

BAB II

DASAR TEORI

2.1. Tinjauan Pustaka

Penelitian ini sebelumnya dilakukan oleh Dahlan Abdullah dan Khairul Azmi, (2016) dari Program Studi Teknik Informatika, Fakultas Teknik, Universitas Malikussaleh, dengan judul “Sistem Pakar Mendiagnosa Gejala Kerusakan Mesin Mobil Toyota Menggunakan Metode *Case Based Reasoning*”, dimana pada penelitian yang dilakukan penulis tersebut mengangkat masalah bagaimana mendiagnosa kerusakan kendaraan mobil dengan cara mengetahui jenis-jenis kerusakan sepeda motor, gejala kerusakan mobil, ciri-ciri kerusakan mesin mobil Toyota, penelitian ini menghasilkan sistem pakar untuk mendiagnosa kerusakan mobil Toyota menggunakan metode *Case Based Reasoning* [8].

Penelitian ini juga dilakukan oleh Asep Abdul Sofyan, Zainul Hakim, Muhammad Iqbal Dzulhaq, dan Ali Mursofi, (2015) dari STMIK Bina Sarana Global, dengan judul “Perancangan Aplikasi Sistem Pakar Deteksi Dini Kerusakan Mobil Toyota Avanza”, penelitian yang dilakukan penulis mengangkat masalah tentang sering kali melihat orang mengeluh, atau mengalami kebingungan jika mobil yang dikendarainya tiba-tiba mogok ditengah jalan, hal ini disebabkan karena pengendara kurang paham bahkan tidak tahu bagaimana cara melakukan tindakan perbaikan, penelitian ini menghasilkan sistem pakar untuk mendiagnosa kerusakan mobil menggunakan metode *Backward Chaining* [1].

Penelitian serupa dilakukan oleh Andri Saputra dan Hengki Andrea Taman (2016) dari STMIK PalComTech Palembang dengan judul “Sistem Pakar Kerusakan Mesin Jahit dengan Metode *Certainty Factor* Berbasis Android”, penelitian ini menggunakan metode *certainty factor* dengan metode tersebut pengguna dapat mengetahui tingkat kepastian kerusakannya. Aplikasi sistem pakar berbasis android ini dapat menjadi sarana untuk mengetahui secara cepat tentang kerusakan mesin jahit dan cara memperbaikinya tanpa harus datang langsung ke pakarnya [9].

Penelitian lainnya juga dilakukan oleh Bosker Sinaga, P.M Hasugian, dan Angelia M. Manurung (2018) dari STMIK Pelita

Nusantara Medan dengan judul “Sistem Pakar Mendiagnosa Kerusakan *Smartphone* Android Menggunakan Metode *Certainty Factor*”. Adapun metode yang digunakan pada sistem pakar ini adalah metode *Certainty Factor*. Pada sistem inii pengguna dapat melakukan konsultasi terhadap keluhan mengenai kerusakan pada *smartphone* [10].

Penelitian lainnya juga dilakukan oleh Dedi Suryadi, Rifki Meiliandaa, Ahmad Fauzan Suryonoa, dan Munadib (2018) dari Fakultas Teknik Universitas Diponegoro dengan judul “Sistem Pakar untuk Mengidentifikasi Kerusakan Mesin Industri Menggunakan Metode *Certainty Factor*”. Pada penelitian ini, data untuk validasi diambil dari data hasil pengujian getaran pada mesin-mesin di industri. Sinyal getaran inilah yang dianalisis jenis kerusakannya oleh pakar dan dirunning oleh aplikasi. Hasil validasi menunjukkan tingkat akurasi aplikasi mencapai 100% [11].

Berbeda dengan penelitian yang ada sebelumnya, penelitian ini bermaksud membuat Sistem Pakar Pendeteksi Kerusakan Mobil Xpander dimana sistem ini dapat mendeteksi kerusakan mobil dengan cara memilih gejala dan memberi nilai pada gejala. Kemudian akan langsung tertampil diagnosa kerusakan, penjelasan kerusakan, dan solusi untuk kerusakan tersebut. Jenis mobil untuk dilakukan penelitian juga tergolong jenis mobil yang baru keluar dari produk Mitsubishi. Terdapat fitur untuk mengetahui kerusakan yang sering terjadi oleh keseluruhan pengguna sistem, dan informasi yang lengkap mengenai mobil xpander. Pada pembuatan sistem ini menggunakan *framework* laravel.

2.2. Landasan Teori

2.2.1. Pengertian Sistem

Sistem didefinisikan menjadi “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untukmelakukan suatu kegiatan atau untuk menyelesaikan suatu sasara tertentu [12].

2.2.2. Pengertian Pakar

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkan untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu

menjelaskan dan mempelajari hal-hal yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya [3].

2.2.3. Pengertian Sistem Pakar

Sistem Pakar adalah adalah sistem yang sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah yang biasa dilakukan oleh para ahli. [13]

2.2.4. Metode *Certainty Factor*

Teori *Certainty Factor* adalah mengakomodasi ketidakpastian pemikiran dari seorang pakar yang di usulkan oleh *Shortlife* dan Buchanan pada tahun 1975. Seorang pakar (misalnya dokter) sering menganalisis informasi yang telah ada dengan ungkapan ketidakpastian, untuk mengakomodasi hal ini kita menggunakan *Certainty Factor* guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi. [9]

Rumus *Certainty Factor*, yaitu :

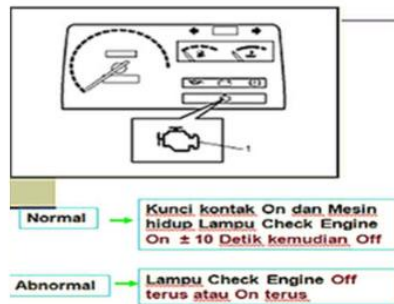
1. Perkalian CF(H,E) :

$$\mathbf{CF(H,E) = CF(E) \times CF(rule)} \quad (2.2)$$

2. Kombinasi CF :

$$\mathbf{CF(1.2) = CF(1) + [CF(2) * (1 - CF(1))]} \quad (2.3)$$

2.2.5. On Board Diagnostic (OBD)



Gambar 2. 1 On Board Diagnostic

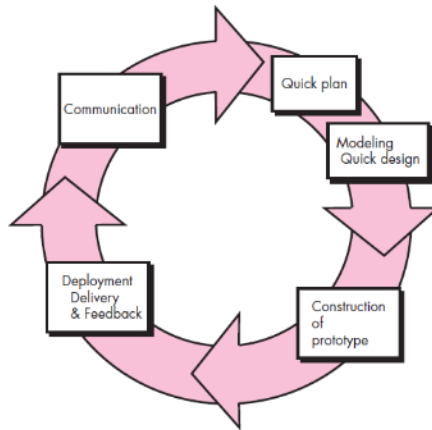
On Board Diagnostic atau OBD ditemukan oleh SAE (Society of Automotive Engineers) dengan melibatkan skema penyeragaman dengan protokol komunikasi komputer untuk kendaraan yang akan dibuat oleh pabrik pembuat mobil, penyeragaman prosedur pengujian, alat pengujian kerusakan, nama komponen, penyeragaman *diagnostic connector*, dan fungsi standarisasi dalam tes kerusakan kendaraan [2].

2.2.6. Rekayasa Perangkat Lunak (RPL)

Pengertian Rekaya Perangkat Lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, Desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa RPL tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan "semua aspek produksi" pada pengertian di atas, mempunyai arti semnua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL. [14]

1.2.5.1 Metode Pengembangan Sistem

Dalam proses pengembangan sistem, penulis menggunakan metode pengembangan sistem *prototype*. *Prototyping* adalah proses pembuatan model sederhana *software* yang mengijjinkan pengguna untuk memiliki gambaran dasar mengenai program serta melakukan pengujian awal. *Prototyping* menyajikan fasilitas bagi para pengembang maupun pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan bagaimana perangkat lunak yang akan dirancang [6].



Gambar 2. 2 Metode *Prototyping*

Metode ini memiliki beberapa tahapan yang dilakukan, yaitu [7]:

1) *Requirment Analysis*

Pada tahap ini pengembang maupun pihak pemakai bersama-sama melakukan identifikasi kebutuhan, mendefinisikan format sistem yang akan dirancang, serta membuat batasan - batasan sistem yang akan dibangun.

2) *Build Prototype*

Pada tahapan kedua ini dibuat prototype dari sebuah sistem yang akan dibangun, namun prototype ini hanya difokuskan untuk penyajian kepada pihak pemakai meliputi inputan dan bentuk laporan yang sesuai dengan keinginan dari pihak pemakai.

3) *Prototype Evaluation*

Tahapan ketiga ini dilakukan oleh pihak pemakai apakah prototype yang dibuat sudah sesuai dengan kebutuhannya atau tidak, jika sesuai kebutuhan maka akan diimplementasikan, namun jika tidak prototype direvisi dan dibuat kembali untuk ditunjukkan kembali kepada pihak pemakai.

4) *Implementation*

Setelah prototype sudah sesuai dengan kebutuhan pemakai dan sudah disepakati bersama, maka tahapan implementasi akan dilakukan yakni dengan membuat program yang sesungguhnya dengan menggunakan bahasa pemrograman yang sesuai.

5) *Testing*

Tahapan selanjutnya adalah setelah sistem yang dibangun menjadi kesatuan program yang utuh dan siap pakai, sebelum digunakan akan dilakukan proses pengujian. Sistem pengujian dilakukan pertama kali dengan pengujian white box, pengujian dilakukan oleh programmer atau sistem analis. Selanjutnya dilakukan pengujian dengan *black box*, pengujian dilakukan oleh user selain pembuat program. Pengujian ini dilakukan untuk mengetahui tingkat error dari program sebelum digunakan oleh pihak pemakai.

6) *System Evaluation*

Pada tahapan ini dilakukan oleh pihak pemakai dengan cara melakukan evaluasi apakah sistem yang didapatkan sudah sesuai dengan yang diharapkan atau belum. Jika sistem sudah sesuai, maka akan diimplementasikan.

7) *Use the System*

Tahap ini adalah tahap penggunaan sistem yang sudah diuji dan juga sudah dievaluasi terlebih dahulu.

1.2.5.2 Metode Pengujian Sistem

Pengujian *Blackbox* testing merupakan pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *blackbox* merupakan perancangan data uji yang didasarkan pada spesifikasi

perangkat lunak, data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah selesai dengan yang diharapkan [15].

Kategori kesalahan yang akan ditemukan dalam pengujian *blackbox*, antara lain yaitu:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

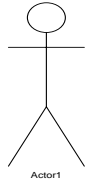

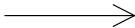

2.2.7. Pengertian UML

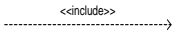
Unified Modeling Language (UML) merupakan salah satu alat bantu yang handal di dunia pengembangan sistem berorientasi objek. Hal ini disebabkan karna UML menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain [16]. UML memiliki banyak diagram, berikut penjelasan tentang diagram UML:

a. *Use case* diagram

adalah pemodelan untuk kelayakan (*behavior*) sistem informasi yang akan dibuat. *Use case* berfungsi untuk mengetahui fungsi apa saja yang terdapat di dalam sistem informasi yang dibuat dan siapa saja yang berhak menggunakan fungsi-fungsi pada sistem tersebut [17]. Berikut simbol-simbol yang sering digunakan dalam *Use Case Diagram* beserta fungsinya :

Tabel 2. 1 Simbol *Use Case Diagram*



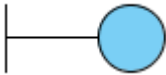



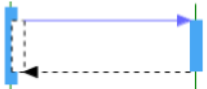
No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Orang proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
2.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan menggunakan kata kerja di awali di awal frase nama nama <i>use case</i> .
3.		<i>Interaction</i>	Digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya <i>interaction</i> dilengkapi dengan <i>teks</i> bernama <i>operation signature</i> yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.
4.		<i>Dependency</i>	Merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah


No	Simbol	Nama	Keterangan
			adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah.
5.		<i>Include</i>	Menunjukkan bahwa suatu bagian dari elemen memicu eksekusi bagian dari elemen lain (yang ada pada tanda panah).
6.		<i>Extend</i>	Menunjukkan bahwa suatu bagian dari elemen bisa disisipkan ke dalam elemen yang lain.
7.		<i>Association</i>	Menggambarkan navigasi antar <i>class</i> , berapa banyak obyek lain yang bisa berhubungan dengan satu obyek, dan apakah suatu <i>class</i> menjadi bagian dari <i>class</i> lainnya.
8.		<i>Generalization</i>	Menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan <i>generalization</i> , <i>class</i> yang lebih spesifik (<i>subclass</i>) akan menurunkan atribut dan operasi dari <i>class</i> yang lebih umum (<i>superclass</i>). Notasi ini digunakan pada konsep <i>inheritance</i> .

b. *Sequence Diagram*

Menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna, display dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram biasa digunakan untuk menggambarkan scenario atau rangkaian Langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu [16]. Berikut simbol-simbol yang sering digunakan dalam Sequence diagram beserta fungsinya.

Tabel 2. 2 Simbol *Sequence Diagram*





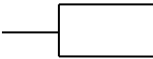
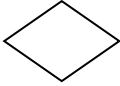
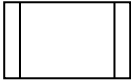
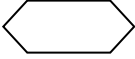
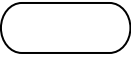
No	Simbol	Nama	Keterangan
1.		<i>Object Lifeline</i>	Menggambarkan <i>object</i> apa saja yang terlibat.
2.		<i>Actor</i>	Menggambarkan <i>actor</i> yang terlibat.
3.		<i>Lifeline Control</i>	Digunakan untuk menggambarkan sebuah <i>form</i> .
4.		<i>Lifeline Entity</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.
5.		<i>Lifeline Control</i>	Digunakan untuk menghubungkan <i>boundary</i> dengan tabel.
6.		<i>Time Active</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
7.		<i>Synchronous</i>	Relasi ini digunakan untuk memanggil operasi atau <i>method</i>

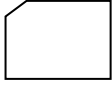

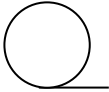

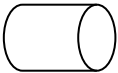

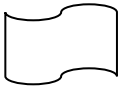

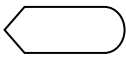
			yang dimiliki oleh suatu objek. <i>Synchronous</i> mengharuskan kita menyelesaikan 1 proses baru kemudian memanggil proses berikutnya.
8.		<i>Asynchronous</i>	Relasi ini digunakan untuk memanggil operasi atau <i>method</i> yang dimiliki oleh suatu objek. <i>Asynchronous</i> memberikan kita fasilitas untuk menjalankan proses lain ketika proses sebelumnya belum selesai.




c. *Flowcharts*

Flowchart merupakan penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah- langkah dan urutan prosedur dari suatu program. Bagan alir (flowchart) adalah bagan (chart) yang menunjukkan alir (flow) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi [18]. Berikut simbol-simbol yang ada di dalam flowcharts beserta fungsinya.

Tabel 2. 3 Simbol *Flowcharts*

No	Simbol	Nama	Keterangan
1.		<i>Input/Output</i>	Mempresentasikan <i>input</i> data atau <i>output</i> data yang diproses atau informasi.
2.		Proses	Mempresentasikan operasi
3.		Penghubung	Keluar ke atau masuk dari bagian lain, khususnya halaman yang sama.
4.		Anak Panah	Mempresentasikan alur kerja.
5.		Penjelasan	Digunakan untuk komentar tambahan.
6.		Keputusan	Keputusan dalam program.
7.		<i>Predefined Process</i>	Rincian operasi berada di tempat lain.
8.		<i>Preparation</i>	Pemberian harga awal.
9.		<i>Terminal Port</i>	Awal/akhir <i>Flowchart</i> .

No	Simbol	Nama	Keterangan
10.		<i>Punched Card</i>	<i>Input/Output</i> yang menggunakan kartu berlubang.
11.		Dokumen	<i>Input/Output</i> dalam format yang dicetak.
12.		<i>Magnetic Tape</i>	<i>Input/Output</i> yang menggunakan pita magnetic.
13.		<i>Magnetic Disk</i>	<i>Input/Output</i> yang menggunakan disk magnetic.
14.		<i>Magnetic Drum</i>	<i>Input/Output</i> yang menggunakan drum magnetic.
15.		<i>On-line Storage</i>	<i>Input/Output</i> yang menggunakan penyimpanan akses langsung.
16.		<i>Punched Tape</i>	<i>Input/Output</i> yang menggunakan pita kertas berlubang.
17.		<i>Manual Input</i>	<i>Input</i> yang dimasukkan secara manual dari keyboard.
18.		<i>Display</i>	<i>Output</i> yang ditampilkan pada terminal.

No	Simbol	Nama	Keterangan
19.		<i>Manual Operation</i>	Operasi manual.
20.		<i>Communication Link</i>	Transmisi data melalui <i>channel</i> komunikasi, seperti telepon
21.		<i>Off-line Storage</i>	Penyimpanan yang tidak dapat diakses oleh komputer secara langsung

2.2.8. Basis Data

Basis data terdiri atas 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya [19].

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya. Sedangkan SQL (Structured Query Language) adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis [12].

Database Management System (DBMS) adalah suatu sistem yang dapat mengorganisasikan *database* pada perangkat komputer. Dengan adanya DBMS selain lebih efisien juga dapat digunakan sebagai media membangun interface yang simpel untuk user. DBMS menggunakan suatu perintah atau intruksi yang berfungsi untuk mengelola *database* [20]. Perintah tersebut dibagi menjadi 2 bagian, yaitu :


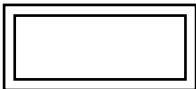
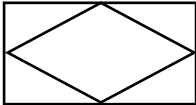
1. DDL (Data Definition Language) adalah suatu intruksi yang lebih ditunjukkan pada database secara keseluruhan, biasanya hanya dapat diakses oleh admin database.
2. DML (Data Manipulation Language) adalah intruksi yang lebih diberikan kepada isi datanya dan dapat diakses secara umum.

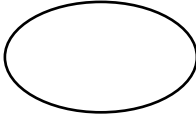
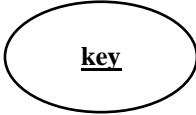

2.2.7.1 Entity Relationship Diagram (ERD)

ERD adalah kumpulan peralatan atau cara mendeskripsikan data atau objek yang nyata dibawa ke entitas (*entity*) dan hubungan (*relationship*) antar entitas menggunakan notasi [21].

Berikut daftar simbol *entity relationship diagram*.

Tabel 2. 4 Simbol *Entity Relationship Diagram*

No	Simbol	Nama	Keterangan
1		<i>Entity</i>	Suatu entitas yang memiliki nama benda, orang atau lokasi digambarkan dengan persegi panjang.
2		<i>Weak Entity</i>	Entitas yang tidak dapat diidentifikasi melalui atributnya sendiri. <i>weak entity</i> bergantung pada <i>entity</i> lain (<i>Owner entity</i>).
3		<i>Associative Entity</i>	Entitas yang digunakan pada banyak antar banyak hubungan.

4		<i>Attribute</i>	Mendeskripsikan karakter entitas
5		<i>Key attribute</i>	Atribut yang berfungsi sebagai kunci.
6		<i>Link</i>	Penghubung antar relasi dengan entitas dan entitas dengan atribut.

Elemen –elemen hubungan entitas:

1. *Entity*
Entity adalah sesuatu di dalam sistem, nyata maupun abstrak ada data tersimpan. Entitas bisa kata benda seperti orang, benda, lokasi, kejadian yang ada unsur waktu di dalamnya.
2. *Relationship*
Relationship adalah hubungan alami yang terjadi antar entitas, biasanya berbentuk kata kerja aktif atau pasif.
3. *Relationship Degree*
Derajat relasi adalah jumlah entitas yang bergabung dalam suatu hubungan.
4. *Atribut value*
Nilai atribut adalah suatu kejadian dari atribut di dalam satu entitas atau hubungan. Ada dua jenis atribut yaitu *identifier* unik (*primary key*) dan *Descriptor* tidak unik (*nonkey attribute*).

5. *Cardinality*

Kardinalitas adalah relasi yang menunjukkan jumlah relasi dengan entitas pada entitas yang lain.

- a. Satu ke satu (1-1)
Suatu entitas hanya dapat berelasi dengan satu entitas lain dan sebaliknya.
- b. Satu ke banyak (1-n)
Suatu entitas dapat berelasi dengan banyak entitas namun entitas pasangannya tidak dapat sebaliknya.
- c. Banyak ke banyak (m-n)

Suatu entitas dapat berelasi dengan banyak entitas dan sebaliknya.

2.2.9. Skala Likert

Pengumpulan data kuesioner dalam menanggapi pertanyaan penilaian sistem dihitung menggunakan pengukuran skala likert. Skala likert diambil dari penamaan Rensis Likert, skala likert merupakan metode pengukuran dari tanggapan negatif ke tanggapan positif atau sebaliknya biasanya tersedia dalam lima tingkatan nilai bahkan ada yang sembilan [22].

Langkah menghitung skala likert:

1. Tentukan total skor
Rumus: $T \times P_n$
Keterangan:
T = total responden
P_n = pilihan angka skor likert
2. Interpretasi skor perhitungan
 $Y = \text{skor tertinggi likert} \times \text{jumlah responden}$
3. Cari interval skor
 $I = 100 / \text{jumlah skor likert}$
4. Interpretasi skor berdasarkan interval
Dari 0 % + interval % sampai 100%
5. Penyelesaian akhir

$$\text{Hasil} = \frac{\text{Total skor}}{y} \times 100\% \quad (2. 1)$$

6. Kategori penilaian berdasarkan interval.

2.2.10. Gejala dan Kerusakan

Data gejala dan kerusakan pada sistem pakar yang dibuat dapat dilihat pada table 2.5

Tabel 2. 5 Tabel Gejala dan Kerusakan

No	Data Kerusakan	Pengertian	Data Gejala
1	Ecu Rusak	ECU biasa disebut 'otak' mobil berisi rangkaian elektronika yang bertugas memerintahkan sistem untuk bekerja. Bila bagian otak mobil ini mengalami kerusakan, bisa dipastikan kerja mobil tidak maksimal, bahkan hal paling buruk yaitu mesin mobil tidak bisa dinyalakan.	Cek engine menyala
			Kendaraan tidak bisa hidup
			Pengapian tidak menyala
			Bahan bakar tidak mengalir
2	Piston Rusak	Ring piston menjadi salah satu komponen dari mesin mobil yang krusial karena berada di jantung mesin. Masalah pada komponen ini bisa merembet pada piston yang mengakibatkan performa kendaraan menjadi menurun bahkan hingga mogok.	Suara mesin kasar, seperti benturan besi dengan besi
			Asap emisi putih pada saat jalan maupun berhenti
			Kuantitas oli berkurang
			Tenaga tidak ada
3	Seal Klep Rusak	Seal klep ini berada di bushing dudukan klep.	Saat pertama kali dihidupkan emisi

		<p>Fungsi seal klep sebagai penyekat antar mesin dengan klep yang berada di ruang bakar. Komponen seal klep terbuat dari karet khusus yang tahan dengan gesekan dan panas mesin. Akan tetapi, karena sering terpapar panas dan penggunaan yang cukup lama, seal klep bisa mengalami kerusakan.</p>	<p>putih seperti oli terbakar</p> <hr/> <p>Kuantitas diukur dari stik oli berkurang</p>
4	Sensor ABS Rusak	<p>„Sistem pengereman ini merupakan sebuah mekanisme yang akan mencegah rem mengunci roda sehingga tidak berputar. Kondisi roda yang terkunci oleh rem justru lebih berbahaya karena sistem pengereman bisa jadi tidak lagi berfungsi karena mobil tetap bergerak karena telah kehilangan grip atau daya cengkram dengan permukaan jalan.</p>	<p>Indikator ABS menyala terus</p> <hr/> <p>Saat di rem roda berhenti, tapi mobil masih jalan meninggalkan bekas</p> <hr/> <p>Stir tidak stabil</p>