA
THEN SUPPORT = BLOCKB5A;
```

```
RULE 28_2
IF   B5A = YA
THEN BLOCKB5A = OK;
```

```
RULE 28_3
IF   B5A = YA
THEN BLOCKB5A = NOT_OK;
```

```
RULE 28_4
IF   B5A = TIDAK
THEN BLOCKB5A = NOT_OK;
```

```
RULE MESSAGE_BLOCKB5A
IF   SUPPORT = BLOCKB5A
THEN MESSAGE = DENGAN_TELITI
      DISPLAY " CEK DAN PERIKSALAH {MESSAGE}.";
```

! ===== RULES 29 =====

```
RULE 29_1
IF   BLOCKB5B = OK AND
     B5B = YA
THEN SUPPORT = BLOCKB5B;
```

```
RULE 29_2
IF   B5B = YA
THEN BLOCKB5B = OK;
```



```
RULE 29_3
IF B5B = YA
THEN BLOCKB5B = NOT_OK;
```

```
RULE 29_4
IF B5B = TIDAK
THEN BLOCKB5B = NOT_OK;
```

```
RULE MESSAGE_BLOCKB5B
IF SUPPORT = BLOCKB5B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = CHOKE TIDAK DITUTUP SECARA TEPAT
2 = KARBURATOR TIDAK DISETEL SECARA TEPAT
3 = SARINGAN UDARA TERSUMBAT {MESSAGE}.";
```

```
! ===== RULES 30 =====
```

```
RULE 30_1
IF BLOCKB5C = OK AND
B5C = YA
THEN SUPPORT = BLOCKB5C;
```

```
RULE 30_2
IF B5C = YA
THEN BLOCKB5C = OK;
```

```
RULE 30_3
IF B5C = YA
THEN BLOCKB5C = NOT_OK;
```

```
RULE 30_4
IF B5C = TIDAK
THEN BLOCKB5C = NOT_OK;
```

```
RULE MESSAGE_BLOCKB5C
IF SUPPORT = BLOCKB5C
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = OLI NAIK DAN TERBAKAR
2 = OLI TURUN DAN TERBAKAR 'EMPAT LANGKAH'
3 = POMPA OLI TIDAK DISETEL DENGAN BAIK 'DUA LANGKAH'
{MESSAGE}.";
```

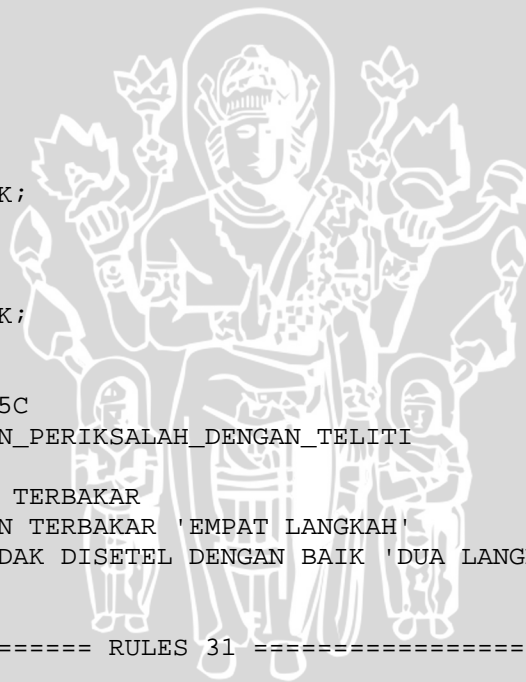
```
! ===== RULES 31 =====
```

```
RULE 31_1
IF BLOCKB6 = OK AND
B6 = YA
THEN SUPPORT = BLOCKB6;
```

```
RULE 31_2
IF B6 = YA
THEN BLOCKB6 = OK;
```

```
RULE 31_3
IF B6 = YA
THEN BLOCKB6 = NOT_OK;
```

```
RULE 31_4
IF B6 = TIDAK
THEN BLOCKB6 = NOT_OK;
```



```
RULE MESSAGE_BLOCKB6
IF SUPPORT = BLOCKB6
THEN MESSAGE = DENGAN_TELITI
    DISPLAY " CEK DAN PERIKSALAH {MESSAGE}.";
```

```
! ===== RULES 32 =====
```

```
RULE 32_1
IF BLOCKB6A = OK AND
    B6A = YA
THEN SUPPORT = BLOCKB6A;
```

```
RULE 32_2
IF B6A = YA
THEN BLOCKB6A = OK;
```

```
RULE 32_3
IF B6A = YA
THEN BLOCKB6A = NOT_OK;
```

```
RULE 32_4
IF B6A = TIDAK
THEN BLOCKB6A = NOT_OK;
RULE MESSAGE_BLOCKB6A
IF SUPPORT = BLOCKB6A
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
    DISPLAY "
        1 = TERLALU PANAS
        2 = KWALITAS BENSIN RENDAH 'ANGKA OKTAN RENDAH'
        3 = TIMING IGNITION TERLALU DINI
        4 = LAPISAN KARBON DI DALAM RUANG BAKAR {MESSAGE}.";
```

```
! ===== RULES 33 =====
```

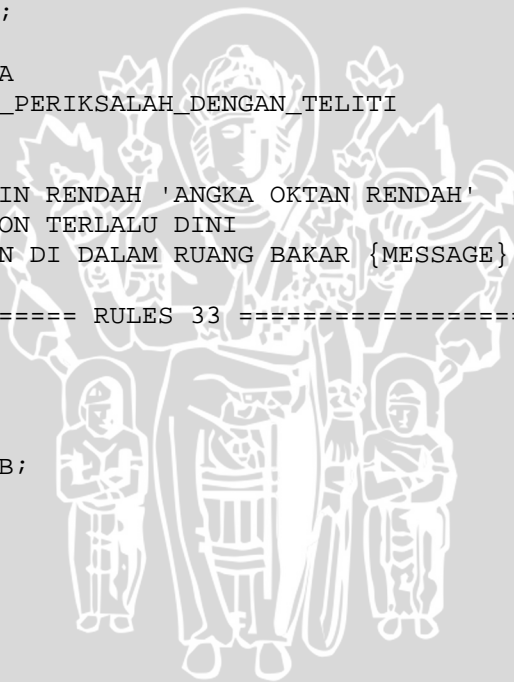
```
RULE 33_1
IF BLOCKB6B = OK AND
    B6B = YA
THEN SUPPORT = BLOCKB6B;
```

```
RULE 33_2
IF B6B = YA
THEN BLOCKB6B = OK;
```

```
RULE 33_3
IF B6B = YA
THEN BLOCKB6B = NOT_OK;
```

```
RULE 33_4
IF B6B = TIDAK
THEN BLOCKB6B = NOT_OK;
```

```
RULE MESSAGE_BLOCKB6B
IF SUPPORT = BLOCKB6B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
    DISPLAY "
        1 = MISFIRING
        2 = TIMING IGNITION TERLAMBAT
        3 = KATUP GAS BUANG
        4 = CAMPURAN BENSIN UDARA TERLALU KAYA {MESSAGE}.";
```



! ===== RULES 34 =====

```
RULE 34_1
IF   BLOCKB6C = OK AND
     B6C = YA
THEN SUPPORT = BLOCKB6C;

RULE 34_2
IF   B6C = YA
THEN BLOCKB6C = OK;

RULE 34_3
IF   B6C = YA
THEN BLOCKB6C = NOT_OK;

RULE 34_4
IF   B6C = TIDAK
THEN BLOCKB6C = NOT_OK;

RULE MESSAGE_BLOCKB6C
IF   SUPPORT = BLOCKB6C
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
           DISPLAY "
           1 = TERLALU PANAS
           2 = CAMPURAN BENSIN UDARA TERLALU MISKIN
           3 = KATUP UDARA MASUK BOCOR {MESSAGE}.";
```

! ===== RULES 35 =====

```
RULE 35_1
IF   BLOCKB7 = OK AND
     B7 = YA
THEN SUPPORT = BLOCKB7;

RULE 35_2
IF   B7 = YA
THEN BLOCKB7 = OK;

RULE 35_3
IF   B7 = YA
THEN BLOCKB7 = NOT_OK;

RULE 35_4
IF   B7 = TIDAK
THEN BLOCKB7 = NOT_OK;

RULE MESSAGE_BLOCKB7
IF   SUPPORT = BLOCKB7
THEN MESSAGE = DENGAN_TELITI
           DISPLAY " CEK DAN PERIKSALAH {MESSAGE}.";
```

! ===== RULES 36 =====

```
RULE 36_1
IF   BLOCKB7A = OK AND
     B7A = YA
THEN SUPPORT = BLOCKB7A;

RULE 36_2
IF   B7A = YA
THEN BLOCKB7A = OK;
```

```
RULE 36_3
IF B7A = YA
THEN BLOCKB7A = NOT_OK;
```

```
RULE 36_4
IF B7A = TIDAK
THEN BLOCKB7A = NOT_OK;
```

```
RULE MESSAGE_BLOCKB7A
IF SUPPORT = BLOCKB7A
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY " CLEARANCE TAPPET TIDAK DISETEL SECARA TEPAT
'EMPAT LANGKAH' {MESSAGE}.";
```

```
! ===== RULES 37 =====
```

```
RULE 37_1
IF BLOCKB7B = OK AND
B7B = YA
THEN SUPPORT = BLOCKB7B;
```

```
RULE 37_2
IF B7B = YA
THEN BLOCKB7B = OK;
```

```
RULE 37_3
IF B7B = YA
THEN BLOCKB7B = NOT_OK;
```

```
RULE 37_4
IF B7B = TIDAK
THEN BLOCKB7B = NOT_OK;
```

```
RULE MESSAGE_BLOCKB7B
IF SUPPORT = BLOCKB7B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = BILA DIASELERASI SECARA CEPAT
2 = BILA DIBEBANI {MESSAGE}.";
```

```
! ===== RULES 38 =====
```

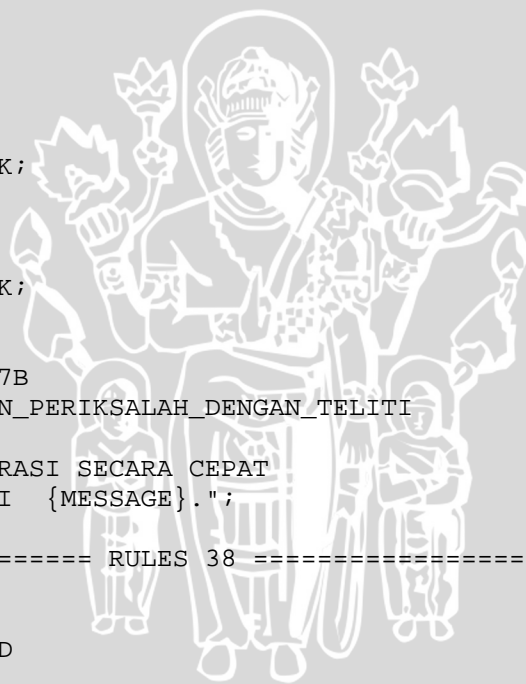
```
RULE 38_1
IF BLOCKB7C = OK AND
B7C = YA
THEN SUPPORT = BLOCKB7C;
```

```
RULE 38_2
IF B7C = YA
THEN BLOCKB7C = OK;
```

```
RULE 38_3
IF B7C = YA
THEN BLOCKB7C = NOT_OK;
```

```
RULE 38_4
IF B7C = TIDAK
THEN BLOCKB7C = NOT_OK;
```

```
RULE MESSAGE_BLOCKB7C
IF SUPPORT = BLOCKB7C
```



```
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY " SUARA KARENA TORAK YANG TELAH AUS {MESSAGE}.";
```

```
! ===== RULES 39 =====
```

```
RULE 39_1
IF BLOCKB7D = OK AND
   B7D = YA
THEN SUPPORT = BLOCKB7D;
```

```
RULE 39_2
IF B7D = YA
THEN BLOCKB7D = OK;
```

```
RULE 39_3
IF B7D = YA
THEN BLOCKB7D = NOT_OK;
```

```
RULE 39_4
IF B7D = TIDAK
THEN BLOCKB7D = NOT_OK;
```

```
RULE MESSAGE_BLOCKB7D
IF SUPPORT = BLOCKB7D
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY " SUARA KARENA PENA TORAK YANG TELAH AUS {MESSAGE}.";
```

```
! ===== RULES 40 =====
```

```
RULE 40_1
IF BLOCKB7E = OK AND
   B7E = YA
THEN SUPPORT = BLOCKB7E;
```

```
RULE 40_2
IF B7E = YA
THEN BLOCKB7E = OK;
```

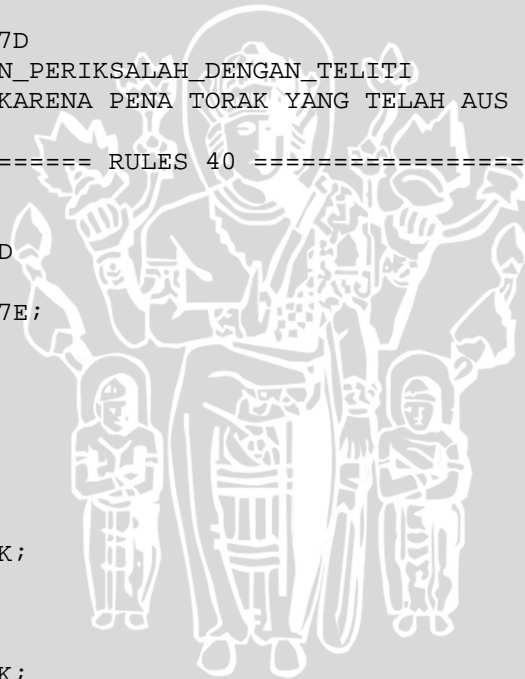
```
RULE 40_3
IF B7E = YA
THEN BLOCKB7E = NOT_OK;
```

```
RULE 40_4
IF B7E = TIDAK
THEN BLOCKB7E = NOT_OK;
```

```
RULE MESSAGE_BLOCKB7E
IF SUPPORT = BLOCKB7E
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY " RANTAI CAM TELAH BERTAMBAH PANJANG
'EMPAT LANGKAH' {MESSAGE}.";
```

```
! ===== RULES 41 =====
```

```
RULE 41_1
IF BLOCKB8 = OK AND
   B8 = YA
THEN SUPPORT = BLOCKB8;
```




```

RULE 41_2
IF B8 = YA
THEN BLOCKB8 = OK;

RULE 41_3
IF B8 = YA
THEN BLOCKB8 = NOT_OK;

RULE 41_4
IF B8 = TIDAK
THEN BLOCKB8 = NOT_OK;

RULE MESSAGE_BLOCKB8
IF SUPPORT = BLOCKB8
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = SLIP
2 = MENAHAN
3 = SUKAR DIKOPEL {MESSAGE}.";

! ===== RULES 42 =====

RULE 42_1
IF BLOCKB8A = OK AND
B8A = YA
THEN SUPPORT = BLOCKB8A;

RULE 42_2
IF B8A = YA
THEN BLOCKB8A = OK;

RULE 42_3
IF B8A = YA
THEN BLOCKB8A = NOT_OK;

RULE 42_4
IF B8A = TIDAK
THEN BLOCKB8A = NOT_OK;

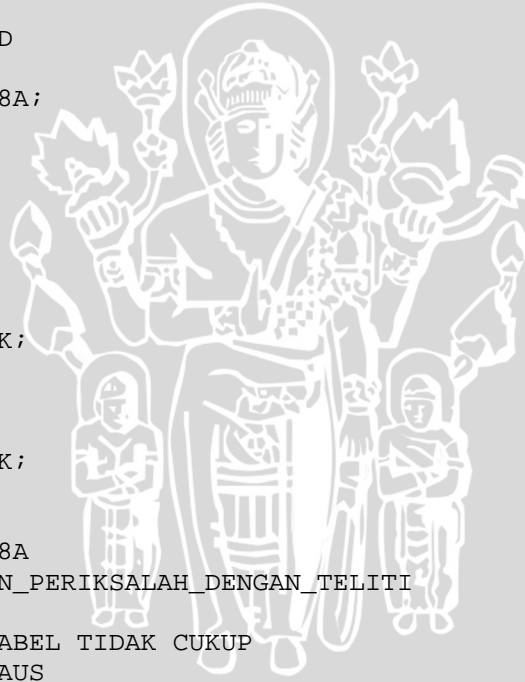
RULE MESSAGE_BLOCKB8A
IF SUPPORT = BLOCKB8A
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = GERAK MAIN KABEL TIDAK CUKUP
2 = PELAT GESEK AUS
3 = PEGAS KOPLING TELAH MENJADI LUNAK
4 = PELAT PELAT KOPLING BERUBAH BENTUK
MENJADI TIDAK RATA {MESSAGE}.";

! ===== RULES 43 =====

RULE 43_1
IF BLOCKB8B = OK AND
B8B = YA
THEN SUPPORT = BLOCKB8B;

RULE 43_2
IF B8B = YA
THEN BLOCKB8B = OK;

```



```
RULE 43_3
IF B8B = YA
THEN BLOCKB8B = NOT_OK;
```

```
RULE 43_4
IF B8B = TIDAK
THEN BLOCKB8B = NOT_OK;
```

```
RULE MESSAGE_BLOCKB8B
IF SUPPORT = BLOCKB8B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = GERAK MAIN KABEL BERLEBIHAN
2 = PELAT GESEK ATAU PELAT KOPLING PATAH
3 = PEGAS KOPLING PATAH {MESSAGE}.";
```

```
! ===== RULES 44 =====
```

```
RULE 44_1
IF BLOCKB8C = OK AND
B8C = YA
THEN SUPPORT = BLOCKB8C;
```

```
RULE 44_2
IF B8C = YA
THEN BLOCKB8C = OK;
```

```
RULE 44_3
IF B8C = YA
THEN BLOCKB8C = NOT_OK;
```

```
RULE 44_4
IF B8C = TIDAK
THEN BLOCKB8C = NOT_OK;
```

```
RULE MESSAGE_BLOCKB8C
IF SUPPORT = BLOCKB8C
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = PELAT GESEK ATAU PELAT KOPLING BERUBAH BENTUK
MENJADI TIDAK RATA
2 = KABEL TIDAK MELUNCUR SECARA HALUS {MESSAGE}.";
```

```
! ===== RULES 45 =====
```

```
RULE 45_1
IF BLOCKB9 = OK AND
B9 = YA
THEN SUPPORT = BLOCKB9;
```

```
RULE 45_2
IF B9 = YA
THEN BLOCKB9 = OK;
```

```
RULE 45_3
IF B9 = YA
THEN BLOCKB9 = NOT_OK;
```

```
RULE 45_4
IF B9 = TIDAK
THEN BLOCKB9 = NOT_OK;
```

```

RULE MESSAGE_BLOCKB9
IF SUPPORT = BLOCKB9
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = SUKAR GANTI GIGI
2 = RODA GIGI TIDAK BERHUBUNGAN DENGAN TEPAT {MESSAGE}.";

```

! ===== RULES 46 =====

```

RULE 46_1
IF BLOCKB9A = OK AND
   B9A = YA
THEN SUPPORT = BLOCKB9A;

```

```

RULE 46_2
IF B9A = YA
THEN BLOCKB9A = OK;

```

```

RULE 46_3
IF B9A = YA
THEN BLOCKB9A = NOT_OK;

```

```

RULE 46_4
IF B9A = TIDAK
THEN BLOCKB9A = NOT_OK;

```

```

RULE MESSAGE_BLOCKB9A
IF SUPPORT = BLOCKB9A
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = KOPLING MENAHAN
2 = MEKANISME PEMINDAH GIGI RUSAK
3 = OPERASI TIDAK TEPAT {MESSAGE}.";

```

! ===== RULES 47 =====

```

RULE 47_1
IF BLOCKB9B = OK AND
   B9B = YA
THEN SUPPORT = BLOCKB9B;

```

```

RULE 47_2
IF B9B = YA
THEN BLOCKB9B = OK;

```

```

RULE 47_3
IF B9B = YA
THEN BLOCKB9B = NOT_OK;

```

```

RULE 47_4
IF B9B = TIDAK
THEN BLOCKB9B = NOT_OK;

```

```

RULE MESSAGE_BLOCKB9B
IF SUPPORT = BLOCKB9B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = RODA GIGI AUS
2 = GARPU PEMINDAH GIGI BENGKOK ATAU PATAH
3 = POROS GARPU PEMINDAH GIGI BENGKOK {MESSAGE}.";

```

! ===== RULES 48 =====

```
RULE 48_1
IF   BLOCKB10 = OK AND
     B10 = YA
THEN SUPPORT = BLOCKB10;

RULE 48_2
IF   B10 = YA
THEN BLOCKB10 = OK;

RULE 48_3
IF   B10 = YA
THEN BLOCKB10 = NOT_OK;

RULE 48_4
IF   B10 = TIDAK
THEN BLOCKB10 = NOT_OK;

RULE MESSAGE_BLOCKB10
IF   SUPPORT = BLOCKB10
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
           DISPLAY "
           1 = GETARAN BERLEBIHAN PADA KECEPATAN TINGGI
           2 = SUARA ABNORMAL {MESSAGE}.";
```

! ===== RULES 49 =====

```
RULE 49_1
IF   BLOCKB10A = OK AND
     B10A = YA
THEN SUPPORT = BLOCKB10A;

RULE 49_2
IF   B10A = YA
THEN BLOCKB10A = OK;

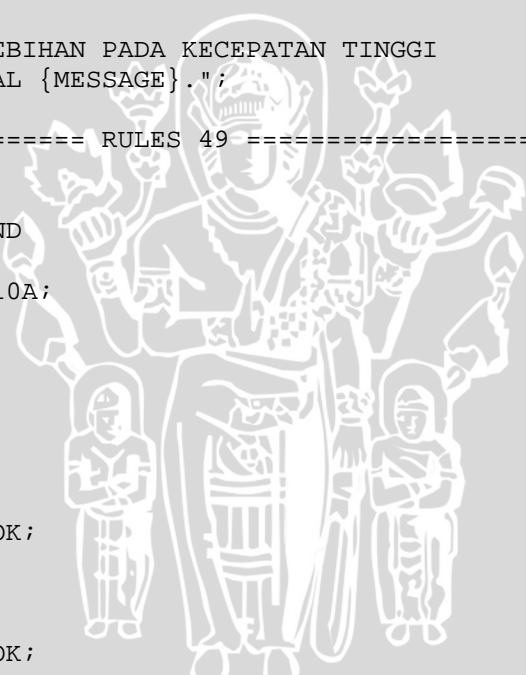
RULE 49_3
IF   B10A = YA
THEN BLOCKB10A = NOT_OK;

RULE 49_4
IF   B10A = TIDAK
THEN BLOCKB10A = NOT_OK;

RULE MESSAGE_BLOCKB10A
IF   SUPPORT = BLOCKB10A
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
           DISPLAY "
           1 = POROS PENGGERAK BENGKOK
           2 = BANTALAN UNIVERSAL JOINT AUS ATAU PATAH
           3 = BAUT PEMEGANG POROS LONGGAR {MESSAGE}.";
```

! ===== RULES 50 =====

```
RULE 50_1
IF   BLOCKB10B = OK AND
     B10B = YA
THEN SUPPORT = BLOCKB10B;
```



```
RULE 50_2
IF B10B = YA
THEN BLOCKB10B = OK;

RULE 50_3
IF B10B = YA
THEN BLOCKB10B = NOT_OK;

RULE 50_4
IF B10B = TIDAK
THEN BLOCKB10B = NOT_OK;

RULE MESSAGE_BLOCKB10B
IF SUPPORT = BLOCKB10B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = UNIVERSAL JOINT AUS
2 = ALUR ALUR 'SPLINES' DARI POROS TELAH AUS
3 = BAUT PEMEGANG POROS LONGGAR ATAU PATAH {MESSAGE}.";

! ===== RULES 51 =====

RULE 51_1
IF BLOCKB11 = OK AND
B11 = YA
THEN SUPPORT = BLOCKB11;

RULE 51_2
IF B11 = YA
THEN BLOCKB11 = OK;

RULE 51_3
IF B11 = YA
THEN BLOCKB11 = NOT_OK;

RULE 51_4
IF B11 = TIDAK
THEN BLOCKB11 = NOT_OK;

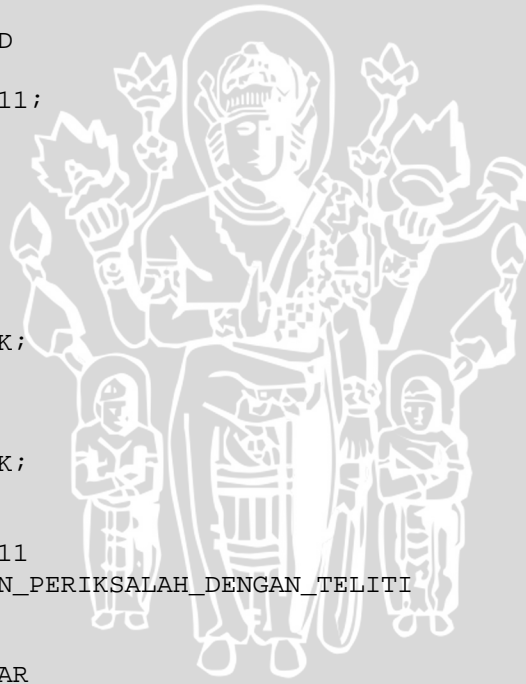
RULE MESSAGE_BLOCKB11
IF SUPPORT = BLOCKB11
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = RANTAI PATAH
2 = RANTAI LONGGAR
3 = SUARA ABNORMAL {MESSAGE}.";

! ===== RULES 52 =====

RULE 52_1
IF BLOCKB11A = OK AND
B11A = YA
THEN SUPPORT = BLOCKB11A;

RULE 52_2
IF B11A = YA
THEN BLOCKB11A = OK;

RULE 52_3
IF B11A = YA
THEN BLOCKB11A = NOT_OK;
```



```
RULE 52_4
IF B11A = TIDAK
THEN BLOCKB11A = NOT_OK;

RULE MESSAGE_BLOCKB11A
IF SUPPORT = BLOCKB11A
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = RANTAI AUS
2 = RANTAI TERLALU TEGANG
3 = RODA BELAKANG 'RUN OUT' {MESSAGE}.";

! ===== RULES 53 =====

RULE 53_1
IF BLOCKB11B = OK AND
B11B = YA
THEN SUPPORT = BLOCKB11B;

RULE 53_2
IF B11B = YA
THEN BLOCKB11B = OK;

RULE 53_3
IF B11B = YA
THEN BLOCKB11B = NOT_OK;

RULE 53_4
IF B11B = TIDAK
THEN BLOCKB11B = NOT_OK;

RULE MESSAGE_BLOCKB11B
IF SUPPORT = BLOCKB11B
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = TEGANGAN RANTAI TIDAK DISETEL SECARA TEPAT
2 = RANTAI AUS
3 = GIGI RANTAI 'SPROCKET' YANG MENGERAKKAN ATAU
DIGERAKKAN TELAH AUS {MESSAGE}.";

! ===== RULES 54 =====

RULE 54_1
IF BLOCKB11CB = OK AND
B11C = YA
THEN SUPPORT = BLOCKB11C;

RULE 54_2
IF B11C = YA
THEN BLOCKB11C = OK;

RULE 54_3
IF B11C = YA
THEN BLOCKB11C = NOT_OK;

RULE 54_4
IF B11C = TIDAK
THEN BLOCKB11C = NOT_OK;
```

```
RULE MESSAGE_BLOCKB11C
IF SUPPORT = BLOCKB11C
THEN MESSAGE = CEK_DAN_PERIKSALAH_DENGAN_TELITI
DISPLAY "
1 = TEGANGAN RANTAI TIDAK DISETEL SECARA TEPAT
2 = RANTAI TIDAK DIBERI OLI
3 = GIGI RANTAI YANG MENGGERAKKAN ATAU
DIGERAKKAN TELAH AUS {MESSAGE}.";
```

```
! ===== RULES 55 =====
```

```
RULE 55_1
IF BLOCKC = OK AND
C = YA
THEN SUPPORT = BLOCKC;
```

```
RULE 55_2
IF C = YA
THEN BLOCKC = OK;
```

```
RULE 55_3
IF C = YA
THEN BLOCKC = NOT_OK;
```

```
RULE 55_4
IF C = TIDAK
THEN BLOCKC = NOT_OK;
```

```
RULE MESSAGE_BLOCKC
IF SUPPORT = BLOCKC
THEN MESSAGE = KERUSAKAN_PADA_NON_MESIN
DISPLAY "
1 = ENGINE TIDAK DAPAT HIDUP DENGAN MOTOR STARTER
2 = ENGINE TIDAK DAPAT HIDUP DENGAN KICK STARTER
3 = PERFORMANCE ENGINE KURANG
4 = KERUSAKAN SISTEM PELUMASAN
5 = GAS BUANG MEMPERLIHATKAN WARNA ABNORMAL
6 = PEMBAKARAN ABNORMAL
7 = SUARA SUARA ABNORMAL
8 = KOPLING
9 = TRANSMISI
10 = POROS PENGGERAK
11 = RANTAI PENGGERAK {MESSAGE}.";
```

```
! ===== RULES 56 =====
```

```
RULE 56_1
IF BLOCKC1 = OK AND
C1 = YA
THEN SUPPORT = BLOCKC1;
```

```
RULE 56_2
IF C1 = YA
THEN BLOCKC1 = OK;
```

```
RULE 56_3
IF C1 = YA
THEN BLOCKC1 = NOT_OK;
```

```
RULE 56_4
IF C1 = TIDAK
THEN BLOCKC1 = NOT_OK;

RULE MESSAGE_BLOCKC1
IF SUPPORT = BLOCKC1
THEN MESSAGE = TEMPATNYA
DISPLAY " MASUKKAN KUNCI PADA {MESSAGE}.";

! ===== RULES 57 =====

RULE 57_1
IF BLOCKC2 = OK AND
C2 = YA
THEN SUPPORT = BLOCKC2;

RULE 57_2
IF C2 = YA
THEN BLOCKC2 = OK;

RULE 57_3
IF C2 = YA
THEN BLOCKC2 = NOT_OK;

RULE 57_4
IF C2 = TIDAK
THEN BLOCKC2 = NOT_OK;

RULE MESSAGE_BLOCKC2
IF SUPPORT = BLOCKC2
THEN MESSAGE = SEGERA
DISPLAY " BELIKAN BENSIN {MESSAGE}.";

! ===== RULES 58 =====

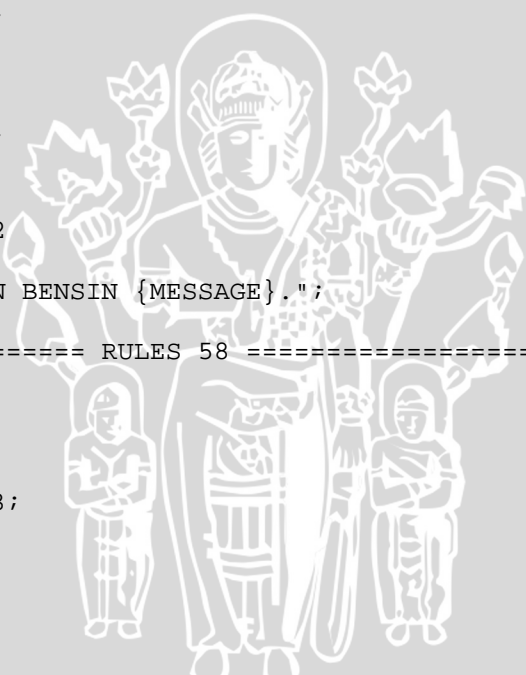
RULE 58_1
IF BLOCKC3 = OK AND
C3 = YA
THEN SUPPORT = BLOCKC3;

RULE 58_2
IF C3 = YA
THEN BLOCKC3 = OK;

RULE 58_3
IF C3 = YA
THEN BLOCKC3 = NOT_OK;

RULE 58_4
IF C3 = TIDAK
THEN BLOCKC3 = NOT_OK;

RULE MESSAGE_BLOCKC3
IF SUPPORT = BLOCKC3
THEN MESSAGE = SEGERA
DISPLAY " BELIKAN BUSI {MESSAGE}.";
```



! ===== RULES 59 =====

```
RULE 59_1
IF   BLOCKC4 = OK AND
    C4 = YA
THEN SUPPORT = BLOCKC4;

RULE 59_2
IF   C4 = YA
THEN BLOCKC4 = OK;

RULE 59_3
IF   C4 = YA
THEN BLOCKC4 = NOT_OK;

RULE 59_4
IF   C4 = TIDAK
THEN BLOCKC4 = NOT_OK;

RULE MESSAGE_BLOCKC4
IF   SUPPORT = BLOCKC4
THEN MESSAGE = SEGERA
    DISPLAY " GANTI KABEL BUSI {MESSAGE}.";
```

! ===== RULES 60 =====

```
RULE 60_1
IF   BLOCKC5 = OK AND
    C5 = YA
THEN SUPPORT = BLOCKC5;

RULE 60_2
IF   C5 = YA
THEN BLOCKC5 = OK;

RULE 60_3
IF   C5 = YA
THEN BLOCKC5 = NOT_OK;

RULE 60_4
IF   C5 = TIDAK
THEN BLOCKC5 = NOT_OK;

RULE MESSAGE_BLOCKC5
IF   SUPPORT = BLOCKC5
THEN MESSAGE = SEGERA
    DISPLAY " BERSIHKAN FILTER UDARA {MESSAGE}.";
```

! ===== RULES 61 =====

```
RULE 61_1
IF   BLOCKC6 = OK AND
    C6 = YA
THEN SUPPORT = BLOCKC6;

RULE 61_2
IF   C6 = YA
THEN BLOCKC6 = OK;
```



```

RULE 61_3
IF C6 = YA
THEN BLOCKC6 = NOT_OK;

RULE 61_4
IF C6 = TIDAK
THEN BLOCKC6 = NOT_OK;

RULE MESSAGE_BLOCKC6
IF SUPPORT = BLOCKC6
THEN MESSAGE = SEGERA
DISPLAY " HILANGKAN PENYUMBATNYA {MESSAGE}. ";

```

! ===== RULES 62 =====

```

RULE 62_1
IF BLOCKC7 = OK AND
C7 = YA
THEN SUPPORT = BLOCKC7;

RULE 62_2
IF C7 = YA
THEN BLOCKC7 = OK;

RULE 62_3
IF C7 = YA
THEN BLOCKC7 = NOT_OK;

RULE 62_4
IF C7 = TIDAK
THEN BLOCKC7 = NOT_OK;

RULE MESSAGE_BLOCKC7
IF SUPPORT = BLOCKC7
THEN MESSAGE = SEGERA
DISPLAY " HILANGKAN AIRNYA {MESSAGE}. ";

```

! ===== RULES 63 =====

```

RULE 63_1
IF BLOCKC8 = OK AND
C8 = YA
THEN SUPPORT = BLOCKC8;

RULE 63_2
IF C8 = YA
THEN BLOCKC8 = OK;

RULE 63_3
IF C8 = YA
THEN BLOCKC8 = NOT_OK;

RULE 63_4
IF C8 = TIDAK
THEN BLOCKC8 = NOT_OK;

RULE MESSAGE_BLOCKC8
IF SUPPORT = BLOCKC8
THEN MESSAGE = SEGERA
DISPLAY " ISI AIR ACCU {MESSAGE}. ";

```



```
! ===== RULES 64 =====
```

```
RULE 64_1
IF   BLOCKC9 = OK AND
     C9 = YA
THEN SUPPORT = BLOCKC9;

RULE 64_2
IF   C9 = YA
THEN BLOCKC9 = OK;

RULE 64_3
IF   C9 = YA
THEN BLOCKC9 = NOT_OK;

RULE 64_4
IF   C9 = TIDAK
THEN BLOCKC9 = NOT_OK;

RULE MESSAGE_BLOCKC9
IF   SUPPORT = BLOCKC9
THEN MESSAGE = F1ZR
     DISPLAY " GUNAKANLAH MOTOR DUA TAK {MESSAGE}.";
```

```
! ===== RULES 65 =====
```

```
RULE 65_1
IF   BLOCKC10 = OK AND
     C10 = YA
THEN SUPPORT = BLOCKC10;

RULE 65_2
IF   C10 = YA
THEN BLOCKC10 = OK;

RULE 65_3
IF   C10 = YA
THEN BLOCKC10 = NOT_OK;

RULE 65_4
IF   C10 = TIDAK
THEN BLOCKC10 = NOT_OK;

RULE MESSAGE_BLOCKC10
IF   SUPPORT = BLOCKC10
THEN MESSAGE = SEGERA
     DISPLAY " ISI OLI SAMPINGNYA {MESSAGE}.";
```

4.2.5 Perancangan User Interface (UI)

Kekuatan SP itu antara lain tergantung pada UI. Diharapkan agar dialog yang terjadi mirip kejadian yang sesungguhnya (manusiawi). Berikut contoh desain UI yang dipersiapkan untuk software *VP-Expert*.

```
! ===== QUESTIONS BLOCK =====
```

```
! ===== ASK 1 =====
```

```
ASK A: "APAKAH ADA TROUBLESHOOTING?";
CHOICES A: YA, TIDAK;
```

! ===== ASK 2 =====

ASK B: "APAKAH TROUBLESHOOTING PADA NON MESIN?";
CHOICES B: YA, TIDAK;

! ===== ASK 3 =====

ASK B1: "ENGINE TIDAK DAPAT HIDUP DENGAN MOTOR STARTER?";
CHOICES B1: YA, TIDAK;

! ===== ASK 4 =====

ASK B1A: "APAKAH MOTOR STARTER BERPUTAR?";
CHOICES B1A: YA, TIDAK;

! ===== ASK 5 =====

ASK B1B: "BERPUTAR?";
CHOICES B1B: YA, TIDAK;

! ===== ASK 6 =====

ASK B1C: "BERPUTAR PERLAHAN LAHAN ATAU TIDAK DAPAT BERPUTAR?";
CHOICES B1C: YA, TIDAK;

! ===== ASK 7 =====

ASK B1D: "APAKAH ENGINE BERPUTAR?";
CHOICES B1D: YA, TIDAK;

! ===== ASK 8 =====

ASK B1E: "APAKAH ENGINE TIDAK BERPUTAR 'MOTOR STARTER BERPUTAR SENDIRI'?";
CHOICES B1E: YA, TIDAK;

! ===== ASK 9 =====

ASK B2: "ENGINE TIDAK DAPAT HIDUP DENGAN KICK STARTER?";
CHOICES B2: YA, TIDAK;

! ===== ASK 10 =====

ASK B2A: "APAKAH KICKSTARTER DAPAT DITEKAN?";
CHOICES B2A: YA, TIDAK;

! ===== ASK 11 =====

ASK B2B: "DAPAT DITEKAN?";
CHOICES B2B: YA, TIDAK;

! ===== ASK 12 =====

ASK B2C: "TIDAK DAPAT DITEKAN ATAU DAPAT DITEKAN SECARA PERLAHAN LAHAN?";
CHOICES B2C: YA, TIDAK;

! ===== ASK 13 =====

ASK B2D: "APAKAH ENGINE DAPAT BERPUTAR?";
CHOICES B2D: YA, TIDAK;



! ===== ASK 14 =====

ASK B2E: "APAKAH ENGINE TIDAK DAPAT BERPUTAR ?";
CHOICES B2E: YA, TIDAK;

! ===== ASK 15 =====

ASK B3: "PERFORMANCE ENGINE KURANG?";
CHOICES B3: YA, TIDAK;

! ===== ASK 16 =====

ASK B3A: "APAKAH ENGINE IDLE SECARA HALUS?";
CHOICES B3A: YA, TIDAK;

! ===== ASK 17 =====

ASK B3B: "IDLE SECARA HALUS TETAPI MEMPUNYAI PERCEPATAN DAN DAYA YANG KURANG?";
CHOICES B3B: YA, TIDAK;

! ===== ASK 18 =====

ASK B3C: "IDLE SECARA KASAR?";
CHOICES B3C: YA, TIDAK;

! ===== ASK 19 =====

ASK B4: "KERUSAKAN SISTEM PELUMASAN?";
CHOICES B4: YA, TIDAK;

! ===== ASK 20 =====

ASK B4A: "KONSUMSI OLI BERLEBIHAN?";
CHOICES B4A: YA, TIDAK;

! ===== ASK 21 =====

ASK B4B: "OLI TERCAMPUR?";
CHOICES B4B: YA, TIDAK;

! ===== ASK 22 =====

ASK B4C: "OLI KOTOR?";
CHOICES B4C: YA, TIDAK;

! ===== ASK 23 =====

ASK B4D: "PEMAKAIAN BAHAN BAKAR BERLEBIHAN?";
CHOICES B4D: YA, TIDAK;

! ===== ASK 24 =====

ASK B4E: "PEMAKAIAN BAHAN BAKAR PADA ENGINE?";
CHOICES B4E: YA, TIDAK;

! ===== ASK 25 =====

ASK B4F: "PEMAKAIAN BAHAN BAKAR PADA CHASSIS?";
CHOICES B4F: YA, TIDAK;



! ===== ASK 26 =====

ASK B4G: "PEMAKAIAN BAHAN BAKAR PADA OPERASI?";
CHOICES B4G: YA, TIDAK;

! ===== ASK 27 =====

ASK B5: "GAS BUANG MEMPERLIHATKAN WARNA ABNORMAL?";
CHOICES B5: YA, TIDAK;

! ===== ASK 28 =====

ASK B5A: "APAKAH TERJADI WARNA GAS BUANG?";
CHOICES B5A: YA, TIDAK;

! ===== ASK 29 =====

ASK B5B: "APA WARNA GAS BUANG HITAM?";
CHOICES B5B: YA, TIDAK;

! ===== ASK 30 =====

ASK B5C: "APA WARNA GAS BUANG PUTIH?";
CHOICES B5C: YA, TIDAK;

! ===== ASK 31 =====

ASK B6: "APAKAH TERJADI PEMBAKARAN ABNORMAL ?";
CHOICES B6: YA, TIDAK;

! ===== ASK 32 =====

ASK B6A: "APAKAH TERJADI KNOCKING?";
CHOICES B6A: YA, TIDAK;

! ===== ASK 33 =====

ASK B6B: "APAKAH TERJADI AFTER FIRE?";
CHOICES B6B: YA, TIDAK;

! ===== ASK 34 =====

ASK B6C: "APAKAH TERJADI BACK FIRE?";
CHOICES B6C: YA, TIDAK;

! ===== ASK 35 =====

ASK B7: "APAKAH TERJADI SUARA SUARA ABNORMAL ?";
CHOICES B7: YA, TIDAK;

! ===== ASK 36 =====

ASK B7A: "APAKAH TERJADI SUARA KLIK KLIK 'CHATTERING'?";
CHOICES B7A: YA, TIDAK;

! ===== ASK 37 =====

ASK B7B: "APAKAH TERJADI SUARA KETUKAN?";
CHOICES B7B: YA, TIDAK;



! ===== ASK 38 =====

ASK B7C: "BILA DIASELERASI SECARA CEPAT?";
CHOICES B7C: YA, TIDAK;

! ===== ASK 39 =====

ASK B7D: "BILA DIBEBANI?";
CHOICES B7D: YA, TIDAK;

! ===== ASK 40 =====

ASK B7E: "APAKAH TERJADI SUARA KELENING?";
CHOICES B7E: YA, TIDAK;

! ===== ASK 41 =====

ASK B8: "APAKAH TERJADI PADA KOPLING ?";
CHOICES B8: YA, TIDAK;

! ===== ASK 42 =====

ASK B8A: "APAKAH KOPLING SLIP ?";
CHOICES B8A: YA, TIDAK;

! ===== ASK 43 =====

ASK B8B: "APAKAH KOPLING MENAHAN ?";
CHOICES B8B: YA, TIDAK;

! ===== ASK 44 =====

ASK B8C: "APAKAH KOPLING SUKAR DIKOPEL ?";
CHOICES B8C: YA, TIDAK;

! ===== ASK 45 =====

ASK B9: "APAKAH TERJADI PADA KOPLING ?";
CHOICES B9: YA, TIDAK;

! ===== ASK 46 =====

ASK B9A: "APAKAH SUKAR GANTI GIGI ?";
CHOICES B9A: YA, TIDAK;

! ===== ASK 47 =====

ASK B9B: "APAKAH RODA GIGI TIDAK BERHUBUNGAN DENGAN TEPAT?";
CHOICES B9B: YA, TIDAK;

! ===== ASK 48 =====

ASK B10: "APAKAH TERJADI PADA POROS PENGGERAK?";
CHOICES B10: YA, TIDAK;

! ===== ASK 49 =====

ASK B10A: "APAKAH TERJADI GETARAN BERLEBIHAN PADA
KECEPATAN TINGGI?";
CHOICES B10A: YA, TIDAK;



! ===== ASK 50 =====

ASK B10B: "APAKAH TERJADI SUARA ABNORMAL?";
CHOICES B10B: YA, TIDAK;

! ===== ASK 51 =====

ASK B11: "APAKAH TERJADI PADA RANTAI PENGGERAK?";
CHOICES B11: YA, TIDAK;

! ===== ASK 52 =====

ASK B11A: "APAKAH RANTAI PATAH?";
CHOICES B11A: YA, TIDAK;

! ===== ASK 53 =====

ASK B11B: "APAKAH RANTAI LONGGAR?";
CHOICES B11B: YA, TIDAK;

! ===== ASK 54 =====

ASK B11C: "APAKAH SUARA ABNORMAL?";
CHOICES B11C: YA, TIDAK;

! ===== ASK 55 =====

ASK C: "APAKAH TROUBLESHOOTING PADA MESIN?";
CHOICES C: YA, TIDAK;

! ===== ASK 56 =====

ASK C1: "APAKAH KUNCI BELUM TERPASANG?";
CHOICES C1: YA, TIDAK;

! ===== ASK 57 =====

ASK C2: "APAKAH BENSINYA HABIS?";
CHOICES C2: YA, TIDAK;

! ===== ASK 58 =====

ASK C3: "APAKAH BUSI MATI?";
CHOICES C3: YA, TIDAK;

! ===== ASK 59 =====

ASK C4: "APAKAH KABEL BUSI PUTUS?";
CHOICES C4: YA, TIDAK;

! ===== ASK 60 =====

ASK C5: "APAKAH FILTER UDARA TERPUTUS?";
CHOICES C5: YA, TIDAK;

! ===== ASK 61 =====

ASK C6: "APAKAH KNALPOTNYA TERSUMBAT?";
CHOICES C6: YA, TIDAK;




```
! ===== ASK 62 =====
ASK C7: "APAKAH KARBURATORNYA KEMASUKKAN AIR?";
CHOICES C7: YA, TIDAK;
```

```
! ===== ASK 63 =====
ASK C8: "APAKAH AIR ACCUNYA HABIS?";
CHOICES C8: YA, TIDAK;
```

```
! ===== ASK 64 =====
ASK C9: "APAKAH MOTOR ANDA EMPAT TAK?";
CHOICES C9: YA, TIDAK;
```

```
! ===== ASK 65 =====
ASK C10: "APAKAH OLI SAMPINGNYA HABIS?";
CHOICES C10: YA, TIDAK;
```

4.2.6 Pemasukan Rancangan ke dalam VP Expert

Setelah rancangan selesai didesain, maka semuanya di inputkan kedalam software melalui text editor yang bernama Notepad. File ini ini harus mempunyai nama ekstensi file.kbs. Contoh file kbs adalah sebagai berikut.

```
! HMO SERVICE SCREENING ASSISTANT
! WITH ADDITIONAL ENHANCEMENTS : CUSTOMIZED CONCLUSIONS
! SAVED AS HMOADD1.KBS

! *****
! SET SCREEN BACKGROUND COLOR TO
LIGHT BLUE
BKCOLOR = 3;
! ELIMINATE RULES AND FACTS WINDOWS
RUNTIME;

! ===== ACTIONS BLOCK =====

ACTIONS
WOPEN 1,3,10,13,60,7 ! DEFINE OPENING WINDOW 1
ACTIVE 1 ! ACTIVATE WINDOW 1
DISPLAY "
PERANCANGAN SISTEM PAKAR UNTUK TROUBLESHOOTING SEPEDA MOTOR
MENGUNAKAN VP-XPERT
```

DEPARTEMEN PENDIDIKAN NASIONAL
 UNIVERSITAS BRAWIJAYA
 FAKULTAS TEKNIK
 JURUSAN MESIN
 MALANG
 2007

```
TEKAN SEMBARANG TOMBOL UNTUK MELANJUTKAN KONSULTASI,~
"
WCLOSE 1 ! REMOVE WINDOW 1
COLOR = 20 ! SET FOLLOWING TEXT COLOR TO
BLINKING RED
DISPLAY " PERHATIAN! "
```

```

COLOR = 4                                ! SET FOLLOWING TEXT TO NON-BLINKING
RED
DISPLAY " TOLONG PASTIKAN PRINTER ANDA MENYALA."
DISPLAY " TEKAN SEMBARANG TOMBOL UNTUK MELANJUTKAN KONSULTASI.~"
CLS                                       ! CLEAR THE SCREEN
COLOR = 0                                ! SET FOLLOWING TEXT TO NORMAL BLACK
WOPEN 1,1,1,6,77,2                       ! DEFINE INSRUCTIONS WINDOW 1
ACTIVE 1                                  ! ACTIVATE WINDOW 1
DISPLAY "                                PETUNJUK
                                GUNAKANLAH TOMBOL ANAK PANAH UNTUK MEMILIH MENU
                                JAWABAN KEMUDIAN TEKAN TOMBOL ENTER,
                                KEMUDIAN TEKAN TOMBOL END UNTUK MEMILIH
                                PILIHAN SELANJUTNYA."
WOPEN 2,8,1,13,77,2                     ! DEFINE CONSULTATION WINDOW 2
ACTIVE 2                                  ! ACTIVATE WINDOW 2
FIND SUPPORT
FIND MESSAGE                             ! SET SUB GOAL TO CUSTOIMIZE MESSAGES
DISPLAY " TEKAN SEMBARANG TOMBOL UNTUK MELANJUTKAN KONSULTASI.~"
;

! ===== RULES BLOCK =====
! ===== RULES 1 =====

RULE 1_1
IF    BLOCKA = OK AND
     A = YA
THEN  SUPPORT = BLOCKA;

RULE 1_2
IF    A = YA
THEN  BLOCKA = OK;

RULE 1_3
IF    A = YA
THEN  BLOCKA = NOT_OK;

RULE 1_4
IF    A = TIDAK
THEN  BLOCKA = NOT_OK;

RULE MESSAGE_BLOCKA
IF    SUPPORT = BLOCKA
THEN  MESSAGE = TROUBLESHOOTING_SEPEDA_MOTOR
      DISPLAY "
      1 = TROUBLSHOOTING DI MESIN
      2 = TROUBLESHOTING DI NON MESIN
      {MESSAGE}.";

! ===== RULES 2 =====

RULE 2_1
IF    BLOCKB = OK AND
     B = YA
THEN  SUPPORT = BLOCKB;

RULE 2_2
IF    B = YA
THEN  BLOCKB = OK;

```



```

RULE 2_3
IF B = YA
THEN BLOCKB = NOT_OK;

RULE 2_4
IF B = TIDAK
THEN BLOCKB = NOT_OK;

RULE MESSAGE_BLOCKB
IF SUPPORT = BLOCKB
THEN MESSAGE = KERUSAKAN_PADA_MESIN
DISPLAY "
1 = KUNCI
2 = BENSIN
3 = BUSI
4 = KABEL BUSI
5 = FILTER UDARA
6 = KNALPOT
7 = KARBURATOR
8 = ACCU
9 = MOTOR
10 = OLI {MESSAGE}.";

```

.....

! ===== RULES 55 =====

```

RULE 55_1
IF BLOCKC = OK AND
C = YA
THEN SUPPORT = BLOCKC;

RULE 55_2
IF C = YA
THEN BLOCKC = OK;

RULE 55_3
IF C = YA
THEN BLOCKC = NOT_OK;

RULE 55_4
IF C = TIDAK
THEN BLOCKC = NOT_OK;

RULE MESSAGE_BLOCKC
IF SUPPORT = BLOCKC
THEN MESSAGE = KERUSAKAN_PADA_NON_MESIN
DISPLAY "
1 = ENGINE TIDAK DAPAT HIDUP DENGAN MOTOR STARTER
2 = ENGINE TIDAK DAPAT HIDUP DENGAN KICK STARTER
3 = PERFORMANCE ENGINE KURANG
4 = KERUSAKAN SISTEM PELUMASAN
5 = GAS BUANG MEMPERLIHATKAN WARNA ABNORMAL
6 = PEMBAKARAN ABNORMAL
7 = SUARA SUARA ABNORMAL
8 = KOPLING
9 = TRANSMISI
10 = POROS PENGGERAK
11 = RANTAI PENGGERAK {MESSAGE}.";

```

.....



```

! ===== RULES 65 =====

RULE 65_1
IF   BLOCKC10 = OK AND
    C10 = YA
THEN SUPPORT = BLOCKC10;

RULE 65_2
IF   C10 = YA
THEN BLOCKC10 = OK;

RULE 65_3
IF   C10 = YA
THEN BLOCKC10 = NOT_OK;

RULE 65_4
IF   C10 = TIDAK
THEN BLOCKC10 = NOT_OK;

RULE MESSAGE_BLOCKC10
IF   SUPPORT = BLOCKC10
THEN MESSAGE = SEGERA
    DISPLAY " ISI OLI SAMPINGNYA {MESSAGE}.";

! ===== QUESTIONS BLOCK =====
! ===== ASK 1 =====
ASK A: "APAKAH ADA TROUBLESHOOTING?";
CHOICES A: YA, TIDAK;

! ===== ASK 2 =====
ASK B: "APAKAH TROUBLESHOOTING PADA NON MESIN?";
CHOICES B: YA, TIDAK;
.....

! ===== ASK 55 =====
ASK C: "APAKAH TROUBLESHOOTING PADA MESIN?";
CHOICES C: YA, TIDAK;
.....

! ===== ASK 65 =====
ASK C10: "APAKAH OLI SAMPINGNYA HABIS?";
CHOICES C10: YA, TIDAK;

```

Gambar 5.4 Contoh File KBS

4.4 Pengujian Sistem

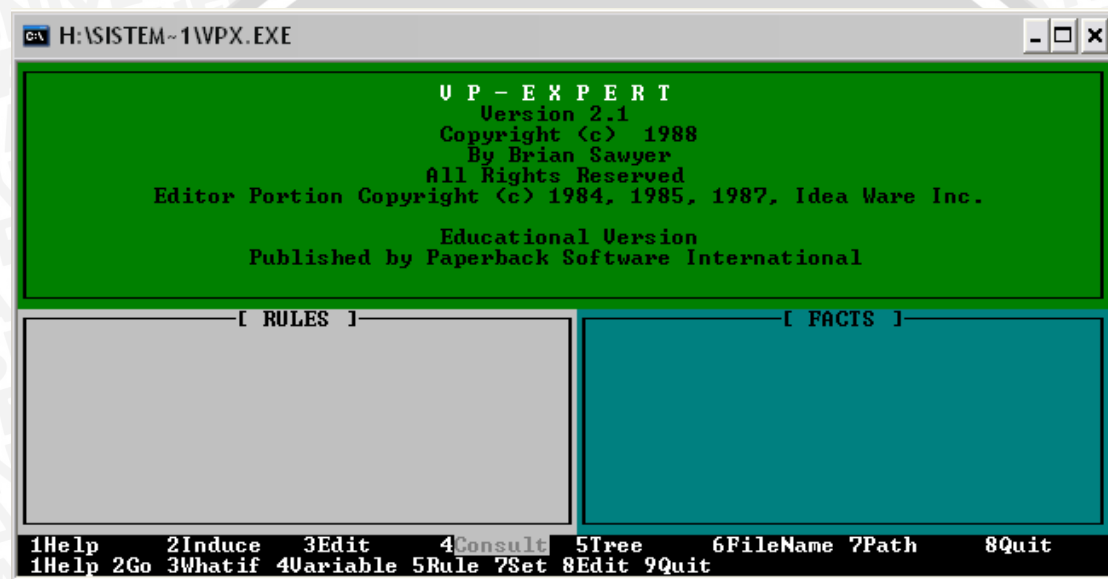
Roger S. Pressman (1999) menyarankan beberapa strategi untuk menguji suatu aplikasi. Strategi pengujian yang paling populer adalah, meliputi:

- Verifikasi
- Validasi

Verifikasi dilakukan untuk menguji fungsi-fungsi yang ada di dalam sistem apakah sesuai dengan rancangan, sedangkan validasi dilakukan untuk menguji fungsi sistem secara keseluruhan.

4.4.1 Verifikasi

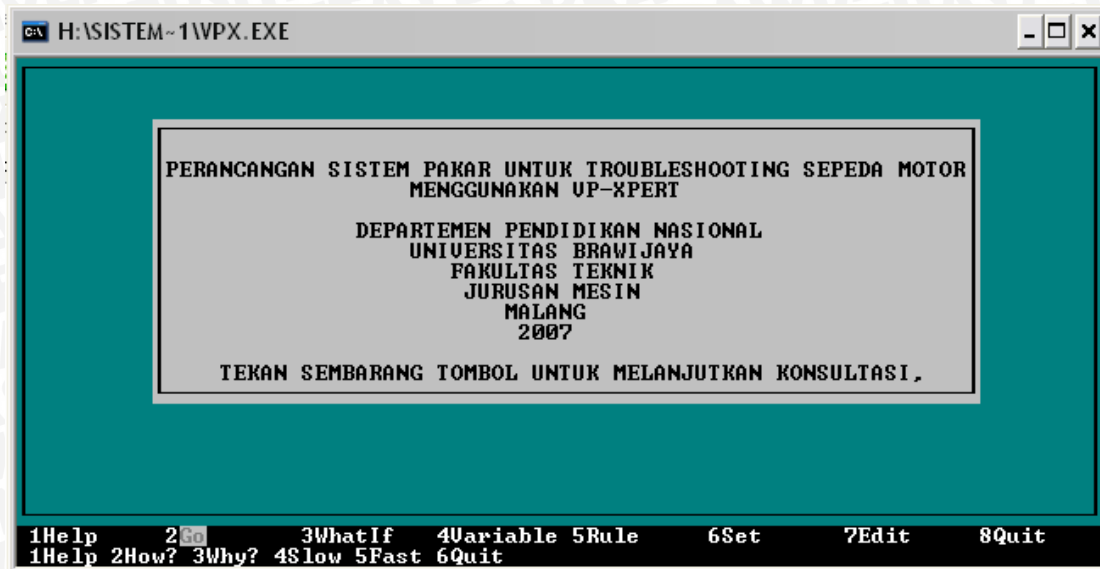
Untuk verifikasi, maka perlu menjalankan konsultasi melalui prototipe yang ada. Untuk dapat menjalankan konsultasi program konsultasi VP-Expert, terlebih dahulu harus menjalankan aplikasi VP-Expert, yaitu dapat dilakukan dengan me-load-nya dari sistem prompt (under DOS) dengan mengetikkan VPX dan menekan tombol ENTER sehingga tampil layar sebagai berikut :



Gambar 4.1 Tampilan VP-Expert

Setelah itu, untuk memulai konsultasi dilakukan dengan langkah-langkah sebagai berikut:

1. Pilih file name dari menu utama untuk membuka daftar file-file basis pengetahuan. Kemudian pilih file dengan nama troubleshooting sepeda motor.KBS dengan menggunakan tombol panah dan menekan ENTER.
2. Ketika menu utama tampil, pilih consult dan selanjutnya pilih go dari menu consult tersebut sehingga muncul layar sebagai berikut:



Gambar 4.2 Konsultasi VP-Expert

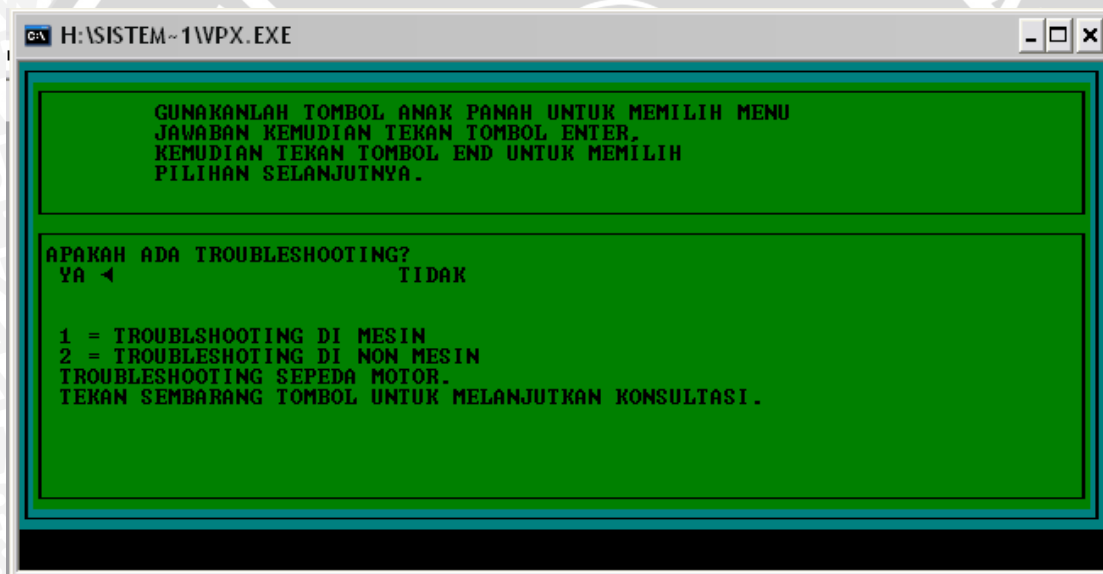
3. Untuk selanjutnya, ikuti perintah yang diberikan pada jendela konsultasi, dalam hal ini adalah dengan menekan sembarang tombol untuk memulai konsultasi. Jawablah pertanyaan-pertanyaan yang diajukan dengan cara :

- Gunakan tombol anak panah untuk berpindah dari pilihan yang satu ke pilihan yang lain.
- Tekan ENTER pada pilihan jawaban yang diinginkan (memunculkan sebuah segitiga di sebelah item yang dipilih tersebut).
- Tekan END untuk mengakhiri pilihan.
- Untuk membatalkan pilihan jawaban sebelum menekan END, caranya dengan menekan tombol DELETE.

Diakhir konsultasi akan diberikan suatu rekomendasi tentang pengecekan troubleshooting sepeda motor bagi user berdasarkan jawaban yang diberikan user pada saat konsultasi sebagai berikut :



Gambar 4.3 Tampilan Pesan VP-Expert



Gambar 4.4 Rekomendasi VP-Expert

Ketik Q (menu quit) untuk kembali ke menu utama VP-Expert atau ketik G (menu go) untuk melanjutkan konsultasi.

- Memberikan Perintah Pada VP-Expert

Pilihan perintah yang disediakan oleh menu VP-Expert dapat dipilih dengan menggunakan salah satu dari keempat cara berikut :

- Mengetikkan huruf pertama dari perintah tersebut.
- Mengetikkan nomor sesuai dengan yang terdapat di sebelah kiri perintah tersebut.
- Mengetikkan tombol fungsi (F1, F2, dan seterusnya) sesuai dengan nomor perintah tersebut.

- Menggunakan tombol anak panah untuk menggerakkan cursor ke perintah yang diinginkan dan diakhiri dengan menekan ENTER.

4.4.2 Validasi

Setelah meload troubleshooting A.KBS awali konsultasi dengan memilih Go dari menu Consult.. Jalankan konsultasi dan jawab pertanyaan yang diberikan berdasarkan ketentuan berikut :

```
! ##### RULES 1 #####
! ===== VALIDATION RULE 1 =====
A = YA
```

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

```
! ===== RECOMMENDATION RULE 1 =====
1 = TROUBLSHOOTING DI MESIN
2 = TROUBLESHOTING DI NON MESIN TROUBLESHOOTING SEPEDA MOTOR
```

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.

```
! ===== VALIDATION RULE 1 =====
A = TIDAK
```

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

```
! ===== RECOMMENDATION RULE 1 =====
TEKAN SEMBARANG TOMBOL UNTUK MELANJUTKAN KONSULTASI
```

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.

```
! ##### RULES 2 #####
! ===== VALIDATION RULE 2 =====
B = YA
```

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

```
! ===== RECOMMENDATION RULE 2 =====
1 = KUNCI
2 = BENSIN
3 = BUSI
4 = KABEL BUSI
5 = FILTER UDARA
6 = KNALPOT
7 = KARBURATOR
8 = ACCU
9 = MOTOR
10 = OLI
```

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.



! ===== VALIDATION RULE 2 =====
 B = TIDAK

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

! ===== RECOMMENDATION RULE 2 =====
 TEKAN SEMBARANG TOMBOL UNTUK MELANJUTKAN KONSULTASI

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.

! ##### RULES 3 #####
 ! ===== VALIDATION RULE 3 =====
 B = YA

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

! ===== RECOMMENDATION RULE 3 =====
 COBA TERUS SEPULUH KALI

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.

! ===== VALIDATION RULE 3 =====
 B = TIDAK

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

! ===== RECOMMENDATION RULE 3 =====
 TEKAN SEMBARANG TOMBOL UNTUK MELANJUTKAN KONSULTASI

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.

.....

! ##### RULES 55 #####
 ! ===== VALIDATION RULE 55 =====
 C = YA

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

! ===== RECOMMENDATION RULE 55 =====

- 1 = ENGINE TIDAK DAPAT HIDUP DENGAN MOTOR STARTER
- 2 = ENGINE TIDAK DAPAT HIDUP DENGAN KICK STARTER
- 3 = PERFORMANCE ENGINE KURANG
- 4 = KERUSAKAN SISTEM PELUMASAN
- 5 = GAS BUANG MEMPERLIHATKAN WARNA ABNORMAL
- 6 = PEMBAKARAN ABNORMAL
- 7 = SUARA SUARA ABNORMAL
- 8 = KOPLING
- 9 = TRANSMISI
- 10 = POROS PENGGERAK
- 11 = RANTAI PENGGERAK

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.



! ===== VALIDATION RULE 55 =====

C = TIDAK

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

! ===== RECOMMENDATION RULE 55 =====

TEKAN SEMBARANG TOMBOL UNTUK MELANJUTKAN KONSULTASI

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.

.....

! ##### RULES 65 #####

! ===== VALIDATION RULE 65 =====

C10 = YA

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

! ===== RECOMMENDATION RULE 65 =====

ISI OLI SAMPINGNYA SEGERA

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.

! ===== VALIDATION RULE 65 =====

C10 = TIDAK

Di akhir konsultasi, sistem akan memberikan rekomendasi dengan pesan :

! ===== RECOMMENDATION RULE 65 =====

TEKAN SEMBARANG TOMBOL UNTUK MELANJUTKAN KONSULTASI

Ternyata rekomendasi ini sesuai dengan formal knowledge atau expert knowledge.



BAB V PENUTUP

5.1 KESIMPULAN

Kesimpulan bertujuan menjawab rumusan masalah yang kemudian didetailkan pada tujuan perancangan seperti ditulis di bab I. Berikut kesimpulan yang didapat:

- Telah diperoleh dan disusun pengetahuan (*Knowledge Acquisition*) tentang kerusakan sepeda motor Yamaha yang terdiri atas:
 - *Formal Knowledge*, misal tentang: karburator, engine, sistim pelumasan
 - *Expert Knowledge*, misal tentang: suara aneh dan warna gas buang. Hasil ada di bab 2.7 dan 2.8
- Atas dasar pengetahuan diatas telah dikembangkan aturan (Rules) yang meliputi himpunan rules antara lain : Sistim penyalaaan, karburator, suara abnormal, rantai penggerak, dst. Hasil ada di bab 4.2.4
- Kemudian atas dasar aturan diatas, maka telah dikembangkan aturan dan user interface yang diperlukan bagi pembuatan prototipe dengan alat VP-Expert. Setelah itu, telah dilaksanakan uji verifikasi dan validasi dari prototipe dengan hasil yang memuaskan. Hasil ada di bab 4.4.1 dan 4.4.2
- Untuk melengkapi pengguna maka telah dibuat manual kecil untuk menjalankan prototipe ini yang dimasukkan di bab 4.2.6 bersama dengan uji verifikasi dan validasi.

5.2 SARAN

Saran-saran yang dapat diberikan sebagai masukan untuk mengembangkan Tugas akhir ini lebih lanjut adalah sebagai berikut :

Bagi kelanjutan riset :

1. Program sebaiknya dikembangkan untuk sub-fokus masalah lainnya, misalnya untuk sistem kompresi; atau bidang yang lain turbin, alat kedokteran, PC, dst.
2. Mencoba menggunakan alat pengembang sistim pakar yang versinya lebih baru.
3. Program *expert system* yang ada akan lebih sempurna jika diintegrasikan dengan database, misalnya database untuk mengetahui ketersediaan sparepart.

Bagi user :

1. Agar prototipe ini dapat dikembangkan menjadi *software* siap pakai (profesional) sehingga dapat membantu para pemilik sepeda motor Yamaha yang awam tentang pengetahuan permesinan.

2. Disarankan agar perusahaan sepeda motor melengkapi user manualnya dengan *software* sistem pakar untuk trouble shooting motornya.



DAFTAR PUSTAKA

- Brian Sawyer; 1987: *VP-Expert : Rule-Based Expert System Development Tool*; Paperback Software International, USA.
- D.G. Dologite; 1993: *Developing Knowledge-Based System Using VP-Expert*; Macmillan Publishing Company, New York.
- Elisa Tjomardi; 2005: *Studi Kajian Alat Pengembangan Software Aplikasi Sistem Pakar Dengan Menggunakan VP-Expert*; Sekolah Tinggi Informatika dan Komputer Indonesia, Malang
- <http://balzach.staff.ugm.ac.id/AI/Diktat%20Kecerdasan%20Buatan.pdf>
- http://media.wiley.com/product_data/excerpt/18/0471293318.pdf
- http://www.media.wiley.com/product_data/excerpt/05/08186720/0818672005.pdf
- <http://www.sbaer.uca.edu/research/sbida/1995/pdf/17.pdf>
- <http://www.scism.sbu.ac.uk/~darlink/ai/lectures/a7.pdf>
- John Durkin; 1993: *Expert System Catalog of Applications*; Intelligence Computer System Inc, USA.
- Muhammad Arhami; 2005: *Konsep Dasar Sistem Pakar*; Penerbit ANDI, Yogyakarta.

