

PENERAPAN METODE FORWARD CHAINING DENGAN TEKNIK REPRESENTASI RULE BASED REASONING UNTUK DIAGNOSA KERUSAKAN MOBIL BERBASIS ANDROID

Zeraida Wele¹, Edy Mulyanto²

¹Program Studi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Dian Nuswantoro Semarang
Jl. Nakula I No 5-11 Semarang 50131
Telp : (024) 3517361, Fax : (024) 3520165
Email : zeraidawele@gmail.com¹, edymul007@gmail.com²

Abstrack

The development of technology recently, the human need for information is also increasing in various fields. The development of science and technology brings great influence among the community in meeting any needs that exist especially in the computer field. Therefore, an application system is needed to provide fast, precise and accurate in supporting the development and needs. Increasing the percentage of recent car sales make information systems is required by the owner of the car to recognize the symptoms of damage. Most of the car owners are not aware of the damage done to the car resulting in damage becomes more severe and require large service fee. Applications diagnose expert system based on android damage to the car was made as a tool to analyze the damage caused to the car by providing accurate information in repairs. The application of methods and techniques of representation forward chaining rule-based reasoning should be able to overcome the problem of damage based on user perceived symptoms. This app utilizes android mobile technology as a device that can be used with a small memory capacity for installation.

Keywords: Car Damage, Expert System, Forward Chaining, Rule Based Reasoning

I. PENDAHULUAN

Melihat kebutuhan manusia yang semakin banyak dan kompleks saat ini dalam bidang ilmu pengetahuan dan teknologi, manusia dituntut untuk lebih maju dan berkembang dalam mengikuti perkembangan teknologi yang telah mengalami banyak perubahan yang sangat pesat. Perkembangan yang ada tentunya tak lepas dari perkembangan dibidang komputer yang memiliki kemampuan untuk

menyelesaikan dan membantu daya ingat manusia yang terbatas dalam menyelesaikan suatu masalah dalam berbagai bidang walaupun setiap penanganannya tidak semua memakai komputer [1].

Seorang pakar adalah orang yang ahli dibidangnya, akan tetapi disisi lain seorang pakar juga mempunyai keterbatasan daya ingat. Kemungkinan yang menjadi salah satu faktornya adalah usia dari seorang pakar tersebut sehingga dapat saja melakukan kesalahan pada hasil diagnosa yang tentunya menghasilkan suatu kesalahan pada solusi masalah yang ada [2].

Perkembangan dalam dunia otomotif belakangan ini sangat meningkat, secara khusus kendaraan mobil yang telah menjadi salah satu kebutuhan masyarakat yang harus terpenuhi. Menurut Presiden Direktur PT Toyota Astra Motor, Johnny Darmawan, menyebutkan tahun lalu merupakan salah satu tahun yang penuh tantangan, baik bagi pasar otomotif nasional maupun Toyota Indonesia. Toyota berhasil melewati 2013 dengan membukukan rekor penjualan sebanyak 434.232 unit atau tumbuh 7,1% dibandingkan dengan 2012. Khusus untuk Toyota Avanza, MPV ini kembali tercatat sebagai kontributor terbesar penjualan, yakni 213.458 unit atau tumbuh 11,1% dibandingkan dengan 2012 sebanyak 192.146 unit dan mencapai 49,15% dari total penjualan Toyota selama 2013 []. Menurut data dari PT. Hasjrat Abadi (TOYOTA) – CV. Kombos Mutiara Palu, penjualan mobil pada bulan September 2014 meningkat sampai 10% dan service berkala meningkat mencapai 15%.

Berdasarkan hal tersebut maka dibuatlah suatu aplikasi berbasis Android yang memudahkan para pemilik mobil dan juga dapat digunakan oleh teknisi atau mekanik untuk mendiagnosa kerusakan yang terjadi pada mobil dan untuk ikut serta dalam pengembangan aplikasi Android yang tentunya dapat bermanfaat bagi pengguna Android lainnya dalam mengidentifikasi kerusakan pada mobil.

II. METODE YANG DIUSULKAN

2.1 Kerusakan Mobil

Ada banyak pemilik mobil yang tidak mau membaca atau mempelajarinya dengan alasan membosankan, padahal itu merupakan petunjuk paling sederhana dan mudah untuk merawat mobil. Selain itu, banyak pemilik mobil yang tidak mengetahui dengan cepat kondisi mobilnya, sehingga tidak segera melakukan perbaikan, saat mobilnya menunjukkan tanda-tanda kerusakan

yang menyebabkan kerusakan tersebut semakin parah [3].

Sekarang ini dapat dilihat bahwa pemilik mobil baru sadar jika sudah terjadi kerusakan yang cukup parah, seperti mobil sama sekali tidak bisa berjalan lagi. Oleh karena itu, agar kondisi mobil selalu normal dan komponen-komponennya bisa tahan lama, pemilik perlu melakukan perawatan seperti mengganti oli setiap bulannya dan service berkala lainnya.

Gejala kerusakan pada mobil bisa dirasakan atau dideteksi saat mobil sedang berjalan, seperti : timbul bunyi saat melewati jalan rusak, laju mobil tersendat-sendat, boros bensin dan oli, suara abnormal dari rem, transmisi macet pada posisi gigi mundur dan lain sebagainya.

2.2 Sistem Pakar

Sistem pakar adalah suatu model dan prosedur yang berkaitan, dalam suatu domain tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar [4]. Sistem pakar adalah program AI (Artificial Intelligence) dengan basis pengetahuan (knowledge base) yang diperoleh dari pengetahuan beberapa pakar atau ahli dalam memecahkan persoalan pada bidang tertentu dan didukung mesin inferensi (Inferensi Engine) yang melakukan penalaran atau pelacakan terhadap sesuatu atau fakta-fakta yang diberikan oleh user lalu dicocokkan (Matching) dengan fakta-fakta dan aturan atau akidah yang ada dibasis pengetahuan setelah dilakukan pencarian, sehingga tercapai kesimpulan [5].

Dengan demikian, seorang awam sekalipun bisa menggunakan sistem pakar untuk memecahkan berbagai persoalan yang ia hadapi, dan bagi seorang ahli sistem pakar dapat dijadikan alat untuk menunjang

aktivitasnya yaitu sebagai asisten yang berpengalaman [6].

2.3 Forward Chaining

Pada penelitian ini metode yang diusulkan adalah metode *Forward Chaining* dengan teknik representasi *Rule Based Reasoning*. Mekanisme dari sistem *forward chaining* dimulai dengan memasukkan sekumpulan fakta yang diketahui kedalam memori kerja, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari *rules* IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka *rule* tersebut dieksekusi [2]. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan kedalam *database*. Setiap kali pencocokkan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokkan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi atau sudah mencapai *goal* atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui.

Bentuk representasi *Rule Based Reasoning* digunakan karena memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu dan pakar dapat menyelesaikan masalah tersebut secara sistematis dan berurutan. Representasi berbasis aturan yang mempunyai pola *if* kondisi/premis *then* aksi/konklusi pada suatu tabel pakar akan memberikan keuntungan pada berbagai aspek, diantaranya mudah dalam memodifikasi, baik perubahan data, penambahan data atau penghapusan data. Dalam hal ini *if* bisa direpresentasikan sebagai gejala-gejala yang dirasakan saat mengendarai mobil dan *then* berupa solusi-solusi yang dicapai.

Contoh pelacakan *forward chaining* :

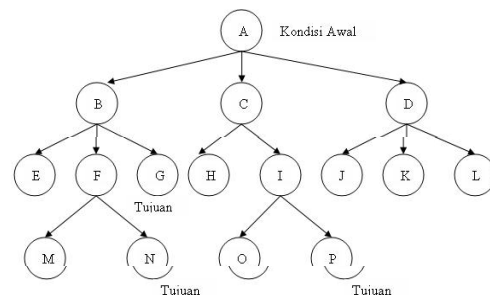
Rule - rule yang diberikan :

- a. R1 : Jika A dan C, maka E

- b. R2 : Jika D dan C maka F
- c. R3 : Jika B dan E maka F
- d. R4 : Jika B maka C
- e. R5 : Jika F maka G

Fakta yang ada : A benar dan B benar

Dalam *Forward Chaining* pencarian dimulai dengan fakta yang diketahui dan mengambil fakta baru menggunakan aturan yang telah diketahui pada sisi Jika. Karena diketahui A dan B benar, sistem pakar mulai dengan mengambil fakta baru menggunakan aturan yang memiliki A dan B pada sisi Jika. Dengan menggunakan R4, sistem pakar mengambil fakta baru C dan menambahkannya ke dalam *assertion base* sebagai benar. Sekarang R1 fire (karena A dan C benar) dan nyatakan E sebagai benar dalam *assertion base* sebagai benar. Karena B dan E keduanya benar (berada dalam *assertion base*), R3 fire dan menetapkan F sebagai benar dalam *assertion base*. Sekarang R5 fire (karena F berada dalam sisi Jika), yang menetapkan G sebagai benar, jadi hasilnya adalah G.



Gambar 2.1 Forward Chaining [7]

Kelebihan dari metode runut maju (*Forward Chaining*) ini diantaranya adalah [8]:

- a. Dapat menghasilkan informasi baru dari jumlah data yang relative sedikit.
- b. Merupakan pendekatan yang baik untuk masalah tertentu seperti perencanaan, pengawasan, pengaturan, dan interpretasi,

- c. Dapat bekerja baik dengan permasalahan yang membutuhkan informasi lebih dulu.

III. IMPLEMENTASI

Berdasarkan data yang sudah diperoleh yang kemudian dikelola melalui tahap analisa data dalam basis pengetahuan, maka data tersebut akan diterapkan pada sebuah aplikasi diagnosa kerusakan mobil yang nantinya dapat dijalankan pada mobile android ataupun PC. Penerapan metode yang digunakan dalam diagnosa kerusakan mobil berbasis android ini adalah tahapan yang sangat penting untuk menentukan berhasil tidaknya aplikasi yang dibuat berdasarkan data yang didapatkan sesuai dengan tujuan atau solusi nantinya.

Antarmuka sistem pakar ini dirancang dengan melihat kebutuhan *user* (*user friendly*) maupun kebutuhan perangkat lunak itu sendiri, sehingga terjadi keseimbangan antar keduanya.

Dibawah ini merupakan gambaran *interface* dari aplikasi yang sudah dibuat.

A. Menu Utama

Pada Tampilan halaman menu utama aplikasi Diagnosa Kerusakan Mobil Toyota Avanza ini ada 4 menu yang terdapat didalamnya, yaitu menu peringatan, umum, diagnosa dan profil. Untuk setiap menu dibuat lebih mudah agar dapat digunakan oleh *user* sehingga para pemula pun dapat memahami dalam menggunakan aplikasi ini dengan baik.



Gambar 3.1 Tampilan Halaman Utama

B. Submenu diagnosa

Sub menu diagnosa merupakan hasil implementasi dari metode *forward chaining* yang sudah dibentuk dalam beberapa aturan (*rule*). Pada halaman ini, pengguna dapat memilih beberapa gejala kerusakan mobil yang dialami. Setelah *user* selesai memilih beberapa gejala, maka klik OK untuk menampilkan jenis kerusakan apa yang terjadi, penyebab kerusakan dan bagaimana solusi pengendaliannya.



Gambar 3.2 Tampilan Menu Diagnosa



Gambar 3.3 Tampilan Hasil Diagnosa



Gambar 3.4 Tampilan Gejala Kerusakan



Gambar 3.5 Tampilan Penyebab Kerusakan



Gambar 3.6 Tampilan Solusi Perbaikan

IV. HASIL & PEMBAHASAN

4.1 Pengujian

Pengujian (*testing*) merupakan bagian atau tahapan yang tidak kalah pentingnya dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Pengujian yang dilakukan pada program sistem pakar diagnosa kerusakan mobil Toyota Avanza ini menggunakan uji *black box*. *Black box testing* merupakan *testing* dimana software tester tidak memiliki akses *source code* atau mengetahui implementasi dari program tersebut untuk mencari adanya kesalahan pada program dan juga tidak diharuskan memiliki pengetahuan tentang programming dan implementasinya. Ketika melakukan *black box testing*, tester akan berinteraksi dengan *user interface* yang menyediakan *input* dan memeriksa *outputnya*, juga menguji performa program atau menguji

function-function yang tidak bekerja dengan benar.

Tabel 3.1 Hasil Pengujian *Blackbox*

Kasus Uji	Langkah Penelitian	Hasil yang diharapkan	Hasil Uji	Status
Menu Peringatan	<ol style="list-style-type: none"> Masuk ke menu utama Pilih dan klik tombol peringatan 	Tampilan halaman menu peringatan	Tampilan halaman menu peringatan	Sesuai
Menu Umum	<ol style="list-style-type: none"> Masuk ke menu utama Pilih dan klik tombol umum 	Tampilan menu umum	Tampilan menu umum	Sesuai
Submenu diagnosa	<ol style="list-style-type: none"> Masuk ke menu utama Pilih dan klik tombol menu diagnosa Pilih dan klik tombol submenu diagnosa 	Tampilan halaman submenu diagnosa	Tampilan halaman submenu diagnosa	Sesuai

4.2 Pemeliharaan Program

Pada saat program sistem pakar diagnosa kerusakan mobil ini digunakan oleh *user* dibutuhkan beberapa pemeliharaan (maintenance) program yaitu sebagai berikut:

1. Melakukan back up data ke media penyimpanan
2. Penyimpanan luar untuk menjaga data program
3. Melakukan perbaikan jika dalam proses operasi terjadi kesalahan (*bugs*) dalam program
4. Melakukan modifikasi terhadap program
5. Melakukan Updating data kerusakan mobil. Updating dilakukan pada saat ada informasi atau yang lainnya yang harus diganti ataupun ditambahkan pada program untuk menjaga agar informasi yang diberikan selalu akurat dan agar program tersebut selalu terlihat baru.

Maintenance adalah tahap dimana sistem yang sudah ada dapat terus dikembangkan mengikuti perkembangan teknologi informasi maupun basis pengetahuan agar tidak menjadi usang dan dapat melakukan perawatan apabila terjadi *fatal error* agar sistem tetap dapat digunakan.

V. PENUTUP

5.1 Kesimpulan

Dari penelitian yang dibuat, maka dapat ditarik kesimpulan sebagai berikut :

1. Aplikasi yang dirancang dapat mendiagnosis kerusakan yang terjadi pada mobil.
2. Aplikasi sistem pakar ini mampu memberikan informasi yang cepat dan tepat dalam penanganan kerusakan yang terjadi pada mobil.
3. Implementasi metode *Forward Chaining* dengan representasi *Rule Based Reasoning* terbukti mampu memberikan diagnosa yang tepat.

4. Pengujian pada platform Android mampu memberikan kemudahan dan kecepatan akses (*loading*) bagi mekanik maupun pemilik mobil dalam menjalankan aplikasi kerusakan yang terjadi pada mobil.

5.2 Penelitian Selanjutnya

Adapun berbagai saran yang bisa digunakan untuk pengembangan penelitian selanjutnya yang dapat memberikan manfaat dalam pelaksanaannya antara lain:

1. Perlu adanya perbaikan-perbaikan kembali demi kesempurnaan aplikasi dan kemudahan pengguna (*user friendly*).
2. Perlu adanya pengembangan selanjutnya dengan menggunakan metode-metode lainnya yang bisa diterapkan guna menghasilkan kesimpulan dengan presentase yang lebih akurat dan variatif.
3. Dapat di tambahkan informasi dan data kerusakan mobil pada program aplikasi sistem pakar ini berupa nama kerusakan, gejala, penyebab dan langkah penanganan/solusi sehingga lebih lengkap dan akurat.
4. *Update* dan penambahan data pada aplikasi ini masih manual, dilakukan diluar dari *interface* aplikasi, akan lebih baik apabila dapat melakukan *update* maupun menambahkan data secara langsung pada *interface* dengan menambah menu admin pada *interface*.

REFERENCES

- [1] Dwi Cahyo, "Pengantar Komputer dan Perkembangannya," 2010.
- [2] T Sutojo, *Kecerdasan Buatan*, Rini W Benedicta, Ed. Semarang, Indonesia: ANDI Yogyakarta, 2011.

- [3] Sudirman Urip, *Deteksi Dini Gejala Kerusakan Pada Mobil*. Jakarta, Indonesia: PT. Kawan Pustaka, 2009.
- [4] Ignizo J. P, *Introduction To Expert System: The Development and Implementation Of Rule-Based Expert System*. New York: McGraw-Hill, 1991.
- [5] Sri Kusumadewi, *Kecerdasan Buatan*, Edisi Pertama ed. Yogyakarta, Indonesia: Graha Ilmu, 2005.
- [6] Sri Kusumadewi, *Artificial Intelligence*. Jogjakarta, Indonesia: Graha Ilmu, 2003.
- [7] Riley Giarratano, *Expert System : Principles and Programming*, Fourth Edition ed.: Course Tecnology, 1998.
- [8] Riskadewi dan Hendrik Antonius, *Penerapan Sistem Pakar Forward Chaining Berbasis Aturan pada Pengawasan Status Penerbangan.*: Integral, 2005.
- [9] Antonius Riskadewi dan Hendrik, *Penerapan Sistem Pakar Forward Chaining Berbasis Aturan pada Pengawasan Status Penerbangan.*: Integral, 2005.