

Rancang Bangun Aplikasi Sistem Pakar Berbasis Android Untuk Mendeteksi Kerusakan Motor Vespa Matic

Saddam Husein Dio Darmawan^{1*}, Rinabi Tanamal²

^{1,2}Program Studi Teknik Informatika, Universitas Ciputra Surabaya, Jawa Timur
Email: ^{1*}Dioshus1996@gmail.com, ²r.tanamal@ciputra.ac.id

(Naskah masuk: 1 Feb 2021, direvisi: 21 Jun 2021, 12 Jul 2021, diterima: 15 Jul 2021)

Abstrak

Vespa Piaggio adalah salah satu motor yang sangat diminati oleh semua kalangan, salah satunya kalangan muda. Kendala pada Vespa ini berfokus hanya pada penjualan unit baru, sedangkan bengkel *service* Vespa ini tidak banyak tersedia. Dikarenakan sulitnya mencari bengkel *service*, pengguna motor Vespa *matic* akan kesulitan dan mungkin mengabaikan perawatan motor ini. Untuk mengantisipasi hal ini dibuatlah sistem pakar identifikasi kerusakan motor Vespa *matic* berbasis Android yang akan membantu para pengguna motor Vespa *matic* untuk mengatasi hal tersebut. Aplikasi sistem pakar ini dibuat berdasarkan fakta dan aturan yang bersumber dari hasil wawancara dan observasi penulis kepada pakar ahli. Lalu diimplementasikan ke dalam metode *forward chaining* yang dimodifikasi agar keputusan yang diambil sesuai dan tepat. *Expert system* ini mengaplikasikan McGoo dan Thinkable sebagai pembuat aplikasi sistem pakar. Hasil dari penelitian ini adalah aplikasi Android untuk mendeteksi kerusakan Vespa *matic* yang dapat menjadi pedoman penggantian *part* secara tepat sehingga tidak memerlukan banyak biaya dan waktu. Dari tingkat kepuasan pengguna terhadap aplikasi adalah sebagai berikut: aplikasi mudah digunakan memiliki nilai 90,6%, hasil solusi kerusakan aplikasi dapat dipahami mempunyai nilai 97,3%, diagnosis yang disampaikan aplikasi sesuai dengan hasil kerusakan Vespa *matic* memiliki nilai 90,6%, tampilan dan tombol-tombol pada aplikasi memberikan respon yang sesuai dan akurat memiliki nilai 84%, aplikasi benar-benar dapat mendeteksi kerusakan Vespa *matic* memiliki nilai 90,6%.

Kata Kunci: Metode *Forward Chaining*, Vespa *Matic*, Mcgoo, Sistem Pakar, Android

Design and Development of Android Application to Detect Vespa Automatic Motorcycle Damages

Abstract

Vespa Piaggio is one of the motorcycles that is in great demand by all circles, including young people. The problem with this Vespa is that it only focuses on selling new units, while there are not many Vespa service centres available. Due to the difficulty of finding a service repair shop, Vespa automatic motorcycle users will find it difficult and may neglect the maintenance of this motorbike. To anticipate this, an expert system for identification to the Vespa matic motor damage based on Android was created which will help Vespa matic motor users to overcome this. This expert system application is made based on facts and rules that are sourced from the results of interviews and observations by author to experts. Then it is implemented into a modified forward chaining method, so that the decisions taken are appropriate. This expert system applies McGoo and Thinkable as an expert system application maker. The result of this research is an Android application to detect Vespa matic damages which can guide in replacing parts appropriately, so that it does not require a lot of cost and time. The level of user satisfaction with the application is as follows: an easy-to-use application has a value of 90.6%, the results of the application damage solution can be understood to have a value of 97.3%, the diagnosis submitted by the application is in accordance with the results of the damage to the Vespa matic has a value of 90.6% the display and buttons on the application provide an appropriate and accurate response having a value of 84%, the application can actually detect damage to the Vespa matic having a value of 90.6%.

Keywords: *Forward Chaining Method, Vespa Matic, Mcgoo, Expert System, Android*

I. PENDAHULUAN

A. Latar Belakang

Penjualan motor di Indonesia memang di dominasi oleh pabrikan Jepang seperti Honda, Yamaha, Suzuki, dan Kawasaki.



Gambar 1. Data Penjualan Motor di Indonesia Tahun 2019
Sumber: Goozir.com

Dari data di Gambar 1, memang Piaggio tidak masuk pada data dikarenakan konsumen Piaggio ini pasarnya hanya untuk kalangan tertentu dan unit penjualannya pun sedikit dibandingkan merek lain. Bahkan *dealer* dan bengkel untuk *service* motor ini juga sedikit keberadaannya. Dilansir dari wawancara sebuah media viva.id kepada direktur utama PT. Piaggio Vespa Indonesia menyatakan bahwa *dealer* Vespa hanya fokus menangani produk terbaru saja [1].

Hal ini mengakibatkan PT. Piaggio Vespa Indonesia tidak memperhatikan *after sales*. Sebagian dampak jika *after sales* yang kurang dari pihak produsen, konsumen cenderung tidak memperhatikan perawatan dan *service*. Padahal kendaraan yang digunakan terus menerus haruslah diservis dan diganti sesuai dengan daya tahan komponen yang dipakai. Jika tidak, komponen akan mengalami kerusakan dan motor tidak bisa digunakan. *After sales* yang utama dari kendaraan bermotor adalah perawatan mesin. Jika pihak *dealer* atau bengkel *service* tidak banyak menyediakan layanan tersebut, tentu konsumen akan kebingungan untuk merawatnya, terlebih lagi jika dalam perjalanan jauh tiba-tiba motor mati dan tidak ada bengkel yang bisa menangani hal ini akan merepotkan pengguna.

Karena bengkel *service* yang terbatas keberadaannya dan pemahaman masyarakat tentang motor ini yang kurang, dibuatlah sistem pakar untuk mengidentifikasi kerusakan motor Vespa *matic* berbasis Android, agar masyarakat dapat mengetahui jenis kerusakan yang dialami, dan motor Vespa *matic* bisa digunakan kembali untuk kebutuhan rutinitas sehari-hari. Pada penelitian sebelumnya yang meneliti mengenai Yamaha, kerusakan mesin dapat dilacak namun terdapat perbedaan mesin yang digunakan sehingga *tree* dan masukannya juga berbeda [2].

Hasil dari sistem pakar ini diharapkan dapat membantu pengguna Vespa *matic* dapat memberikan kemudahan dalam hal perawatan, menghemat biaya *service* jika mengalami

kerusakan ringan (diganti sendiri atau bengkel non khusus Vespa), dan tidak terjadi kesalahan penggantian suku cadang karena kelalaian.

Sistem pakar aplikasi Android yang dikerjakan mengadopsi metode *forward chaining* yang telah dimodifikasi agar pemilihan keputusan sesuai dan tepat sasaran. Semua data yang diperoleh diambil berdasarkan dari observasi dan wawancara pada pakar ahli. Lalu diterapkan pada sistem di dalam aplikasi Android. Dengan demikian diharapkan solusi yang baik dapat memecahkan masalah ini dengan menggunakan teknologi [3].

II. LANDASAN TEORI

A. Jenis Kerusakan Motor Vespa Matic

Lima gejala motor Vespa *matic* ini adalah yang paling sering terjadi yaitu:

1. Mesin tidak mau menyala (mogok)
2. Mesin berisik
3. Mengeluarkan asap tebal
4. Tidak enak dikendarai
5. Kelistrikan dan lampu

B. Sistem Pakar

Sistem Pakar merupakan sistem informasi yang berisi pengetahuan seorang pakar sehingga dapat digunakan untuk konsultasi. Pengetahuan seorang pakar yang dimiliki oleh Sistem Pakar ini digunakan sebagai dasar untuk menjawab pertanyaan (konsultasi). Kepakaran diperoleh melalui rangkaian pelatihan, membaca atau menerima informasi, dan pengalaman.

Tujuan Sistem Pakar adalah mentransfer kepakaran seorang pakar ke komputer, kemudian melanjutkannya dari komputer ke orang lain (yang bukan pakar) [4].

C. Pohon Keputusan

Pohon keputusan adalah struktur *flowchart* yang mempunyai *tree* (pohon), dimana setiap simpul internal menandakan suatu tes atribut, setiap cabang merepresentasikan hasil tes, dan simpul daun merepresentasikan kelas atau distribusi kelas.

D. Forward Chaining

Algoritma *forward-chaining* adalah satu dari dua metode utama *reasoning* (pemikiran) ketika menggunakan *inference engine* (mesin pengambil keputusan). *Forward-chaining* mulai bekerja dengan data yang tersedia dan menggunakan aturan-aturan inferensi untuk mendapatkan data yang lain sampai sasaran atau kesimpulan didapatkan. Mesin inferensi yang menggunakan *forward-chaining* mencari aturan-aturan inferensi sampai menemukan satu dari *antecedent* (dalil hipotesa atau klausa IF - THEN) yang benar.

E. McGoo

McGoo atau biasa disebut *ES-Builder Web* adalah aplikasi untuk mengambil keputusan dari sejumlah data-data yang bisa didapat dari pakar atau pengetahuan. *Website software* ini

dapat diakses secara gratis. McGoo sendiri dapat di akses pada website secara *online*. Hampir semua permasalahan dapat dibantu oleh *ES-Builder* ini. *ES* ini sangat berguna sekali untuk membantu penyusunan dan gambaran keputusan yang diambil secara detail. Permasalahan pengambilan keputusan di sistem *ES-Builder* ini bisa diangkat dari permasalahan kita sehari-hari, tergantung apa yang akan anda putuskan untuk memecahkan sebuah masalah [5].

F. Thinkable

Thinkable adalah suatu aplikasi atau *tools IDE open source* seperti *App Inventor*. Saat ini, *Thinkable* yang satu-satunya tersedia dalam pembuatan aplikasi berbagai jenis *mobile* yakni *Android* dan *iOS* dalam keperluan *programmer* atau *developer mobile*. Namun, saya fokuskan ke *mobile Android*. *Tools* ini menggunakan *block programming*. Dengan kata lain, tidak perlu mengetik kode program secara manual untuk membuat aplikasi [6].

III. METODE PENELITIAN

A. Metode Penelitian

Pada penelitian ini terdapat 2 jenis metode untuk mengumpulkan data-data yang akan digunakan untuk merancang basis pengetahuan sistem pakar. Metode pertama dengan pengumpulan data primer yaitu melalui wawancara bersama pakar dan yang kedua data sekunder yaitu didapatkan dari sumber buku jurnal dan literatur penelitian sebelumnya.

1. Data Primer

Untuk memperoleh data ini menggunakan wawancara secara langsung pada orang yang ahli dibidangnya (pakar). Wawancara yang dilakukan melalui sesi tanya jawab dan mendapat data dan informasi yang lebih akurat terkait dengan penelitian. Pada pengumpulan data pakar ada 2 pakar ahli yang menjadi narasumber, yaitu:

1. Ayiek Assegaf yaitu sebagai *owner* bengkel *CRT Speed* di kota Surabaya.
2. M Zainal yang berprofesi sebagai montir di *IM Garage* Surabaya.

Dua narasumber berikut adalah orang yang ahli dan berkompeten pada perbaikan motor *Vespa matic*.

2. Data Sekunder

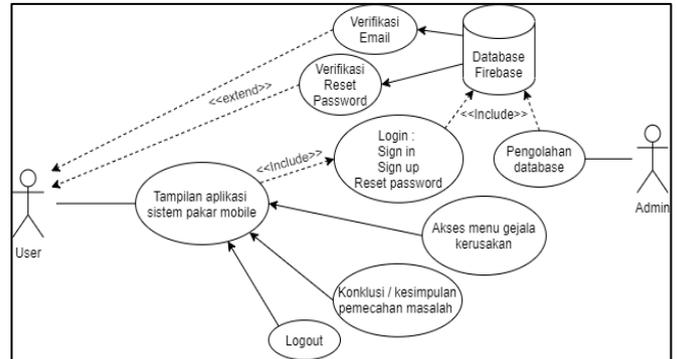
Pengetahuan ini diperoleh dengan mencari data yang mempunyai keterkaitan dengan masalah yang diangkat oleh penulis. Lalu data digunakan untuk dasar dari penulisan penelitian. Cara mendapatkan data tersebut dengan melalui jurnal, penelitian sebelumnya dan pencarian pada situs-situs di *internet*.

B. Desain Sistem Aplikasi

Penggunaan desain sistem aplikasi berguna untuk mengetahui gambaran bentuk sistem yang di kerjakan serta alur dari *Knowledge Base* aplikasi ini seperti apa.

1. Desain Aplikasi Use Case Diagram

Use Case Diagram merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor [7]. Dengan memakai *Use Case Diagram* pada Gambar 2 mempermudah menggambarkan alur kerja sistem dari aplikasi sistem pakar ini.

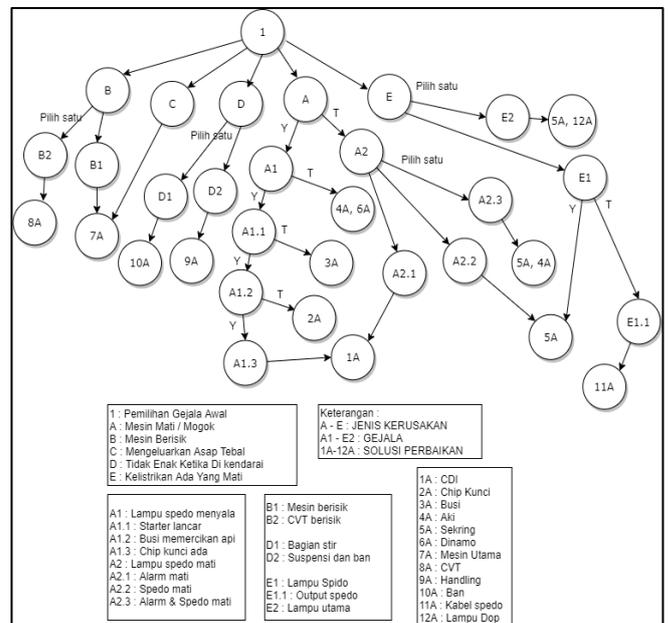


Gambar 2. Use Case Diagram Detail Sistem Aplikasi

User sebagai pengguna atau pemakai aplikasi sistem pakar dengan perangkat *mobile*. *User* dapat memilih gejala langsung dengan perangkat *mobile* dan akan langsung memberikan konklusi atau pemecahan masalah terkait gejala yang dialami oleh *user*. Sedangkan *admin* bertugas untuk mengolah *database* dari *user* yang nantinya bisa digunakan untuk keperluan pengembangan sistem berikutnya. Berikut fitur-fitur yang ada pada aplikasi.

2. Desain Pohon Keputusan/Decision Tree

Pohon keputusan pada Gambar 3 adalah alat untuk menyusun sistem pakar. Dimana *knowledge base* yang sudah diperoleh dari pakar ahli akan digambarkan alurnya sesuai dengan jenis opsi gejala kerusakan lalu dicarikan solusi perbaikannya.



Gambar 3. Desain Sistem Decision Tree/Pohon Keputusan

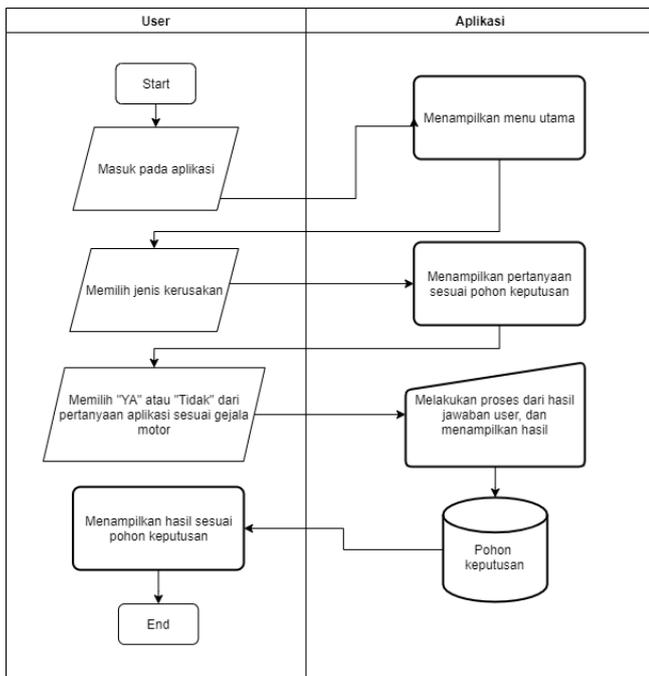
Tahapan awal yang dilakukan untuk membuat *tree* ini adalah dengan mewawancarai dua pakar teknisi Vespa untuk menentukan pertanyaan awal yang harus dibuat untuk menyusun *tree*. Pertanyaan utama yang telah disusun oleh peneliti untuk membuat pohon keputusan yaitu:

1. Mesin tidak mau menyala (Mogok)-A?
2. Mesin berisik-B?
3. Mengeluarkan asap tebal-C?
4. Tidak enak dikendarai-D?
5. Kelistrikan lampu dan panel spidometer-E?

Berdasarkan dari lima pilihan jenis kerusakan ini maka didapatkan jenis-jenis gejala, yang nantinya akan menentukan jenis kerusakan dan solusi perbaikan.

Pada diagnosis kerusakan motor Vespa *matic*, dijabarkan setiap detail alur permasalahan dari kerusakan motor Vespa *matic*. Dengan menu utama awal (1) lalu *user* memilih kerusakan yang terjadi pada opsi (A, B, C, D, E) dan selanjutnya aplikasi akan menanyakan gejala yang terjadi pada motor. Contoh *user* memilih B-mesin berisik, kemudian dari dua opsi dipilih B2-CVT Berisik, sesuai dengan pemilihan maka ada tanda kerusakan yaitu 8A-Kerusakan CVT. Dan akhirnya aplikasi akan menampilkan solusi: biasanya suara berisik CVT bisa diperbaiki oleh semua bengkel, hal yang paling mudah adalah mengganti *seatbelt* CVT dan ganti oli CVT terlebih dahulu.

C. Desain Arsitektur



Gambar 4. Sistem Arsitektur

Pada Gambar 4 desain cara kerja aplikasi digambarkan melalui *Use Case Diagram* untuk mempermudah mengidentifikasi gejala kerusakan Vespa *matic*, di bawah ini:

1. *User* mencoba mengoperasikan aplikasi sistem pakar lalu mencoba memasukkan jenis gejala dan kerusakan yang dialami kedalam aplikasi tersebut.

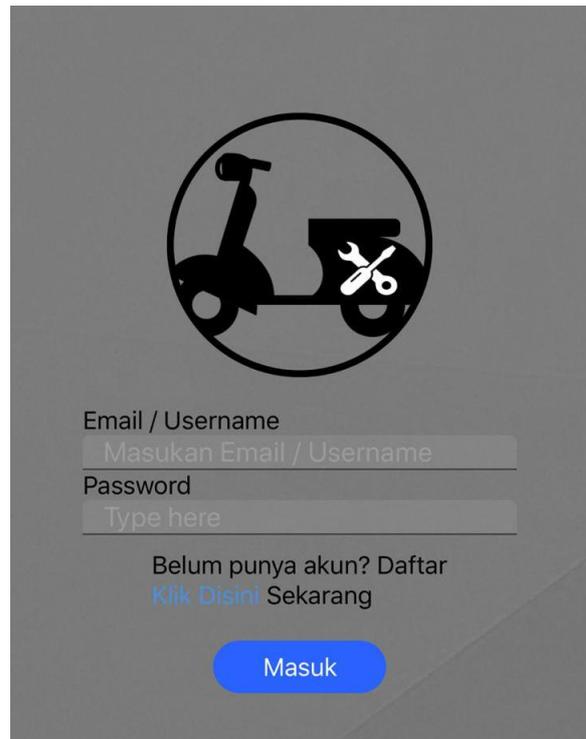
2. Aplikasi sistem pakar pada menu utama memaparkan pertanyaan-pertanyaan jenis gejala dan kerusakan yang dialami oleh pengguna. Pertanyaan yang dipaparkan telah ditentukan berdasarkan pohon keputusan yang sudah dibuat, kemudian *user* memilih berdasarkan gejala kerusakan yang ada. Setiap pertanyaan yang dijawab menentukan solusi dan hasil akhir dari penanganan dari setiap jenis kerusakan motor yang di alami.
3. Pertanyaan yang dijawab oleh *user* akan dicarikan solusi akhir dari masalah kerusakan serta cara menangani motor tersebut.

D. Desain User Interface

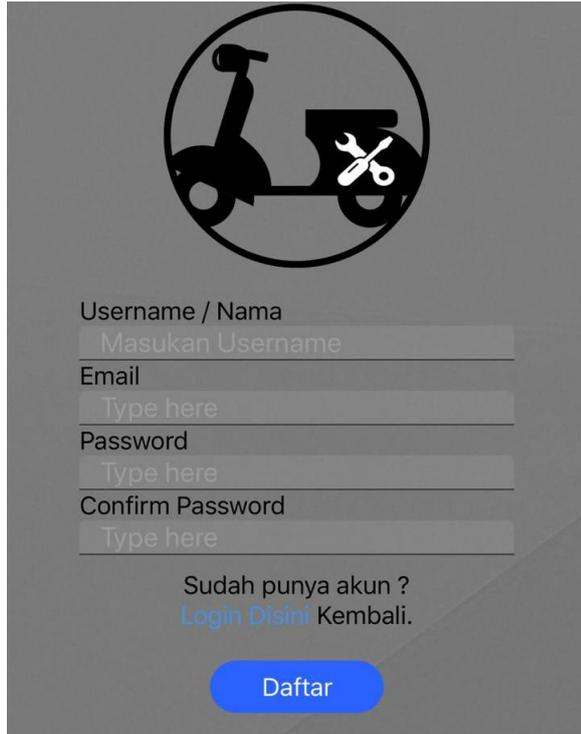
Aplikasi Kerusakan Vespa *matic* akan dipresentasikan dalam beberapa detail setiap halaman, yaitu terbagi jadi 4 bagian halaman.

1. Halaman *Login*

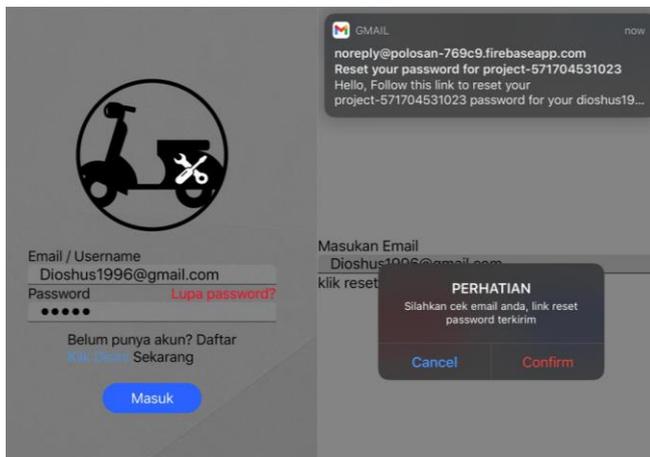
Halaman *Login* aplikasi memuat fitur antara lain *Sign In*, *Sign Up*, dan *Reset Password* yang akan ditampilkan seperti pada Gambar 5, 6, dan 7 di bawah ini. Saat memulai menggunakan aplikasi akan di hadapkan halaman *login* untuk memulai. Fitur ini berguna untuk pengembang aplikasi untuk membantu mengirim seperti notifikasi *email* untuk penggantian oli dan *parts*, atau mengembangkan fitur lain kelangkah selanjutnya jika diperlukan.



Gambar 5. Halaman *Login*



Gambar 6. Halaman Sign In



Gambar 7. Halaman Reset Password

2. Halaman Menu Utama

Pada Gambar 8, halaman ini untuk pemilihan gejala kerusakan. Pada halaman menu utama dibuat dengan *list* yang sudah di buat pada *McGoo ES-Builder*, dengan memilih salah satu gejala kerusakan motor Vespa dan selanjutnya akan diarahkan ke halaman selanjutnya untuk melakukan pengambilan keputusan, dan selanjutnya sesuai dengan data yang di dapat dari pakar ahli.

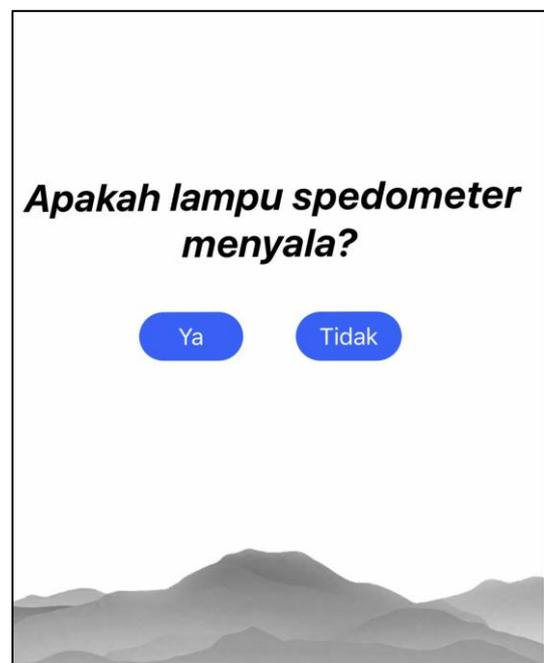
Pilih gejala kerusakan Vespa Matic anda



Gambar 8. Halaman Menu Utama

3. Halaman Penentuan Gejala

Pemilihan gejala tabel menggunakan “*Listview*” pada terlebih dahulu dan selanjutnya akan masuk pada halaman penentuan gejala seperti terlihat pada Gambar 9. Pada halaman ini untuk menentukan jenis gejala dan kerusakan menggunakan kata kunci “Ya”, dan “Tidak”. Dengan pertanyaan yang diberikan *user* akan memilih kedua opsi pilihan tersebut.



Gambar 9. Halaman Penentuan Gejala

4. Halaman Kerusakan

Di Gambar 10, informasi kerusakan dan hasil solusi yang terjadi dari gejala-gejala yang dipilih. Hasil tersebut disimpulkan dan ditampilkan di halaman keimpulan ini.



Gambar 10. Halaman Hasil Kerusakan

IV. PEMBAHASAN

Aplikasi kerusakan Vespa *matic* ini dibuat agar dapat berjalan secara akurat dan sesuai seperti apa yang di harapkan oleh pakar. Dengan melakukan uji aplikasi yang di proses dan diuji keakuratannya bersama pakar ahli yang berkompeten di bidang ini. Berdasarkan uji *User Acceptance Testing (UAT)* yang dilakukan melalui pertanyaan yang diberikan pada responden, penulis menyimpulkan bahwa pengguna atau *user* sangat setuju dengan adanya aplikasi yang mempermudah mengidentifikasi kerusakan Vespa *matic* ini.

A. Uji Akurasi

Uji akurasi ini bertujuan untuk menguji kesesuaian dan keakuratan metode *forward chaining* atau pohon keputusan dengan diagnose oleh pakar. Pada Tabel 1 dijelaskan komparasi kedua pengujian ini sesuai atau tidak. Pengujian berdasarkan dengan aplikasi kerusakan motor Vespa *matic* dengan bantuan data-data sistem yang diperoleh. Berikut tabel uji keakuratan didalam Tabel 1 di bawah ini.

Tabel 1. Hasil Uji Akurasi Aplikasi

No	Jenis Gejala Yang Di Alami Motor	Hasil Pengujian Aplikasi	Hasil Pengujian Pakar	Hasil Sesuai Atau Tidak Sesuai
1.	Lampu <i>speedometer</i> menyala, motor dapat di- <i>starter</i>	Kerusakan CDI	Kerusakan CDI	Sesuai

namun tidak merespon untuk menyala, busi memercikan api, *chip* kontak ada.

2.	Lampu <i>speedometer</i> menyala, motor dapat di- <i>starter</i> namun tidak merespon untuk menyala, busi memercikan api, <i>chip</i> kontak tidak ada	Masalah <i>Chip</i> Kunci	Masalah <i>Chip</i> Kunci	Sesuai
----	--	---------------------------	---------------------------	--------

3.	Lampu <i>speedometer</i> menyala, motor dapat di- <i>starter</i> namun tidak merespon untuk menyala, busi tidak memercikan api.	Kerusakan Busi	Kerusakan Busi	Sesuai
----	---	----------------	----------------	--------

4.	Lampu <i>speedometer</i> menyala, <i>starter</i> motor tersendat-sendat / tidak kuat.	Kerusakan Dinamo <i>starter</i>	Kerusakan Dinamo <i>starter</i>	Sesuai
----	---	---------------------------------	---------------------------------	--------

5.	Lampu <i>speedometer</i> tidak menyala, lampu alarm mati	Kerusakan CDI	Kerusakan CDI	Sesuai
----	--	---------------	---------------	--------

6.	Lampu <i>speedometer</i> tidak menyala, lampu di <i>speedometer</i> mati	Kerusakan sekring	Kerusakan sekring	Sesuai
----	--	-------------------	-------------------	--------

7.	Lampu <i>speedometer</i> tidak menyala, lampu alarm & lampu <i>speedometer</i> mati.	Kerusakan sekring dan aki	Kerusakan sekring dan aki	Sesuai
----	--	---------------------------	---------------------------	--------

8.	Mesin berisik, mesin utama	Masalah mesin	Masalah mesin	Sesuai
9.	Mesin berisik, bagian CVT	Masalah CVT	Masalah CVT	Sesuai
10.	Mengeluarkan asap tebal	Masalah mesin	Masalah mesin	Sesuai
11.	Tidak enak di kendarai, bagian kemudi dan stir	Masalah <i>handling</i>	Masalah <i>handling</i>	Sesuai
12.	Tidak enak di kendarai, bagain suspensi dan ban	Masalah suspensi dan ban	Masalah suspensi dan ban	Sesuai
13.	Kelistrikan & lampu, indikator <i>speedometer</i> , lampu <i>speedometer</i> mati	Masalah sekring dan aki	Masalah sekring dan aki	Sesuai
14.	Kelistrikan & lampu, indikator <i>speedometer</i> , lampu <i>speedometer</i> hidup, jarum indikator <i>speedo</i> mati	Kabel <i>speedo</i>	Kabel <i>speedo</i>	Sesuai
15.	Kelistrikan & lampu, lampu utama, sein, klakson	Masalah lampu bohlam	Masalah lampu bohlam	Sesuai

Pada hasil pengujian akurasi komparasi antara hasil pengujian aplikasi dan pengujian oleh pakar dan hasil perbandingan semua uji adalah semuanya sesuai.

B. Hasil Uji Coba Penggunaan Aplikasi

Pada pengujian ini penulis mengajukan 5 pertanyaan untuk diajukan pada *user*. Saat sedang mengoperasikan aplikasi. Pertanyaan ini berfungsi untuk melihat tingkat kepuasan pengguna dalam mengoperasikan aplikasi kerusakan Vespa *matic*.

Jumlah *user* atau pengguna adalah 15 responden, yang telah mengoperasikan aplikasi. Total jumlah skor penilaian dari 15 orang dapat dilihat pada Tabel 2 di bawah ini. Cara menilai hasil responden memanfaatkan metode Skala *Likert* yang biasa menggunakan skala dari nilai terkecil 1 sampai dengan

terbesar 5. *Perfect score* adalah bila semua mengisi 5 dan total *score* adalah 75.

Tabel 2. Total Skor Hasil Kuesioner

No.	Pertanyaan	Penilaian					Jml skor
		1	2	3	4	5	
1	Apakah aplikasi mudah digunakan?				7	8	68
2	Apakah hasil solusi kerusakan aplikasi dapat dipahami?				2	13	73
3	Apakah diagnosis yang disampaikan aplikasi sesuai dengan hasil kerusakan Vespa <i>matic</i> ?			1	4	10	68
4	Apakah tampilan dan tombol-tombol pada aplikasi memberikan respon yang sesuai dan akurat?				3	12	63
5	Apakah aplikasi benar-benar dapat mendeteksi kerusakan Vespa <i>matic</i> ?			1	5	9	68

Berdasarkan perhitungan total skor dari hasil kuisoner peneliti akan melanjutkan pada tahap perhitungan dengan metode Skala *Likert* agar gambaran hasil lebih jelas dan mudah dimengerti pada Tabel 3. Skala *Likert* adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei [8].

Tabel 3. Hasil Penilaian Skala *Likert*.

No	Pertanyaan	Nilai Persentase	Ket
1.	Apakah aplikasi mudah digunakan?	90,6%	Sangat Setuju
2	Apakah hasil solusi kerusakan aplikasi dapat dipahami?	97,3%	Sangat Setuju
3	Apakah diagnosis yang disampaikan aplikasi sesuai	90,6%	Sangat Setuju

dengan hasil kerusakan Vespa <i>matic</i> ?			
4	Apakah tampilan dan tombol-tombol pada aplikasi memberikan respon yang sesuai dan akurat?	84%	Sangat Setuju
5	Apakah aplikasi benar-benar dapat mendeteksi kerusakan Vespa <i>matic</i> ?	90,6%	Sangat Setuju

C. Pembahasan

Aplikasi kerusakan Vespa *matic* ini dibuat agar dapat berjalan secara akurat dan sesuai seperti apa yang diharapkan oleh pakar. Dengan melakukan uji aplikasi yang di proses dan diuji keakuratannya bersama pakar ahli yang berkompeten dibidang ini.

Berdasarkan uji *User Acceptance Testing* (UAT) yang dilakukan melalui pertanyaan yang diberikan pada responden, penulis menyimpulkan bahwa *user* atau pengguna sangat setuju dengan adanya aplikasi yang mempermudah mengidentifikasi kerusakan Vespa *matic* ini.

V. KESIMPULAN & SARAN

A. Kesimpulan

Dari semua proses yang sudah dilalui dari awal hingga akhir, penulis membuat kesimpulan dari aplikasi sistem pakar yang telah dikerjakan. Pembuatan aplikasi sistem pakar berbasis *mobile* dapat dibuat dengan pengetahuan data-data pakar yang diolah kembali dengan metode *forward chaining* melalui pohon keputusan dan dirancang serta diimplementasikan menggunakan aplikasi *website builder* Thinkable.

Hasil observasi dan wawancara bersama pakar diolah menjadi syarat dari penentu masalah dan solusi yang kemudian diimplementasikan menjadi pohon keputusan pada *Expert System Builder* atau McGoo, untuk memudahkan merancang aplikasi sistem pakar yang akan dibuat.

Dari aplikasi sistem pakar kerusakan Vespa *matic*, didapatkan bahwa solusi untuk masalah kerusakan Vespa *matic* bisa dengan data-data pengetahuan yang diperoleh dari pakar ahli. Hasil analisa yang diukur menggunakan Skala *Likert* dapat disimpulkan bahwa tingkat kepuasan pengguna terhadap aplikasi adalah sebagai berikut: Apakah aplikasi mudah digunakan memiliki nilai 90,6%, Apakah hasil solusi

kerusakan aplikasi dapat dipahami mempunyai nilai 97,3%, Apakah diagnosis yang disampaikan aplikasi sesuai dengan hasil kerusakan Vespa *matic* memiliki nilai 90,6%, Apakah tampilan dan tombol-tombol pada aplikasi memberikan respon yang sesuai dan akurat memiliki nilai 84%, Apakah aplikasi benar-benar dapat mendeteksi kerusakan Vespa *matic* memiliki nilai 90,6%.

B. Saran

Dari hasil yang didapatkan bahwa sistem pakar tidak hanya berguna untuk mendeteksi diagnosis kerusakan motor Vespa *matic* pada bidang segmentasi otomotif saja. Banyak bidang lain yang bisa digunakan untuk melakukan diagnosis.

REFERENSI

- [1] Rendara, S. & Pius, Y. (2018). *Vespa Matik Susah Dijual Lagi, Ini Komentar Piaggio*. Diakses dari <https://www.viva.co.id/otomotif/motor/1026458-Vespa-matik-susah-dijual-lagi-ini-komentar-piaggio> pada tanggal 3 Agustus 2020.
- [2] Hareka, W. A. & Tanamal, R. (2018). Rancang Bangun Aplikasi Sistem Pakar Berbasis Android Untuk Memprediksi Kerusakan Pada Mesin Sepeda Motor Yamaha R25. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, Vol. 2(2), pp. 200-207.
- [3] Tanamal, R., Nurdiansyah, Y. & Firdaus, F. (2019). Inventory Support System for Retail Shop. *International Conference on Electrical Systems, Technology and Information*. Vol. 188(1). DOI:10.1051/e3sconf/202018800020
- [4] Widiyanto, M.H. (2019). *Sistem Pakar Pada Era Industri 4.0*. Diakses dari <https://binus.ac.id/bandung/2019/11/sistem-pakar-pada-era-industri-4-0/> pada tanggal 16 Agustus 2020.
- [5] McGoo. (2016). *ES-Builder*. Diakses dari <https://www.mcgoo.com.au> pada tanggal 10 Agustus 2020.
- [6] Lestari, D.A., (2018). Pengenalan Dalam Pembuatan Aplikasi Android Dengan Thinkable. Diakses dari <https://www.scribd.com/document/395026366/Pengenalan-Dalam-Pembuatan-Aplikasi-Android-Dengan-Thinkable/> pada tanggal 21 Agustus 2020.
- [7] Sasmito, G.W. (2017). Penerapan Metode Waterfall Pada Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Pengembangan IT (JPIT)*. Vol. 2(1).
- [8] Taluke, D., Lakat, R.S.M. & Sembel, A. (2019). Analisis Prefensi Masyarakat Dalam Pengelolaan Ekosistem Mangrove di Pesisir Pantai Kecamatan Loloda Kabupaten Halmahera Barat. *Jurnal Spasial*, Vol. 6(2).