

**SISTEM PAKAR MENDETEKSI KERUSAKAN
SEPEDA MOTOR SUZUKI NEX F1 BERBASIS
*ANDROID***

SKRIPSI



Oleh
Erlina Sinaga
140210283

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2019**

**SISTEM PAKAR MENDETEKSI KERUSAKAN
SEPEDA MOTOR SUZUKI NEX F1 BERBASIS
*ANDROID***

SKRIPSI
Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana



Oleh
Erlina Sinaga
140210283

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2019**

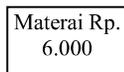
HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Putera Batam maupun di perguruan tinggi lain.
2. Skripsi ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam,

Yang membuat pernyataan,



Erlina Sinaga

140210283

**SISTEM PAKAR MENDETEKSI KERUSAKAN SEPEDA MOTOR
SUZUKI NEX F1 BERBASIS ANDROID**

Oleh:

Erlina Sinaga

140210283

SKRIPSI

**Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana**

Telah disetujui oleh Pembimbing pada tanggal

Seperti tertera di bawah ini

Batam, 09 Agustus 2018

Yulia, S.Kom.,M.Kom

Pembimbing

ABSTRAK

Kebanyakan pengendara cenderung untuk menyerahkan sepeda motornya kepada mekanik tanpa mengetahui apa sebenarnya yang terjadi dengan kendaraanya. Jika dikaitkan dengan tuntutan masa depan yang bukan hanya bersifat kompetitif tapi juga sangat terkait dengan berbagai kemajuan teknologi dan informasi maka sistem pembelajaran yang dikembangkan harus mampu secara cepat memperbaiki kekurangan sarana penunjang yang ada. Salah satu cara yang dapat dikembangkan adalah yaitu mengubah sistem informasi yang penuh dengan sistem pengetahuan yang lebih efektif dan efisien dengan dukungan sarana dan prasarana yang memadai. Jadi Aplikasi Sistem Pakar Berbasis Android untuk analisa kerusakan Sepeda motor Suzuki Nex F1 sebagai penunjang perbaikan dan pengetahuan pada Teknik Sepeda Motor dirancang dengan tujuan untuk member informasi dan diagnose mengenai kerusakan yang bisa terjadi dan dialami oleh pengguna sepeda motor, berdasarkan gejala-gejala dan pertanyaan-pertanyaan yang diajukan. Sistem ini merupakan sistem berbasis android menggunakan metode *Forward chaining* model representasi yang digunakan adalah dengan aplikasi starUML, dan dibangun dengan JAVA dan tool Eclipse, berdasarkan hasil pengujian dan percobaan yang telah menunjukkan bahwa sistem pakar dapat membantu pengguna sepeda motor untuk mendeteksi kerusakan sepeda motor Suzuki Nex F1.

Kata Kunci: Sitem pakar, deteksi kerusakan, Suzuki Nex F1, *forward chaining*, berbasis android

ABSTRACT

Most motorists tend to give up their motorbikes to mechanics without knowing what exactly happens with the vehicle. If it is associated with future demands that are not only competitive but also strongly related to various advances in technology and information, the learning system developed must be able to quickly improve the lack of existing supporting facilities. One way that can be developed is to change information systems that are full of knowledge systems that are more effective and efficient with the support of adequate facilities and infrastructure. Become an Android-based Expert System Application for analysis of damage Nex F1 Suzuki motorbikes as supporting improvements and knowledge in Motorcycle Engineering are designed with the aim to provide information and diagnosis regarding damage that can occur and be experienced by motorcycle users, based on symptoms and questions question asked. This system is an android-based system using the Forward chaining method of representation model used is the StarUML application, and built with JAVA and Eclipse tools, based on the results of tests and experiments that have shown that expert systems can help motorcycle users to detect damage to Suzuki Nex motorbikes F1.

Keywords: *Expert system, damage detection, Suzuki Nex F1, forward chaining, android based*

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa yang telah melimpahkan segala Rahmat dan KaruniaNYA, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu syarat untuk menyelesaikan program studi strata satu (S1) pada program studi Teknik Informatika Putera Batam. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu kritik dan saran akan senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini tidak akan selesai tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Tuhan Yang Maha Esa yang telah memberkati dan memberikan Hikmat kepada penulis dan mencukupkan segala kebutuhan penulis.
2. Rektor Universitas Putra Batam Ibu Nur Elfi Husda, S.Kom., M.SI.
3. Ketua Program Studi Teknik Informatika Bapak Andi Maslan S.T., M.SI.
4. Ibu Yulia, S.Kom., M.Kom, selaku pembimbing skripsi pada program Studi Teknik Informatika Universitas Putra Batam.
5. Seluruh Dosen dan Staff Universitas Putra Batam
6. Kedua orang tua penulis terutama Ibu yang selalu memberikan dukungan dan motivasi serta Doa nya untuk menyelesaikan skripsi ini.
7. Saudara dan Saudari penulis yang tetap memberikan dukungan penuh kepada penulis sehingga skripsi ini bisa selesai dan dengan hasil yang memuaskan.
8. Sahabat saya Debora Pestaria Dongoran yang selalu memberikan dukungan dalam menyelesaikan skripsi ini.
9. Sanri Sianturi yang memberikan motivasi dan dukungannya di dalam menyelesaikan skripsi ini.

10. Irwan Iskandar Sihaloho yang selalu memberikan dukungannya untuk menyelesaikan skripsi ini.
11. Epi Julianti sebagai teman yang memberikan dukungan selama proses penyelesaian skripsi ini.
12. Mori Herlina Sianipar yang juga memberikan dukungannya untuk menyelesaikan skripsi ini.
13. Alfredo Sinaga yang memberi dukungan untuk penyelesaian skripsi ini.
14. Riris Asianna Samosir yang sama-sama berjuang mengerjakan skripsi ini.
15. Seluruh teman-teman dari Unit Kegiatan Mahasiswa Kristen Universitas Putera Batam yang memberikan Doa dan Dukungannya untuk menyelesaikan skripsi ini.
16. Semua teman-teman teknik informatika angkatan tahun 2014 yang memberikan dukungan dalam menyelesaikan skripsi ini.

Semoga Tuhan Yang Maha Esa membalas kebaikan yang dan selalu mencurahkan berkatNYA atas kebaikan yang telah penulis terima.

Batam, 11 Februari 2019

Penulis

DAFTAR ISI

	Halaman
HALAMAN SAMPUL DEPAN	
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN.....	iii
HALAMAN PENGESAHAN.....	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN	
1.1. Latar Belakang Penelitian	1
1.2. Identifikasi Masalah	5
1.3. Pembatasan Masalah	5
1.4. Perumusan Masalah.....	6
1.5. Tujuan Penelitian.....	6
1.6. Manfaat Penelitian.....	7
BAB II TINJAUAN PUSTAKA	
2.1. Teori Dasar	10
2.1.1. Kecerdasan Buatan.....	10
2.1.2. <i>Fuzzy logic</i>	11
2.1.2. Pengertian Jaringan Syaraf Tiruan	12
2.1.3. Sistem Pakar (<i>Expert Sytem</i>)	14
2.1. Variabel	26
2.2. Software Pendukung	31
2.3.1. <i>UML (Unified Modelling Language)</i>	31
2.3.1. <i>Android</i>	37
2.3.4. Java.....	39
2.3.4. <i>Eclipse</i>	40
2.4. Penelitian Terdahulu	40
2.5. Kerangka Pemikiran.....	44
BAB III METODE PENELITIAN	
3.1. Desain Penelitian.....	46
3.2. Teknik Pengumpulan Data	49
3.3. Operasional Variabel.....	51
3.4. Perancangan Sistem.....	52
3.4.1. Desain Basis Pengetahuan.....	52

3.4.2. Struktur Kontrol (Mesin Inferensi)	60
3.4.3. Desain UML (<i>Unified Modelling Language</i>).....	60
3.4.4. Knowledge Base.....	77
3.4.5. Desain Anta muka	78
3.5. Lokasi dan Jadwal penelitian	83
3.5.1. Lokasi.....	83
3.5.2. Jadwal Penelitian.....	84

BAB IV HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian	85
4.2.1. Pengujian Validasi Sistem.....	91
4.2.2. Pengujian Dengan Pakar	93

BAB V SIMPULAN DAN SARAN

5.1. Simpulan.....	95
5.2. Saran	96

DAFTAR PUSTAKA	97
-----------------------------	-----------

DAFTAR RIWAYAT HIDUP

SURAT BALASAN IZIN PENELITIAN

LAMPIRAN

DAFTAR TABEL

	Halaman
Tabel 2. 1 Defenisi Sistem Pakar	14
Tabel 2. 2 Tabel Keputusan	22
Tabel 2. 3 Simbol Use Case Diagram	33
Tabel 2. 4 Simbol Activity Diagram	34
Tabel 2. 5 Simbol Sequence Diagram.....	35
Tabel 2. 6 Simbol Class Diagram	36
Tabel 3. 1 Operasional Variabel.....	51
Tabel 3. 2 Gejala dan Solusi	53
Tabel 3. 3 Aturan.....	54
Tabel 3. 4 Tabel Keputusan	56
Tabel 3. 5 Jadwal Penelitian.....	84
Tabel 4. 1 Menu Beranda	91
Tabel 4. 2 Pengujian Menu Deteksi dan Solusi	91
Tabel 4. 3 Pengujian Hasil Diagnosa	92
Tabel 4. 4 Menu Tentang	92
Tabel 4. 5 Menu Artikel.....	92
Tabel 4. 6 Menu Not Found	92
Tabel 4. 7 Hasil Diagnosa Pakar dan Diagnosa Sistem	94

DAFTAR GAMBAR

	Halaman
Gambar 2. 1	Komponen-komponen sistem pakar..... 18
Gambar 2. 2	Pohon Keputusan..... 22
Gambar 2. 3	Piston..... 26
Gambar 2. 4	Klep 27
Gambar 2. 5	Karburator 28
Gambar 2. 6	Busi..... 29
Gambar 2. 7	Injektor 30
Gambar 2. 8	Logo StarUML 32
Gambar 2. 9	StarUML..... 32
Gambar 2. 10	Logo Android 39
Gambar 2. 11	Logo Java 40
Gambar 2. 12	Logo Eclipse..... 40
Gambar 2. 13	Kerangka Pemikiran..... 45
Gambar 3. 1	Desain Penelitian..... 47
Gambar 3. 2	Pohon Keputusan..... 58
Gambar 3. 3	Use case diagram..... 61
Gambar 3. 4	Activity Diagram Mengelola Knowledge Base 62
Gambar 3. 5	Activity Diagram Execute FileApk..... 63
Gambar 3. 6	Activity Diagram Memperbarui Versi 64
Gambar 3. 7	Activity Diagram Sharing FileApk 65
Gambar 3. 8	Activity Diagram Mengunduh FileApk 66
Gambar 3. 9	Activity Diagram Melakukan Instalasi 67
Gambar 3. 10	Activity Diagram Melakukan Deteksi..... 68
Gambar 3. 11	Activity Diagram Melihat Menu Artikel..... 69
Gambar 3. 12	Activity Diagram Melihat Menu Tentang..... 70
Gambar 3. 13	Sequence diagram mengelola Knowledge base 71
Gambar 3. 14	Sequence Diagram Execute File Apk..... 72
Gambar 3. 15	Sequence Diagram memperbarui Versi..... 72
Gambar 3. 16	Sequence diagram sharing file apk..... 73
Gambar 3. 17	Sequence Diagram Mengunduh File Apk 74
Gambar 3. 18	Sequence Diagram Melakukan Instalasi 74
Gambar 3. 19	Sequence Diagram Melakukan Deteksi 75
Gambar 3. 20	Sequence Diagram Melihat Menu Artikel 75
Gambar 3. 21	Sequence Diagram Melihat Menu Tentang..... 76
Gambar 3. 22	Desain Knowledge Base..... 77
Gambar 3. 23	Rancangan Form Beranda 78
Gambar 3. 24	Rancangan Form Deteksi 79

Gambar 3. 25 Rancangan Form Hasil Deteksi.....	80
Gambar 3. 26 Rancangan Form Artikel	81
Gambar 3. 27 Rancangan Form Tentang	82
Gambar 3. 28 Rancangan Form Not Found	83
Gambar 4. 1 Beranda.....	85
Gambar 4. 2 Diagnosa.....	86
Gambar 4. 3 Hasil Deteksi dan solusi	87
Gambar 4. 4 Tentang.....	88
Gambar 4. 5 Artikel.....	89
Gambar 4. 6 Form Not Found	90

DAFTAR LAMPIRAN

LAMPIRAN 1 FORM WAWANCARA

LAMPIRAN 2 FOTO WAWANCARA

LAMPIRAN 3 KODING PROGRAM

BAB I

PENDAHULUAN

1.1. Latar Belakang Penelitian

Dewasa ini motor adalah salah satu kendaraan yang banyak diminati oleh masyarakat, dan motor bukan barang mewah lagi di jaman sekarang karena terbukti hampir semua masyarakat memiliki motor untuk alat transportasi mereka. Motor menjadi alat transportasi yang paling dinamis dan cepat saat ini dibanding dengan alat transportasi lainnya, hal ini dibuktikan dengan lebih banyaknya pengendara sepeda motor dibandingkan dengan pengendara transportasi lainnya di jalan. Sepeda motor harganya masih bisa dijangkau oleh masyarakat luas, perawatan dan penggunaannya pun lebih mudah dibandingkan dengan mobil. Hal ini yang membuat sebagian masyarakat luas lebih memilih menggunakan kendaraan sepeda motor. Dalam bidang transportasi, kita telah mengenal sepeda motor sebagai salah satu alat transportasi yang biasa digunakan oleh masyarakat. Hal ini dikarenakan selain praktis dan hemat jika dibandingkan dengan membayar biaya angkutan umum, alasan lain yang mempengaruhi perkembangan sepeda motor di Indonesia adalah untuk mempercepat seseorang apabila akan melakukan perpindahan dari suatu tempat ke tempat dan terlebih jika sedang terjadi kemacetan dengan mengendarai motor maka masyarakat pada umumnya akan lebih mudah untuk menerjang kemacetan tersebut.

Saat ini banyak perusahaan yang bergerak dibidang transportasi seperti sepeda motor bersaing dan berlomba-lomba menawarkan produknya. Masing-masing perusahaan memberikan keunggulan yang terbaik dari produk yang ditawarkan kepada konsumen, agar perusahaan tersebut dapat merebut pasar persaingan. Di mata konsumen produksi sepeda motor yang mempunyai kualitas dari segi model, ketersediaan suku cadang, bengkel resmi, desain produk, performa mesin dan harga jual kembali menjadi faktor-faktor pendukung dalam menentukan pilihan mereka. Demikian halnya dengan produk motor matik yang juga banyak diminati oleh masyarakat karena motor matik lebih praktis untuk digunakan dibandingkan dengan sepeda motor manual sehingga ini membuat perusahaan yang memproduksi produk sepeda motor matik dengan masing-masing keunggulan motor yang mereka produksi. Demikian halnya dengan sepeda motor matik Suzuki Nex F1 yang tergolong sangat terjangkau yang menjadi salah satu kelarisan pemasaran sepeda motor ini, selain itu sepeda motor matik Suzuki F1 mengusung mesin berkapasitas 113 cc, dan Suzuki Nex F1 mengadopsi sebuah tampilan yang *sporty* dan juga akan terasa nyaman karena Suzuki telah mendesain sepeda motor ini dengan jarak pijak ke tanah sekitar 135 mm dan tampilannya terlihat lebih elegan.

Terlepas dari keunggulan motor matik tersebut , banyak pengguna sepeda motor yang mengeluhkan permasalahan yang sering terjadi pada sepeda motor yang mereka kendarai. Terlebih pengguna sepeda motor yang disibukkan dengan pekerjaan sehari-harinya yang membuat mereka tidak mempunyai waktu untuk melakukan *service* ke bengkel motor, yang seharusnya motor tersebut harus di

service, akan tetapi pengguna mengabaikan hal tersebut karena kendala waktu dari pengguna. Selain itu kurangnya pengetahuan pengguna masalah kerusakan sepeda motor menjadi salah satu kendala bagi pengguna sepeda motor, dan itu membuat mereka mendatangi bengkel motor yang ada, akan tetapi karena pengguna kurang mengetahui kerusakan motor dan itu membuat pengguna mempercayakan sepenuhnya kerusakan motor mereka kepada bengkel yang pengguna datangi, akan tetapi mengingat banyaknya persaingan bengkel yang ada sekarang ini, terkadang pengguna mendatangi bengkel motor yang kurang jujur dalam pelayanan yang mereka lakukan, dan karena keterbatasan pengguna motor mengenai kerusakan motor sehingga membuat pengguna harus tetap melakukan *service* sepeda motor mereka. Faktor ekonomi juga sangat berpengaruh kepada pengguna motor untuk melakukan *service* sepeda motor yang pengguna pakai, karena mahalnya biaya untuk melakukan *service* motor, sehingga membuat pengguna menunda-nunda perawatan sepeda motor yang pengguna kendarai.

Untuk membuat suatu sistem yang memudahkan pengguna dalam perawatan kerusakan sepeda motor dengan menggunakan sistem pakar. Sistem pakar adalah salah satu cabang dari *AI* yang membuat penggunaan secara luas *knowledge* yang khusus untuk menyelesaikan masalah tingkah 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. Teknologi sistem pakar ini meliputi bahasa sistem pakar, program dan perangkat keras yang dirancang untuk membantu pengembangan dan pembuatan.

Metode yang digunakan dalam penelitian ini adalah metode *Forward chaining*. Salah satu metode yang paling umum digunakan untuk merepresentasikan pengetahuan adalah dalam bentuk tipe aturan “Jika..Maka”(IF..THEN). Dan menurut penelitian *forward chaining* merupakan grup dari *multiple* inferensi yang melakukan pencarian dari suatu masalah kepada solusinya, selain itu *forward chaining* merupakan proses peruntukan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir (Ivon Idiego,2010.) Jadi metode *forward chaining* dimulai dari informasi masukan (*if*) dahulu kemudian menuju konklusi (*then*) atau dapat dimodelkan sebagai berikut : *IF* (informasi masukan) *THEN* (konklusi). Informasi masukan dapat berupa data, bukti, temuan atau pengamatan, sedangkan konklusi dapat berupa tujuan, penjelasan, atau diagnosa.

Selain metode *fordward chaining* yang digunakan, di dalam penelitian ini juga menggunakan aplikasi berbasis *android* untuk membantu pengguna sepeda motor dalam mendeteksi kerusakan sepeda motor. *Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi.

Maka dari permasalahan di atas, peneliti mengangkat sebuah judul penelitian yang berjudul : **“SISTEM PAKAR MENDETEKSI KERUSAKAN SEPEDA MOTOR SUZUKI NEX F1 BERBASIS *ANDROID*”**.

1.2. Identifikasi Masalah

Dari latar belakang di atas, adapun identifikasi masalah adalah sebagai berikut:

1. Minimnya pengetahuan masyarakat tentang kerusakan sepeda motor matik Suzuki Nex F1.
2. Mahalnya biaya perawatan dan *service* secara berkala motor Suzuki Nex F1.
3. Kadang pengguna mendapatkan *service* motor yang kurang jujur.
4. Pengguna terkendala dengan waktu untuk melakukan *service* langsung ke bengkel.
5. Belum adanya suatu sistem yang dirancang untuk mendeteksi kerusakan sepeda motor Suzuki Nex F1 berbasis *android*.

1.3. Pembatasan Masalah

Agar penelitian ini tidak menyimpang dari yang seharusnya, adapun batasan masalah sebagai berikut:

1. Penelitian ini membahas tentang gejala kerusakan sepeda motor Suzuki Nex F1.
2. Penelitian ini juga membahas kendala yang dihadapi pengguna sepeda motor dalam melakukan *service*.
3. Sistem yang dirancang adalah sistem pakar dengan menggunakan metode *forward chaining* dan berbasis *android*.

4. Penelitian ini mewawancarai seorang pakar dan pengambilan data di *dealer* motor Suzuki Roda Mas Makmur Motor Batu aji.
5. Penelitian ini akan memberikan solusi bagi pengguna sepeda motor untuk mengetahui gejala kerusakan motor.

1.4. Perumusan Masalah

Berdasarkan pembatasan masalah di atas maka, adapun perumusan masalah dalam penelitian ini sebagai berikut:

1. Bagaimana merancang suatu sistem pakar berbasis *android* terhadap kerusakan sepeda motor Suzuki Nex F1?
2. Bagaimana aplikasi sistem pakar berbasis *android* untuk mendeteksi kerusakan sepeda motor ini dapat memberikan solusi dan kemudahan bagi pengguna sepeda motor ?
3. Bagaimana sistem pakar dapat membantu untuk mendeteksi kerusakan sepeda motor matik Suzuki Nex F1?

1.5. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Sistem pakar ini dirancang untuk mendeteksi kerusakan sepeda motor Suzuki Nex F1.
2. Sistem pakar membantu menemukan solusi kerusakan dengan aplikasi yang akan dirancang.

3. Sistem pakar berbasis android memberikan solusi bagi pengguna agar memudahkan dalam mendeteksi kerusakan sepeda motor Suzuki Nex F1.

1.6. Manfaat Penelitian

Manfaat penelitian dapat dikategorikan menjadi dua, yaitu manfaat teoritis dan manfaat praktis. Secara teoritis manfaat penelitian adalah:

- a. Aspek teoritis
 1. Memperdalam penelitian penulis secara teori tentang sistem pakar.
- b. Aspek Praktis
 1. Bagi Penulis

Hasil penelitian dapat dijadikan pengalaman guna menambah wawasan penulis dalam membuat sebuah sistem pakar untuk selanjutnya dapat diterapkan untuk membuat aplikasi sistem pakar kerusakan motor.

2. Bagi Masyarakat

Diharapkan dengan adanya aplikasi sistem pakar ini dapat memudahkan masyarakat dalam mengenal dan mengetahui kerusakan sepeda motor berbasis android yang dialami masyarakat.

BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

Supaya penelitian ini dapat berjalan dengan baik, maka diperlukan landasan bagi jalannya penelitian berupa teori yang telah ada. Dalam penelitian ini, akan menjelaskan secara singkat tentang kecerdasan buatan dimana kecerdasan Buatan itu sendiri terdiri dari beberapa cabang yaitu Jaringan Saraf Tiruan, Sistem Pakar dan Logika *Fuzzy*.

2.1.1. Kecerdasan Buatan

Kecerdasan buatan berasal dari bahasa Inggris "*Artificial Intelligence*" disingkat AI, yaitu *Intelligence* adalah kata sifat yang berarti cerdas, sedangkan *Artificial* artinya buatan. Kecerdasan buatan yang dimaksud adalah merujuk pada mesin yang mampu berpikir, mampu mengambil keputusan seperti yang dilakukan manusia. Alan Turing ahli matematika berkebangsaan Inggris yang dijuluki bapak komputer modern dan pembongkar sandi Nazi dalam era Perang Dunia ke II 1950, menetapkan definisi *Artificial Intelligence* "Jika komputer tidak dapat dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan (Sutojo, 2011). Kecerdasan buatan memungkinkan komputer untuk berfikir dan menirukan proses belajar manusia sehingga informasi baru dapat diserap sebagai pengetahuan dan

proses pembelajaran serta dapat digunakan sebagai acuan di masa yang akan datang.

2.1.2 Fuzzy logic

2.1.2.1. Pengertian Logika Fuzzy

Konsep tentang *Fuzzy logic* pertama kali diperkenalkan oleh Prof. Astor Zadeh pada 1962. Logika fuzzy adalah metodologi sistem control pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kacil, embedded sistem, jaringan PC, multi-chanel atau workstation berbasis akuisisi data, dan sistem control. (Sutojo, 2011, p. 211)

2.1.2.2. Operasi Himpunan Fuzzy

Operasi himpunan *Fuzzy* diperlukan untuk proses inferensi atau penalaran. Dalam hal ini yang dioperasikan adalah derajat keanggotaannya. Berikut beberapa operasi dasar yang paling sering digunakan untuk mengombinasi dan memodifikasi himpunan *fuzzy*.

1. Operasi Gabungan (*Union*) sering disebut operator OR dari himpunan fuzzy A dan B, dan dalam operasi gabungan disebut sebagai *Max*.
2. Operasi Irisan (*Intersection*) yang sering disebut dengan operator AND dari himpunan A dan B dan dalam sistem logika *fuzzy*, operasi irisan disebut sebagai *Min*.
3. Operator Komplemen adalah komponen dari himpunan fuzzy A yang sering disebut NOT adalah dengan fungsi keanggotaan untuk setiap x elemen X. (Sutojo, 2011, pp. 227–228)

2.1.2.3. Metode Logica Fuzzy

Metode dari *Logica Fuzzy* sendiri terdiri dari metode diantaranya adalah:

1. Metode Tsukamoto: Secara umum bentuk model *fuzzy* Tsukamoto adalah If(X IS A) and (Y IS B) Then (Z IS C). Dimana A, B dan C adalah himpunan *Fuzzy*. Dalam inferensinya, metode Tsukamoto menggunakan tahapan berikut:
 - a. Fuzzyfikasi
 - b. Pembentukan basis pengetahuan Fuzzy (Rule dalam bentuk IF..THEN)
 - c. Mesin Inferensi
 - d. Defuzzyfikasi.
2. Metode Mamdani adalah metode yang paling sering digunakan dalam aplikais karena strukturnya yang sederhana yaitu menggunakan operasi MIN-MAX atau MAX-PRODUCT.
3. Metode Sugeno yaitu output sistem berupa konstanta atau persamaan linier. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada 1985.(Sutojo, 2011, pp. 233–237)

2.1.2 Pengertian Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah paradigma pengelola informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (neuron), bekerja serentak untuk menyelesaikan masalah

tertentu (Sutojo, 2011, p. 283). Kelebihan-kelebihan yang diberikan oleh JST adalah:

1. Belajar Adaptive, kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. *Self-organization*, sebuah JST dapat membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real-time operation*, perhitungan JST dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.(Sutojo, 2011, p. 284).

Selain mempunyai kelebihan tersebut, JST juga mempunyai kelemahan-kelemahan yaitu sebagai berikut:

1. Tidak efektif untuk melakukan operasi-operasi numeric dengan presisi tinggi
2. Tidak efisien untuk operasi algoritma aritmatik, operasi logika, dan simbolis,
3. Untuk beroperasi JST butuh pelatihan sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.(Sutojo, 2011, p. 284).

2.1.3 Sistem Pakar (*Expert Sytem*)

2.1.4.1 Defenisi Sistem Pakar (Expert System)

Sistem pakar adalah salah satu teknik kecerdasan buatan yang meniru penalaran manusia. Pemecahan masalah yang kompleks biasanya hanya dilakukan oleh seorang pakar. Sistem pakar mencoba mencari penyelesaian yang memuaskan, yaitu dengan penyelesaian yang bagus agar pekerjaan dapat berjalan meskipun itu bukan penyelesaian yang optimal (Hartati Sri dan Iswanti Sari, 2008. 2) . Defenisi sistem pakar bisa di lihat pada tabel 2.1.

Tabel 2. 1 Defenisi Sistem Pakar

Sumber	Definisi
Martin dan Oxman (1998)	Sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam pemecahan masalah, dimana biasanya hanya dapat diselesaikan seorang pakar.
Iqnizio (1991)	Sistem pakar merupakan bidang yang didirikan oleh sistem berbasis pengetahuan (<i>Knowledge base system</i>), yang memungkinkan komputer dapat berpikir dan mengambil kesimpulan dari sejumlah kaidah.
Turban dan Aronson (2001)	Sistem yang menggunakan pengetahuan manusia yang kemudian dimasukkan kedalam komputer dan untuk dapat memecahkan masalahnya akan dikerjakan oleh seorang pakar

Sumber : (Hartati Sri dan Iswanti Sari, 2008, p. 3)

2.1.4.2 Kategori Permasalahan Sistem Pakar

Biasanya aplikasi sistem pakar menyentuh beberapa area permasalahan sebagai berikut menurut (Sutojo, 2011, p. 162).

1. Interpretasi, menghasilkan deskripsi situasi berdasarkan data-data masukan.

2. Prediksi, memperkirakan akibat yang mungkin terjadi dari situasi yang ada.
3. Diagnosis, menyimpulkan suatu keadaan berdasarkan gejala yang diberikan (*symptoms*).
4. Desain, melakukan perancangan berdasarkan kendala yang diberikan.
5. *Planning*, merencanakan tindakan yang akan dilakukan.
6. *Monitoring*, membandingkan hasil pengamatan dengan proses perencanaan.
7. *Debugging*, menentukan penyelesaian dari suatu kesalahan sistem.
8. Reparasi, melaksanakan rencana perbaikan.
9. *Instruction*, melakukan instruksi untuk diagnosis, debugging, dan perbaikan kinerja.
10. Kontrol, melakukan kontrol terhadap hasil interpretasi, diagnosis, debugging, monitoring, dan perbaikan tingkah laku sistem.

2.1.4.3 Manfaat dan Kekurangan Sistem Pakar

Menurut (Sutojo, 2011, p. 160) sistem pakar menjadi sangat populer karena sangat banyaknya kemampuan dan manfaat yang diberikan, diantaranya:

1. Meningkatkan produktivitas.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan proses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.

8. Meningkatkan kapabilitas sistem komputer..
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dan pelatihan.
11. Meningkatkan kemampuan untuk menyelesaikan masalah.

Selain manfaat, ada juga kekurangan yang ada pada sistem pakar, diantaranya:

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar.

2.1.4.4 Komponen Sistem Pakar

Menurut Giarratano dan Riley (2005) dalam (Hartati & Iswanti, 2008, p. 3) sistem pakar sebagai suatu program yang berfungsi untuk menirukan pakar manusia dan harus bisa melakukan hal-hal yang dapat dikerjakan oleh seorang pakar. Untuk membangun sistem yang seperti itu maka komponen yang harus dimiliki adalah sebagai berikut:

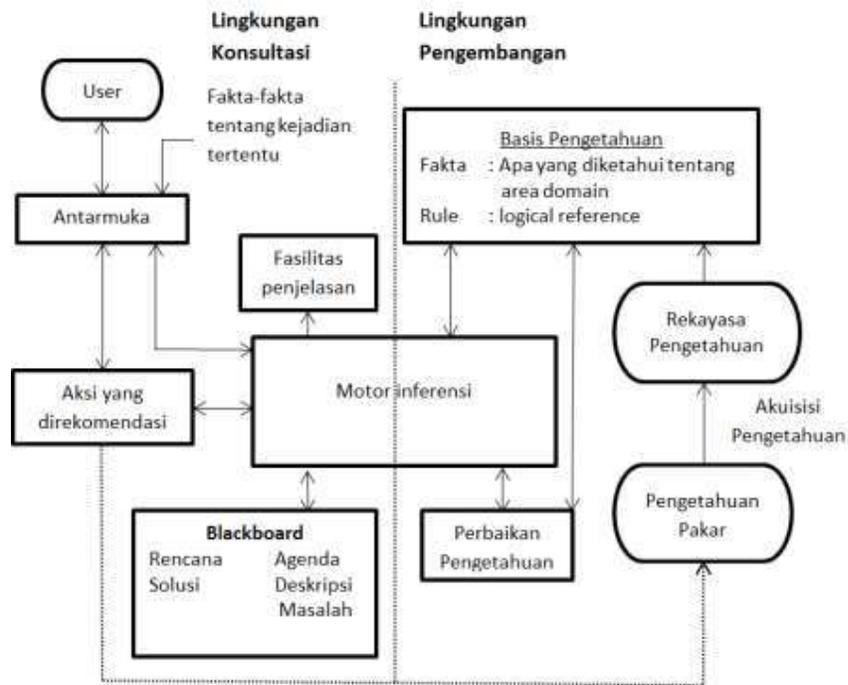
1. Antar muka pengguna (*user interface*),
2. Basis pengetahuan (*knowledge base*),
3. Mekanisme inferensi (*inference machine*),
4. Memori kerja (*working memory*).

Sedangkan untuk menjadikan sistem pakar menjadi lebih menyerupai seorang pakar yang berinteraksi dengan pemakai, maka dilengkapi dengan fasilitas berikut:

1. Fasilitas penjelasan (*explanation facility*),
2. Fasilitas akuisisi pengetahuan (*knowledge acquisition facility*).

2.1.4.5 Struktur Sistem Pakar

Menurut (Sutojo, 2011, p. 166) ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledgebase* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. Gambar 2.1 menunjukkan komponen-komponen yang penting dalam sebuah sistem pakar.



Gambar 2. 1 Komponen-komponen sistem pakar

Sumber: (Sutojo, 2011, p. 166)

Keterangan:

1. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa di proses oleh *computer* dan menaruhnya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan). Sumber-sumber pengetahuan bisa diperoleh dari pakar, buku.

2. Basis Pengetahuan (*Knowledge Base*).

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu:

- a. Fakta, misalnya situasi, kondisi atau permasalahan yang ada.
- b. *Rule* (Aturan), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

3. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan.

4. Daerah Kerja (*Blackboard*)

Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi, sistem pakar membutuhkan *Blackboard*, yaitu area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard*, yaitu:

- a. Rencana: bagaimana menghadapi masalah.
- b. Agenda: aksi-aksi potensial yang sedang menunggu untuk dieksekusi.
- c. Solusi: calon aksi yang akan dibangkitkan.

5. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling bagus bila disajikan dalam bahasa alami (*natural*

language) dan dilengkapi dengan grafik, menu, dan formulir elektronik.

Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

6. Subsistem Penjelasan (*Explanation Subsystem / Justifier*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil dan kemampuan ini penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

7. Sistem Perbaikan Pengetahuan (*knowledge refining system*)

Berfungsi memperbaiki pengetahuan (*knowledge refining system*) dari seorang pakar diperlukan untuk menganalisis pengetahuan, dan memperbaiki pengetahuan sehingga dapat dipakai pada masa mendatang.

8. Pengguna (*User*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada.

2.1.4.6 Representasi Pengetahuan

Pemrosesan yang dilakukan oleh sistem pakar merupakan pemrosesan pengetahuan, bukan pemrosesan data seperti yang dikerjakan dengan pemrograman secara konvensional yang kebanyakan dilakukan oleh sistem informasi. Pengetahuan (*knowledge*) adalah pemahaman secara praktis maupun teoritis terhadap suatu obyek atau domain tertentu (Hartati Sri dan Iswanti Sari, 2008, p. 18). Berikut ini adalah contoh struktur kaidah *IF-THEN* yang

menghubungkan objek (Adedeji, 1992 *dalam* (Hartati Sri dan Iswanti Sari, 2008, p. 25):

1. *IF* premis *THEN* konklusi
2. *IF* masukan *THEN* keluaran
3. *IF* kondisi *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan
7. *IF* aksi *THEN* reaksi
8. *IF* gejala *THEN* diagnosa

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan (diagnosa) (Hartati Sri dan Iswanti Sari, 2008, p. 25).

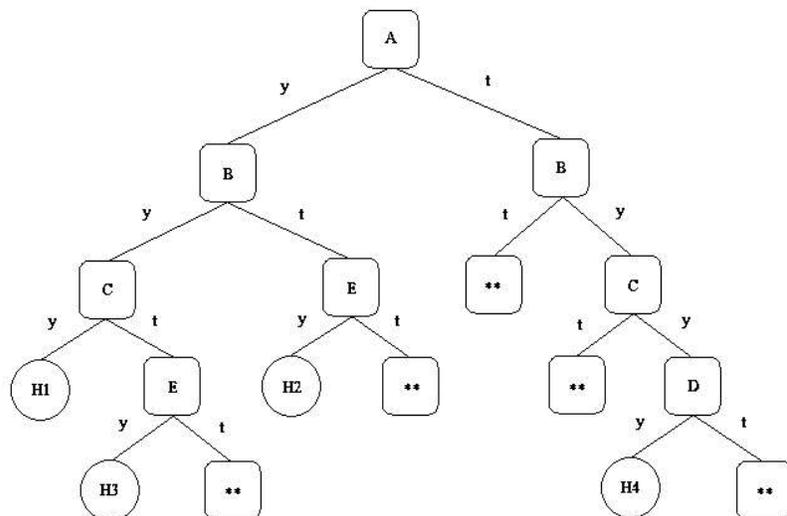
Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk tabel keputusan kemudian

dibuat pohon keputusannya. Berikut ini adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan (Hartati Sri dan Iswanti Sari, 2008, p. 26)

Tabel 2. 2 Tabel Keputusan

Hipotesa	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
Evidence A	Ya	Ya	Ya	tidak
Evidence B	Ya	Tidak	Ya	ya
Evidence C	Ya	Tidak	Tidak	ya
Evidence D	Tidak	Tidak	Tidak	ya
Evidence E	Tidak	Ya	Ya	tidak

Sumber: (Hartati dan Iswanti 2008: 32)



Gambar 2. 2 Pohon Keputusan

Sumber: (Hartati Sri dan Iswanti Sari, 2008, p. 33)

Keterangan:

A = evidence A, H1 = hipotesa 1, y = ya

B = evidence B, H2 = hipotesa 2, t = tidak

$C = \text{evidence } C$, $H3 = \text{hipotesa } 3$, $** = \text{tidak menghasilkan hipotesa tertentu}$

$D = \text{evidence } D$, $H4 = \text{hipotesa } 4$

Dilihat dari gambar 2.3, masing-masing *node* yang mewakili *evidence* tertentu untuk kondisi “y” dan “t” sudah tidak mengarah pada *evidence* yang sama. Hal ini berarti jawaban pengguna yang berbeda akan mengarah pada pertanyaan yang berbeda pula.

Kaidah yang dapat dihasilkan berdasarkan pohon keputusan pada gambar 2.3 adalah sebagai berikut:

1. Kaidah 1: *IF A AND B AND C THEN H1*
2. Kaidah 2: *IF A AND B AND E THEN H3*
3. Kaidah 3: *IF A AND E THEN H2*
4. Kaidah 4: *IF D AND B AND C THEN H4*

Model representasi pengetahuan kaidah produksi banyak digunakan pada aplikasi sistem pakar karena model representasi ini mudah dipahami dan bersifat deklaratif sesuai dengan jalan pikiran manusia dalam menyelesaikan suatu masalah, dan mudah diinterpretasikan.

2.1.4.7 Mesin Inferensi (Inferensi Engine)

Menurut (Hartati Sri dan Iswanti Sari, 2008, p. 5) mesin inferensi merupakan otak dari sistem pakar, bisa dikatakan sebagai mesin pemikir (*Thinking Machine*). Pada prinsipnya mesin inferensi inilah yang akan mencari solusi dari suatu masalah. Menurut Kusri (2006: 35) inferensi merupakan proses

untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar, proses inferensi dilakukan dengan suatu modul yang disebut Mesin Inferensi (*Inference Engine*). Ada dua metode inferensi yang penting dalam sistem pakar, yaitu *forward chaining* (runut maju) dan *backward chaining* (runut mundur).

2.1.4.8 Forward Chaining (Runut Maju)

Menurut (Sutojo, 2011, p. 171) *Forward Chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam *database*. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi.

Konsep ini dapat juga disebut sebagai pencarian yang dimotori data (*data driven search*). Runut maju melakukan proses perunutan (penalaran) dimulai dari premis-premis atau informasi masukan (*IF*) terlebih dahulu kemudian menuju konklusi atau *derived information (THEN)*. Konsep ini dapat dimodelkan sebagai berikut:

IF (informasi masukan)

THEN (konklusi)

Informasi masukan dapat berupa suatu pengamatan sedangkan konklusi dapat berupa diagnosa sehingga dapat dikatakan jalannya penalaran runut maju dimulai dari pengamatan menuju diagnosa. Pada metode ini, sistem tidak melakukan praduga apapun terhadap konklusi, namun sistem akan menerima semua gejala yang diberikan pengguna lalu sistem akan memeriksa gejala-gejala tersebut dan selanjutnya mencocokkan dengan konklusi yang sesuai (Hartati Sri dan Iswanti Sari, 2008, p. 45).

2.1.4.9 Validasi Sistem

Menurut (A.S & M.Shalahuddin, 2011, pp. 211–214) validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat digunakan dan telah sesuai dengan yang diharapkan. Adapun pendekatan dalam melakukan pengujian untuk validasi sistem yaitu:

1. *Black-Box Testing* (Pengujian Kotak Hitam)

Black-box testing adalah pengujian sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program yang bertujuan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari sistem atau perangkat lunak telah sesuai dengan spesifikasi yang dibutuhkan. Pengujian pada *black-box testing* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem apakah telah sesuai dengan spesifikasi yang dibutuhkan.

2. *White-Box Testing* (Pengujian Kotak Putih)

White-box testing adalah pengujian sistem atau perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. Pengujian pada *white-box testing* dilakukan dengan memeriksa logika dari kode program.

2.1 Variabel

Variabel adalah objek penelitian, atau apa saja yang menjadi fokus di dalam suatu penelitian untuk di pelajari dan di peroleh kesimpulannya.(Sudaryono, 2014, p. 16).

Pada penelitian ini, peneliti hanya berfokus pada bagian mesin dari sepeda motor Suzuki nex F1. Peneliti mengangkat gejala kerusakan yang terjadi pada bagian mesin. Beberapa diantaranya yaitu:

1. Kerusakan Piston



Gambar 2. 3 Piston
(Sumber: Data Penelitian, 2018)

Piston adalah komponen mesin yang membentuk ruang bakar bersama-sama dengan silinder blok dan silinder head, piston juga dapat melakukan gerakan naik turun untuk melakukan siklus kerja mesin. Piston berfungsi untuk mempertahankan kerapatan antara piston dengan dinding silinder agar tidak ada kebocoran gas dari ruang bakar ke dalam bak mesin. Berikut adalah gejala-gejala yang terjadi pada piston:

- 1) Mesin motor sulit hidup
- 2) Tenaga mesin kurang atau lemah
- 3) Suara mesin kasar
- 4) Gas motor tersendat-sendat

Cara memperbaiki kerusakan piston yang rusak adalah harus mengganti piston tersebut. Namun sebelum menggantinya pastikan bahwa piston yang baru sesuai dengan spesifikasi motor Suzuki nex F1.

2. Kerusakan klep



Gambar 2. 4 Klep

(Sumber: Data Penelitian, 2018)

Klep atau yang sering juga dikenal sebagai katup adalah salah satu komponen pada mesin motor yang mempunyai sifat dinamis yang biasanya terpasang dikepala silinder. Fungsi katup adalah untuk mengatur masuknya gas baru dan keluarnya gas buang sisa pembakaran pada mesin motor. Gejala yang terjadi akibat kerusakan pada klep motor yaitu:

- 1) Mesin motor sulit hidup.
- 2) Motor mengeluarkan asap tebal berwarna hitam.
- 3) Oli mesin cepat habis.
- 4) Bahan bakar motor menjadi boros.

Solusinya adalah membersihkan klep, *siting* dan bos klep menggunakan bensin, Bila tidak terjadi perubahan gantilah katup tersebut dengan yang baru dan melakukan pemasangan dengan hati-hati.

3. Kerusakan Karburator



Gambar 2. 5 Karburator
(Sumber: Data Penelitian, 2018)

Karburator merupakan komponen untuk mencampur udara dan bahan bakar. Fungsi karburator ada dua yaitu pertama, mengatur RPM untuk mencampur udara dan bahan bakar sesuai dengan perbandingan. Kedua, karburator juga harus mampu menyumplai bensin dengan perbandingan yang ideal pada segala RPM.

Gejala-gejala yang terjadi pada karburator yaitu:

- 1) Mesin motor sulit hidup.
- 2) Motor mengeluarkan asap tebal berwarna hitam.
- 3) Knalpot mengeluarkan suara (knalpot nembak)
- 4) Kecepatan motor sulit untuk bertambah

Solusinya adalah putar setelan sekrup udara searah jarum jam atau arah menutup, yaitu memaksudkan memperbanyak campuran udara dengan bahan bakar. Kemudian bersihkan karburator dari kotoran yang menempel.

4. Kerusakan Busi



Gambar 2. 6 Busi

(Sumber: Data Penelitian, 2018)

Busi memiliki peranan penting dalam sistem pengapian pada kendaraan sepeda motor, busi berfungsi untuk mengubah tegangan listrik dengan yang disalurkan koil, menjadi menjadi percikan api yang akan membakar campuran bahan bakar dan udara diruang mesin. Kerusakan yang terjadi akibat busi rusak yaitu:

- 1) Mesin motor sulit hidup.
- 2) Tenaga mesin kurang atau lemah.
- 3) Suara mesin kasar.
- 4) Mesin mati mendadak ketika di jalan.

Solusinya adalah membersihkan kerak-kerak hitam yang menempel pada bagian kepala busi, jika masih tetap rusak, maka busi perlu diganti yang baru.

5. Kerusakan Injektor



Gambar 2. 7 Injektor

(Sumber: Data Penelitian, 2018)

Mempunyai fungsi yaitu untuk metode pencampuran bahan bakar dengan udara pada kendaraan bermotor untuk menghasilkan pembakaran yang sempurna.

Kerusakan yang disebabkan oleh injector pada motor yaitu:

- 1) Mesin sulit distarter,
- 2) Asap knalpot lebih pekat
- 3) Mesin sering mati mendadak.
- 4) Mesin mogok dan tidak mampu dihidupkan

Solusi untuk memperbaiki dari kerusakan injector adalah maka perlu mengecek apakah bensin sampai keruang bakar, Cek bagian-bagian yang terhubung dengan *injector*, bila tidak ada perubahan gantilah *injector* dengan yang baru.

2.2 Software Pendukung

2.3.1. UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah sebuah standarisasi bahasa pemodelan perangkat lunak yang dibangun menggunakan teknik pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual yang menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML (*Unified Modeling Language*) juga berarti bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodeologi tertentu,

meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (A.S & M.Shalahuddin, 2011, p. 135). Salah satu aplikasi untuk memudahkan menggunakan UML adalah StarUML.



Gambar 2. 8 Logo StarUML

(Sumber: <http://staruml.sourceforge.net/image/staruml-logo.jpg>)

StarUML merupakan salah satu *CASE (Computer-Aided Software Engineering) tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upperCASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak (Rosa & M.Shalahuddin, 2013, pp. 122–123).



Gambar 2. 9 Interface StarUML

(Sumber : <http://staruml.io>)

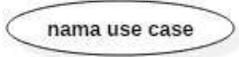
UML mendefinisikan diagram-diagram adalah sebagai berikut:

1. *Use case diagram*
2. *Activity diagram*
3. *Sequence diagram*
4. *Class diagram*

2.3.1.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. *Use case diagram* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, *meng-create* sebuah daftar belanja, dan sebagainya (Yasin, 2012: 198).

Tabel 2. 3 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
<p>Aktor/<i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
<p>asosiasi/<i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor

Ekstensi/ <i>extend</i> <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu. Arah panah mengarah pada <i>use case</i> yang ditambahkan.
generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
Menggunakan/ <i>include/uses</i> <<include>>  <<uses>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya <i>use case</i> ini. Arah panah mengarah pada <i>use case</i> yang ditambahkan

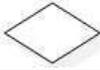
Sumber: (A.S & M.Shalahuddin, 2011, p. 156)

2.3.1.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau manu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktifitas yang dapat dilakukan oleh sistem.

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktifitas memiliki sebuah status awal
Aktifitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja

Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
<i>Swimlane</i> atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi

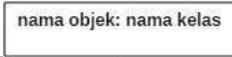
Sumber: (A.S & M.Shalahuddin, 2011, pp. 162–163)

2.3.1.3 Sequence Diagram

Menurut (A.S & M.Shalahuddin, 2011, p. 137) diagram sequen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Tabel 2. 5 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor/ <i>actor</i>  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Garis hidup/ <i>lifeline</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor

Objek  nama objek: nama kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat

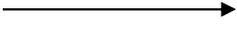
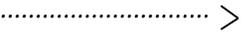
Sumber: (A.S & M.Shalahuddin, 2011, p. 209)

2.3.1.4 Class Diagram

Menurut (A.S & M.Shalahuddin, 2011, p. 122) *Class Diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Tabel 2. 6 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i>  Nama_ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Semua-bagian (<i>whole-part</i>)

Sumber: (A.S & M.Shalahuddin, 2011, pp. 123–124)

2.3.1 *Android*

Android merupakan *software* yang digunakan pada perangkat *mobile* yang mencakup sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh *Google*. Sehingga android mencakup keseluruhan sebuah aplikasi, mulai dari sistem operasi sampai pada pengembangan aplikasi itu sendiri. Pengembangan aplikasi pada platform *Android* ini menggunakan dasar bahasa pemrograman Java. Platform pengembangan aplikasi *Android* ini bersifat *open-source* atau terbuka, sehingga dapat dapat mengembangkan kemampuan untuk membangun

aplikasi yang kaya dan inovatif.(EMS, 2015). Fitur yang terdapat dalam android antara lain adalah:

1. *Android run-time*, yaitu terdiri atas *library java* dan *dalvik virtual machine*.
2. *Open graphics library*, yaitu *application program interface* yang digunakan untuk membuat grafis 2D dan 3D.
3. *Webkit*, yaitu *engine* dari *web browser* yang digunakan untuk menampilkan isi *website* dan menyederhanakan tampilan dari proses *loading*.
4. *SQLite*, yaitu *engine* dari relasional *database* yang dapat diintegrasikan dengan aplikasi.

Menurut (EMS, 2015) Kelebihan dari penggunaan sistem *android*, yaitu:

1. *Multitasking*, dapat menjalankan aplikasi secara bersamaan.
2. Terdapat notifikasi ketika ada panggilan atau sms, yaitu ketika ada sms atau *email* yang masuk, akan terdapat notifikasi pada ponsel.
3. Dukungan ribuan aplikasi terpercaya melalui situs *Google Play*, yaitu berfungsi mendapatkan berbagai aplikasi yang diperlukan.
4. Penggunaan *widget* pada *home screen*, yaitu memudahkan dan mempercepat pengguna ketika membuka aplikasi.



Gambar 2. 10 Logo Android
(Sumber: (EMS, 2015, p. 2)

2.3.4 Java

Java adalah bahasa pemrograman yang dapat dijalankan di komputer maupun telepon genggam. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam *p-code (bytecode)* dan dapat dijalankan pada berbagai *Mesin Virtual Java (JVM)*. Java merupakan bahasa pemrograman yang bersifat umum dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin (Bambang Hariyanto, 2014: 4).

Karena fungsionalitasnya yang memungkinkan aplikasi *java* mampu berjalan di beberapa *platform* sistem operasi yang berbeda. *Java* memiliki beberapa kelebihan yaitu :

1. Bahasa sederhana.
2. Bahasa orientasi objek.
3. Bahasa *statically Typed*.
4. Bahasa dikompilasi.
5. Bahasa yang aman.
6. Bahasa *independen* terhadap *platform*.
7. Bahasa *multithreading*.
8. Bahasa yang didukung *garbage collector*.
9. Bahasa yang tangguh.
10. Bahasa yang mampu diperluas.



Gambar 2. 11 Logo Java
(Sumber: Bambang Hariyanto, 2014: 2)

2.3.4 Eclipse

Eclipse adalah tool editor yang berdiri sendiri yang dipakai dalam pemrograman android baik berbasis java atau dengan pemaketan HTML (EMS, 2015, p. 35).



Gambar 2. 12 Logo Eclipse
Sumber : (EMS, 2015, p. 37)

2.4 Penelitian Terdahulu

Penelitian pertama yang dilakukan (Anggraheni & Uly Wartaty, 2014) yang berjudul “*Sistem Pakar Untuk Mendiagnosa Kerusakan Sepeda Motor Non Injeksi Pada Bengkel Gemilang Jaya Motor Kabupaten Pacitan*”. Pada penelitian ini, sistem pakar untuk mendeteksi kerusakan pada sepeda motor *non injeksi* ini merupakan suatu sistem untuk mempermudah pemilik motor mendeteksi kerusakan pada sepeda motor. Sehingga pemilik dapat mengetahui lebih dini kerusakan pada sepeda motor dan dapat melakukan tindakan awal

sebelum ditindak lanjuti oleh mekanik ataupun dapat menangani kerusakan-kerusakan ringan pada sepeda motor.

Penelitian kedua dilakukan oleh (Maria Shusanti F, 2010) yang berjudul **“Sistem Pakar Untuk Mendeteksi Kerusakan Pada Sepeda Motor 4-tak Dengan Menggunakan Metode Backward Chaining”** : Kendaraan sepeda motor merupakan suatu alat transportasi yang banyak digunakan masyarakat pada umumnya. Pertumbuhan jumlah sepeda motor sangat pesat seiring dengan tingkat ekonomi dan kebutuhan masyarakat terhadap alat transportasi yang murah dan terjangkau sehingga banyak masyarakat yang memilih sepeda motor 4-tak sebagai kendaraan pribadinya untuk digunakan dalam aktivitas sehari-hari. Penelitian ini akan menggambarkan serta menjelaskan sistem pakar yang akan digunakan untuk mendeteksi kerusakan sepeda motor 4-tak dengan menggunakan metode backward chaining dengan jelas. Peneliti akan menggunakan metode *backward chaining* sebagai metode inferensi untuk sistem pakar ini. sistem pakar untuk mendeteksi kerusakan sepeda motor 4-tak dengan menggunakan metode *backward chaining* dapat menggantikan tenaga ahli atau pakar dalam mendeteksi kerusakan sepeda motor 4-tak sehingga kerusakan tersebut dapat diatasi dengan cepat. Selain itu dengan adanya fitur untuk menambah pengetahuan yang hak aksesnya diberikan kepada pengguna pakar maka sistem pakar ini akan dapat terus berkembang sesuai dengan keadaan dan kebutuhan. Kelebihan dari penggunaan metode *backward chaining* adalah metode ini memiliki sifat yang cocok dalam melakukan diagnosa dan dalam pencarian solusi menggunakan metode ini tidak memerlukan waktu yang lama bila pengguna memberikan

informasi yang tepat, tetapi sebaliknya metode ini memiliki kelemahan dalam pencarian solusi, bila pemilihan hipotesa atau kesimpulan tidak tepat maka si pengguna tidak dapat menemukan langsung solusi yang dibutuhkannya sehingga pengguna harus berulang kali melakukan konsultasi.

Penelitian ketiga dilakukan oleh (Sodiq & Shinta, 2016) dengan judul **“Rancang bangun sistem pakar untuk Diagnosa kerusakan pada motor matic dengan Metode *Forward Chaining*”**. Penelitian ini menekankan pada diagnosis dari masalah yang terjadi pada sepeda motor, khususnya teknologi Honda Vario CVT atau lebih dikenal dengan sebutan motor matic untuk mengkhususkan pada transmisi / CVT. Metode pengumpulan data penelitian mulai menggunakan metode observasi (observasi), dan literatur atau yang diketahui literatur. Desain sebuah sistem pakar untuk mendeteksi kerusakan pada motor dibuat dengan menggunakan metode *forward chaining* dengan Bahasa pemrograman Microsoft Visual Basic adalah bahasa pemrograman yang dibuat Microsoft Access. Sistem pakar membuatnya mudah pengguna motor matic untuk menentukan kerusakan pada motor maticnya secara mendalam dan menentukan segala jenis sparebagian yang harus diganti dengan penerapan gejala kerusakan yang tepat. Buka mereka pengetahuan tentang komponen apa saja yang terdapat pada motor matic dan akhirnya konsumen merasa puas dengan aplikasi tersebut.

Penelitian keempat yang dilakukan oleh (Anggri Sartika Wiguna, 2017) **“Sistem pakar Diagnosa Kerusakan Sepeda Motor Matic Injeksi menggunakan metode *Forward Chaining* berbasis android”**. Perkembangan industri sepeda motor *matic* injeksi di Indonesia mengalami perkembangan yang

signifikan, sepeda motor matic injeksi yang lebih irit bahan bakar dan ramah lingkungan, dengan tingginya pengguna sepeda motor matic injeksi saat ini timbul permasalahan

bahwa tidak semua pengguna motor matic injeksi memiliki kemampuan melakukan perbaikan terhadap kerusakan sepeda motornya. Dengan kemajuan teknologi smartphone saat ini, memunculkan suatu ide atau gagasan aplikasi sistem pakar ke dalam aktivitas mutu pelayanan smartphone. Sistem yang akan dibuat adalah “Sistem pakar diagnosa kerusakan sepeda motor matic injeksi menggunakan metode *forward chaining* berbasis mobile” aplikasi ini akan menggunakan metode *forward chaining*. Sistem pakar ini diharapkan dapat membantu pengguna mengetahui kerusakan dan melakukan perbaikan sepeda motornya lebih awal sebelum terjadi kerusakan yang berkelanjutan. Sistem ini dibangun menggunakan aplikasi *Basic4 android*.

Penelitian kelima dari jurnal Internasional yang diteliti oleh (Martinez & Ramirez, 2014) “*Using an improved rule match algorithm in an expert system detect broken driving rules for an energy-efficiency and safety relevant driving system*” yaitu *Vehicles have been so far improved in terms of energy-efficiency and safety mainly by optimising the engine and the power train. However, there are opportunities to increase energy-efficiency and safety by adapting the individual driving behaviour in the given driving situation. In this paper, an improved rule match algorithm is introduced, which is used in the expert system of a human-centred driving system. The goal of the driving system is to optimise*

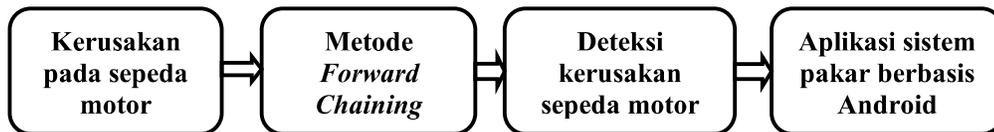
the driving behaviour in terms of energy-efficiency and safety by giving recommendations to the driver. The improved rule match algorithm checks the incoming information against the driving rules to recognise any breakings of a driving rule. The needed information is obtained by monitoring the driver, the current driving situation as well as the car, using in-vehicle sensors and serial-bus systems. On the basis of the detected broken driving rules, the expert system will create individual recommendations in terms of energy-efficiency and safety, which will allow eliminating bad driving habits, while considering the driver needs.

2.5 Kerangka Pemikiran

Menurut Sekaran (2006) dalam (Sudaryono, 2014) mengemukakan bahwa kerangka berpikir adalah model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah penting. Secara teoritis kerangka berpikir yang baik akan menjelaskan peraturan antar variabel yang akan diteliti. Kerangka berpikir di dalam penelitian perlu dikemukakan apabila penelitian hanya membahas satu variabel atau lebih secara mandiri, disamping mengemukakan deskripsi teoritis untuk masing-masing variabel, peneliti juga harus menyatakan argumentasi terhadap besaran variabel yang diteliti. (Sudaryono, 2014). Selanjutnya dalam mengemukakan bahwa kerangka berpikir yang baik memuat hal-hal berikut:

1. Variabel yang diteliti harus dijelaskan.

2. Diskusi dalam kerangka berpikir harus dapat menunjukkan dan menjelaskan hubungan atau keterkaitan antar variabel yang di teliti, serta teori yang mendasarinya.
3. Diskusi juga harus dapat menunjukkan dan menjelaskan apakah hubungan antar variabel itu positif atau negatif, berbentuk simetris, kausal, atau interaktif (timbal balik atau umpan balik)
4. Kerangka berpikir selanjutnya dinyatakan dalam bentuk diagram sehingga pihak lain dapat memahami kerangka berpikir yang dikemukakan dalam penelitian. Berikut adalah kerangka berpikir dari penelitian ini. (Sudaryono, 2015:21)



Gambar 2. 13 Kerangka Pemikiran
Sumber: Olahan Peneliti (2018)

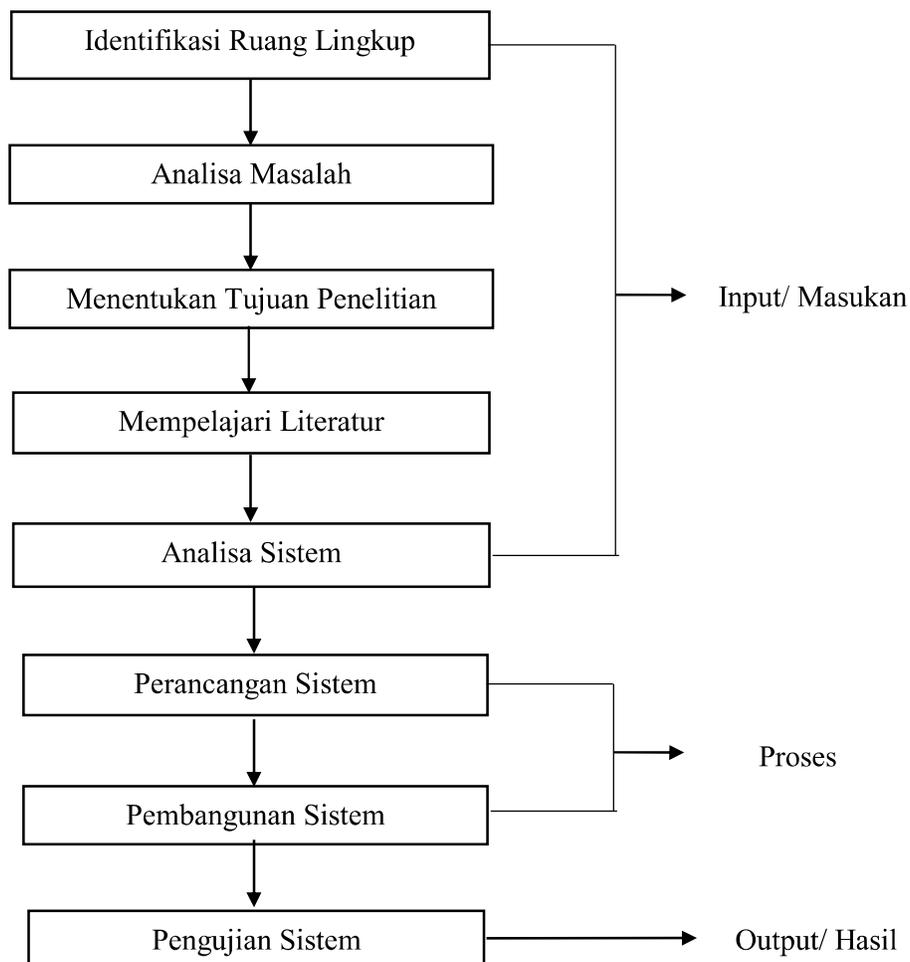
BAB III

METODE PENELITIAN

Metode penelitian adalah cara ilmiah untuk untuk mendapatkan data yang digunakan untuk tujuan dan kegunaan tertentu, seperti untuk penemuan, pengembangan, dan pembuktian. Data nantinya diperoleh melalui kriteria penelitian yang valid, yaitu untuk menunjukkan derajat ketetapan antara data yang sesungguhnya terjadi pada objek dengan data yang dapat dikumpulkan oleh penelitian.

3.1. Desain Penelitian

Desain penelitian bertujuan untuk melaksanakan penelitian sehingga dapat diperoleh suatu logika. Suatu desain penelitian adalah suatu rencana tentang cara dalam melakukan penelitian tersebut. Karena itu desain penelitian erat kaitannya dengan proses penelitian. Dalam penelitian ini menggunakan penelitian konklusif. Penelitian konklusif adalah penelitian yang didesain untuk membantu mengambil keputusan dalam menentukan, mengevaluasi dan memilih alternatif terbaik dalam memecahkan suatu masalah. Langkah-langkah penelitian yang akan dilakukan terlihat pada gambar 3.1 sebagai berikut:



Gambar 3. 1 Desain Penelitian
(Sumber: Data Penelitian, 2018)

Adapun langkah-langkah yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Identifikasi ruang lingkup

Identifikasi ruang lingkup adalah menentukan batasan-batasan dari masalah yang diteliti. Identifikasi ruang lingkup bertujuan untuk menjaga konsistensi dari

penelitian ini sehingga penelitian ini lebih terarah, sehingga tujuan dari penelitian yang diharapkan tercapai.

2. Analisa Masalah

Analisa masalah adalah proses pengenalan dan menganalisa masalah ataupun objek penelitian. Analisa masalah adalah salah satu proses penelitian yang boleh dikatakan paling penting dari proses lainnya. Pada tahap ini, penulis akan mencoba menganalisa masalah, mencari tahu hal-hal apa saja yang menjadi faktor penyebab timbulnya masalah tersebut. Merumuskan masalah menjadi bentuk pertanyaan-pertanyaan, yang kemudian menghasilkan solusi.

3. Menentukan tujuan penelitian

Menentukan tujuan penelitian merupakan hal yang penting. Dengan adanya tujuan maka penelitian akan menjadi terarah dan setiap proses didalam penelitian akan menjadi lebih mudah. Adapun tujuan penelitian akan memberikan dampak bagi masyarakat luas mengenai penelitian yang sedang dilakukan oleh penulis.

4. Mempelajari *literatur*

Pada tahap ini dilakukan pengumpulan data untuk lebih mengetahui masalah yang diteliti. Dari data yang diperoleh akan diketahui mengenai sistem yang akan dirancang. Dalam melakukan penelitian ini metode yang dilakukan dalam pengambilan data salah satunya adalah mempelajari *literatur*. Mempelajari *literatur* artinya mencari bahan-bahan pendukung berupa buku-buku dan jurnal yang berhubungan dengan objek atau masalah yang sedang diteliti.

5. Analisa Sistem

Analisa sistem merupakan fase pengembangan sistem yang menentukan sistem apa yang akan dibangun untuk memecahkan masalah yang sudah ada. Pada tahap ini peneliti akan melakukan analisa terhadap sistem yang akan dibangun akan dapat memecahkan masalah yang sudah ada.

6. Perancangan sistem

Pada tahap ini perancangan sistem pakar berdasarkan aturan-aturan yang telah dibuat dengan tujuan mendapatkan hasil terbaik didalam mendeteksi kerusakan berdasarkan gejala-gejala yang terjadi pada sepeda motor.

7. Pembangunan sistem

Pembangunan sistem merupakan proses membangun dan mengimplementasikan hasil penelitian kedalam sebuah aplikasi sistem pakar dengan menggunakan metode, alat, taktik, dan alat bantu pengembangan sistem tertentu.

8. Pengujian sistem

Pada tahapan ini aplikasi telah selesai dibuat/ dirancang, maka dibutuhkan sebuah pengujian terhadap kinerja baik dari segi ketepatan deteksi maupun cepat atau lambatnya respon dari aplikasi yang telah selesai dibuat. Pengujian berupa efektifitas dari aplikasi dan kecepatan serta ketepatan dari aplikasi sistem pakar.

3.2. Teknik Pengumpulan Data

Teknik pengumpulan data adalah cara yang digunakan peneliti untuk mendapatkan data yang berkaitan dengan pokok pembahasan dalam penelitian

yang sedang dilakukan. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah:

1. Wawancara

Menurut (Noor, 2011) wawancara adalah teknik pengumpulan data yang dilakukan dengan berhadapan secara langsung dengan orang atau pakar yang akan diwawancarai. Untuk mendapatkan data-data yang berkaitan dengan penelitian, peneliti melakukan wawancara langsung dengan Bapak Sriyono yang bekerja sebagai kepala mekanik yang khusus menangani permasalahan sepeda motor di bengkel PT.Roda Mas Makmur Motor. Pedoman wawancara yang digunakan berupa garis-garis besar permasalahan yang akan ditanyakan seperti gejala kerusakan motor, penyebab kerusakan motor dan solusi perbaikan motor.

2. Studi Literatur

Menurut (Noor, 2011) studi literatur diperoleh melalui mengumpulkan, membaca dan memahami referensi teoritis yang berasal dari buku-buku teori, buku elektronik (*e-book*), jurnal-jurnal penelitian, dan sumber pustaka lainnya. Sifat utama dari studi literatur adalah tak terbatas pada ruang dan waktu sehingga memberi peluang kepada peneliti untuk mengetahui hal-hal yang pernah terjadi di waktu yang lalu. Studi literatur bertujuan untuk menemukan variabel yang akan diteliti, membedakan hal-hal yang sudah dilakukan dan menentukan hal yang perlu dilakukan, melakukan sintesa dan memperoleh perspektif baru, dan menentukan makna dan hubungan antar variabel.

3.3. Operasional Variabel

Menurut Sudaryono (Sudaryono, 2015, p. 16), variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi dan kesimpulannya. Variabel harus didefinisikan secara operasional agar lebih mudah dicari hubungannya antara satu variabel dengan lainnya dan pengukurannya. Adapun manfaat operasional variabel antara lain untuk mengidentifikasi kriteria yang sedang didefinisikan. Dalam penelitian ini operasional variabel yang digunakan adalah kerusakan sepeda motor matic pada Suzuki Nex F1. Jenis kerusakan yang diteliti penulis adalah kerusakan yang umum terjadi pada sepeda motor matic. Terdapat tiga indikator yang disajikan pada tabel 3.1 sebagai berikut:

Tabel 3. 1 Operasional Variabel

Variabel	Kode Kerusakan	Indikator/ Kerusakan
Kerusakan Komponen Mesin Pada Sepeda Motor Suzuki Nex F1.	K01	Kerusakan pada piston
	K02	Kerusakan Pada Klep
	K03	Kerusakan Pada Karburator
	K04	Kerusakan Pada Busi
	K05	Kerusakan Pada Injektor

(Sumber : Data Penelitian, 2018)

3.4. Perancangan Sistem

Perancangan sistem yaitu kegiatan untuk mengkonstruksi sebuah sistem yang memberikan kepuasan akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performa maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat (A.S & M.Shalahuddin, 2011, p. 21).

3.4.1. Desain Basis Pengetahuan

Sumber pengetahuan dan fakta diperoleh melalui wawancara dengan teknisi motor dan studi literatur tentang materi yang berkaitan dengan motor matic. Sumber pengetahuan dan fakta yang didapat berupa data-data yang berhubungan dengan motor matic, penyebab kerusakan motor matic dan solusi mengatasinya. Pengetahuan dan fakta tersebut ditampilkan dalam tabel indikator, tabel penyebab dan solusi, tabel penyebab, tabel gejala dan tabel aturan.

Tabel 3. 2 Gejala dan Solusi

Kode Kerusakan	Kode Gejala	Gejala	Solusi
K01	G001	Mesin motor sulit hidup	1. Ganti Piston dengan yang baru, pastikan bahwa piston sesuai dengan spesifikasi motor.
	G002	Tenaga mesin kurang atau lemah	
	G003	Suara mesin kasar	
	G004	Gas motor tersendat-sendat	
K02	G001	Mesin motor sulit hidup	1. Membersihkan klep, <i>siting</i> dan bos klep menggunakan bensin. 2. Bila tidak terjadi perubahan, ganti klep tersebut dengan yang baru dan melakukan pemasangan dengan hati-
	G005	Motor mengeluarkan asap tebal berwarna hitam.	
	G006	Oli mesin cepat habis	
	G007	Bahan bakar motor menjadi boros.	
K03	G001	Mesin motor sulit hidup	1. Putar setelan sekrup udara searah jarum jam atau arah menutup,yaitu memperbanyak campuran udara dengan bahan bakar. 2. Kemudian bersihkan karburator dari kotoran yang menempel.
	G005	Motor mengeluarkan asap tebal berwarna hitam	
	G008	Knalpot menegeluarkan suara (knalpot nembak)	
	G009	Kecepatan motor sulit bertambah	
K04	G001	Mesin motor sulit hidup.	1. Bersihkan kerak-kerak hitam yang menempel pada bagian kepala busi. 2. Jika masih tidak ada percikan api dari busi, gantilah dengan busi yang baru.
	G002	Tenaga mesin kurang atau lemah	
	G003	Suara mesin kasar	
	G0010	Mesin mati mendadak ketika dijalan	
K05	G0011	Mesin sulit distarter	1. Mengecek apakah bensin sampai keruang bakar, 2. Cek bagian-bagian yang terhubung dengan <i>injector</i> .
	G0012	Keluar asap dari knalpot	

	G0013	Mesin sering mati mendadak	3. bila tidak ada perubahan gantilah <i>injector</i> dengan yang baru.
	G0014	Mesin mogok dan tidak mampu dihidupkan	

(Sumber: Data Penelitian, 2018)

Sistem pakar yang menggunakan metode *forward chaining* pada penelitian ini digunakan untuk mendeteksi kerusakan motor Suzuki Nex F1 sehingga data solusi tidak diberikan kode. Data solusi hanya sebagai keterangan tambahan yang digabungkan ke dalam tabel penyebab dan solusi (tabel 3.3). Dalam **Tabel 3.2** menjelaskan nama kode gejala, kode kerusakan, dan solusinya untuk memudahkan dalam pengolahan data.

Data aturan merupakan data yang berisi relasi antara data-data indikator, penyebab kerusakan dan gejala kerusakan yang telah diberi kode sebelumnya. Relasi antar data tersebut disusun berdasarkan sumber pengetahuan dan fakta yang telah didapatkan. Data aturan ini disusun untuk memudahkan peneliti dalam menyusun kaidah yang akan digunakan sebagai basis pengetahuan dalam sistem pakar pada penelitian ini. Susunan data aturan yang digunakan dapat dilihat pada tabel dibawah ini.

Tabel 3. 3 Aturan

Kode Kerusakan	Kode Gejala
K01	G001, G002, G003, G004
K02	G001, G005, G006, G007
K03	G001, G005, G008, G009
K04	G001, G002, G003, G0010
K05	G0011, G0012, G0013, G014

Sumber: Data Penelitian (2018)

Berdasarkan data aturan yang telah disusun, maka kaidah yang akan digunakan dalam sistem pakar dan tabel keputusannya adalah sebagai berikut:

1. Kaidah 1: *IF G001 AND G002 AND G003 AND G004 THEN K01*
2. Kaidah 2: *IF G001 AND G005 AND G006 AND G007 THEN K02*
3. Kaidah 3: *IF G001 AND G005 AND G008 AND G009 THEN K03*
4. Kaidah 4: *IF G001 AND G002 AND G003 AND G0010 THEN K04*
5. Kaidah 5: *IF G0011 AND G0012 AND G0013 AND G0014 THEN K05*

Berdasarkan kaidah (*rule*) tersebut, maka dapat dijelaskan bahwa :

1. Jika gejala kerusakan yang timbul adalah mesin motor sulit hidup, tenaga mesin kurang atau lemah, suara mesin kasar dan gas motor tersendat-sendat maka penyebabnya adalah kerusakan piston (Kerusakan pada komponen mesin).
2. Jika gejala kerusakan yang timbul adalah mesin motor sulit hidup, motor mengeluarkan asap tebal berwarna hitam, oli mesin cepat habis dan bahan bakar motor menjadi boros maka penyebabnya adalah kerusakan klep (Kerusakan pada komponen mesin).
3. Jika gejala kerusakan yang timbul adalah mesin motor sulit hidup, motor mengeluarkan asap tebal berwarna hitam, knalpot mengeluarkan suara (knalpot nembak) dan kecepatan motor sulit untuk bertambah boros maka penyebabnya adalah kerusakan karburator (Kerusakan pada komponen mesin).
4. Jika gejala kerusakan yang timbul adalah mesin motor sulit hidup, tenaga mesin kurang atau lemah, suara mesin kasar dan mesin mati mendadak ketika

dijalanmaka penyebabnya adalah kerusakan busi (Kerusakan pada sistem kelistrikan).

5. Jika gejala kerusakan yang timbul adalah bola lampu motor depan atau lampu sen atau lampu rem belakang tidak menyala maka penyebabnya adalah kerusakan bola lampu (Kerusakan pada sistem kelistrikan).
6. Jika gejala kerusakan yang timbul adalah bola lampu motor depan atau lampu sen atau lampu rem belakang tidak menyala, bola lampu bagus, tapi lampu tetap tidak hidup maka penyebabnya adalah kerusakan soket lampu (Kerusakan pada sistem kelistrikan).
7. Jika gejala kerusakan yang timbul adalah bola lampu motor depan atau lampu sen atau lampu rem belakang tidak menyala, bola lampu bagus, tapi lampu tetap tidak hidup dan stater motor tidak berfungsi maka penyebabnya adalah kerusakan baterai ((Kerusakan pada sistem kelistrikan).
8. Jika gejala kerusakan yang timbul adalah mesin sulit distater, keluar asap dari knalpot, dan mesin sering mati mendadak maka penyebabnya adalah kerusakan injektor (Kerusakan pada sistem injeksi).

Berdasarkan kaidah yang telah dibuat tersebut maka tabel keputusannya adalah berikut:

Tabel 3. 4 Tabel Keputusan

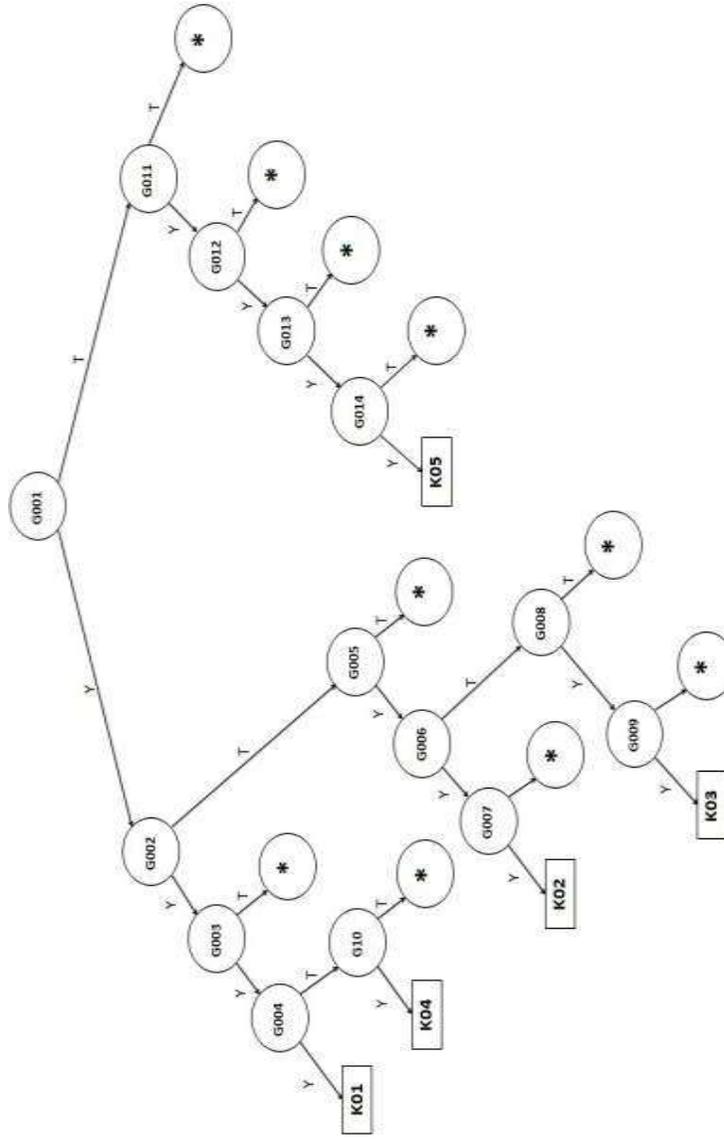
Penyebab Gejala	K01	K02	K03	K04	K05
G001	√	√	√	√	
G002	√		√	√	
G003	√			√	

G004	√				
G005		√	√		
G006		√			
G007		√			
G008			√		
G009			√		
G0010				√	
G0011					√
G0012					√
G0013					√
G0014					√

Sumber: Data Penelitian (2018)

Pada **Tabel 3.4** diatas, baris gejala diberi tanda centang untuk kolom kode penyebab yang memenuhi aturan dari masing-masing gejala. Hal ini dibuat untuk memudahkan dalam menyusun aturan kaidah produksi sistem pakar yang akan dibuat.

Setelah disusun tabel keputusan (**Tabel 3.4**) diatas maka dapat dibuat pohon keputusan berikut ini:



Gambar 3. 2 Pohon Keputusan
(Sumber: Data Penelitian, 2018)

Keterangan:

Y: Ya

T: Tidak

*: Tidak ada data /teridentifikasi

G001 – G0014: Keterangan lengkap pada **Tabel 3.2**

K01 – K05: Keterangan lengkap pada **Tabel 3.4**

Data gejala adalah keadaan awal dalam sistem saat melakukan penelusuran sebelum mendapatkan kesimpulan. Pohon keputusan pada gambar 3.2 digunakan untuk memperlihatkan hubungan terkait antar gejala yang ada. Arah penelusuran pada pohon keputusan tersebut dimulai dari simpul akar (yang paling atas) ke bawah. Alur penelusuran sistem pakar dimulai dari G001, yaitu mesin motor sulit hidup. Gejala ini dipilih sebagai keadaan awal dalam penelusuran karena gejala ini adalah gejala yang paling mudah diperiksa dan diketahui.

Proses penelusuran selanjutnya tergantung bagaimana jawaban yang diberikan pengguna. Jika pengguna memberikan jawaban “ya”, maka penelusuran menuju simpul kiri pada level berikutnya (G002) dan jika pengguna memberikan jawaban “tidak”, maka penelusuran menuju simpul kanan pada level berikutnya (G0011). Begitu seterusnya sampai penelusuran menemukan simpul K (K01, K02, dst) atau simpul *. Simpul * berarti kerusakan motor anda tidak ditemukan dan sistem pakar akan menampilkan pesan silahkan jumpai pakar yang lebih ahli atau bawa ke tempat service motor terdekat. Apabila ingin melakukan deteksi kembali dari keadaan awal, cukup dengan menekan tombol deteksi ulang.

3.4.2. Struktur Kontrol (Mesin Inferensi)

Struktur kontrol dalam sistem pakar kerusakan motor ini menggunakan metode *forward chaining*. Adapun Langkah-langkah yang digunakan dalam proses penelusurannya adalah:

1. Mengajukan pertanyaan tentang gejala kerusakan motor kepada pengguna.
2. Menyimpan sementara jawaban pengguna tentang gejala kerusakan motor dan kemungkinan penyebab kerusakan motor ke dalam memori sementara. (di dalam *knowledge base*)
3. Memeriksa gejala-gejala yang ada dengan aturan yang telah dibuat. Jika konklusi cocok maka simpan hasil ke dalam memori tetap (di dalam *knowledge base*) dan jika belum memenuhi konklusi, ulangi langkah 1 sampai dengan langkah 3. Jika semua pertanyaan telah diberikan namun belum memenuhi konklusi apapun, keluarkan pesan *default* atau *looping*.
4. Menampilkan hasil atau solusi dari deteksi kerusakan motor.

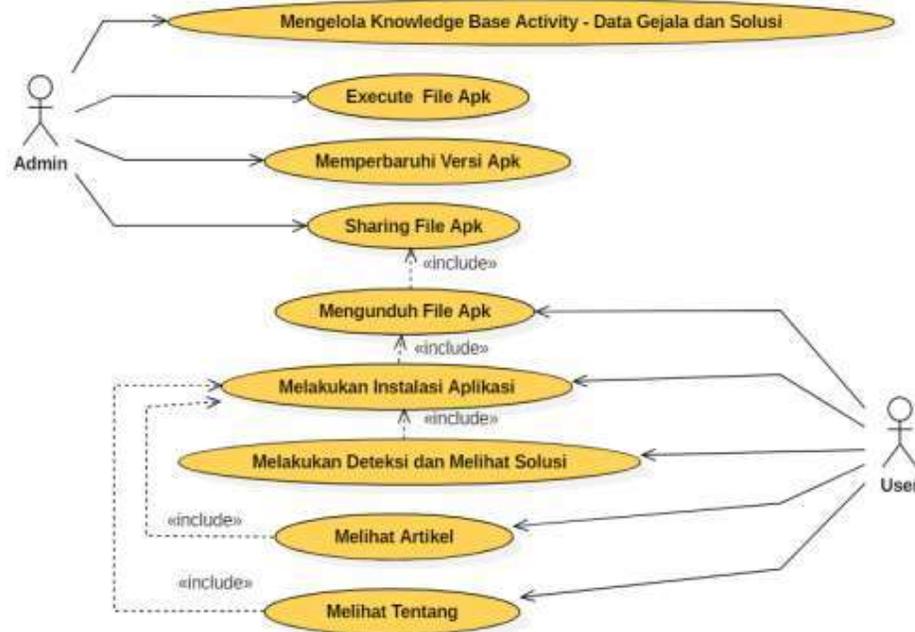
3.4.3. Desain UML (*Unified Modelling Language*)

Desain sistem pada penelitian ini menggunakan bahasa pemodelan *Unified Modelling Language (UML)* yang digambarkan dengan bantuan aplikasi *StarUML*. Diagram *UML* dalam penelitian ini yaitu:

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. Diagram ini mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang

akan dibuat (Rosa & M.Shalahuddin, 2013). *Use case diagram* yang dirancang untuk sistem pakar dalam penelitian ini dapat dilihat pada gambar dibawah ini.



Gambar 3.3 *Use case diagram*
(Sumber: Data Penelitian, 2018)

Terdapat 2 aktor yaitu admin dan user. Admin melakukan interaksi dengan sistem berupa mengelola knowledge base activity, execute file apk, memperbaruhi versi apk, sharing file apk. Sedangkan user berinteraksi dengan sistem yaitu mengunduh file apk, melakukan instalasi, melakukan deteksi kerusakan motor dan melihat hasil atau solusi dari deteksi, melihat artikel, melihat tentang.

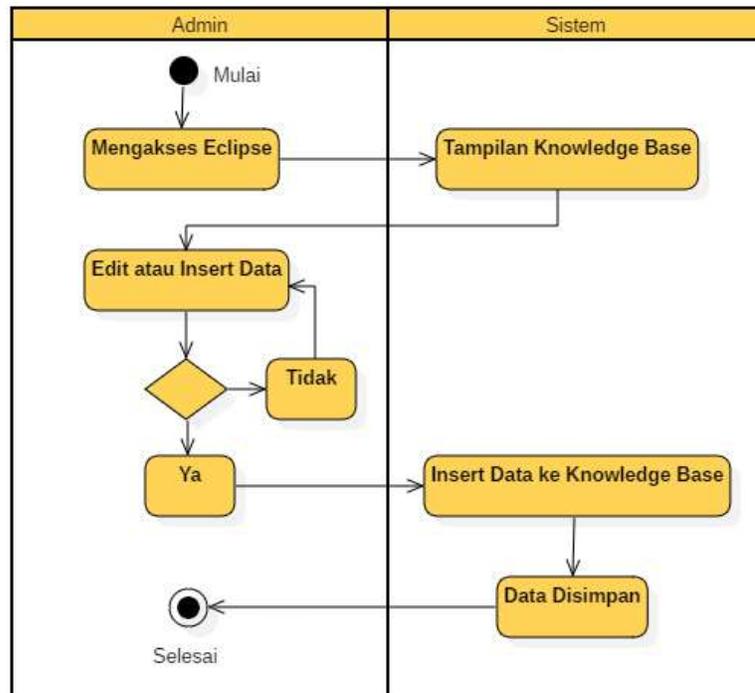
2. *Activity Diagram*

Activity diagram menggambarkan *workflow* dari sistem. *Activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor (Rosa & M.Shalahuddin, 2013). Adapun *Activity diagram* dalam penelitian ini adalah:

2.1 Admin

a. Activity Diagram Mengelola Knowledge Base

Activity diagram mengelola knowledge base merupakan UML yang menggambarkan kegiatan admin pada pengelolaan knowledge base.

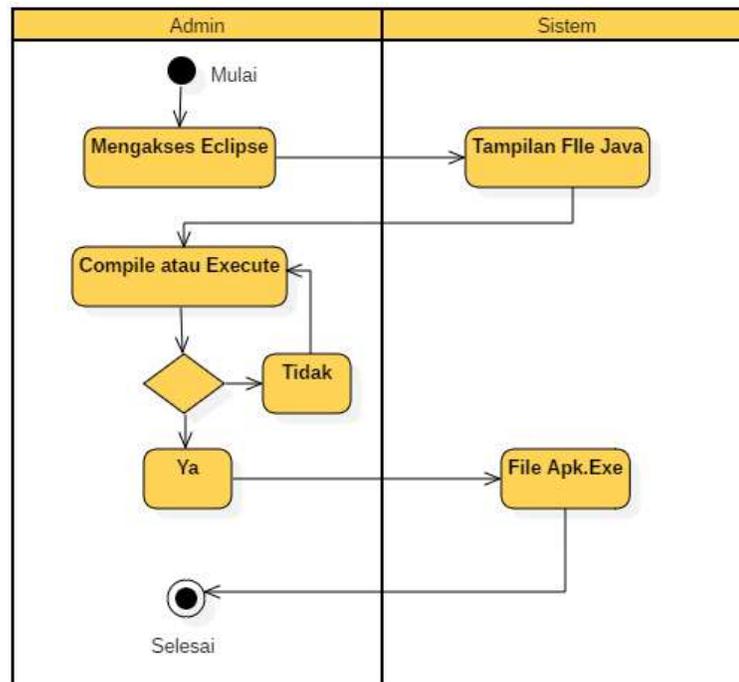


Gambar 3. 4 Activity Diagram Mengelola Knowledge Base
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.4** di atas, proses mengelola knowledge base yaitu admin mengakses aplikasi eclipse dan setelah aplikasi terbuka, admin dapat melakukan pengelolah knowledge base sistem.

b. Activity Diagram Execute FileApk

Activity diagram execute file apk merupakan UML yang menggambarkan kegiatan admin pada mengexecute file apk android.

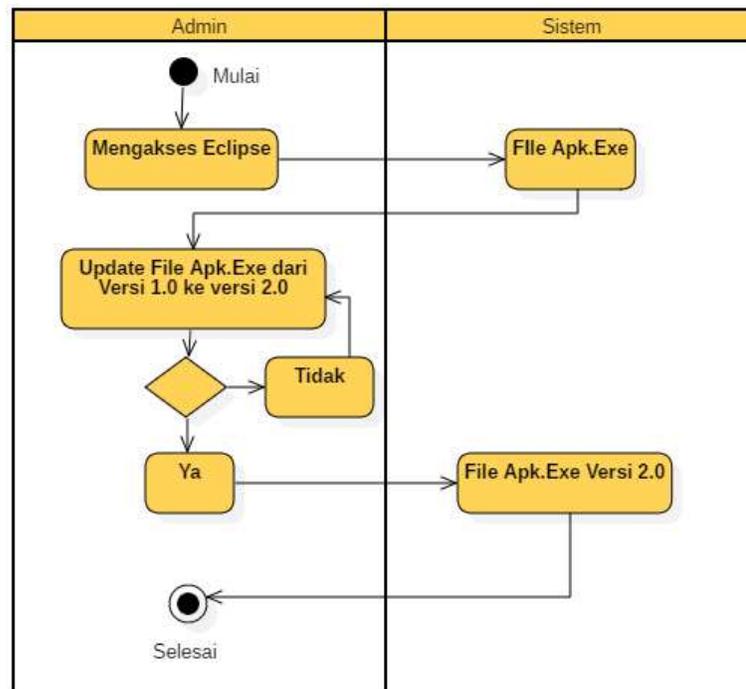


Gambar 3.5 Activity Diagram Execute FileApk
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.5** di atas, proses *execute file apk* yaitu admin mengakses aplikasi eclipse dan setelah aplikasi terbuka dan menampilkan *file java*, *admin* akan melakukan *compile atau execute file java* menjadi *file apk android*.

c. Activity Diagram Memperbarui Versi

Activity diagram memperbarui versi merupakan UML yang menggambarkan kegiatan *admin* pada memperbaruhi *versifileapkandroid*.

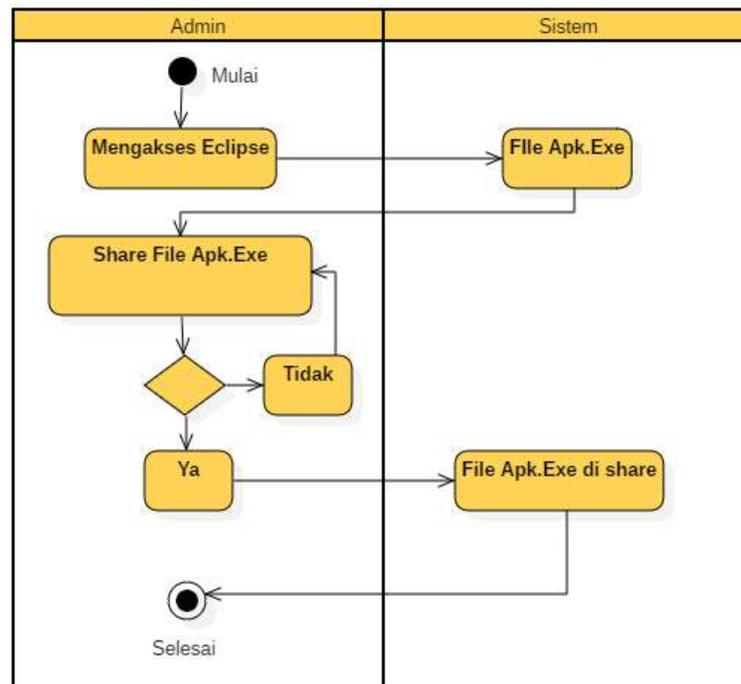


Gambar 3. 6 *Activity Diagram* Memperbarui Versi
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.6** di atas, proses memperbarui *versi* yaitu *admin* mengakses aplikasi eclipse dan setelah aplikasi terbuka akan menampilkan *file apk*, *admin* akan melakukan *update fileapk* menjadi *versi fileapk* yang terbaru.

d. *Activity Diagram Sharing File Apk*

Activity diagram sharing file apk merupakan *UML* yang menggambarkan kegiatan *admin* pada *mensharing file apk android*.



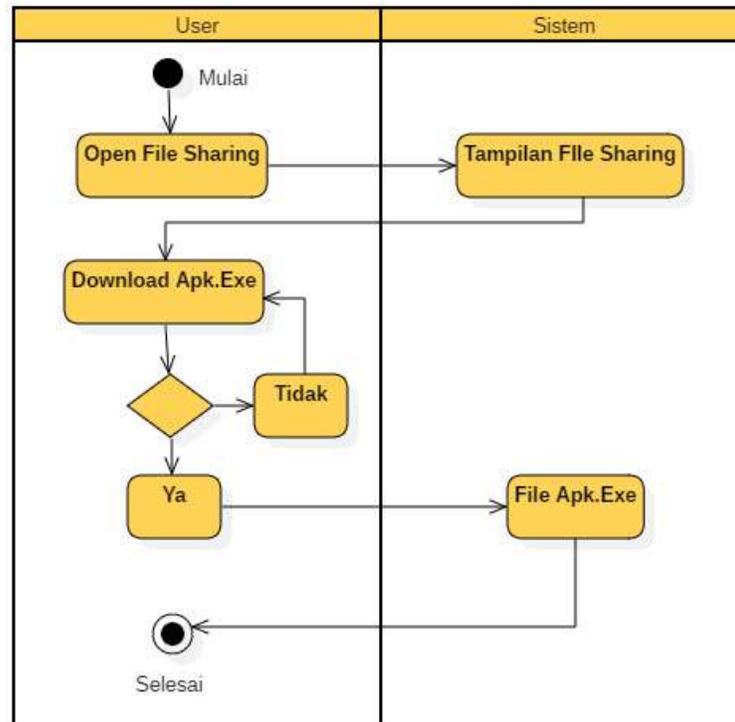
Gambar 3.7 Activity Diagram Sharing FileApk
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.7** di atas, proses *sharing file apk* yaitu *admin* mengakses aplikasi eclipse dan setelah aplikasi terbuka akan menampilkan *file apk*, *admin* akan melakukan *sharing file apk* kepada pengguna.

2.2 User

a. Activity Diagram Mengunduh File.Apk

Activity diagram mengunduh *file apk* merupakan *UML* yang menggambarkan kegiatan *user* pada mengunduh *file apk android*.

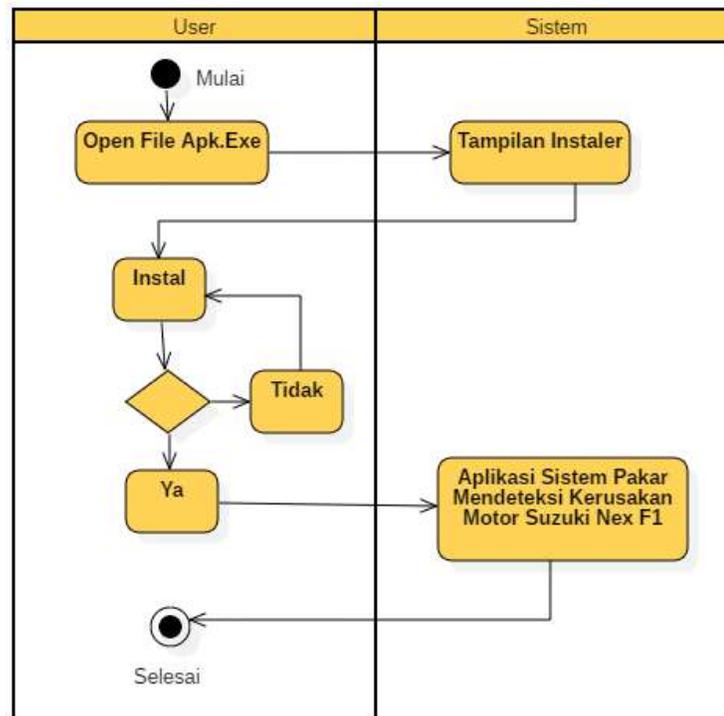


Gambar 3. 8 *Activity Diagram* Mengunduh FileApk
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.8** di atas, proses mengunduh *file apk* yaitu user *open file sharing* lalu *mendownload file apk.exe*.

b. *Activity Diagram* Melakukan *Instalasi*

Activity diagram melakukan *instalasi* merupakan *UML* yang menggambarkan kegiatan *user* melakukan *instalasi file apk.exe*.

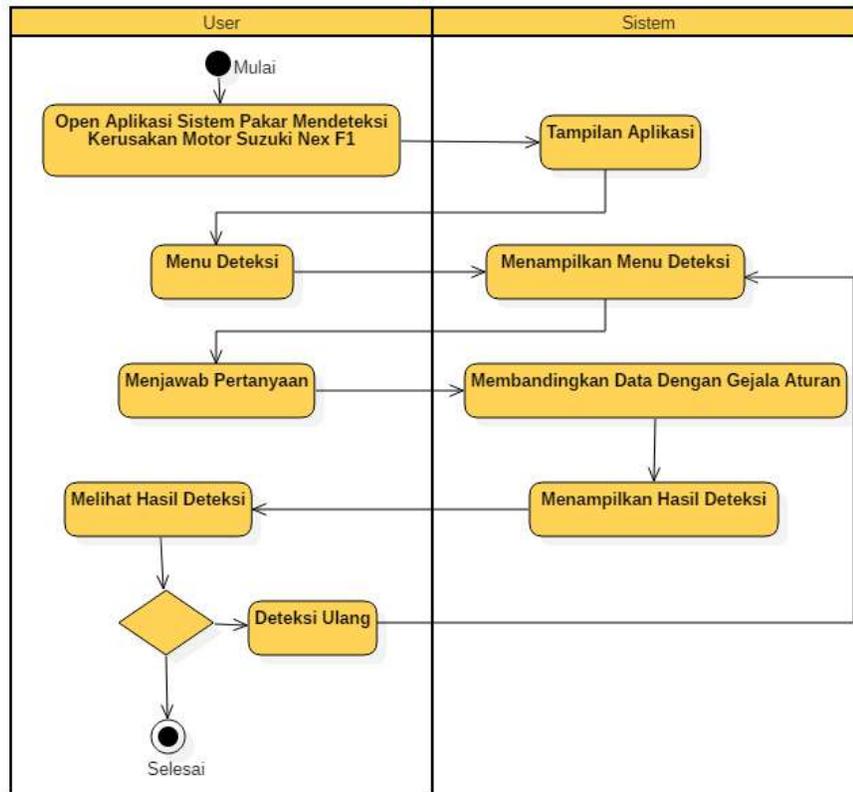


Gambar 3.9 *Activity Diagram* Melakukan Instalasi
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.9** di atas, proses menginstal *apk* yaitu *user* akan *open file apk.exe* yang telah *didownload* lalu melakukan *intalasi file apk.exe* sehingga sistem pakar mendeteksi kerusakan motor Suzuki nex F1 *terinstal* di Hp Android.

c. *Activity Diagram* Melakukan Deteksi dan Melihat Solusi

Activity diagram melakukan deteksi merupakan *UML* yang menggambarkan kegiatan *user* melakukan deteksi kerusakan motor.

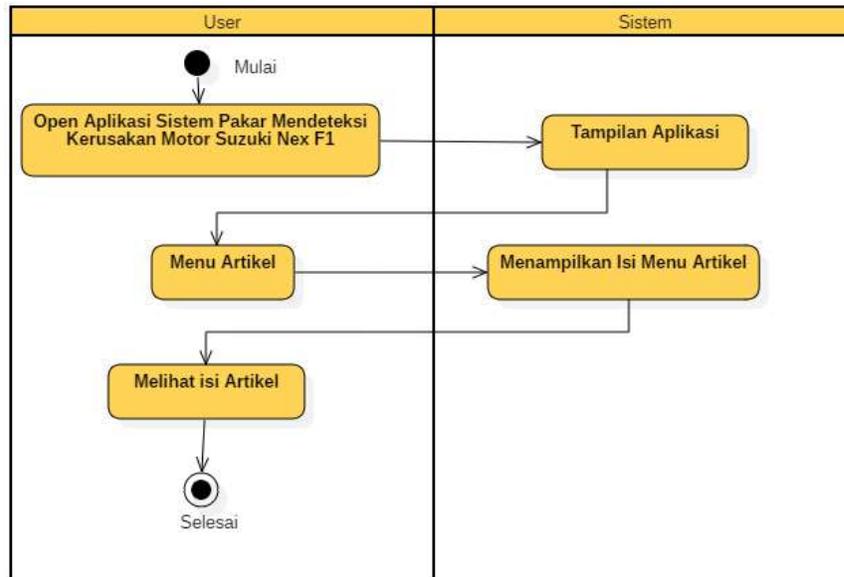


Gambar 3. 10 Activity Diagram Melakukan Deteksi
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.10** di atas, proses deteksi yaitu *user* akan *open file* sistem pakar yang sudah di *instal* lalu *user* akan memilih menu deteksi, pada menu deteksi *user* akan menjawab pertanyaan gejala-gejala motor yang timbul dan diakhirnya akan menampilkan hasil deteksi.

i. Activity Diagram Melihat Menu Artikel

Activity diagram melihat artikel merupakan UML yang menggambarkan kegiatan *user* melihat artikel motor.

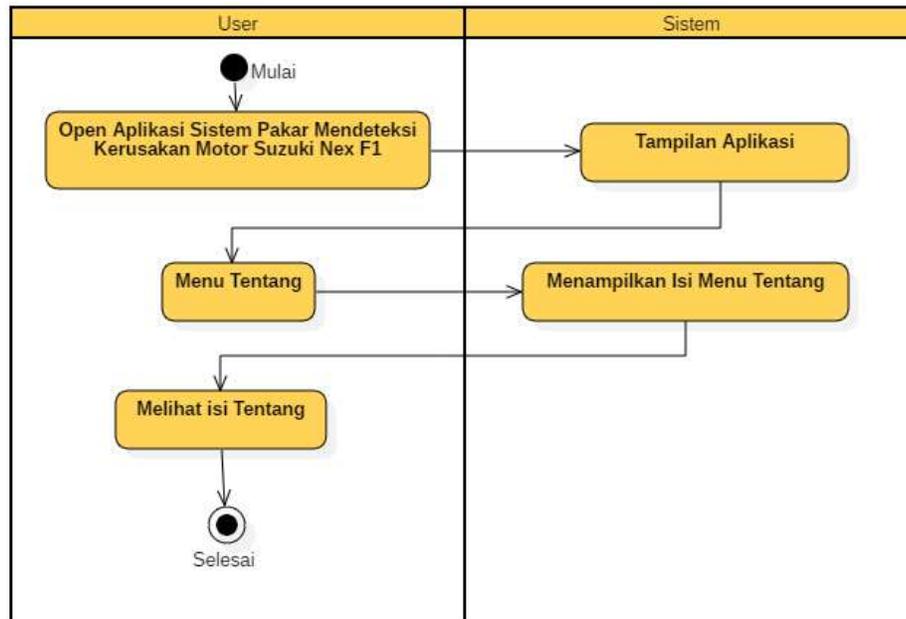


Gambar 3.11 *Activity Diagram* Melihat Menu Artikel
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.11** di atas, proses melihat artikel yaitu *user* akan *open file* sistem pakar yang sudah di *instal* lalu *user* akan memilih menu artikel, pada menu artikel *user* akan melihat artikel-artikel mengenai kerusakan motor.

ii. *Activity Diagram* Melihat Menu Tentang

Activity diagram melihat menu tentang merupakan *UML* yang menggambarkan kegiatan user melihat tentang sistem pakar.



Gambar 3.12 *Activity Diagram* Melihat Menu Tentang
(Sumber: Data Penelitian, 2018)

Pada **Gambar 3.12** di atas, proses melihat tentang yaitu *user* akan *open file* sistem pakar yang sudah di *instal* lalu *user* akan memilih menu tentang, pada menu tentang *user* akan melihat *versi* dari pada sistem pakar tersebut.

Pada **Gambar 3.13** di atas, proses melihat menu profil yaitu *user* akan *open file* sistem pakar yang sudah di *instal* lalu *user* akan memilih menu profil, pada menu profil *user* akan melihat profil pembuat sistem pakar tersebut.

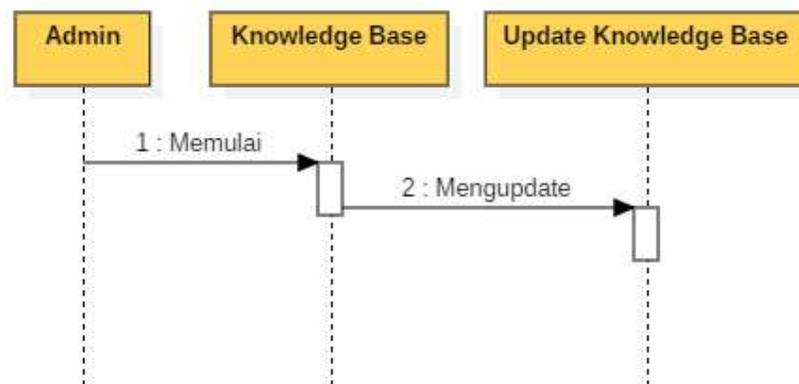
3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Rosa & M.Shalahuddin, 2013). Adapun *sequence diagram* dalam penelitian ini adalah.

1. Admin

1.1 *Sequence diagram* mengelola *knowledge base*

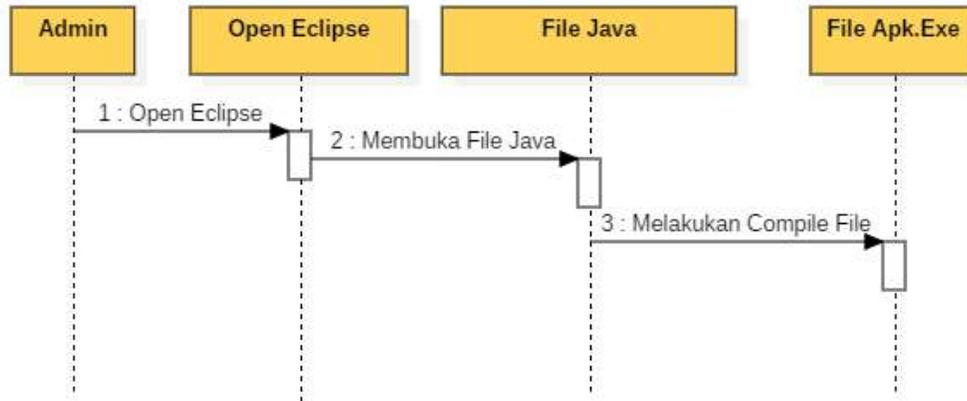
Sequence diagram mengelola *knowledge base* merupakan urutan waktu kegiatan admin saat mengelola *knowledge base*. Berikut ini gambar *sequence diagram* mengelola *knowledge base* (**Gambar 3.14**):



Gambar 3. 13 *Sequence diagram* mengelola *Knowledge base*
(Sumber: Data Penelitian, 2018)

1.2 *Sequence Diagram* Execute File Apk

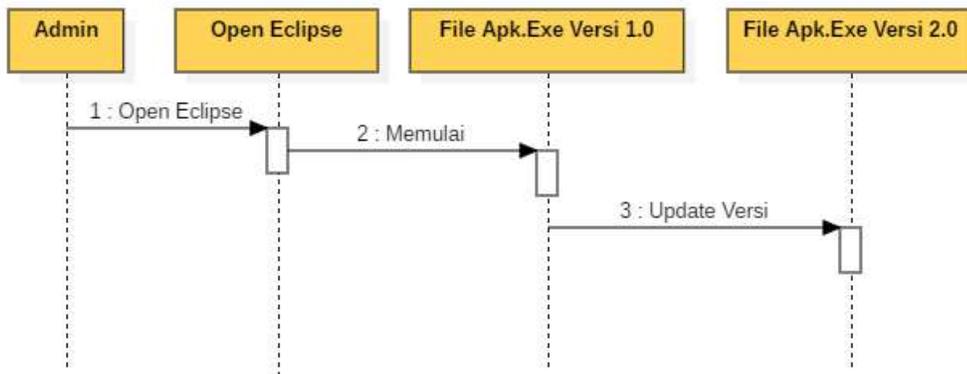
Sequence diagram execute file Apk merupakan urutan waktu kegiatan *admin* saat mengexecute file Apk. Berikut ini gambar *sequence diagram* execute file Apk (**Gambar 3.15**):



Gambar 3. 14 *Sequence Diagram Execute File Apk*
(Sumber: Data Penelitian, 2018)

1.3 *Sequence Diagram Memperbarui Versi*

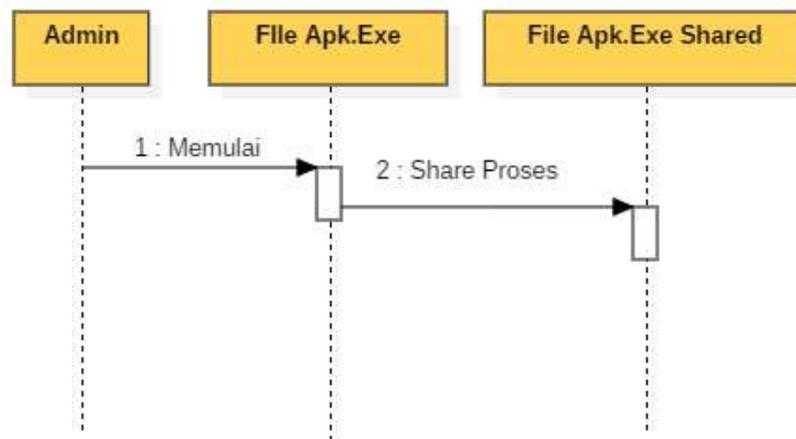
Sequence diagram memperbarui versi merupakan urutan waktu kegiatan *admin* saat memperbarui *versi*. Berikut ini gambar *sequence diagram* memperbarui *versi* (**Gambar 3.16**):



Gambar 3. 15 *Sequence Diagram memperbarui Versi*
(Sumber: Data Penelitian, 2018)

1.4 Sequence Diagram Sharing File Apk

Sequence diagram sharing file apk merupakan urutan waktu kegiatan *admin* saat *mensharing file apk*. Berikut ini gambar *sequence diagram sharing file apk* (**Gambar 3.17**):

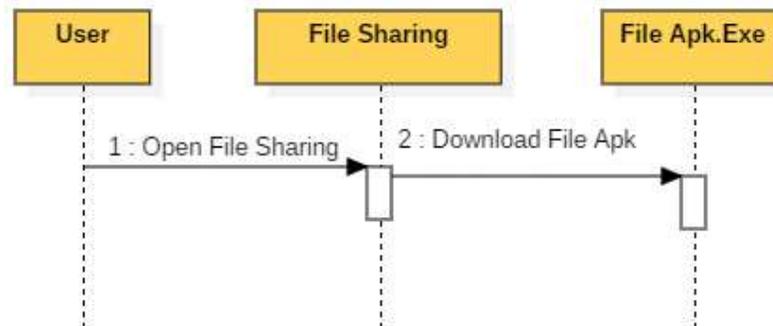


Gambar 3. 16 *Sequence diagram sharing file apk*
(Sumber: Data Penelitian, 2018)

2. User

2.1 *Sequence Diagram Mengunduh File Apk*

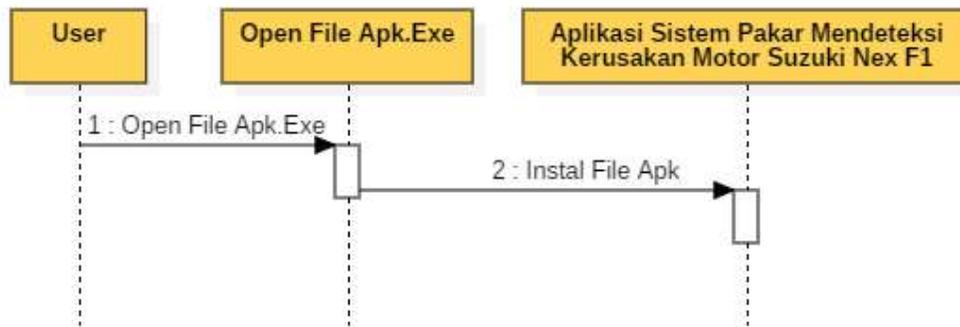
Sequence diagram Mengunduh File Apk merupakan urutan waktu kegiatan pengguna (*user*) saat *Mengunduh File Apk*. Berikut ini gambar *sequence diagram Mengunduh File Apk* (**Gambar 3.18**):



Gambar 3. 17 *Sequence Diagram* Mengunduh File Apk
(Sumber: Data Penelitian, 2018)

2.2 *Sequence Diagram* Melakukan Instalasi

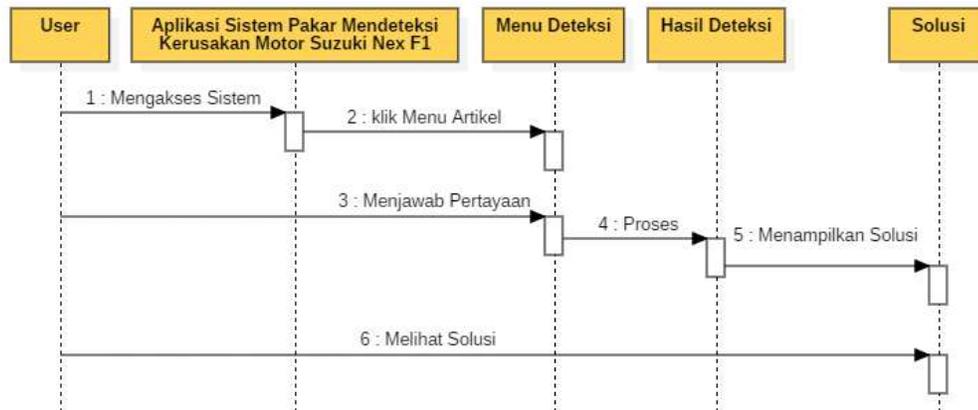
Sequence diagram Melakukan *Instalasi* merupakan urutan waktu kegiatan pengguna (*user*) saat melakukan *Instalasi file apk*. Berikut ini gambar *sequence diagram* Melakukan *Instalasi* (**Gambar 3.19**):



Gambar 3. 18 *Sequence Diagram* Melakukan Instalasi
(Sumber: Data Penelitian, 2018)

2.3 *Sequence Diagram* Melakukan Deteksi dan Melihat Solusi

Melakukan Deteksi merupakan urutan waktu kegiatan pengguna (*user*) saat Melakukan Deteksi kerusakan motor dan melihat hasil dari deteksi kerusakan motor. Berikut ini gambar *sequence diagram* Melakukan Deteksi (**Gambar 3.20**):



Gambar 3. 19 *Sequence Diagram* Melakukan Deteksi
(Sumber: Data Penelitian, 2018)

2.3 *Sequence Diagram* Melihat Menu Artikel

Sequence diagram melihat menu artikel merupakan urutan waktu kegiatan pengguna (*user*) saat melihat menu *artikel*. Berikut ini gambar *sequence diagram* melihat menu *artikel* (**Gambar 3.21**):



Gambar 3. 20 *Sequence Diagram* Melihat Menu Artikel
(Sumber: Data Penelitian, 2018)

2.3 Sequence Diagram Melihat Menu Tentang

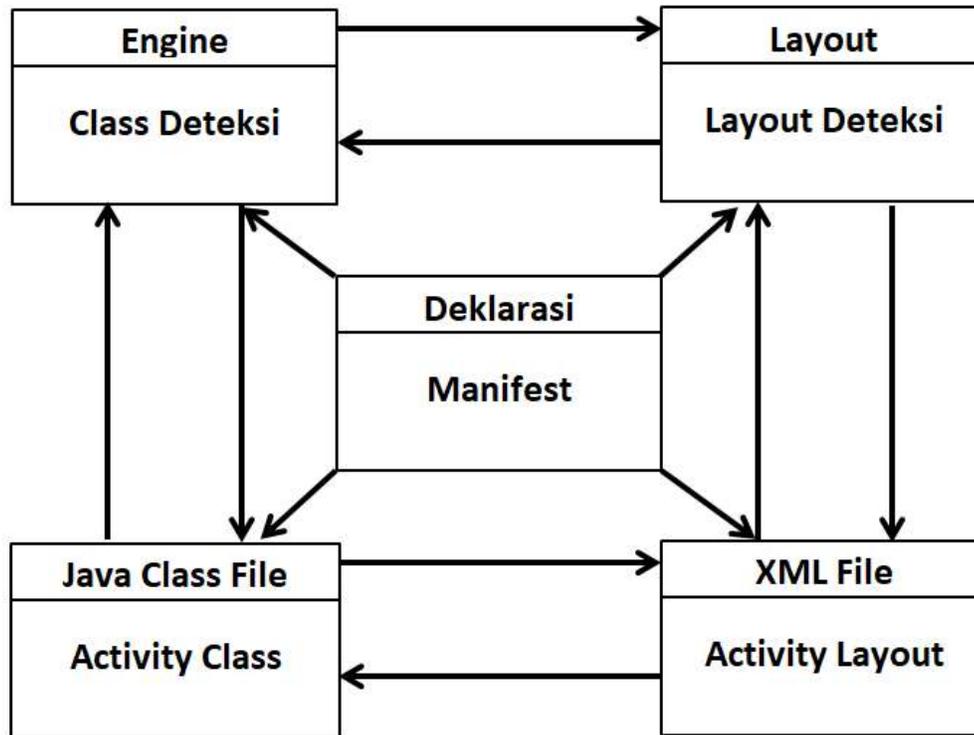
Sequence diagram melihat menu tentang merupakan urutan waktu kegiatan pengguna (*user*) saat melihat menu tentang. Berikut ini gambar *sequence diagram* melihat *menu tentang* (**Gambar 3.22**):



Gambar 3. 21 *Sequence Diagram* Melihat Menu Tentang
(Sumber: Data Penelitian, 2018)

3.4.4. Knowledge Base

Dalam penelitian ini, peneliti membuat desain *database* seperti berikut:



Gambar 3. 22 Desain Knowledge Base
(Sumber: Data Penelitian, 2018)

Tabel yang digunakan dalam sistem pakar ini terdiri dari 5 tabel, yaitu tabel *Class Deteksi* sebagai *Engine*, *Layout Deteksi* sebagai tampilan, *Manifest* sebagai Deklarasi, *Activity Class* sebagai *Java Class File* dan *Activity Layout* sebagai *XML File*. Awalnya *Manifest* akan memperkenalkan atau mendeklarasikan *Class Java* kemudian *XML File* sebagai *Layout* dihubungkan ke *Main Activity*. Lalu *Main Activity* dihubungkan kembali ke *Class Java* lainnya disertai dengan *XML File*. Semua tabel saling berhubungan satu dengan yang

lainnya untuk menghasilkan aplikasi sistem pakar mendeteksi kerusakan motor Suzuki Nex F1.

3.4.5. Desain Antarmuka

Adapun desain tampilan sistem pakar mendeteksi kerusakan motor Suzuki Nex F1 adalah:

1. Rancangan *Form* Beranda

Form beranda memiliki beberapa tampilan, yaitu, *header*, logo, judul *content*, menu utama dan *text area*.



Gambar 3. 23 Rancangan Form Beranda
(Sumber: Data Penelitian, 2018)

2. Rancangan *Form* Deteksi

Form deteksi digunakan oleh pengguna untuk mendeteksi kerusakan dengan sistem pakar. Sistem akan mengajukan beberapa pertanyaan tentang gejala-gejala kerusakan yang dialami oleh pengguna.

The diagram shows a rectangular form with a header section at the top. The header contains a box labeled 'Logo' on the left and the text 'Header' on the right. Below the header is a large central area containing the text 'Pertanyaan :'. Underneath this text are two buttons: 'Ya' on the left and 'Tidak' on the right. At the bottom left of the form, there is a label 'Created By'.

Gambar 3. 24 Rancangan Form Deteksi
(Sumber: Data Penelitian, 2018)

3. Rancangan *Form* Hasil Deteksi

Form hasil deteksi digunakan untuk menampilkan hasil deteksi yang berisi penyebab kerusakan dan solusi yang diberikan oleh sistem pakar.



The diagram shows a rectangular form layout. At the top, there is a header section divided into two parts: a box labeled 'Logo' on the left and the text 'Header' on the right. Below the header, the main content area contains the text 'Penyebab :', followed by 'Solusi :'. In the lower right portion of this area, there are two stacked rectangular buttons: the top one is labeled 'Deteksi Ulang' and the bottom one is labeled 'Menu Awal'. At the bottom left corner of the form, the text 'Created By' is present.

Gambar 3. 25 Rancangan Form Hasil Deteksi
(Sumber: Data Penelitian, 2018)

4. Rancangan *Form* Artikel

Form Artikel berisi tentang kumpulan artikel-artikel yang berguna bagi pengguna sebagai tambahan informasi dengan hal-hal yang berhubungan dengan motor Suzuki Nex F1.

Logo	Header
Judul Artikel :	
Content Artikel :	
Judul Artikel :	
Content Artikel :	
Judul Artikel :	
Content Artikel :	
Menu Awal	
Created By	

Gambar 3. 26 Rancangan Form Artikel
(Sumber: Data Penelitian, 2018)

5. Rancangan *Form* Tentang

Form Tentang berisi *versi* dari sistem pakar deteksi kerusakan motor dan terdapat keterangan bahwa setiap *knowledge base* diperbarui maka versi akan berubah seperti dari Versi 1.0 ke Versi 2.0 dan seterusnya.



The image shows a wireframe of a form layout. At the top, there is a header section containing a 'Logo' box on the left and the word 'Header' on the right. Below the header, the main content area contains the following text elements from top to bottom: 'Versi Sistem Pakar', 'Keterangan:', 'Pemilik Sistem', and 'Profil Pakar'. At the bottom of the main content area, there is a 'Menu Awal' button box. Below the button box, the text 'Created By' is positioned.

Gambar 3. 27 Rancangan Form Tentang
(Sumber: Data Penelitian, 2018)

6. Rancangan *Form Not Found*

Form Not Found berisi versi sistem pakar dimana ketika deteksi tidak ditemukan maka akan muncul *form not found* di dalam aplikasi dan pengguna harus kembali ke menu awal.



The image shows a wireframe for a 'Not Found' form. It consists of a rectangular frame divided into several sections. At the top left, there is a box labeled 'Logo'. To its right, the word 'Header' is centered. Below the logo and header, the text 'Tidak Terdeteksi' is centered. At the bottom left, there is a box labeled 'Menu Awal'. Below this box, the text 'Created By' is centered.

Gambar 3. 28 Rancangan Form Not Found
Sumber : Data penelitian, 2018

3.5. Lokasi dan Jadwal penelitian

Setiap rancangan penelitian perlu dilengkapi dengan jadwal kegiatan yang akan dilaksanakan yang berisi jadwal kegiatan apa saja yang akan dilakukan selama penelitian.

3.5.1. Lokasi

Penelitian ini dilakukan di Dealer Motor Suzuki PT.Roda Mas Makmur Motor yang beralamat di Komp Mitra Raya Blok B1.No.8-9 Batu Aji, Batam Alasan peneliti memilih perusahaan ini sebagai lokasi penelitian adalah:

1. Ketersediaan data untuk melakukan penelitian.
2. Mudah mendapatkan data yang dibutuhkan.
3. Efisiensi biaya dan waktu.

3.5.2. Jadwal Penelitian

Tabel 3. 5 Jadwal Penelitian

No	Kegiatan	Tahun 2017/2018																	
		April				Mei				Juni				Juli				Agus	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	
1	Pengajuan Judul	■	■																
2	Penyusunan Bab I		■	■	■	■													
3	Penyusunan Bab II					■	■	■	■										
4	Penyusunan Bab III									■	■	■	■						
5	Penyusunan Bab IV													■	■	■	■		
6	Penyusunan Bab V, Daftar Pustaka, Lampiran																	■	■

Sumber: Data Penelitian (2018)