

APLIKASI ANALISA MASALAH MESIN MOTOR BEBEK MENGUNAKAN METODE BACKWARD CHAINING

Angga Sulchan Saputra¹⁾, Bayu Kuncoro Jati²⁾, Sumdoro Fajar Utomo³⁾

^{1), 2), 3)} Teknik Informatika STMIK AMIKOM Yogyakarta

Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281

Email : angga.7227@students.amikom.ac.id¹⁾, bayu.ja@students.amikom.ac.id²⁾,
sundoro.u@students.amikom.ac.id³⁾

Abstrak

Setiap orang pasti mempunyai sepeda motor dan tidak sedikit juga mempunyai masalah dengan kerusakan sepeda motor. Selain itu, jangkauan bengkel sepeda motor bebek masih terhitung jauh dari tempat tinggal

Tujuan dari penulisan ini adalah untuk menyelesaikan permasalahan pengguna yang mempunyai masalah dengan kerusakan pada sepeda motor dimanapun berada. Dengan merancang aplikasi yang ramah terhadap pengguna (*user friendly*) yang berbasis komputer untuk mendeteksi kerusakan pada sepeda motor berdasarkan gejala yang terjadi. Metode penulisan yang dipakai adalah metode **Backward Chaining**. dengan pelacakan menggunakan metode **Backward Chaining** yang berjumlah 8 rule akan ditemukan masalah dengan fakta yang ada dan kemudian dicoba untuk mencocokkan dengan tujuan yang diharapkan.

Setelah penelitian ditemukan hasil berupa aplikasi sistem pakar dengan menggunakan bahasa pemrograman java yang dapat membantu masalah yang dialami oleh pengguna motor bebek yang masih awam.

Kata kunci: Backward Chaining, Sistem Pakar, motor bebek,

1. Pendahuluan

Pada zaman modern ini, alat transportasi menjadi kebutuhan yang paling dasar. Dengan menggunakan alat transportasi, segala aktivitas menjadi ringan. Sebagian masyarakat menggunakan alat transportasi sepeda motor untuk melakukan segala aktivitas. Dengan seiringnya berjalan waktu, sepeda motor juga memerlukan perawatan agar tetap bisa digunakan. Banyak pengendara sepeda motor tidak mengetahui kerusakan pada sepeda motor tersebut. Kerusakan akan menjadi fatal bila tidak ditangani dengan cepat. Oleh karena itu, kami membuat aplikasi untuk menganalisis jenis kerusakan sepeda motor berdasarkan gejala-gejala fakta dengan menggunakan system pakar *backward chaining*. Dapat ditemukan berbagai rumusan masalah, yakni, bagaimana cara mengetahui penyebab kerusakan pada sepeda motor?, bagaimana cara memperoleh informasi tentang jenis kerusakan yang dialami sepeda motor dengan bahasa yang mudah dimengerti oleh pengguna?, bagaimana memberikan solusi dengan akurat memecahkan masalah kerusakan sepeda motor?, dan

bagaimana merancang dan membangun sebuah aplikasi dengan metode *backward chaining* yang mudah dimengerti (*user friendly*)? Aplikasi ini bertujuan untuk mengetahui jenis kerusakan pada sepeda motor tanpa mekanik, menambah ilmu pengetahuan bagi pengguna tentang masalah teknis sepeda motor, membuat aplikasi yang dapat menganalisis kerusakan pada sepeda motor berdasarkan informasi fakta.

2. Pembahasan

Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer digunakan menyelesaikan masalah seperti yang dipikirkan oleh sang pakar (orang yang punya keahlian khusus). Dengan kata lain sistem pakar bagian dari kecerdasan buatan dimana komputer dapat berfikir dan mengambil kesimpulan dari sekumpulan aturan. Sebagai contoh, montir adalah seorang yang punya keahlian dan pengalaman dalam menyelesaikan kerusakan mesin motor berdasarkan keluhan dari pelanggan. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan memberikan penatalaksanaan suatu kerusakan. Sistem pakar ada 2 komponen utama yaitu mesin inferensi dan basis pengetahuan. Basis pengetahuan merupakan tempat penyimpanan pengetahuan dalam memori computer dimana pengetahuan ini diambil dari pengetahuan pakar. Sistem pakar akan dipandang saat mampu mengambil keputusan sama seperti yang dilakukan oleh sang pakar. Otak dari aplikasi sistem pakar adalah mesin inferensi. Mesin inferensi dapat menuntun pengguna untuk memasukkan fakta hingga diperoleh suatu kesimpulan.[1]

Backward Chaining

Backward chaining (runut balik) adalah metode penalaran dimulai dengan tujuan kemudian merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut. Runut balik juga disebut sebagai *goal-driven reasoning*, merupakan cara efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur. Penelusuran didasarkan pada suatu keyakinan bahwa ada kemungkinan konklusi dari daftar konklusi merupakan salah satu tujuan atau konklusi terpilih berdasarkan fakta yang diberikan oleh user. Sistem dengan urutan tertentu akan mengambil sebuah konklusi sebagai calon konklusinya. Misal urutannya adalah sesuai dengan urutan konklusi. Awalnya sistem akan mengambil

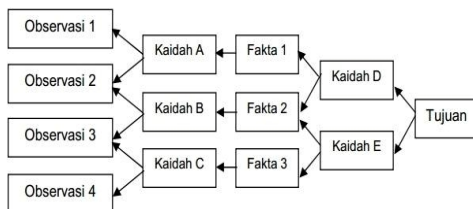
hipotesis bahwa konklusinya adalah konklusi 1. Untuk membuktikan hipotesisnya, sistem akan mencari premis-premis aturan yang mengandung konklusi 1. Setelah itu sistem akan meminta umpan balik kepada user mengenai premis-premis yang ditemukan tersebut. Untuk konklusi 1 premisnya adalah premis 1, premis 2 dan premis 3, maka sistem akan mencari tahu apakah user memilih premis-premis tersebut.

Keuntungan dengan menggunakan metode *backward chaining* ialah:

1. *Backward chaining* terfokus pada tujuan yang diberikan. Prosedur ini akan menanyakan hal-hal yang perlu saja dan ini merupakan kenyamanan bagi user.
2. Bila *forward chaining* mencoba semua kemungkinan dari informasi yang ada, *backward chaining* mencoba menyelesaikan masalah dengan mencari basis pengetahuan yang relevan dengan masalah sekarang.
3. *Backward chaining* merupakan pendekatan yang baik untuk menyelesaikan suatu diagnostik, preskripsi, dan *debugging*. [2][3]

Arsitektur Sistem Pakar

Komponen utama yang harus ada dalam sebuah sistem pakar adalah *knowledge base* (basis pengetahuan), *inference engine* (mesin penarik kesimpulan), *explanation subsystem* (subsistem penjelas output) dan *user interface*. Juga adanya diagram yang ditunjukkan untuk menjelaskan tujuan dari sistem pakar tersebut



Gambar 1. Diagram Sistem Pakar

Knowledge Acquisition

Knowledge acquisition adalah proses mendapatkan pengetahuan dari seorang pakar dan biasanya ditampilkan oleh pengetahuan (*knowledge engineer*). Pengolah pengetahuan mewancarai pakar-pakar dan mengumpulkan pengetahuan yang ada dari manusia. Pengetahuan atau data-data yang dikumpulkan disebut sebagai *knowledge base*. Tahap-tahap dalam *knowledge acquisition* dibagi menjadi lima tahapan, yaitu:

1. Identifikasi Merupakan tahap mengidentifikasi permasalahan dan karakteristik utamanya.
2. Konseptualisasi tahap penentuan konsep, informasi dan relasi yang digunakan serta menentukan bagaimana representasi yang akan digunakan.

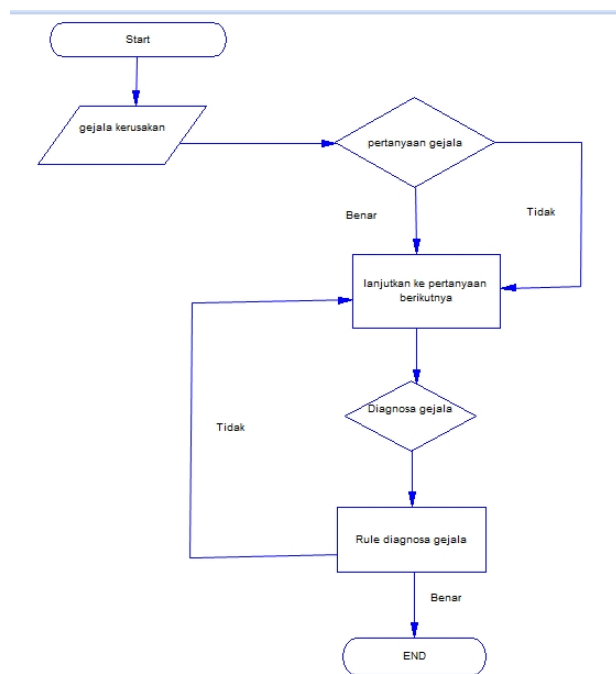
3. Formalisasi merupakan tahap perancangan struktur Untuk mengorganisasikan pengetahuan dan merepresentasikannya ke *knowledge base*.

4. Implementasi merupakan tahap pengkodean pengetahuan yang telah diolah ke dalam komputer.

5. Pengujian merupakan tahap pengujian kebenaran dari pengetahuan yang telah dibentuk.

Flowchart Sistem

Bagan Alir Sistem (*System Flowchart*) Bagan alairsistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada dalam di dalam sistem, bagan alir sistem menunjukkan apa yang dikerjakan di sistem



Gambar 2. Flowchart Sistem

Metode Penelitian

Metodologi penelitian yang akan digunakan dalam membangun aplikasi sistem pakar adalah metode *waterfall* (Pressman, 2005).

Metode pendekatan *waterfall* yaitu metode yang menggunakan pendekatan secara sistematis dan urutan mulai dari level mendefinisikan kebutuhan sistem sampai maintenance.

Metode *waterfall* memiliki beberapa tahapan yaitu :

1. Mendefinisikan kebutuhan, maksudnya adalah mengumpulkan kebutuhan dan entitas yang diperlukan untuk menyusun sejumlah kecil analisa informasi, baik strategi maupun area bisnis.

2. Menganalisis kebutuhan, berarti terjemahan dari tahap pertama, yang menguraikan definisi dari perangkat lunak diantaranya kebutuhan sistem, aplikasi yang digunakan, *interface*, bentuk proses pengolahan informasi, performansi yang diharapkan, pendokumentasian dan lain-lain yang terkait dengan definisi dan pemfokusan persoalan rekayasa perangkat lunak.

3. Mendesain sistem dan perangkat lunak, merupakan tahap penjabaran multifungsi dari analisa kebutuhan, prosesnya melalui tahapan struktur data, arsitektur perangkat lunak, representasi *interface*, algoritma, dan lain-lain.

4. Koding, yaitu pembuatan program atau menerjemahkan hasil rancangan ke dalam bahasa pemrograman tertentu. Penulisan kode program sesuai dengan desain yang dibuat, sehingga bisa menghasilkan aplikasi yang bermanfaat bagi pengguna.

5. Pengujian sistem dan integrasi, yaitu melakukan pengujian terhadap aplikasi yang telah dibuat dengan menyesuaikan kebutuhan, sekaligus mengintegrasikan komponen dalam sistem tersebut.

6. Implementasi dan *maintenance*, yaitu mengaplikasikan sistem yang sudah terintegrasi dan melakukan perawatan atau perbaikan bila ada kekeliruan

Pengumpulan Data

Pembuatan aplikasi didasari oleh data-data yang telah kami peroleh dari hasil observasi gejala kerusakan motor, berikut data ditampilkan menggunakan tabel:

Tabel 1. Table aturan gejala kerusakan motor

Rule 1	If motor susah dihidupkan dengan menggunakan starter elektrik atau secara manual : And tenaga yang dihasilkan lemah And Mesin cepat panas And busi mati tidak ada percikan And Keluar asap putih pada knalpot And suara kasar pada mesin Then kerusakan pada piston
Rule 2	If motor susah dihidupkan dengan menggunakan starter elektrik atau secara manual : And tenaga yang dihasilkan lemah And mesin tersendat-sendat And busi mudah mati And warna percikan pada busi berwarna merah Then kerusakan pada CDI
Rule 3	If motor susah dihidupkan dengan menggunakan starter elektrik atau secara manual : And mesin tidak stasioner(gas tidak konstan) And keluar asap hitam pada knalpot And bahan bakar boros And oli cepat

	habis Then kerusakan pada katup atau klep
Rule 4	If ornament motor mati saat dihidupkan : And lampu posisi transmisi persneling/gigi mati And jarum bensin tidak normal And jarum kecepatan mati/tidak normal And odometer tidak berjalan Then kerusakan pada speedometer
Rule 5	If motor tersendat-sendat saat digunakan : And susah/keras saat mengganti transmisi persneling /gigi motor And suara menjadi kasar saat mengganti transmisi And tidak bisa mengganti transmisi persneling/gigi ke 1 dari top gear(puncak transmisi) Then kerusakan pada sistem Rotary Transmisi
Rule 6	If starter elektrik bersuara keras : And suara kasar pada dynamo starter And saat dihidupkan terdapat bunyi selip pada starter elektrik Then kerusakan pada starter elektrik
Rule 7	If dihidupkan dengan menggunakan starter elektrik tidak ada bunyi : And lampu sein tidak terang And klakson tidak bunyi And lampu tidak menyala terang And suara starter lemah Then kerusakan pada Aki motor
Rule 8	If saat dijalankan motor oleng/tidak stabil : And ban motor tekanan angin kurang And ban motor bocor And ban motor gundul Then kerusakan pada ban sepeda motor

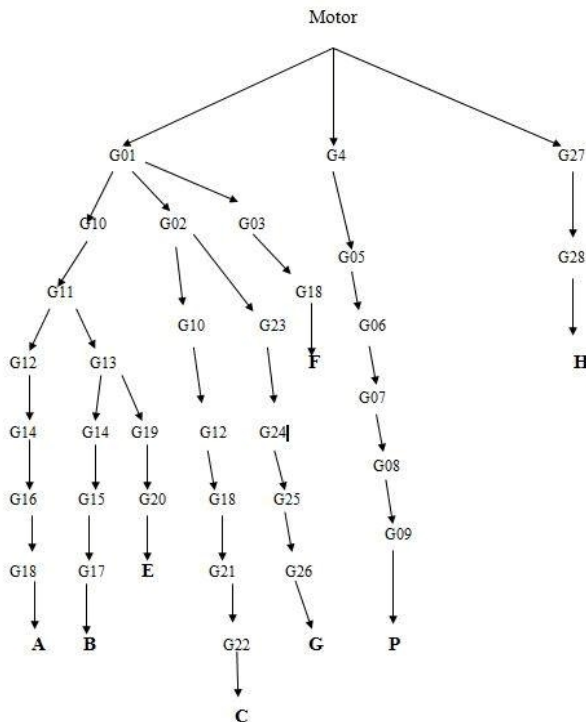
Tabel 2. Tabel Keputusan

2	Kode	Gejala kerusakan	A	B	C	D	E	F	G	H	I
3	G01	motor susah dihidupkan baik dengan starter atau manual	*	*	*			*			
4	G02	saat dihidupkan menggunakan starter elektrik tidak ada bunyi			*				*		
5	G03	saat dihidupkan menggunakan starter elektrik terdapat bunyi selip tidak mau berputar						*			
6	G04	saat dihidupkan lampu latar belakang penunjuk kecepatan mati				*					
7	G05	lampu transmisi persneling mati				*					
8	G06	jarum penunjuk kecepatan mati/tidak normal				*					
9	G07	jarum penunjuk bensin mati/tidak normal				*					
10	G08	odometer tidak jalan				*					
11	G09	suara klakson lemah/mati				*					
12	G10	tenaga motor menjadi lemah	*	*	*						
13	G11	tarikan/akselerasi menjadi lemah	*	*	*		*				
14	G12	mesin cepat panas	*	*	*						
15	G13	mesin tersendat-sendat saat jalan	*	*	*		*				
16	G14	mesin sering macet	*	*	*						
17	G15	mesin mati saat terkena air	*	*	*						

18	G16	keluar asap putih pada knalpot	*								
19	G17	suara ledakan saat digunakan		*	*						
20	G18	suara mesin kasar	*	*	*			*	*		
21	G19	susah/keras saat mengganti transmisi						*			
22	G20	transmisi top gear (puncak) tidak bisa kembali ke transmisi awal						*			
23	G21	bahan bakar boros			*						
24	G22	oli cepat habis			*						
25	G23	lampu sein menyala redup						*	*		
26	G24	lampu utama tidak menyala terang						*	*		
27	G25	lampu rem mati						*	*		
28	G26	lampu kota belakang mati						*	*		
29	G27	motor susah dikendalikan/oleng							*	*	
30	G28	ban sering selip saat melewati jalanan basah							*	*	
31	G29								*	*	
32	G30								*	*	

Pohon Keputusan

Pohon keputusan adalah salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasi oleh manusia. Pohon keputusan adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. Konsep dari pohon keputusan adalah mengubah data menjadi pohon keputusan dan aturan-aturan keputusan. Manfaat utama dari penggunaan pohon keputusan adalah kemampuannya untuk mem-break down proses pengambilan keputusan yang kompleks menjadi lebih simpel sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan. Pohon Keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan.[4]



Gambar 1. Tampilan pohon keputusan

Desain dan Implementasi

Implementasi adalah pelaksanaan sebuah aplikasi ini akan menampilkan implementasi rancangan antarmuka. Implementasi dirancang menjadi satu tampilan, yaitu rancangan antarmuka user. Implementasi antarmuka user terdiri atas beberapa menu pilihan antara lain menu pakar, menu utama atau home, dan menu materi-materi yang berhubungan dengan bebatuan. Sedangkan pada implementasi rancangan antarmuka admin, terdiri dari login admin, menu utama admin, menu edit dan hapus.

Tahapan perancangan aplikasi telah diselesaikan, mulai dari perancangan sistem rancangan input-output, rancangan database, dan juga rancangan antar muka pengguna (user interface). Semua rancangan digunakan untuk mempermudah dalam penjabaran sistem kedalam bahasa pemrograman, atau aplikasi yang akan dibuat. Desain form user pada saat membuka aplikasi sistem pakar tersebut dengan tampilan yang menarik dan mudah dipahami oleh user.



Gambar 2. Halaman depan aplikasi

Tampilan diagnosa dimana user mengisi form tersebut agar diketahui kerusakan pada motor.

Gambar 3. Halaman diagnosis kerusakan motor

Gambar 4. Halaman diagnosis akhir kerusakan motor

Gambar 5. Halaman hasil akhir

Metode pengujian yang digunakan adalah dengan menggunakan metode *black-box*, *Black-Box*. Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program.

Tabel 3. Tabel pengujian blackbox pada aplikasi

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Test Form Menu utama	Menampilkan layout menu utama setelah user membuka aplikasi	Pengguna dapat masuk pada layout menu utama setelah membuka aplikasi	Layout utama dapat tampil setelah user membuka aplikasi	Valid
2	Test Form diagnosa	Menampilkan form diagnose gejala kerusakan pada motor bebek	Pengguna dapat masuk ke form diagnose gejala kerusakan pada motor bebek	Form diagnose gejala kerusakan pada motor bebek dapat ditampilkan dan user dapat memilih gejala yang terdapat pada motor bebeknya.	Valid
3	Test Form peringatan	Menampilkan form peringatan akibat penggunaan tidak mengisi form diagnosa	Pengguna dapat menampilkan form peringatan setelah tidak mengisi form diagnosa	Form peringatan dapat ditampilkan oleh pengguna setelah pengguna tidak mengisi form diagnosa	Valid
4	Test Form Hasil Diagnosa	Menampilkan form hasil diagnosa setelah user mengisi form diagnosa	Pengguna dapat menampilkan form hasil diagnose setelah mengisi form diagnosa	Form hasil diagnose dapat ditampilkan setelah user mengisi form diagnosa	Valid

Tabel 3 di atas menjelaskan hasil pengujian yang telah dilakukan pada aplikasi ini, dan hasil dari pengujian tersebut telah sesuai dengan pengujian yang dilakukan.[5]

3. Kesimpulan

Pembuatan aplikasi sistem pakar dengan *backward chaining* untuk mendeteksi kerusakan sepeda motor menggunakan bahasa pemrograman java. Aplikasi ini dapat membantu pengguna motor yang masih awam dengan masalah teknis tanpa harus menemui sang mekanik. Hasil akhir akan didapat setelah memilih gejala kerusakan motor yang dialami dengan cepat, murah, dan mandiri. Semua pasti ada kekurangan dan kelebihan termasuk aplikasi ini. Oleh karena itu, kami mengharapkan saran dari pembaca demi kesempurnaan dan kelengkapan aplikasi ini. Semoga aplikasi ini dapat memberikan banyak bermanfaat untuk masyarakat pengguna sepeda motor

Daftar Pustaka

- [1] Suparman, Mengenal Artificial Intelligence, Andi Offset, Yogyakarta, 1991.
- [2] Kusriani, Sistem Pakar Teori dan Aplikasi, Yogyakarta: Andi 2006.
- [3] Kusumadewi, Sri. Artificial Intelligence (Teknik Dan Aplikasinya) Yogyakarta: Graha Ilmu. 2003
- [4] <http://dua7an.blogspot.co.id/2013/12/tentang-pohon-keputusan-decision-tree.html>

- [5] <http://www.dasarpendidikan.com/2013/06/black-box-testing-dan-contoh-pengujian.html>

Biodata Penulis

Angga Sulchan Saputra, sedang menempuh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta.

Bayu Kuncoro Jati, sedang menempuh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta.

Sundoro Fajar Utomo, sedang menempuh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta.