

**SISTEM PAKAR DIAGNOSA KERUSAKAN SEPEDA
MOTOR YAMAHA NMAX**

SKRIPSI



**Oleh:
Yulius Andi Setiawan
130210170**

**FAKULTAS TEKNIK DAN KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
TAHUN 2018**

**SISTEM PAKAR DIAGNOSA KERUSAKAN SEPEDA
MOTOR YAMAHA NMAX**

SKRIPSI
Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana



Oleh
Yulius Andi Setiawan
130210170

FAKULTAS TEKNIK DAN KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
TAHUN 2018

PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, dan/atau magister), 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 telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam, 8 Februari 2018

Yang membuat pernyataan,

Materai

Yulius Andi Setiawan
NPM: 130210170

**SISTEM PAKAR DIAGNOSA KERUSAKAN SEPEDA MOTOR
YAMAHA NMAX**

Oleh
Yulius Andi Setiawan
130210170

SKRIPSI
Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana

Telah disetujui oleh Pembimbing pada tanggal
seperti tertera di bawah ini

Batam, 8 Februari 2018

Sunarsan Sitohang, S.Kom., M.TI.
Pembimbing

ABSTRAK

Pada saat ini, alat transportasi sudah jelas menjadi kebutuhan yang amat mendasar. Sudah banyak orang-orang menggunakan alat transportasi untuk melakukan aktivitasnya sehari-hari, *mobilitas* hampir tidak mungkin dilakukan jika tidak menggunakan alat transportasi. Berbicara alat transportasi, alat transportasi darat lah yang sering dijadikan pilihan masyarakat untuk mobilisasi. Apalagi untuk motor Yamaha Nmax yang notabenehnya sangat memudahkan pengendaranya untuk melintasi jalan-jalan yang sering terjadi kemacetan. Kerusakan pada motor Yamaha Nmax terjadi akibat kelalaian dalam melakukan perawatan. Pemilik motor biasanya baru menyadari kerusakan setelah motor mereka tidak dapat beroperasi sebagaimana mestinya. Oleh karena itu dalam penggunaan motor kemungkinan besar membutuhkan perawatan berkala. Hal inilah yang mendorong pembangunan sistem pakar untuk mengidentifikasi kerusakan motor Yamaha Nmax. Sistem Pakar adalah salah satu bagian dari kecerdasan buatan yang mengandung pengetahuan dan pengalaman yang dimasukkan oleh satu banyak pakar ke dalam suatu area pengetahuan tertentu sehingga setiap orang dapat menggunakannya untuk memecahkan berbagai masalah yang bersifat spesifik dalam hal ini adalah permasalahan pada diagnosis kerusakan pada motor Yamaha Nmax. Sistem Pakar Diagnosa Kerusakan Motor Yamaha Nmax ini menggunakan metode *forward chaining*, yang bertujuan menelusuri gejala yang ditampilkan dalam bentuk pertanyaan-pertanyaan agar dapat mendiagnosa gejala kerusakan motor Yamaha Nmax yang berbasis pada *web*. *Forward chaining* merupakan pendekatan yang baik untuk masalah tertentu seperti perencanaan, pengawasan dan pengaturan. Hasil pengujian sistem menunjukkan bahwa sistem pakar ini berfungsi dengan cukup baik dengan tingkat keakuratan mencapai 86%. Sistem pakar kerusakan sepeda motor Yamaha Nmax dengan metode *forward chaining* berjalan dengan cukup baik dan akurat sehingga dapat digunakan mekanik untuk membantu menganalisa dalam mendeteksi kerusakan sepeda motor Yamaha Nmax di bengkel Marfen Utama Motor , Tanjung Piayu, Batam.

Kata kunci: Sistem pakar, *forward chaining*, *web*, Yamaha Nmax

ABSTRACT

At this time, transportation is obviously a very basic necessity. Already many people use the means of transportation to perform their daily activities, mobility is almost impossible if not using the means of transportation. Speaking of means of transportation, land transportation is often used as a community choice for mobilization. Especially for the Yamaha Nmax motor that notabenenya greatly facilitate the rider to cross the streets that often occur congestion. Damage to Yamaha Nmax motor due to negligence in doing maintenance. Motor owners usually only realize the damage after their motor can not operate properly. Therefore, in the use of motors most likely require regular maintenance. This is what prompted the development of expert systems to identify damage to Yamaha Nmax motor. Expert System is one part of artificial intelligence that contains knowledge and experience inserted by one expert in a certain area of knowledge so that everyone can use it to solve various problems that are specific in this case is the problem on the diagnosis of damage on the motor Yamaha Nmax . Expert System Diagnosis of Damage Yamaha Motor Nmax is using forward chaining method, which aims to trace the symptoms displayed in the form of questions in order to diagnose symptoms of damage to Yamaha Nmax motor based on the web. Forward chaining is a good approach to certain issues such as planning, supervision and regulation. System test results show that this expert system works pretty well with a level of accuracy of 86%. Yamaha Nmax motorcycle damage expert system with forward chaining method is running well enough and accurate so it can be used mechanically to help analyze in detecting damage of motorcycle Yamaha Nmax at workshop of Marfen Utama Motor, Tanjung Piayu, Batam.

Keywords: Expert system, forward chaining, web, Yamaha Nmax

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran yang membangun akan senantiasa penulis terima dengan senang hati. Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam.
2. Ketua Program Studi Teknik Informatika Universitas Putera Batam Bapak Andi Maslan, S.T., M.SI.
3. Sunarsan Sitohang, S.Kom., M.TI. selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam.
5. Bapak Gunawan selaku nara sumber penelitian.
6. Istriku Roro Kinasih yang terus memberikan dukungan.
7. Kedua orangtua penulis, Bapak dan Ibu Djumono yang selalu menyemangati.

8. Seluruh teman-teman seperjuangan selama kuliah yang telah bersedia memberikan saran, masukan dan kritikan yang membangun demi lancarnya penyusunan skripsi ini.

Semoga Tuhan Yang Maha Esa membalas setiap kebaikan dengan berkat damai sejahtera yang berkelimpahan, Amin.

Batam, 8 Februari 2018

Yulius Andi Setiawan
NPM: 130210170

DAFTAR ISI

HALAMAN SAMPUL DEPAN
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN	iii
HALAMAN PENGESAHAN	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Penelitian	1
1.2 Identifikasi Masalah	3
1.3 Pembatasan Masalah	4
1.4 Perumusan Masalah	5
1.5 Tujuan Penelitian	5
1.6 Manfaat Penelitian	5
1.6.1 Manfaat Teoritis	5
1.6.2 Manfaat Praktis	6
BAB II TINJAUAN PUSTAKA	7
2.1 Teori Dasar.....	7
2.1.1 Kecerdasan Buatan (<i>Artificial Intelligence</i>).....	7
2.1.2 Logika <i>Fuzzy</i> (<i>fuzzy logic</i>)	10
2.1.3 Jaringan Syaraf Tiruan (<i>artificial neural network</i>)	11
2.1.4 Sistem Pakar.....	14
2.1.4.1 Komponen Sistem Pakar	19
2.1.4.2 <i>Rule</i> Sebagai Teknik Representasi Pengetahuan.....	23
2.1.4.3 Mesin Inferensi.....	28
2.1.5 Pelacak Maju (<i>Forward Chaining</i>).....	30

2.1.6 Basis Data (<i>Database</i>)	32
2.1.7 UML (<i>Unified Modeling Language</i>).....	33
2.1.8 Bahasa Pemrograman <i>Web</i>	41
2.2 Variabel Penelitian	46
2.2.1 Sepeda Motor	47
2.2.2 Mengenal Sepeda Motor Matic	47
2.2.3 Klasifikasi Motor	50
2.2.3.1 Klasifikasi Motor Berdasarkan Sistem Penyalaan	50
2.2.3.2 Klasifikasi Motor Berdasarkan Susunan Silinder	52
2.2.3.3 Klasifikasi Motor Berdasarkan Lokasi Katup.....	53
2.2.3.4 Klasifikasi Motor Berdasarkan Proses Pasokan Udara	53
2.2.3.5 Klasifikasi Motor Berdasarkan Sistem Pendingin	55
2.2.3.6 Klasifikasi Motor Berdasarkan Siklus Operasi / Langkah kerja.....	55
2.2.4 Ruang Bakar dan Pengapian	59
2.2.4.1 Ruang Bakar.....	59
2.2.4.2 Sistem pengapian	61
2.2.5 Sistem Transmisi.....	63
2.2.6 Klep dan <i>Filter</i>	64
2.2.6.1 Klep.....	64
2.2.6.2 Filter	64
2.2.7 Kelistrikan	64
2.3 <i>Software</i> Pendukung	66
2.3.1 StarUML	66
2.3.2 <i>XAMPP</i>	68
2.3.3 Sublime Text	69
2.3.4 phpMyAdmin	70
2.3.5 Mozilla Firefox.....	70
2.3.7 Mendeley.....	71
2.4 Penelitian Terdahulu	72
2.5 Kerangka Pemikiran.....	79
BAB III METODE PENELITIAN	81
3.1 Desain Penelitian.....	82
3.2 Teknik Pengumpulan Data.....	86
3.3 Operasional Variabel.....	87

3.4 Perancangan Sistem	91
3.4.1 Desain Basis Pengetahuan	91
3.4.2 Pembentukan Aturan.....	97
3.4.3 Struktur Kontrol (Mesin Inferensi)	102
3.4.4 Desain UML (<i>Unified Modeling Language</i>).....	104
3.4.5 Desain Database	140
3.4.6 Desain Antarmuka.....	142
3.5 Lokasi dan Jadwal Penelitian	151
3.5.1 Lokasi Penelitian.....	151
3.5.2 Jadwal Penelitian.....	152
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	154
4.1 Hasil Penelitian	154
4.1.1 Tampilan Sistem Pakar	154
4.2 Pembahasan.....	166
4.2.1 Black Box Testing.....	167
4.2.2. Pengujian Sistem Dengan Pakar (Mekanik)	171
BAB V SIMPULAN DAN SARAN.....	174
5.1 Simpulan.....	174
5.2 Saran.....	174
DAFTAR PUSTAKA	

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Jaringan Syaraf Dengan Lapis Tunggal	13
Gambar 2. 2 Jaringan Syaraf Dengan Lapisan Banyak.....	13
Gambar 2. 3 Jaringan Syaraf Dengan Lapis Kompetitif	14
Gambar 2. 4 Metode <i>Forward Chaining</i>	15
Gambar 2. 5 Metode <i>Backward Chaining</i>	16
Gambar 2. 6 Komponen Utama Sistem Pakar	20
Gambar 2. 7 Arsitektur Sistem Pakar.....	21
Gambar 2. 8 Pohon Keputusan <i>Forward Chaining</i>	25
Gambar 2. 9 Alternatif Pohon Keputusan	27
Gambar 2. 10 Operasi Sistem <i>Forward Chaining</i>	30
Gambar 2. 11 Diagram UML	35
Gambar 2. 12 Logo PHP	41
Gambar 2. 13 Logo HTML	42
Gambar 2. 14 Logo CSS	44
Gambar 2. 15 Logo <i>JavaScript</i>	46
Gambar 2. 16 Logo <i>jQuery</i>	46
Gambar 2. 17 Motor Bensin atau Motor <i>Otto</i>	51
Gambar 2. 18 Motor <i>Diesel</i>	51
Gambar 2. 19 Susunan Silinder Motor.....	52
Gambar 2. 20 Supercarjer untuk meningkatkan tekanan pasokan udara	54
Gambar 2. 21 <i>Turbocarjer</i> digunakan untuk meningkatkan tekanan pasokan udara dengan memanfaatkan tekanan gas buang untuk memutar <i>turbin</i>	54
Gambar 2. 22 (a) Motor siklus 2-tak (b) Motor siklus4-tak.....	55
Gambar 2. 23 Prinsip kerja motor 2-langkah (a) langkah hisap dan kompresi (b) langkah daya dan buang	56
Gambar 2. 24 Prinsip kerja motor 4-langkah	59
Gambar 2. 25 Ruang bakar motor bensin	60
Gambar 2. 26 <i>V-belt</i>	63
Gambar 2. 27 Logo <i>StarUML</i>	66
Gambar 2. 28 Logo <i>XAMPP</i>	68
Gambar 2. 29 Logo Sublime Text.....	69
Gambar 2. 30 Logo <i>phpMyAdmin</i>	70
Gambar 2. 31 Logo <i>Mozilla Firefox</i>	71
Gambar 2. 32 Logo Mendeley	72
Gambar 2. 33 Kerangka Pemikiran.....	80
Gambar 3. 1 Desain Penelitian.....	82
Gambar 3. 2 Pohon Keputusan Sistem Pakar.....	102
Gambar 3. 3 <i>Flowchart</i> Mesin Inferensi Sistem Pakar	103

Gambar 3. 4 Use Case Diagram	105
Gambar 3. 5 <i>Activity Diagram</i> Melihat Beranda Utama.....	107
Gambar 3. 6 <i>Diagram Activity</i> Melihat Petunjuk	107
Gambar 3. 7 <i>Activity Diagram</i> Daftar	108
Gambar 3. 8 <i>Activity Diagram</i> Login User	109
Gambar 3. 9 <i>Activity Diagram</i> Konsultasi	110
Gambar 3. 10 <i>Activity Diagram</i> Melihat Riwayat Konsultasi	111
Gambar 3. 11 <i>Activity Diagram</i> Logout User	112
Gambar 3. 12 <i>Activity Diagram</i> Login Menu Admin	113
Gambar 3. 13 <i>Activity Diagram</i> Mengelola Data Gejala	114
Gambar 3. 14 <i>Activity Diagram</i> Mengelola Data Kerusakan	115
Gambar 3. 15 <i>Activity Diagram</i> Mengelola Data Bagian Kerusakan	117
Gambar 3. 16 <i>Activity Diagram</i> Mengelola Data Aturan	118
Gambar 3. 17 <i>Activity Diagram</i> Mengelola Data Admin	120
Gambar 3. 18 <i>Activity Diagram</i> Mengelola Data Riwayat Pengunjung	121
Gambar 3. 19 <i>Activity Diagram</i> Mengelola Data Riwayat Konsultasi	122
Gambar 3. 20 <i>Activity Diagram</i> Logout Admin	123
Gambar 3. 21 <i>Sequence Diagram</i> Melihat Beranda Utama.....	124
Gambar 3. 22 <i>Sequence Diagram</i> Melihat Petunjuk.....	125
Gambar 3. 23 <i>Sequence Diagram</i> Daftar	125
Gambar 3. 24 <i>Sequence Diagram</i> Login User	126
Gambar 3. 25 <i>Sequence Diagram</i> Konsultasi	127
Gambar 3. 26 <i>Sequence Diagram</i> Melihat Riwayat Konsultasi.....	128
Gambar 3. 27 <i>Sequence Diagram</i> Logout User	129
Gambar 3. 28 <i>Sequence Diagram</i> Login	130
Gambar 3. 29 <i>Sequence Diagram</i> Mengelola Data Gejala	131
Gambar 3. 30 <i>Sequence Diagram</i> Mengelola Data Kerusakan.....	132
Gambar 3. 31 <i>Sequence Diagram</i> Mengelola Data Bagian Kerusakan	134
Gambar 3. 32 <i>Sequence Diagram</i> Mengelola Data Aturan.....	135
Gambar 3. 33 <i>Sequence Diagram</i> Mengelola Data Admin.....	137
Gambar 3. 34 <i>Sequence Diagram</i> Mengelola Data Riwayat Pengunjung	138
Gambar 3. 35 <i>Sequence Diagram</i> Mengelola Riwayat Konsultasi.....	139
Gambar 3. 36 <i>Sequence Diagram</i> Logout Admin.....	140
Gambar 3. 37 Desain <i>Physical Data Model</i>	141
Gambar 3. 38 Rancangan Halaman Beranda	142
Gambar 3. 39 Rancangan Halaman Petunjuk	143
Gambar 3. 40 Rancangan Halaman Pendaftaran.....	144
Gambar 3. 41 Rancangan Halaman Beranda Konsultasi	144
Gambar 3. 42 Rancangan Halaman Konsultasi.....	145
Gambar 3. 43 Rancangan Halaman Riwayat Konsultasi	145
Gambar 3. 44 Rancangan Halaman Hasil Diagnosa	146
Gambar 3. 45 Rancangan Halaman Data Gejala.....	147
Gambar 3. 46 Rancangan Halaman Data Kerusakan	148
Gambar 3. 47 Rancangan Halaman Data Bagian Kerusakan.....	148
Gambar 3. 48 Rancangan Halaman Data Aturan	149
Gambar 3. 49 Rancangan Halaman Data Admin	150

Gambar 3. 50 Rancangan Halaman Data Riwayat Pengunjung.....	150
Gambar 3. 51 Rancangan Halaman Data Riwayat Konsultasi.....	151
Gambar 3. 52 Tampilan Denah Lokasi Penelitian	152
Gambar 4. 1 Tampilan Halaman Beranda	155
Gambar 4. 2 Tampilan Halaman Petunjuk.....	156
Gambar 4. 3 Tampilan Halaman Pendaftaran	157
Gambar 4. 4 Tampilan Halaman Beranda Konsultasi.....	158
Gambar 4. 5 Tampilan Halaman Konsultasi	159
Gambar 4. 6 Tampilan Halaman Riwayat Konsultasi.....	160
Gambar 4. 7 Tampilan Halaman Hasil Diagnosa	160
Gambar 4. 8 Tampilan Halaman Beranda Admin.....	161
Gambar 4. 9 Tampilan Halaman Data Gejala	162
Gambar 4. 10 Tampilan Halaman Data Kerusakan	162
Gambar 4. 11 Tampilan Halaman Data Bagian Kerusakan	163
Gambar 4. 12 Tampilan Halaman Data Aturan	164
Gambar 4. 13 Tampilan Halaman Data Admin	164
Gambar 4. 14 Tampilan Halaman Data Riwayat Pengunjung	165
Gambar 4. 15 Tampilan Halaman Data Riwayat Konsultasi	166

DAFTAR TABEL

	Halaman
Tabel 2. 1 Tabel Keputusan	25
Tabel 2. 2 Tabel Alternatif Keputusan	26
Tabel 2. 3 Simbol <i>Use Case Diagram</i>	36
Tabel 2. 4 Simbol <i>Activity Diagram</i>	38
Tabel 2. 5 Simbol <i>Sequence Diagram</i>	39
Tabel 3. 1 Tabel hubungan Variabel dan Indikator.....	87
Tabel 3. 2 Tabel Kerusakan dan Solusi Kerusakan sepeda motor Yamaha Nmax	88
Tabel 3. 3 Tabel nama indikator kerusakan sepeda motor Yamaha Nmax.....	92
Tabel 3. 4 Tabel kerusakan dan solusi kerusakan sepeda motor Yamaha Nmax .	92
Tabel 3. 5 Tabel gejala kerusakan sepeda motor Yamaha Nmax	95
Tabel 3. 6 Tabel Data Aturan	96
Tabel 3. 7 Aturan Inference	97
Tabel 3. 8 Tabel relasi gejala, kerusakan dan bagian kerusakan	101
Tabel 3. 9 Tabel Jadwal Penelitian	153
Tabel 4. 1 Tabel Pengujian Validitas Sistem	167
Tabel 4. 2 Tabel Hasil Diagnosa Pakar dan Diagnosa Sistem	171

DAFTAR LAMPIRAN

RIWAYAT HIDUP

SURAT PENELITIAN

FORMULIR WAWANCARA DAN PENGUJIAN SISTEM

FOTO WAWANCARA DAN PENGUJIAN SISTEM

BAB I

PENDAHULUAN

1.1 Latar Belakang Penelitian

Sepeda motor pada saat ini menjadi salah satu sarana transportasi pilihan terbaik masyarakat. Banyak pengendara sepeda motor yang tidak mengetahui kendala kerusakan sepeda motor. Sebagian besar pengendara cenderung menyerahkan kendala kerusakan sepeda motor kepada mekanik tanpa mengetahui bahwa sebenarnya kerusakan tersebut merupakan kerusakan sederhana atau tidak terlalu rumit untuk diperbaiki. Permasalahan lain yang sering terjadi dan ditemukan berdasarkan informasi yang didapatkan dari beberapa mekanik yang bekerja pada salah satu bengkel di kota Batam, yaitu bengkel Marfen Utama Motor yang beralamat di Ruko Bukit Sentosa C 9-10, Jalan Jendral S. Parman, Tanjung Piayu, Kecamatan Sei Beduk, Kota Batam, Kepulauan Riau 29433. Adalah bahwa mekanik juga masih mendapat kendala dalam hal menentukan kerusakan yang terjadi, untuk mendapatkan keputusan penanganan, dan kemudian menentukan pilihan spareparts yang tepat pada sepeda motor pelanggannya. Pengguna sepeda motor matic yang semakin banyak menyebabkan meningkatnya usaha-usaha yang berhubungan dengan sepeda motor matic, salah satunya adalah usaha perbaikan sepeda motor Yamaha Nmax. Perkembangan usaha yang semakin maju dan meluas tetapi tidak diiringi dengan perkembangan tenaga ahli atau pakar yang menguasai keahlian untuk memperbaiki sepeda motor Yamaha

Nmax. Permintaan konsumen yang meningkat tetapi tidak dapat diatasi dengan tenaga ahli yang sedikit.

Berdasarkan penjelasan di atas, penulis melihat adanya masalah saat mekanik melakukan analisa kerusakan pada sepeda motor pelanggannya ketika sudah letih dan menurunkan daya analisa, sehingga dapat berakibat kesalahan menentukan kerusakan yang terjadi pada sepeda motor pelanggannya. Oleh karena itu diperlukan suatu sistem pakar yang dapat membantu mekanik dalam mengatasi permasalahan pada saat proses analisa kerusakan sepeda motor yang ditanganinya, sehingga dapat mengurangi resiko kesalahan dalam menentukan kerusakan sepeda motor pelanggannya yang berakibat kesalahan dalam tindakan perbaikan atas kerusakannya.

Dalam jurnal penelitian (Janssen, Grinter, & Roberts, 2011: 70) mengemukakan bahwa *The expert system's role is in the preparation to capture the knowledge of the experts and the data from the customer's requirements. The system has the capability to compile the collected data and form the appropriate rules for choosing fragrance notes for the products. In order to identify the hidden pattern of the customer's needs, the artificial neural networks technique has been applied to classify the fragrance notes based upon a list of selected information.* Jadi disini dapat kita ketahui bahwa sistem pakar berfungsi untuk menyimpan pengetahuan para ahli dan data dari kebutuhan pengguna sistem pakar.

Untuk menarik suatu kesimpulan, sistem pakar melakukan proses inferensi (Triyanto & Fadlil, 2014: 703) menyatakan bahwa Inferensi yaitu kemampuan menarik kesimpulan berdasarkan pengalaman. Dalam jurnal penelitian

(Primantoro & Fadlil, 2014: 683) dikemukakan bahwa runut maju (Forward Chaining) berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan hasil

Sistem pakar tersebut akan mengadopsi pengetahuan mekanik ke dalam komputer dan digunakan oleh mekanik bengkel tersebut untuk mencari solusi dalam mengatasi masalah pada saat melakukan analisa kerusakan sepeda motor ketika kondisi daya analisa sudah mulai menurun atau kondisi sudah letih. Berdasarkan latar belakang yang telah diuraikan di atas, yaitu adanya permasalahan pada bengkel Marfen Utama Motor, maka dalam penelitian ini penulis mengambil judul “SISTEM PAKAR DIAGNOSA KERUSAKAN SEPEDA MOTOR YAMAHA NMAX”.

1.2 Identifikasi Masalah

Berdasarkan dari latar belakang diatas maka permasalahan yang timbul dapat diidentifikasi sebagai berikut:

1. Kurangnya pengetahuan masyarakat terhadap kerusakan sepeda motor.
2. Mekanik yang biasa menangani kerusakan motor menjadi kurang teliti dalam menganalisa kerusakan yang terjadi pada sepeda motor pelanggannya ketika sudah lelah dan dalam tekanan pelanggan.

3. Perkembangan usaha yang semakin maju dan meluas tetapi tidak diiringi dengan perkembangan tenaga ahli atau pakar yang menguasai keahlian untuk memperbaiki sepeda motor Yamaha Nmax.
4. Belum tersedianya sistem pakar yang dapat digunakan oleh mekanik bengkel dalam membantu menganalisa atau untuk mendeteksi kerusakan sepeda motor Yamaha Nmax di bengkel Marfen Utama Motor.

1.3 Pembatasan Masalah

Pada pembuatan aplikasi ini perlu didefinisikan batasan masalah mengenai sejauh mana pembuatan aplikasi ini akan dikerjakan. Beberapa batasan masalah tersebut antara lain:

1. Sistem hanya melakukan deteksi permasalahan yang terjadi pada kendaraan sepeda motor Yamaha Nmax.
2. Gejala yang ditanyakan oleh sistem berdasarkan pada jenis kerusakan pada 4 indikator yaitu: masalah pada ruang bakar dan pengapian, masalah pada CVT, masalah pada klep dan filter, dan masalah kelistrikan.
3. Tempat penelitian hanya pada bengkel Marfen Utama Motor yang beralamat di Ruko Bukit Sentosa C 9-10, Jalan Jendral S. Parman, Tanjung Piayu, Kecamatan Sei Beduk, Kota Batam, Kepulauan Riau 29433.
4. Objek penelitian adalah motor Yamaha Nmax.
5. Sistem ini dibuat berbasis *web*

1.4 Perumusan Masalah

Dari permasalahan yang telah dijelaskan sebelumnya, maka dapat dirumuskan segala permasalahan yang akan dijadikan pokok pembahasan di dalam penelitian ini, yaitu bagaimana merancang dan membangun sebuah aplikasi sistem pakar dengan menggunakan metode *forward chaining* untuk mendeteksi kerusakan pada kendaraan sepeda motor Yamaha Nmax ?

1.5 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang bangun serta menghasilkan sistem pakar untuk mendeteksi kerusakan pada sepeda motor Yamaha Nmax berbasis web dengan metode *forward chaining* dengan menggunakan kaidah-kaidah sistem pakar yang ada. Sehingga nantinya dapat difungsikan untuk mendeteksi kerusakan sepeda motor Yamaha Nmax

1.6 Manfaat Penelitian

Penelitian ini diharapkan mampu memberikan manfaat baik dari aspek keilmuan (teoritis) maupun dari aspek kegunaan (praktis). Adapun manfaat yang akan diperoleh dari penelitian ini adalah sebagai berikut:

1.6.1 Manfaat Teoritis

Manfaat teoritis yang akan diperoleh dari penelitian ini adalah sebagai berikut:

1. Dapat menjadi referensi untuk bidang penelitian sistem pakar

2. Dapat memberikan pemahaman lebih jauh tentang gejala-gejala kerusakan yang terjadi pada sepeda motor Yamaha Nmax.

1.6.2 Manfaat Praktis

Manfaat praktis yang akan diperoleh dari penelitian ini adalah sebagai berikut:

1. Mendeteksi lebih cepat kerusakan pada sepeda motor Yamaha Nmax .
2. Mekanik mengetahui lebih cepat kerusakan pada motor sehingga pengerjaan bisa lebih cepat.
3. Membantu masyarakat awam untuk mengetahui kerusakan yang terjadi pada sepeda motor Yamaha Nmax serta bagaimana cara menanganinya.
4. Dapat menghemat waktu dalam menyelesaikan masalah.

BAB II KAJIAN PUSTAKA

2.1 Teori Dasar

Kecerdasan tiruan adalah ilmu yang berdasarkan proses manusia berfikir, maka penelitian bagaimana proses manusia berpikir adalah hal yang pokok. Pada saat ini para peneliti hanya mulai mengerti sedikit dari proses berfikir tersebut, tetapi sudah cukup diketahui untuk membuat asumsi-asumsi yang pasti tentang bagaimana cara berfikir dan menggunakan asumsi-asumsi tersebut untuk mendesain suatu program komputer yang mempunyai kecerdasan tiruan. Sistem Pakar adalah sistem yang paling banyak digunakan aplikasinya dalam membantu manusia dalam menyelesaikan masalah-masalah di dunia nyata. Sebuah sistem pakar harus berkemampuan ekstensif dan dapat menghitung kerugian dalam memecahkan sebuah persoalan. Jadi, sistem pakar merupakan sub bagian dari bidang kecerdasan tiruan, karena dalam membuat sistem pakar membutuhkan cara dan pikiran manusia yang pakar. Untuk mendukung penelitian ini, maka pada bagian ini akan dijelaskan teori mengenai kecerdasan buatan (*Artificial Intelligence*) dan beberapa sub disiplin ilmunya seperti Logika *Fuzzy* (*Fuzzy Logic*), Jaringan Saraf Tiruan (JST), SistemPakar (*Expert System*).

2.1.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan

artificial artinya buatan. Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berfikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia (Sutojo, dkk, 2011: 1).

Kecerdasan buatan atau *Artificial Intelligence* (AI) adalah suatu sistem informasi yang berhubungan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia di dalam sebuah sistem teknologi informasi, sehingga sistem tersebut memiliki kecerdasan seperti yang dimiliki oleh manusia. Sistem tersebut dikembangkan untuk mengembangkan metode dan sistem untuk menyelesaikan masalah, biasanya diselesaikan melalui aktifitas intelektual manusia (Husda, 2012: 192-193).

Perkembangan kecerdasan buatan berawal pada 1965, John McCarty melakukan penelitian bersama 3 rekannya yaitu Minsky, Claude Shannon dan Nathaniel Rochester di Dartmouth College. Selama 2 bulan mereka melakukan penelitian dalam bidang *automata*, jaringan saraf dan pembelajaran intelegensia. Hasilnya adalah program yang dinamakan *Principia Mathematica* yang mampu berfikir non-numerik dan menyelesaikan masalah pemikiran. Hal ini menjadikan McCarty disebut sebagai *The Father of AI* atau bapak AI (Suyanto, 2014: 5) .

Menurut Husda (2012: 196-198) sistem kecerdasan buatan yang banyak dikembangkan saat ini adalah :

1. Sistem Pakar (*Expert System*)

Sistem pakar yaitu program konsultasi (*advisory*) yang mencoba menirukan proses penalaran seorang pakar / ahli dalam memecahkan masalah yang rumit.

2. Pemrosesan Bahasa Alami (*Natural Language Processing*)

Pemrosesan Bahasa Alami (*Natural Language Processing*) yang memberi kemampuan pengguna komputer untuk berkomunikasi dengan komputer dalam bahasa mereka sendiri (bahasa manusia).

3. Pemahaman Ucapan/Suara (*Speech/Voice Understanding*)

Adalah teknik agar komputer dapat mengenali dan memahami bahasa ucapan.

4. Sistem Sensor dan Robotika

Sistem sensor, seperti sistem visi dan pencitraan, serta sistem pengolahan sinyal, merupakan bagian dari robotika. Sebuah robot, yaitu perangkat elektromekanik yang diprogram untuk melakukan tugas manual, tidak semuanya merupakan bagian dari AI. Robot yang cerdas biasanya mempunyai perangkat sensor, seperti kamera, yang mengumpulkan informasi mengenai operasi dan lingkungannya. Kemudian bagian AI robot tersebut menerjemahkan informasi tadi dan merespon serta beradaptasi jika terjadi perubahan lingkungan.

5. Komputer Visi

Merupakan kombinasi dari pencitraan, pengolahan citra, pengenalan pola serta proses pengambilan keputusan. Tujuan utama dari komputer visi adalah untuk menerjemahkan suatu pemandangan.

6. *Intelligent Tutoring / Intelligent Computer-Aided Instruction*

Merupakan komputer yang mengajari manusia. Dengan menambahkan aspek kecerdasan di dalam komputer, dapat tercipta komputer “guru” yang dapat

mengatur teknik pengajarannya untuk menyesuaikan dengan kebutuhan “murid” secara individual.

7. Mesin Belajar (*Machine Learning*)

Yang berhubungan dengan sekumpulan metode untuk mencoba mengajari / melatih komputer untuk memecahkan masalah atau mendukung usaha pemecahan masalah dengan menganalisa kasus-kasus yang telah terjadi. Dua metode mesin belajar yang paling populer adalah Komputasi *Neural* dan Logika Samar.

8. Aplikasi lain dari AI misalnya untuk merangkum berita, pemrograman komputer secara otomatis, atau menerjemahkan dari suatu bahasa ke bahasa yang lain, serta aplikasi dalam permainan (Ingat pertandingan catur antara Grand Master Anatoly Karpov dengan komputer *Deep Thought* dari IBM).

2.1.2 Logika Fuzzy (*fuzzy logic*)

Logika *fuzzy (fuzzy logic)* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan komputer, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem *control*. Metode ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya (Sutojo, dkk, 2011: 211).

Menurut Sutojo, dkk (2011:233-237) ada beberapa metode yang digunakan dalam sistem inferensi *fuzzy*, yaitu:

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan yaitu: *fuzzifikasi*, pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*), mesin inferensi menggunakan fungsi implikasi MIN (*Minimum*), dan *defuzzifikasi* menggunakan metode Rata-rata (*Average*).

2. Metode Mamdani

Metode Mamdani menggunakan operasi MIN-MAX atau MAX-PRODUCT. Terdapat 4 tahapan untuk mendapatkan *output* pada metode Mamdani, yaitu: *fuzzifikasi*, pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*), aplikasi fungsi implikasi menggunakan fungsi MIN (*Minimum*) dan komposisi antar-*rule* menggunakan fungsi MAX (*Maximum*) dengan menghasilkan himpunan *fuzzy* baru, dan *defuzzifikasi* menggunakan metode *Centroid* (Titik Tengah).

3. Metode Sugeno

Pada metode Sugeno, mempunyai *output* sistem berupa konstanta atau persamaan linear. Metode Sugeno menggunakan beberapa tahapan dalam inferensinya, yaitu: *fuzzifikasi*, pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*), mesin inferensi menggunakan fungsi implikasi MIN (*Minimum*), dan *defuzzifikasi* menggunakan metode rata-rata (*Average*) (Sutojo, dkk, 2011: 233-237).

2.1.3 Jaringan Syaraf Tiruan (*artificial neural network*)

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak

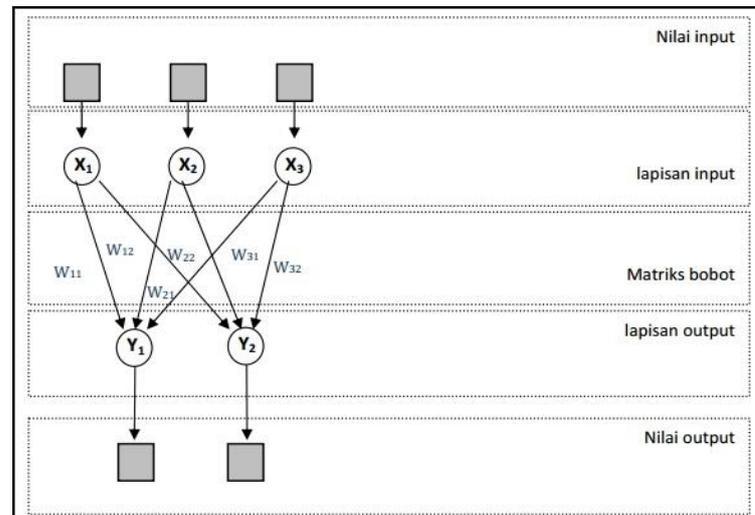
manusia. Elemen utamanya adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu. Cara kerja jaringan saraf tiruan sama seperti cara kerja manusia, yaitu belajar melalui contoh (Sutojo, dkk, 2011: 238).

Beberapa contoh aplikasi jaringan saraf tiruan adalah implementasi di bidang kedokteran, yaitu pemodelan dan diagnosis sistem kardiovaskular, hidung elektronik, dan dokter instan; dan implemetasi di bidang bisnis, yaitu jaringan saraf tiruan yang diintegrasikan dengan merek dagang *The Airline Marketing Tactician* (AMT) menggunakan *back-propagation* untuk membantu kontrol pemasaran dari alokasi kursi penerbangan (Sutojo, dkk, 2011: 286-288).

Menurut Sutojo, dkk (2011: 292-295) tiga arsitektur jaringan yang sering digunakan dalam jaringan saraf tiruan adalah :

1. Jaringan lapisan tunggal

Jaringan ini terdiri dari 1 lapisan *input* dan 1 lapisan *output*, yang mana setiap unit dalam lapisan *input* selalu terhubung dengan setiap unit yang terdapat pada lapisan *output*. Berikut ini adalah gambar arsitektur jaringan lapis tunggal:

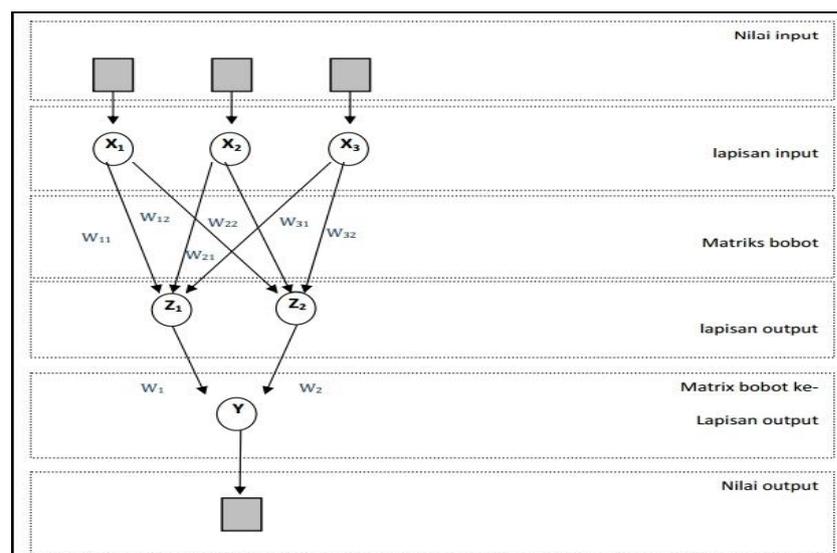


Gambar 2. 1 Jaringan Syaraf Dengan Lapis Tunggal

Sumber: Sutojo dkk 2011

2. Jaringan lapisan banyak

Jaringan ini mempunyai 3 jenis lapisan, yaitu lapisan *input*, lapisan *tersembunyi*, dan lapisan *output*. Jaringan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan jaringan lapisan tunggal. Berikut ini adalah gambar arsitektur jaringan lapisan banyak:

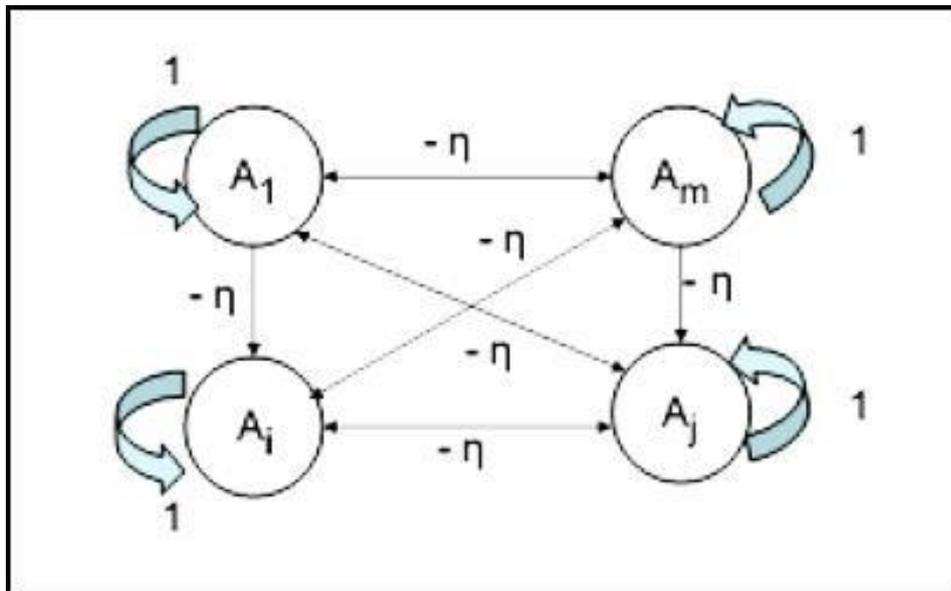


Gambar 2. 2 Jaringan Syaraf Dengan Lapisan Banyak

Sumber: Sutojo dkk., 2011

3. Jaringan dengan lapisan kompetitif

Jaringan ini memiliki bobot yang telah ditentukan dan tidak memiliki proses pelatihan. Jaringan ini digunakan untuk mengetahui *neuron* pemenang dari sejumlah *neuron* yang ada sehingga sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif.



Gambar 2. 3 Jaringan Syaraf Dengan Lapis Kompetitif
Sumber: Sutojo dkk, 2011

2.1.4 Sistem Pakar

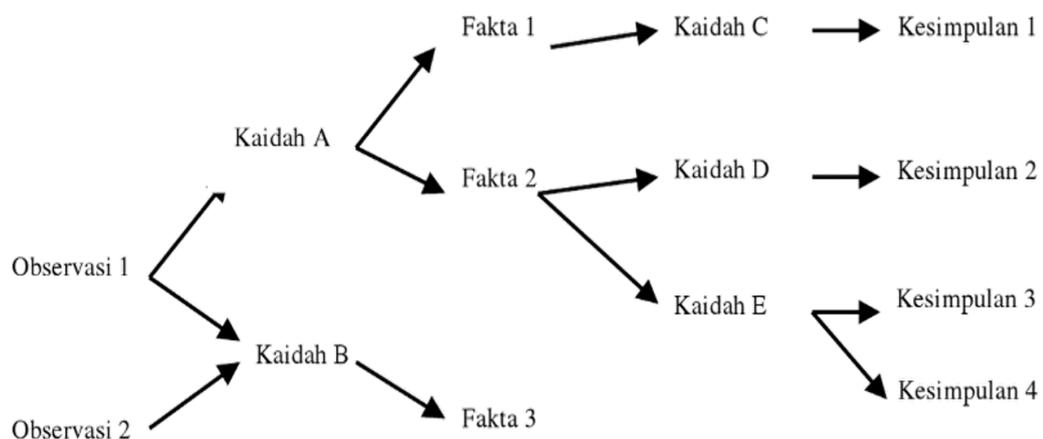
Sistem pakar adalah suatu program komputer yang mengandung pengetahuan dari satu atau lebih pakar manusia mengenai suatu bidang spesifik (Husda, 2012: 181). Sedangkan menurut Hartati dan Iswanti (2008: 2) sistem pakar merupakan salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia.

Menurut Arhami, sistem pakar adalah salah satu cabang dari *Artificial Intelligence* (kecerdasan buatan) yang membuat penggunaan secara luas

knowledge yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Menurut Marimin, sistem pakar adalah sistem perangkat lunak komputer yang menggunakan ilmu, fakta, dan teknik berfikir dalam pengambilan keputusan untuk menyelesaikan masalah-masalah yang biasanya hanya dapat diselesaikan oleh tenaga ahli dalam bidang yang bersangkutan (Husda, 2012: 181-182).

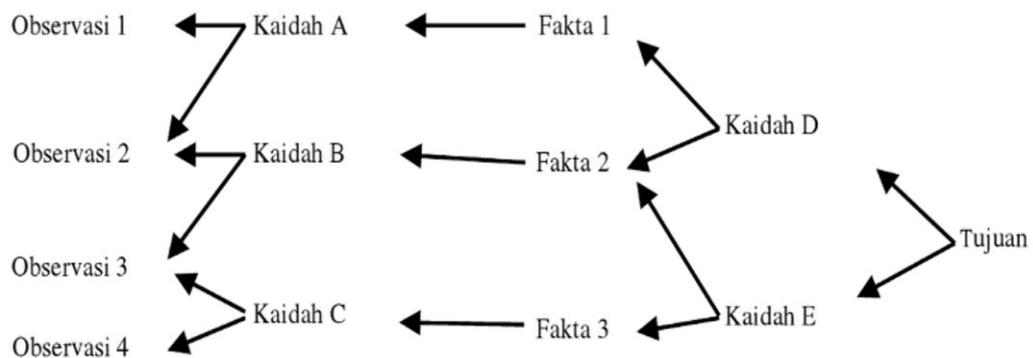
Mesin inferensi dalam sistem pakar adalah bagian yang mengandung mekanisme fungsi berfikir dan pola penalaran sistem yang digunakan oleh seorang pakar. Ada dua teknik inferensi yang digunakan dalam sistem pakar yaitu pelacakan ke belakang (*backward chaining*) yang memulai penalaran dari kesimpulan hipotesa menuju fakta yang mengandung hipotesa tersebut. Dan yang kedua yakni pelacakan ke depan (*forward chaining*) yang merupakan kebalikan dari pelacakan ke belakang yaitu memulai dari sekumpulan data menuju kesimpulan (Hayadi, 2016: 7).

Berikut ini adalah gambar teknik *inferensi* dengan metode *forward chaining*:



Gambar 2. 4 Metode *Forward Chaining*
Sumber: Hayadi, 2016

Berikut ini adalah gambar teknik *inferensi* dengan metode *backward chaining*:



Gambar 2. 5 Metode *Backward Chaining*
Sumber: Hayadi, 2016

Metode inferensi sistem pakar dipengaruhi oleh tiga macam teknik penelusuran yaitu (Hayadi, 2016: 8):

- Depth-first search*: melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan.
- Breadth-first search*: melakukan penelusuran dengan bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah.
- Best-first search*: bekerja berdasarkan kombinasi kedua metode sebelumnya.

Ada 4 tipe penjelasan yang digunakan dalam sistem pakar, yaitu (Suwondo, 2014: 91):

- Penjelasan mengenai jejak aturan yang menunjukkan status konsultasi.
- Penjelasan mengenai bagaimana sebuah keputusan diperoleh.
- Penjelasan mengapa sistem menanyakan suatu pertanyaan.
- Penjelasan mengapa sistem tidak memberikan keputusan seperti yang dikehendaki pengguna.

Menurut (Husda, 2012: 182) ciri-ciri sistem pakar adalah:

- Memiliki informasi yang handal.

2. Mudah dimodifikasi
3. Dapat digunakan dalam berbagai jenis komputer.
4. Memiliki kemampuan belajar beradaptasi.

Menurut Hayadi (2016: 5-6) sistem pakar terdiri dari beberapa konsep yang harus dimiliki. Konsep dasar dari suatu pakar sebagai berikut:

1. Keahlian

Adalah suatu pengetahuan khusus yang diperoleh dari latihan, belajar, dan pengetahuan untuk memecahkan masalah.

2. Ahli (*Expert*)

Melibatkan kegiatan mengenali dan memformulasikan permasalahan, memecahkan masalah secara cepat dan tepat, menerangkan pemecahannya, belajar dari pengalaman, merestrukturisasi pengetahuan, memecahkan aturan serta menentukan relevansi.

3. Mentransfer keahlian (*Transferring Expertise*)

Adalah proses pentransferan keahlian dari seorang pakar kedalam komputer agar dapat digunakan oleh orang lain yang bukan pakar.

4. Menyimpulkan aturan (*Inferencing Rule*)

Merupakan kemampuan komputer yang telah diprogram. Penyimpulan ini dilakukan oleh mesin inferensi yang meliputi prosedur tentang penyelesaian masalah.

5. Peraturan (*Rule*)

Diperlukan karena mayoritas dari sistem pakar bersifat *rule based sistem*, yang berarti pengetahuan disimpan dalam bentuk peraturan.

6. Kemampuan menjelaskan (*Explanation Capability*)

Adalah karakteristik dari sistem pakar yang memiliki kemampuan menjelaskan atau memberi saran mengapa tindakan tertentu dianjurkan atau tidak dianjurkan.

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud antara lain: pembuatan keputusan (*decision making*), pemaduan pengetahuan (*knowledge using*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*) dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar (Suwondo, 2014: 90).

Sistem pakar merupakan sebuah program komputer yang memiliki kemampuan seperti seorang pakar, ada banyak keuntungan bila menggunakan sistem pakar yaitu (Husda, 2012: 186) :

1. Menjadikan pengetahuan dan nasehat lebih mudah didapat.
2. Meningkatkan *output* dan produktifitas.
3. Menyimpan kemampuan dan keahlian seorang pakar.
4. Meningkatkan penyelesaian masalah yang khusus.
5. Meningkatkan reliabilitas.
6. Memberikan respon (jawaban) yang cepat.
7. Merupakan panduan yang cerdas.

8. Dapat bekerja dengan informasi yang kurang lengkap dan mengandung ketidakpastian.
9. Sebagai basis data cerdas.

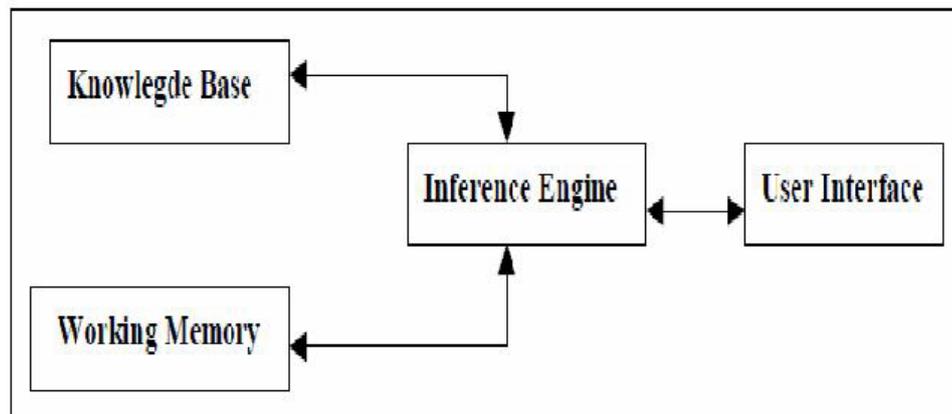
Selain mempunyai banyak keuntungan, sistem pakar juga memiliki beberapa kelemahan, diantaranya adalah (Husda, 2012: 186-187) :

1. Masalah dalam mendapatkan pengetahuan.
2. Untuk membuat suatu sistem pakar yang berkualitas tinggi sangat sulit dan memerlukan biaya yang sangat besar.
3. Boleh jadi sistem pakar tidak dapat membuat keputusan.
4. Sistem pakar tidaklah 100% menguntungkan, walaupun seseorang tetap tidak sempurna atau tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan. Sehingga dalam hal ini peran manusia tetap merupakan *factor* yang dominan.

Kelemahan-kelemahan dari sistem pakar tersebut bukanlah sama sekali tidak bisa diatasi, tetapi dengan terus melakukan perbaikan dan pengolahan berdasarkan pengalaman yang telah ada maka hal itu diyakini akan dapat diatasi, walaupun dalam waktu yang panjang dan terus-menerus.

2.1.4.1 Komponen Sistem Pakar

Komponen sistem pakar terbagi menjadi empat bagian, untuk lebih jelasnya dapat kita gambarkan sebagai berikut (Harison dan Alexyusandera, 2014: 9):



Gambar 2. 6 Komponen Utama Sistem Pakar
Sumber: Harison dan Alexyusandera, 2014

1. *Knowledge Base* (Basis Pengetahuan)

Knowledge Base merupakan inti dari program sistem pakar karena basis pengetahuan itu merupakan presentasi pengetahuan atau *knowledge representation* basis pengetahuan adalah sebuah basis data.

2. *Working Memory* (Basis Data atau Memori Kerja)

Working memory adalah bagian yang mengandung semua fakta-fakta baik fakta awal pada saat sistem beroperasi maupun fakta-fakta pada saat pengambilan kesimpulan sedang dilaksanakan selama sistem pakar beroperasi basis data berada di dalam memori kerja.

3. *Inference Engine* (Mesin Inferensi)

Inference Engine adalah bagian yang menyediakan mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini akan menganalisa masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. Mesin ini juga akan dimulai

pelacakannya dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.

4. *User Interface* (Antarmuka Pemakai)

Antarmuka pemakai adalah bagian penghubung antara program sistem pakar dengan pemakai. Pada bagian ini memungkinkan pengguna untuk memasukkan instruksi dan informasi ke dalam sistem pakar serta menerima penjelasan dan kesimpulan.

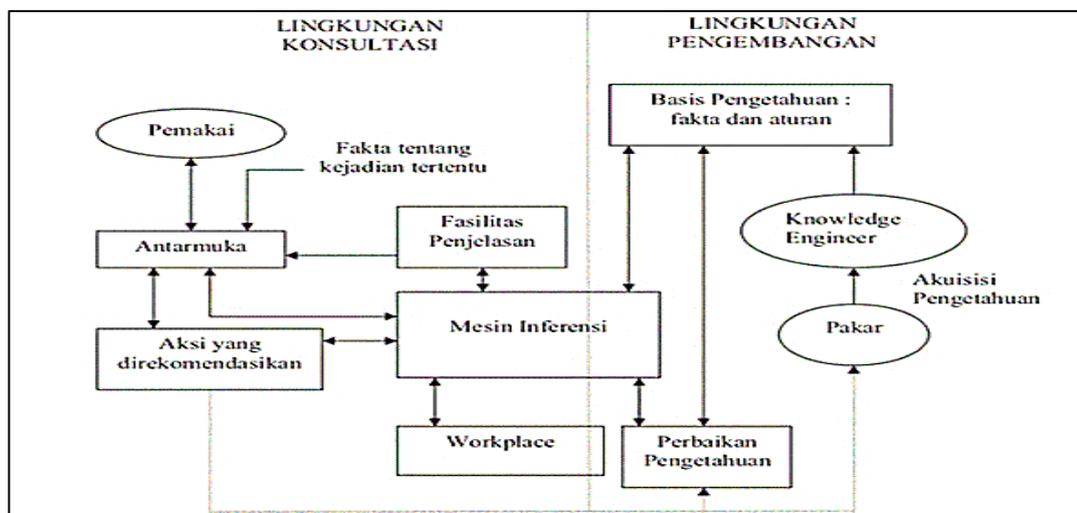
Menurut Husda (2012: 182) terdapat dua bagian utama yang menyusun sistem pakar, yaitu:

1. Lingkungan pengembangan (*Development Environment*)

Lingkungan pengembangan digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar.

2. Lingkungan konsultasi (*Consultation Environment*)

Lingkungan konsultasi digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan pakar.



Gambar 2. 7 Arsitektur Sistem Pakar

Sumber: Husda, 2012

Komponen-komponen yang terdapat dalam arsitektur/struktur sistem pakar adalah (Husda, 2012:183-186):

1. Antarmuka Pengguna (*User Interface*)

Merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem.

2. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Terdapat 2 elemen dasar, yaitu: fakta (informasi tentang obyek dalam area permasalahan tertentu) dan aturan (informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui).

3. Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah akumulasi, *transfer*, dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam komputer.

4. Mesin/Motor Inferensi (*Inference Engine*)

Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan.

5. *Workplace/Blackboard*

Workplace merupakan area dari sekumpulan memori kerja (*working memory*), digunakan untuk merekam kejadian yang sedang berlangsung termasuk dalam keputusan sementara.

6. Fasilitas Penjelasan

Fasilitas penjelasan adalah elemen tambahan yang akan meningkatkan kemampuan sistem pakar. Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif.

7. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya dan juga mengevaluasi apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa mendatang.

2.1.4.2 Rule Sebagai Teknik Representasi Pengetahuan

Basis pengetahuan adalah komponen yang berisi sekumpulan kaidah yang berasal dari pengetahuan dalam domain tertentu dan secara umum disajikan dalam bentuk kaidah produksi (*IF...THEN...*). Pengetahuan pakar yang disajikan dalam format tertentu didapat dari sekumpulan pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti buku-buku, jurnal ilmiah, majalah, maupun dokumentasi tercetak lainnya (Hartati dan Iswanti, 2008: 5). Setiap *rule* terdiri dari dua bagian, yaitu bagian *IF* disebut *evidence* (fakta-fakta) dan bagian *THEN* disebut hipotesis atau kesimpulan (Sutojo, dkk., 2010 dalam Hayadi, 2011: 9). Menurut Adedeji (1992) dalam Hartati dan Ismawati (2008: 25) berikut ini adalah contoh struktur kaidah *IF-THEN* yang menghubungkan obyek:

IF premis *THEN* konklusi

IF masukan *THEN* keluaran

IF kondisi *THEN* tindakan

IF antesenden *THEN* konsekuen

IF data *THEN* hasil

IF tindakan *THEN* tujuan

IF aksi *THEN* reaksi

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. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. 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 (Hartati dan Iswanti, 2008:25-26).

Menurut Hartati dan Iswanti, (2008: 26-39) sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari *domain* tertentu disajikan dalam bentuk tabel keputusan kemudian dibuat pohon keputusannya sehingga bisa menggambarkan kondisi pada sistem pakar.

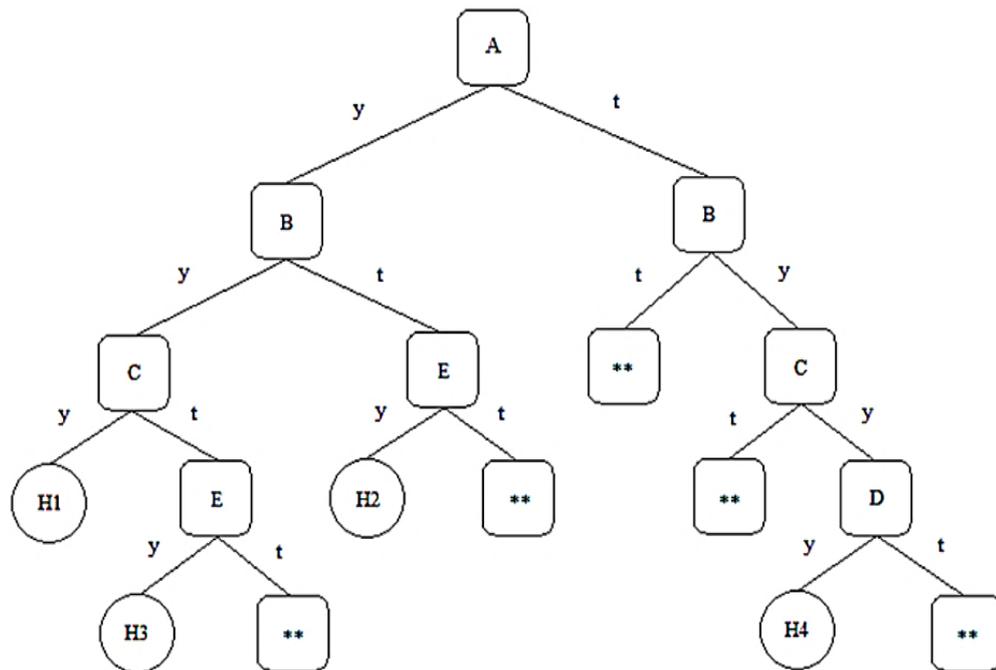
Berikut ini adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan:

Tabel 2. 1 Tabel Keputusan

Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>	Ya	Ya	ya	Tidak
<i>Evidence B</i>	Ya	Tidak	ya	Ya
<i>Evidence C</i>	Ya	Tidak	tidak	Ya
<i>Evidence D</i>	Tidak	Tidak	tidak	Ya
<i>Evidence E</i>	Tidak	Ya	ya	Tidak

Sumber: Hartati dan Iswanti, 2008

Berikut ini adalah contoh gambar pohon keputusan *forward chaining*:



Keterangan:

A = *evidence A* H1 = hipotesa 1 y = ya
 B = *evidence B* H2 = hipotesa 2 t = tidak
 C = *evidence C* H3 = hipotesa 3 ** = tidak menghasilkan hipotesa
 D = *evidence D* H4 = hipotesa 4
 E = *evidence E*

Gambar 2. 8 Pohon Keputusan *Forward Chaining*

Sumber: Hartati dan Iswanti, 2008

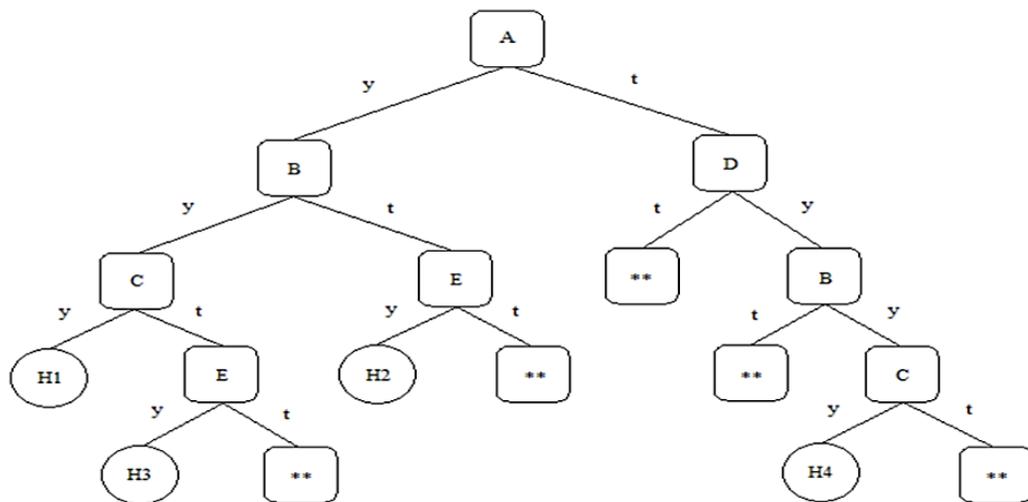
Dari Gambar 2.8 dapat diketahui bahwa hipotesa H1 terpenuhi jika memenuhi *evidence* A, B, dan C. Hipotesa H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. Hipotesa H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. Hipotesa H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi *node* (*evidence*) di atasnya, notasi “t” artinya tidak memenuhi. Dalam sesi konsultasi pada sistem pakar, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada Gambar 2.8 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak” maka sistem akan menanyakan *evidence* B. Ini berarti jawaban pengguna tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan. Berikut ini adalah tabel keputusan untuk mengatasi hal tersebut:

Tabel 2. 2 Tabel Alternatif Keputusan

Hipotesa · Evidence	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>	Ya	Ya	Ya	tidak
<i>Evidence D</i>	Tidak	Tidak	tidak	ya
<i>Evidence B</i>	Ya	Tidak	Ya	ya
<i>Evidence C</i>	Ya	Tidak	tidak	ya
<i>Evidence E</i>	Tidak	Ya	Ya	Tidak

Sumber: Hartati dan Iswanti, 2008

Berdasarkan Tabel 2.2 dapat dihasilkan pohon keputusan (Gambar 2.9) sebagai berikut:



Keterangan:

A = *evidence* A

B = *evidence* B

C = *evidence* C

D = *evidence* D

E = *evidence* E

H1 = hipotesa 1

H2 = hipotesa 2

H3 = hipotesa 3

H4 = hipotesa 4

y = ya

t = tidak

** = tidak menghasilkan hipotesa

Gambar 2.9 Alternatif Pohon Keputusan

Sumber: Hartati dan Iswanti, 2008

Dilihat dari Gambar 2.9 di atas, 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.

Berdasarkan pohon keputusan pada Gambar 2.9 diatas, maka kaidah yang dapat dihasilkan adalah sebagai berikut:

Kaidah 1: *IF A AND B AND C THEN H1*

Kaidah 2: *IF A AND B AND E THEN H3*

Kaidah 3: *IF A AND E THEN H2*

Kaidah 4: *IF D AND B AND C THEN H4*

2.1.4.3 Mesin Inferensi

Mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, biasa dikatakan sebagai mesin pemikir (*thinking machine*) (Hartati dan Iswanti, 2008: 5).

Menurut Hartati dan Iswanti (2008: 45), dalam melakukan proses inferensi, sistem pakar memerlukan adanya proses pengujian kaidah-kaidah dalam urutan tertentu untuk mencari suatu kondisi yang sesuai dengan kondisi awal atau untuk memastikan kondisi yang sedang berjalan sudah dimasukkan ke dalam *database*. Proses pengujian itu disebut dengan perunutan atau penalaran, yaitu proses pencocokan fakta atau kondisi tertentu yang tersimpan dalam basis pengetahuan maupun pada memori kerja dengan kondisi yang dinyatakan dalam premis atau bagian kondisi pada suatu kaidah atau aturan.

Terdapat beberapa konsep penalaran yang dapat digunakan dalam mesin inferensi (Hartati dan Iswanti, 2008: 45-47), yaitu:

1. Penalaran maju (*forward chaining*)

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.

2. Penalaran mundur (*backward chaining*)

Secara umum, konsep ini diaplikasikan ketika tujuan ditentukan sebagai kondisi atau keadaan awal. Konsep ini disebut juga *goal-driven search*. Arah penalaran atau peruntukan dalam konsep ini berlawanan dengan *forward chaining*. Konsep ini dapat dimodelkan sebagai berikut:

Tujuan

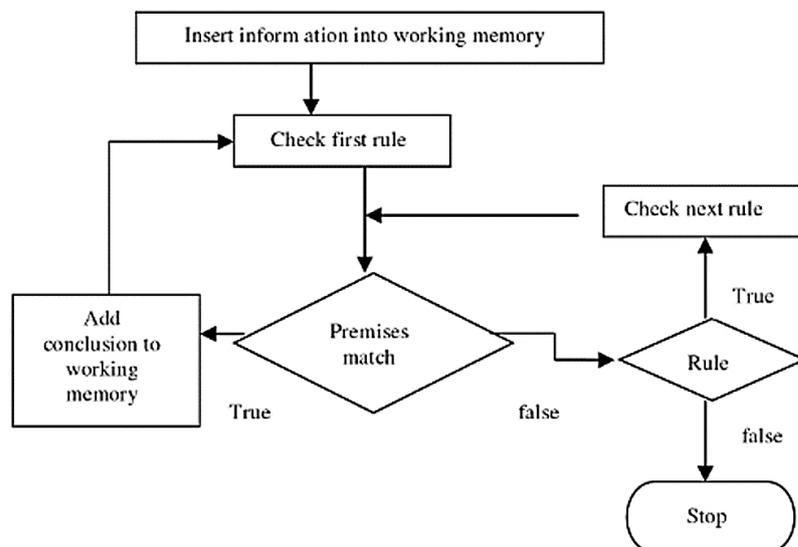
IF (kondisi)

Proses penalaran pada *backward chaining* dimulai dari tujuan kemudian merunut balik ke jalur yang mengarah ke tujuan tersebut, untuk membuktikan bahwa bagian kondisi pada kaidah atau aturan benar-benar terpenuhi. Proses *internal* selalu memeriksa konklusi (tujuan) terlebih dahulu sebagai praduga awal, kemudian memeriksa dan memastikan gejala-gejala (kondisi) telah terpenuhi dan selanjutnya mengeluarkan konklusi sebagai *out put*. Jika sistem menemukan ada bagian kondisi yang tidak terpenuhi maka sistem akan memeriksa konklusi (tujuan) pada aturan atau kaidah berikutnya.

2.1.5 Pelacak Maju (*Forward Chaining*)

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 (Sutojo, dkk.,2011: 171).

Operasi dari sistem *forward chaining* dimulai dengan memasukkan sekumpulan fakta yang diketahui kedalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui (Hayadi, 2016: 11).



Gambar 2. 10 Operasi Sistem *Forward Chaining*
Sumber: Hayadi, 2011

Menurut Hayadi (2016: 11-12) langkah-langkah yang harus dilakukan dalam membuat sistem *forwardchaining* berbasis aturan, yaitu:

1. Pendefinisian masalah

Tahapan ini meliputi pemilihan domain masalah dan akuisisi pengetahuan.

2. Pendefinisian data *input*

Sistem *forwardchaining* memerlukan data awal untuk memulai inferensi.

3. Pendefinisian struktur pengendali data Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan pengaktifan suatu aturan.

4. Penulisan kode awal

Tahap ini berguna untuk menentukan apakah sistem telah menangkap domain pengetahuan secara efektif dalam struktur aturan yang baik.

5. Pengujian sistem

Pengujian sistem dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.

6. Perancangan antarmuka

Antarmuka adalah suatu komponen penting dari suatu sistem. Perancangan antarmuka dibuat bersama-sama dengan pembuatan basis pengetahuan.

7. Penegembangan sistem

Pengembangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan *prototype* sistem.

8. Evaluasi sistem

Pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.

2.1.6 Basis Data (*Database*)

Rosa A.S dan Shalahuddin. M (2011: 44) sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data atau informasi yang sudah diolah dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. *Database Management System (DBMS)* atau sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data (Rosa A.S dan Shalahuddin. M, 2011: 45).

Persyaratan minimal sistem aplikasi disebut *DBMS* (Rosa A.S dan Shalahuddin. M, 2011: 45).

, yaitu:

1. Menyediakan fasilitas untuk mengelola akses data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data yang dilakukan.
4. Mampu menangani *backup* data.

Alur hidup atau *Database Life Cycle (DBLC)* terdapat beberapa fase (Rosa A.S dan Shalahuddin. M, 2011:48-49), yaitu:

1. Analisis kebutuhan (*requirement analysis*)

Hal-hal yang harus dilakukan pada tahapan ini adalah:

- a. Didefinisikan dengan mewawancarai produsen dan pemakai data
 - b. Membuat kontrak spesifikasi basis data
 - c. *Entity Relationship Diagram (ERD)*
2. Desain logik basis data (*logical database design*)

Pada tahap ini harus dibuat rencana logik basis data. Biasanya pada tahap ini dibuat *Conceptual Data Model (CDM)*.

3. Desain fisik basis data (*physical database design*)

Pada tahap ini harus dibuat rencana fisik basis data. Biasanya pada tahap ini dibuat *Physical Data Model (PDM)*.

4. Implementasi

- a. Membuat *QuerySQL*
- b. Aplikasi *DBMS* atau *file*

2.1.7 UML (*Unified Modeling Language*)

UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa A.S dan Shalahuddin. M, 2011: 113).

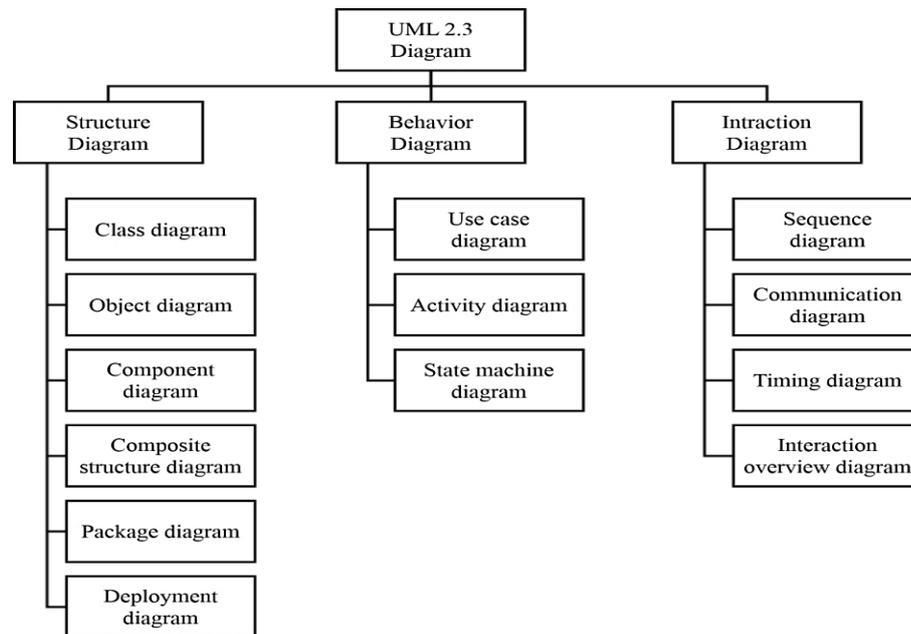
Rosa A.S dan M. Shalahuddin (2011: 118) menyatakan bahwa *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai

sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek.

Pada *UML 2.3* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori, (Rosa A.S dan M. Shalahuddin, 2011: 121) kategori tersebut adalah:

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Terdiri dari 6 macam diagram antara lain: *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.
2. *Behaviour diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Terdiri dari 3 macam diagram antara lain: *use case diagram*, *activity diagram*, dan *state machine diagram*.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Terdiri dari 4 macam diagram antara lain: *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction overview diagram*.

Berikut ini adalah gambar pembagian kategori dan macam-macam diagram tersebut:



Gambar 2. 11 Diagram UML

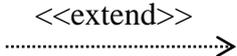
Sumber: Rosa A.S dan Shalahuddin. M, 2011

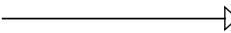
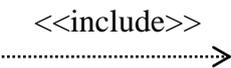
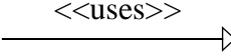
Dalam penelitian ini, 3 diagram yang akan digunakan sebagai desain untuk menggambarkan sistem pakar yang dibuat yaitu:

1. *Use case diagram*

Use case atau diagram *diagram use case* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada 2 hal utama yang terdapat pada *use case* yaitu aktor dan *use case*. Berikut ini adalah simbol-simbol yang digunakan dalam *use case diagram* (Rosa A.S dan Shalahuddin. M, 2011: 130-131).

Tabel 2. 3 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p data-bbox="323 461 440 495"><i>Use case</i></p> 	<p data-bbox="703 461 1342 712">Fungsionalitas yang disediakan sistem sebagai <i>unit-unit</i> yang saling bertukar pesan antar <i>unit</i> atau aktor; dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p data-bbox="323 786 475 819">Aktor/<i>actor</i></p> 	<p data-bbox="703 786 1342 1182">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 <i>actor</i></p>
<p data-bbox="323 1263 580 1296">asosiasi/<i>association</i></p> 	<p data-bbox="703 1263 1342 1442">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i></p>
<p data-bbox="323 1516 528 1550">Ekstensi/<i>extend</i></p> 	<p data-bbox="703 1516 1342 1986">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. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>

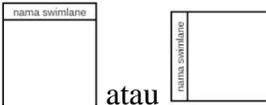
Tabel 2.3 Lanjutan	
<p>generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan/<i>include/uses</i></p>  	<p>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.</p>

Sumber: Rosa A.S dan Shalahuddin. M

2. *Activity diagram*

Diagram aktifitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Berikut ini adalah simbol-simbol yang ada pada *activity diagram* (Rosa A.S dan Shalahuddin. M, 2011: 134) di bawah:

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

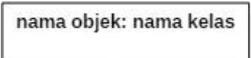
Sumber: Rosa A.S dan Shalahuddin. M

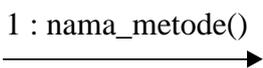
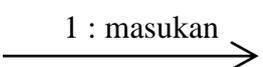
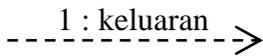
3. *Sequence diagram*

Diagram sekuen (*sequence diagram*) menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang

dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya *diagram* sekuen adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut ini adalah simbol-simbol yang ada pada *sequence diagram* (Rosa A.S dan Shalahuddin. M, 2011: 137-138).

Tabel 2. 5 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p data-bbox="320 1111 480 1144">Aktor/<i>actor</i></p> 	<p data-bbox="670 1111 1348 1440">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>
<p data-bbox="320 1496 576 1529">Garis hidup/<i>lifeline</i></p> 	<p data-bbox="670 1496 1337 1675">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>
<p data-bbox="320 1749 405 1783">Objek</p> 	<p data-bbox="670 1749 1225 1783">Menyatakan objek yang berinteraksi pesan</p>

Tabel 2.5 Lanjutan	
<p>Waktu aktif</p> 	<p>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</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat</p>
<p>pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju</p>
<p>pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Arah panah mengarah pada objek penerima Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri</p>

Sumber: Rosa A.S dan Shalahuddin. M

2.1.8 Bahasa Pemrograman Web

Bahasa pemrograman yang digunakan dalam pembuatan sistem pakar *forward chaining* untuk mendeteksi kerusakan sepeda motor Yamaha Nmax adalah:

1. *PHP: Hypertext Preprocessor (PHP)*



Gambar 2. 12 Logo PHP
Sumber: Saputra, 2012

Menurut Saputra (2012: 91) ada 4 macam format yang bisa digunakan untuk memulai pemrograman *PHP* didalam kode pemrograman yaitu:

- a. `<?php ?>`
- b. `<? ?>`
- c. `<script language="php"> </script>`
- d. `<% %>`

Aturan penulisan variabel dalam *PHP* yang perlu diketahui adalah sebagai berikut (Saputra, 2012: 92):

- a. Penulisan variabel harus diawali dengan simbol *dollar* (\$).
- b. Karakter pertama setelah simbol *dollar*, tidak boleh menggunakan angka (harus huruf).

- c. Setelah simbol *dollar* (\$) dan huruf, maka karakter selanjutnya boleh menggunakan angka.
2. *HTML (Hyper Text Markup Language)*



Gambar 2. 13 Logo HTML
Sumber: Saputra, 2012

HTML merupakan singkatan dari *Hyper Text Markup Language*. *HTML* bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengolah tampilan pada halaman *website*. Menurut wikipedia, *HTML* digunakan untuk menampilkan berbagai informasi didalam sebuah penjelajah *web* internet dengan *formatting hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi (Saputra, 2012: 1). Menurut Saputra (2012: 4-6) beberapa elemen-elemen wajib yang ada pada file *HTML* apabila kita ingin membangun suatu pondasi kerangka *website*:

- a. Elemen *HTML*

Elemen ini merupakan tag dasar apabila kita ingin memulai suatu dokumen *HTML*. Tag ini merupakan perintah wajib bagi pemrogram *web* untuk menulis tag pertama dalam dokumen *html*. Contoh tag-nya adalah:

`</html>` dan diakhiri dengan `</html>`

b. Elemen *Head*

Head merupakan tag berikutnya setelah elemen *html*, yang berfungsi untuk menuliskan keterangan tentang dokumen *web* yang akan ditampilkan. Contoh format penulisannya:

```
<html>
<head>
</head>
</html>
```

c. Elemen *Title*

Element *Title* merupakan suatu elemen yang harus dituliskan didalam elemen *head* yang digunakan untuk memberikan judul/informasi pada *cation browser web* tentang topik/tema atau judul dari suatu dokumen *web* yang ditampilkan pada *browser*. Contoh struktur penggunaannya:

```
<html>
<head>
    <title> Tuliskan Judul disini </title>
</head>
</html>
```

d. Elemen *Body*

Elemen *body* merupakan bagian utama dalam dokumen *web*. Jika kita ingin menampilkan suatu teks atau informasi (konten), maka kita harus meletakkan teks tersebut pada elemen *body*. Struktur elemennya sebagai berikut:

```
<html>
<head>
```

```
<title> Tulis Judul Disini </title>  
  
</head>  
  
<body>  
  
Tuliskan Konten Disini  
  
</body>  
  
</html>
```

3. CSS (*Cascading Style Sheet*)



Gambar 2. 14 Logo CSS
Sumber: Saputra, 2012

Menurut Saputra (2012: 27) CSS (*Cascading Style Sheet*) merupakan bahasa pemrograman *web* yang didesain khusus untuk mengendalikan dan membangun berbagi komponen didalam *web* sehingga tampilan *web* lebih rapi, terstruktur dan seragam. Tujuan utama dari CSS adalah memisahkan konten utama dengan tampilan dokumen lainnya (*html* dan sejenisnya). Tujuan lainnya adalah untuk mempercepat pembuatan halaman *web*.

Keuntungan yang menggunakan CSS adalah (Saputra, 2012: 28-29):

1. Memisahkan pembuatan dokumen (CSS dan HTML).

2. Mempermudah dan mempersingkat pembuatan dan pemeliharaan dokumen *web*.
3. Akses *web* lebih cepat saat *loading* (mempercepat pembacaan HTML).
4. Fleksibel, interaktif, tampil lebih menarik dan nyaman dipandang.
5. Lebih kecil ukuran *file* sehingga *bandwidth* yang digunakan juga otomatis menjadi lebih kecil.
6. Dapat digunakan pada semua *web browser*.

Menurut Saputra (2012: 29-32) untuk menggunakan CSS, setidaknya ada 3 cara yang bisa kita gunakan, yaitu:

a. *Embedded Style Sheet*,

Merupakan cara penulisan kode dimana penulis CSS dilakukan pada tag *html*, yaitu pada tag `<style> ... </style>` dan sebelum tag `<body>`.

b. *Inline Style Sheet*,

Metode *Inline style sheet*, yang merupakan salah satu cara penggunaan CSS langsung pada tag *html* yang dibutuhkan saja. Cara ini dilakukan karena hanya sedikit properti yang dibutuhkan.

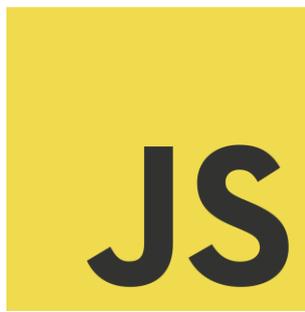
c. *Linked Style Sheet*,

Metode ini merupakan cara pengerjaan dimana antara kode css dan *html* telah dipisahkan. Untuk menggunakan kode css yang telah dipisahkan ini, maka dalam kode *html* dibuat skrip yang isinya adalah memanggil *file css* tersebut untuk digunakan dalam kode *html*. Contoh *link* untuk memanggil kode css ini:

```
<link rel="stylesheet" href="NamaFile.css" type="text/css">
```

4. *JavaScript dan jQuery*

JavaScript adalah bahasa *scripting* handal yang berjalan pada sisi *client*. *JavaScript* merupakan sebuah bahasa *scripting* yang dikembangkan oleh *Netscape*. Untuk menjalankan *script* yang ditulis dengan *JavaScript* kita membutuhkan *JavaScript-enablebrowser* yaitu *browser* yang mampu menjalankan *JavaScript* (Saputra, 2012: 3). Berikut ini adalah logo *JavaScript* dan *jQuery*:



Gambar 2. 15 Logo *JavaScript*
Sumber: Saputra, 2012



Gambar 2. 16 Logo *jQuery*
Sumber: Saputra, 2012

2.2 Variabel Penelitian

Variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi

dan kesimpulannya (Sudaryono, 2015: 16). Obyek yang digunakan dalam penelitian ini adalah mesin sepeda motor Yamaha Nmax.

2.2.1 Sepeda Motor

Sepeda motor terdiri dari beberapa komponen dasar. Bagaikan kita manusia, kita terdiri atas beberapa bagian, antara lain bagian rangka, pencernaan, pengatur sirkulasi darah, panca indera dan lain sebagainya. Maka sepeda motorpun juga seperti itu, ada bagian-bagian yang membangunnya sehingga ia menjadi sebuah sepeda motor. Secara kelompok besar maka komponen dasar sepeda motor terbagi atas sistem mesin, sistem kelistrikan, dan rangka/*chassis* (Jama, 2008: 17-18).

2.2.2 Mengenal Sepeda Motor Matic

Sepeda motor masih menjadi alat transportasi yang paling populer bagi sebagian besar masyarakat Indonesia. Pendapatan masyarakat yang relatif masih rendah, infrastuktur lalu lintas yang belum memadai dan kemudahan dalam pembiayaan ditenggarai menjadi faktor penyebab penggunaan sepeda motor. Tingginya animo masyarakat terhadap keberadaan sepeda motor telah menempatkan Indonesia sebagai pangsa pasar paling potensial. Pabrikan seolah berlomba melancarkan jurus untuk merebut pangsa pasar. Salah satunya adalah menambah varian baru sepeda motor yang didukung oleh *hi-tech* yang disesuaikan dengan karakteristik masyarakat Indonesia.

Tipe matic merupakan sepeda motor keluaran terbaru yang mendapatkan apresiasi luar biasa di tengah-tengah pasar sepeda nasional saat ini. Bahkan menurut data dari Asosiasi Industri Sepeda Motor Indonesia (AISI), sepeda motor

matic mengalami pertumbuhan yang cukup signifikan dari tahun ke tahun. Ini, tentunya tidak lepas dari kejelian dari pabrikan dalam menangkap tuntutan masyarakat yang membutuhkan sepeda motor yang efisien, praktis, dan nyaman untuk dikendarai. Tipe matic memberikan kenyamanan karena menggunakan transmisi otomatis, sehingga pengendara tidak perlu repot memindahkan gigi (persneling). Selain tingkat kenyamanan, kelebihan lain yang terdapat pada sepeda motor tipe matic adalah biaya perbaikan dan perawatan cenderung lebih murah bila dibandingkan dengan tipe sepeda motor lainnya. Misalnya, apabila terjadi kerusakan pada bagian penghubung transmisi, cukup V-belt-nya aja yang diganti, tentunya dengan harga yang relatif terjangkau. Lain halnya pada tipe selain matic, jika terjadi kerusakan pada bagian transmisi, satu set rantai roda harus diganti. Dengan makin banyaknya pabrikan yang terjun pada segmen ini, bukan tidak mungkin kedepan akan terjadi peralihan dari tipe bebek ke tipe matic.

Tranmisi otomatis atau dikenal dengan sebutan *Continuous Variable Transmission (CVT)* telah lama digunakan didunia otomotif termasuk sepeda motor. Penggunaan komponen CVT asalnya dari paten DAF Belanda yang tentunya saja dasar teknologinya telah melewati masa perlindungan paten. Di Tanah Air, penggunaan trasmisi otomatis pada sepeda motor telah dimulai pada tahun 1980-an. Pada tahun 1984, pabrikan asal Taiwan memasukkan skutik Adly. Skutik asal Taiwan tersebut merupakan sepeda motor pertama bertransmisi CVT mesin 2-tak berkapasitas 50 cc.

Namun, masyarakat saat itu belum mengerti teknologi CVT. Produk ini dianggap terlalu maju di era itu, terlebih lagi jaringan penjualan dan kesediaan

suku cadangnya tidak merata. Terbukti sambutan masyarakat terhadap kehadiran sepeda motor Adly tidak begitu sukses. Tahun 1992 Piaggio coba menawarkan kembali skuter bertransmisi CVT lewat Corsa. Penjualan Corsa ternyata tidak menggembirakan hingga akhirnya dihentikan produksinya pada tahun 1998.

Selain modelnya yang dianggap ketinggalan dibandingkan dengan sepeda motor Jepang, sepeda motor ini tidak laku karena masyarakat belum siap menerima produk sepeda motor bertransmisi CVT. Kultur masyarakat Indonesia yang cenderung konservatif dalam menerapkan teknologi baru menjadikan penggemar skuter matic lebih sedikit dibandingkan dengan sepeda motor tipe bebek. Selain itu, desain dan model yang kurang mencerminkan kebutuhan masyarakat dan kurangnya promosi menjadi penyebab kurang lakunya sepeda motor tipe matic di pasaran.

Setelah Piaggio, produsen Kymco mencoba membuka pasar skutik dengan produk bernama Trend pada tahun 2000. Mesin 4-tak berkapasitas 125 cc dan tampilan futuristik dijadikan andalan sebagai pemikat. Dari sinilah konsumen mulai melirik sepeda motor bertransmisi CVT dan meledak ketika Yamaha Mio muncul pada tahun 2004.

Lain dulu lain sekarang. Anggapan sulit perawatan dan suku cadang tak lagi ditakuti, terkalahkan oleh kepraktisan yang ditawarkan. Apalagi jalan-jalan di kota besar sekarang semakin padat dan cukup merepotkan jika menggunakan motor bertransmisi manual. Perkembangan teknologi yang cukup pesat, model yang mengikuti selera pasar, dan adanya perubahan dalam gaya *berkendara (life style motorcycle)* menjadi pemicu untuk menggunakan sepeda motor matic.

Itulah sekilas sejarah hadirnya skutik bertransmisi CVT ke Indonesia. Kini hampir semua produsen menghadirkan produk sepeda motor bertransmisi CVT.

Melihat pasar yang ada, beberapa pabrikan telah mengeluarkan tipe matic dengan menonjolkan fitur-fitur dan kecanggihan teknologi yang berbeda, sebut saja misalnya matic dengan kapasitas cc yang besar, menggunakan radiator sebagai alat pendingin, fitur *side stand switch* (mematikan mesin saat standar dibuka), dan parking *brake lock key shutter* (Subandrio, 2009: 1- 4).

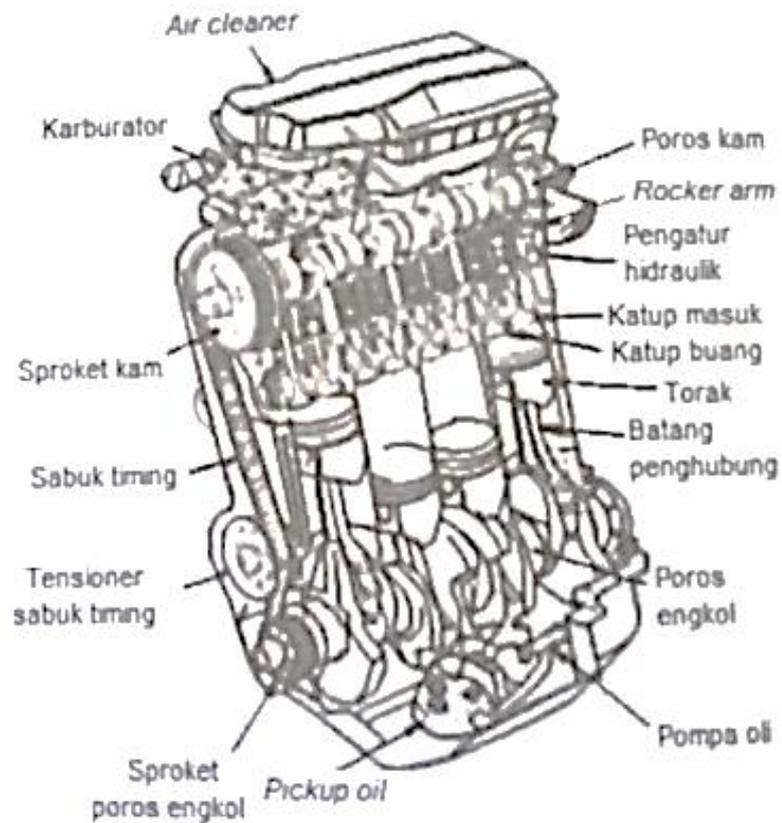
2.2.3 Klasifikasi Motor

Motor dapat dibedakan menjadi beberapa golongan antara lain berdasarkan sistem penyalaan, susunan silinder, lokasi katup, proses pasokan udara. Di bawah ini adalah penjelasannya (Kristanto, 2015: 1):

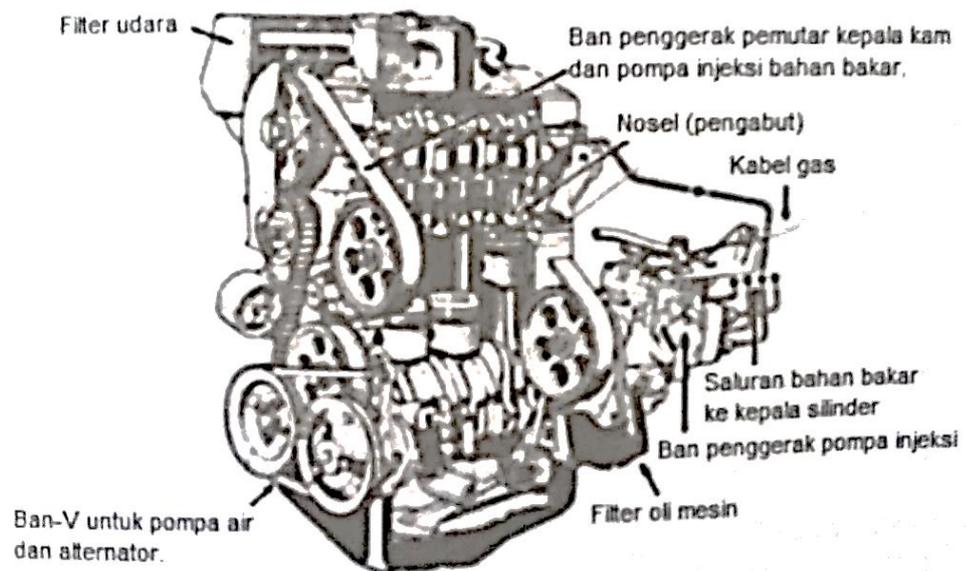
2.2.3.1 Klasifikasi Motor Berdasarkan Sistem Penyalaan

Berdasarkan sistem penyalaan, motor dapat diklasifikasikan menjadi (Kristanto, 2015: 2):

1. Motor bensin atau motor *otto* (Gambar 2.17)
2. Motor *Diesel* (Gambar 2.18)



Gambar 2. 17 Motor Bensin atau Motor *Otto*
Sumber: Kristanto, 2015

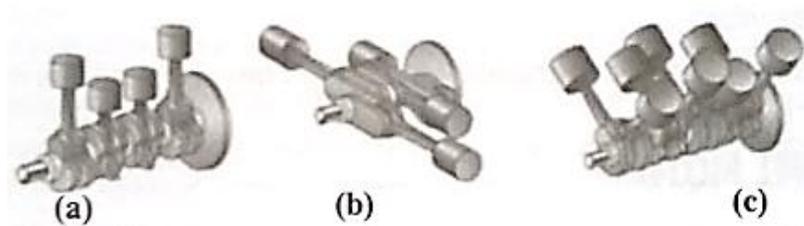


Gambar 2. 18 Motor *Diesel*
Sumber: Kristanto, 2015

Pada motor bensin, campuran udara-bahan bakar dinyalakan oleh percikan bunga api listrik di antara kedua elektroda busi, sehingga motor bensin juga dikenal sebagai motor pengapian percik (*Spark Ignition Engines*). Pada motor diesel atau motor penyalan kompresi (*Compression Ignition Engines*) terjadi proses penyalan sendiri bahan bakar yang diinjeksikan ke dalam aliran udara panas dengan tekanan dan temperatur tinggi akibat proses kompresi di dalam silinder.

2.2.3.2 Klasifikasi Motor Berdasarkan Susunan Silinder

Berdasarkan susunan silindernya, pada umumnya motor bakar dapat dibedakan menjadi 3 jenis yaitu sebagai berikut (Kristanto, 2015: 3):



Gambar 2. 19 Susunan Silinder Motor
Sumber: Kristanto, 2015

1. Motor Sebaris (*In-line*). Gambar 2.19a menunjukkan motor yang pistonnya disusun berderet satu sama lain. Model ini adalah yang paling umum, membutuhkan banyak ruang karena panjang panjang blok silinder yang diperlukan untuk tempat torak, terutama pada motor dengan jumlah silinder lebih dari 4.
2. Motor V. Gambar 2.19b, silinder ditata dalam bentuk V. Poros engkol dapat dibuat lebih pendek dan lebih kompak. Ketika penyalan (*firing*) terjadi pada

setiap *bank* (deretan *piston* dalam satu baris), mereka membantu untuk menyeimbangkan satu sama lain dan menyediakan kelancaran dalam pengiriman tenaga.

3. Motor-datar (*flat engine*). Gambar 2.19c, kadang-kadang disebut motor *boxer* atau motor *horizontal* berlawanan. Pada motor jenis ini, torak ditata mendatar di setiap sisi poros engkol. Ini berarti pada motor 4 silinder *horizontal* berlawanan akan terdapat dua torak disetiap sisi poros engkol. Sebuah mesin datar menyediakan pusat gravitasi rendah dan pergerakan poros engkol relatif singkat karena hanya ada dua piston berdampingan. Ini membuat mesin menjadi kompak.

2.2.3.3 Klasifikasi Motor Berdasarkan Lokasi Katup

Menurut Kristanto (2015: 3) berdasarkan lokasi katupnya motor dapat dibedakan menjadi:

1. Motor katup samping.
2. Motor katup di kepala.
3. Motor katup kombinasi (disamping dan di atas kepala).

2.2.3.4 Klasifikasi Motor Berdasarkan Proses Pasokan Udara

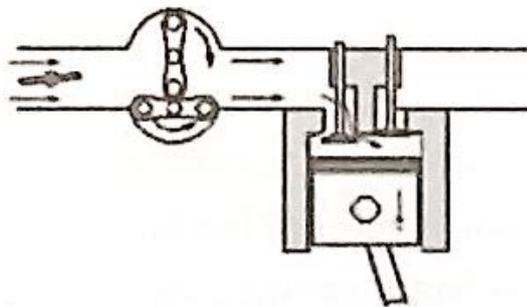
Berdasarkan proses pasokan udara motor dapat dibedakan menjadi (Kristanto, 2015:3):

- a. Alami

Proses pemasukan udara kedalam silinder dilakukan secara alami tanpa menggunakan sistem tekanan dorong (*boost pressure*) udara masukan, dan hanya mengandalkan tekanan hisap motor.

b. Supercarjer

Tekanan udara masuk ditingkatkan dengan kompresor yang digerakkan oleh poros engkol motor (Gambar 2.20).

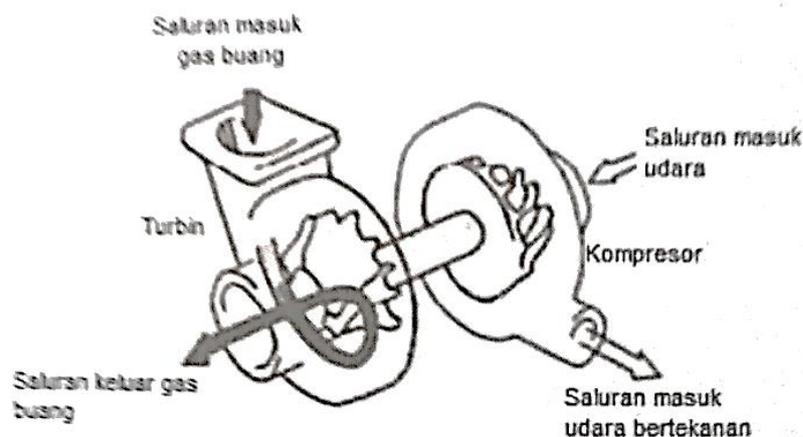


Gambar 2. 20 Supercarjer untuk meningkatkan tekanan pasokan udara

Sumber: Kristanto, 2015

c. Turbocarjer

Tekanan udara masukan ditingkatkan dengan turbin kompresor yang digerakkan oleh gas buang motor (Gambar 2.21).



Gambar 2. 21 *Turbocarjer* digunakan untuk meningkatkan tekanan pasokan udara dengan memanfaatkan tekanan gas buang untuk memutar *turbin*

Sumber: Kristanto, 2015

d. Kompresi melalui karter (bak engkol)

Siklus motor dua langkah dengan menggunakan bak engkol (karter) sebagai penekan udara masukan.

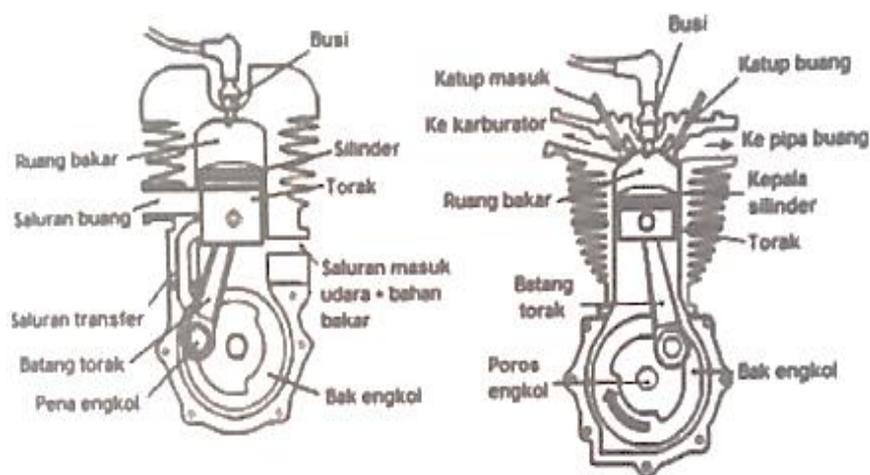
2.2.3.5 Klasifikasi Motor Berdasarkan Sistem Pendingin

Menurut Kristanto (2015: 4), berdasarkan sistem pendingin motor dibedakan menjadi:

1. Motor berpendingin udara
2. Motor berpendingin cairan yang didinginkan, yang pada umumnya adalah air.

2.2.3.6 Klasifikasi Motor Berdasarkan Siklus Operasi / Langkah kerja

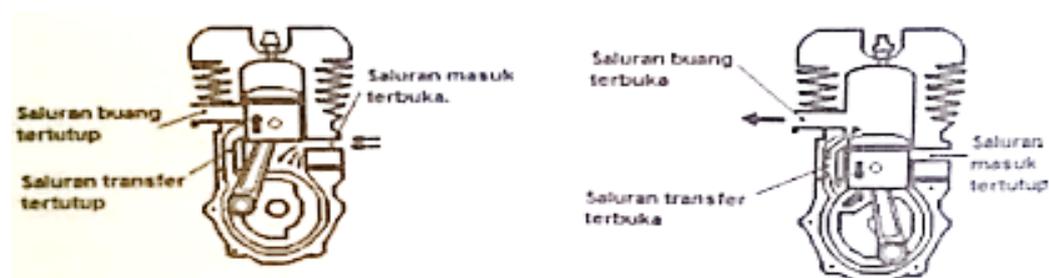
Berdasarkan siklus operasinya, motor pembakaran dalam, baik motor bensin maupun motor diesel, dibedakan menjadi motor 2-langkah (2-tak) dan motor 4-langkah (4-tak) yaitu sebagai berikut (Kristanto, 2015: 9):



Gambar 2. 22 (a) Motor siklus 2-tak (b) Motor siklus 4-tak
Sumber: Kristanto, 2015

1. Motor 2 Langkah

Menurut Kristanto (2015: 9) Motor 2 langkah (Gambar 2.22a) hanya membutuhkan satu kali putaran poros engkol untuk menyelesaikan satu siklus di dalam silinder. Kerja (langkah daya) dihasilkan ada setiap putaran poros engkol. Motor 2 langkah beroperasi tanpa katup. Sebagai pengganti katup, kebanyakan mesin dua langkah menggunakan lubang saluran di dinding silinder yang dibuka dan ditutup oleh torak ketika bergerak naik-turun di dalam silinder. Beberapa motor 2 langkah menggunakan katup pasif atau kelopak penutup yang disebut katup buluh untuk membantu menutup bak engkol setelah campuran udara/bahan bakar dihisap. Jika seluruh proses pada motor 4 langkah berlangsung di atas torak, maka motor 2 langkah juga memanfaatkan area di bawah torak (*crankcase*) untuk mempercepat operasi. Proses kompresi pada motor 2 langkah terjadi dua kali setiap putaran. Kompresi utama atau *pre-compression* berlangsung di dalam bak engkol, dimana campuran udara-bahan bakar ditarik ke dalam bak engkol dan selanjutnya dikompresi melalui gerakan engkol dan masuk ke ruang bakar. Kompresi kedua berlangsung di dalam silinder dan ruang bakar sehingga dihasilkan tekanan tinggi untuk menyalakan campuran udara-bahan bakar dengan bantuan busi. Prinsip kerja motor dua langkah ditunjukkan dalam Gambar 2.23.



Gambar 2. 23 Prinsip kerja motor 2-langkah (a) langkah hisap dan kompresi (b) langkah daya dan buang

Sumber: Kristanto, 2015

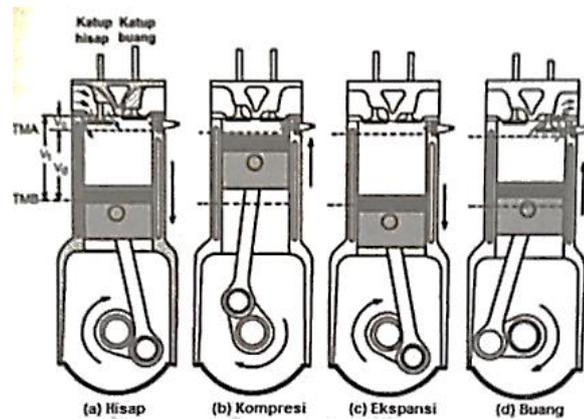
- a. Langkah Hisap, saat torak bergerak menuju ke TMA (setengah putaran pertama atau 180°), terjadi kevakuman di ruang engkol (bagian bawah torak) dan saluran masuk terbuka. Karena adanya beda tekanan tersebut, udara luar dihisap masuk dan bercampur dengan bahan bakar di *karburator* yang selanjutnya masuk ke ruang engkol. Proses ini disebut langkah hisap.
- b. *Pre-compression*, saat torak bergerak kebawah (setengah putaran kedua atau 180°) menutup lubang saluran masuk dan meremas (*Squeezes*) campuran udara/bahan bakar di dalam bak engkol sehingga tekanannya meningkat. Saat torak terus bergerak ke bawah, saluran *transfer* pada dinding silinder terbuka. Campuran bertekanan didesak naik ke atas di atas *piston*. (Pada titik ini, saluran buang di dinding silinder di seberang saluran *transfer* juga terbuka).
- c. Langkah kompresi, piston bergerak ke TMA (menutup saluran *transfer* dan buang) dan memampatkan campuran udara/bahan bakar ke dalam ruang bakar.
- d. Langkah daya, tepat sebelum torak mencapai TMA, busi menyalakan campuran udara/bahan bakar dan tekanan tinggi yang diciptakannya mendorong torak ke TMB untuk melakukan langkah daya.
- e. Langkah buang, saat torak bergerak kebawah melakukan langkah daya, saluran buang dan campuran yang masuk dari saluran *transfer* membantu mendorong gas terbakar keluar dari mesin. Proses ini disebut pembilasan.

2. Motor 4 Langkah

Menurut Kristanto (2015: 10) motor 4 langkah (Gambar 2.22b) membutuhkan dua kali putaran poros engkol untuk menyelesaikan satu siklus di dalam silinder. Dengan kata lain, setiap silinder membutuhkan empat langkah torak pada dua putaran poros engkol untuk melingkupi siklusnya. Prinsip kerja motor empat langkah ditunjukkan dalam Gambar 2.24.

- a. Langkah hisap, diawali dengan posisi torak TMA dan berakhir dengan posisi torak di TMB, yang mana menghisap campuran segar kedalam silinder. Untuk meningkatkan massa campuran yang dihisap, katup masuk terbuka sesaat sebelum langkah hisap dimulai dan menutup setelah berakhirnya langkah tersebut.
- b. Langkah kompresi, ketika kedua katup tertutup dimana campuran di dalam silinder dimampatkan dan *volumenya* diperkecil. Menjelang akhir langkah kompresi, pembakaran diaktifkan dan tekanan silinder naik dengan cepat.
- c. Langkah daya, atau langkah ekspansi, diawali dengan posisi torak di TMA dan berakhir di TMB ketika temperatur dan tekanan gas yang tinggi mendorong torak kebawah dan memaksa poros engkol untuk berutar. Ketika torak mendekati TMB, katup buang terbuka untuk mengawali proses buang dan tekanan silinder turun mendekati tekanan buang.
- d. Langkah buang, dimana sisa gas yang dibakar keluar dari silinder disebabkan tekanan silinder yang pada hakikatnya lebih tinggi dibanding tekanan buang. Gas kemudian di dorong keluar oleh torak ketika bergerak

ke TMA. Ketika torak mendekati TMA, katup masukan terbuka. Sesaat setelah TMA, katup buang menutup dan siklus dimulai lagi.



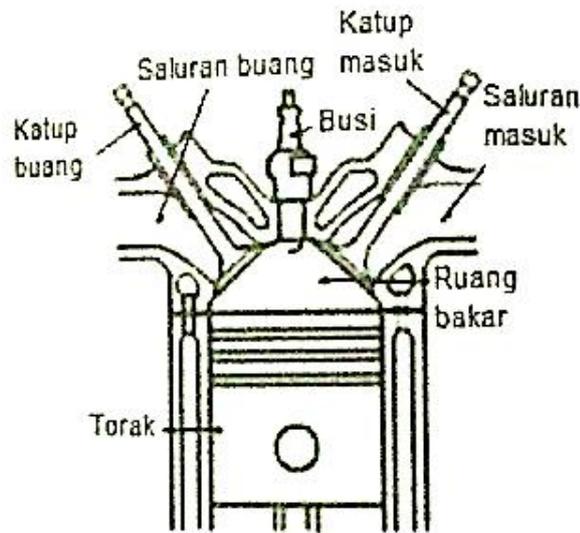
Gambar 2. 24 Prinsip kerja motor 4-langkah
Sumber: Kristanto, 2015

Untuk memperoleh keluaran daya yang lebih tinggi dari ukuran motor yang ditentukan, dan desain katup yang lebih sederhana, dikembangkan siklus dua langkah. Siklus dua langkah juga dapat digunakan baik untuk motor pengapian percikan maupun pengapian kompresi.

2.2.4 Ruang Bakar dan Pengapian

2.2.4.1 Ruang Bakar

Ruang bakar adalah area di dalam silinder dimana campuran bahan bakar/udara dimampatkan dan kemudian dinyalakan. Ruang ini dibatasi oleh kepala silinder dan kepala torak (*piston crown*) dengan posisi torak di TMA (titik mati atas) pada saat langkah kompresi (Kristanto, 2015: 156), dapat dilihat pada (gambar 2.25) dibawah ini:



Gambar 2. 25 Ruang bakar motor bensin
Sumber: Kristanto, 2015

Jama, dkk (2008: 22) mengungkapkan bahwa nilai kalor (panas) bahan bakar perlu kita ketahui, agar neraca kalor dari motor dapat dibuat. Efisiensi atau tidak kerjanya suatu motor, ditinjau atas dasar nilai kalor bahan bakarnya. Nilai kalor mempunyai hubungan dengan berat jenis. Pada umumnya makin tinggi berat jenis maka makin rendah nilai kalornya. Pembakaran dapat berlangsung dengan sempurna, tetapi juga dapat tidak sempurna. Pembakaran yang kurang sempurna dapat berakibat:

1. Kerugian panas dalam motor menjadi besar, sehingga efisiensi motor menjadi turun, usaha dari motor menjadi turun pula pada penggunaan bahan bakar yang tetap.
2. Sisa pembakaran dapat menyebabkan pegas-pegas piston melekat pada alurnya, sehingga ia tidak berfungsi lagi sebagai pegas torak.

3. Sisa pembakaran dapat pula melekat pada lubang pembuangan antara katup dan dudukannya, terutama pada katup buang, sehingga katup tidak dapat menutup dengan rapat.
4. Sisa pembakaran yang telah menjadi keras yang melekat antara *piston* dan dinding silinder, menghalangi pelumasan, sehingga piston dan silinder mudah aus.

Efisiensi bahan bakar dan efisiensi panas sangat menentukan bagi efisiensi motor itu sendiri. Masing-masing motor mempunyai efisiensi yang berbeda.

2.2.4.2 Sistem pengapian

Menurut Kristanto (2015: 183) sistem pengapian terbagi menjadi 3 jenis yaitu :

1. Sistem pengapian mekanik, dibedakan menjadi pengapian baterai menggunakan pemutus kontak (*platina*) di dalam distributor untuk memulai percikan dan pengapian magnet (*magneto ignition*) yang menggunakan magnet.
2. Sistem pengapian elektronik, tidak membutuhkan pemutus kontak. Ketika sinyal diterima, pemantikan api dilakukan oleh transistor yang terdapat di dalam modul pengapian. Pengapian elektronik dibedakan menjadi pengapian lecutan kapasitis (*Capacitive Discharge Ignition, CDI*) dan pengapian koil *transistor*.

3. Sistem pengapian langsung, (*Direct Ignition System*) tidak memiliki distributor dan menggunakan koil untuk memasok tegangan tinggi secara langsung ke busi.

Pada kendaraan bermotor yang sudah menerapkan sistem injeksi, memiliki bagian yang berfungsi untuk mengontrol dan mengatur pasokan udara dan bahan bakar ke dalam ruang pembakaran secara efektif dan efisien. Bagian kontrol ini terdapat sensor (berupa elektronik) yang akan mengatur jumlah udara dan bahan bakar secara homogen sesuai dengan kebutuhan mesin. Selama sensor bekerja dengan baik, kemungkinan kerusakan sangat kecil. Sistem *throttle body* pasokan bahan bakar yang terletak di throttle body langsung ke ruang asupan sedangkan sistem titik tunggal akan memasok bahan bakar dari *injektor* tunggal.

Pada motor matic injeksi membutuhkan perangkat bernama injector, yang bertugas menyuplai campuran bahan bakar dengan udara. Sistem injeksi merupakan teknologi penerus sistem karburator pada kendaraan bermotor. Cara kerja motor matic injeksi apabila pada sistem karburator, kendaraan membutuhkan penyetelan yang tepat agar bisa mendapatkan campuran bahan bakar dan udara atau AFR (*Air-fuel ratio*) yang optimal, sistem injeksi sudah terprogram secara komputer untuk mendapatkan rasio AFR yang optimal. Supaya bisa mendapatkan AFR yang optimal, injektor mengandalkan program komputer untuk mengontrol AFR nya. Perangkat elektronik yang bertugas untuk mengontrol kerja injektor ini bernama ECM atau *Electronic Control Module* (Anggri Sartika Wiguna, 2017: 26)

2.2.5 Sistem Transmisi

Sistem transmisi merupakan bagian komponen mesin sepeda motor yang berfungsi sebagai pemindah tenaga dari mesin ke roda belakang. Sepeda motor matic menggunakan sistem transmisi otomatis, yaitu tenaga dari *crankshaft* (poros engkol) diteruskan ke roda belakang lewat bantuan dua buah *pulley* yang dihubungkan dengan *V-belt*. Pada sistem transmisi otomatis tidak diperlukan adanya pemindahan gigi (persneling) seperti pada sepeda motor pada umumnya.

Teknologi yang digunakan pada sistem transmisi otomatis dikenal dengan sebutan CVT. Pada teknologi ini, tenaga dari mesin dapat tersalurkan dengan sempurna ke roda belakang dengan menyesuaikan perubahan kecepatan dan perubahan torsi kendaraan, tentunya dengan ratio yang sangat tepat, sehingga percepatan yang dihasilkan lebih konstan dan bebas hentakan. Transmisi CVT disalurkan melalaui sabuk yang disebut *V-belt*. Sabuk *Vbelt* terbuat dari campuran serat dan bahan kimia dengan karet khusus yang mempunyai daya tahan tinggi, awet, dan efisien (Subandrio, 2009: 19)



Gambar 2. 26 *V-belt*
Sumber: Subandrio, 2009

2.2.6 Klep dan *Filter*

2.2.6.1 Klep

Setelan klep (*valve clearance*) mengatur masuknya bahan bakar ke dalam ruang bakar dan mengatur pembuangan gas sisa pembakarannya. Jika setelan klep terlalu rapat, saat memuai bisa terjadi kebocoran kompresi, namun jika terlalu renggang akan berisik dan bukaannya pun tidak maksimal (Rickieno, 2008: 40).

2.2.6.2 Filter

Filter udara harus dibersihkan dari kotoran agar udara bersih yang tercampur dengan bahan bakar dan juga alirannya akan lancar (Rickieno, 2008:41). Solusi untuk permasalahan yang terjadi pada *filter* atau saringan yaitu dengan melihat kondisi *filter* tersebut, jika masih bagus (tidak sobek) maka hanya cukup dibersihkan. Tetapi jika ditemukan sobek, patah atau sejenisnya, maka *filter* harus diganti.

2.2.7 Kelistrikan

Setiap sepeda motor dilengkapi dengan beberapa rangkaian sistem kelistrikan. Umumnya sebagai sumber listrik utama sering digunakan baterai, namun ada juga yang menggunakan *flywheel magnet (alternator)* yang menghasilkan pembangkit listrik arus bolak-balik atau AC (*alternating current*). Bagian-bagian yang termasuk sistem kelistrikan pada sepeda motor antara lain; sistem *starter*, sistem pengapian (*ignition system*), sistem pengisian (*charging system*), dan sistem penerangan (*lighting system*) seperti lampu kepala/depan

(*headlight*), lampu belakang (*tail light*), lampu rem (*brake light*), lampu sein/tanda belok (*turn signal lights*), klakson (*horn*) dan lampu-lampu instrumen/*indikator* (Jama, dkk, 2008: 85).

Mekanisme kelistrikan dipakai untuk menghasilkan daya pembakaran untuk proses kerja mesin dan sinyal untuk menunjang keamanan berkendara. Jadi semua komponen yang berhubungan langsung dengan energi listrik dikelompokkan menjadi bagian kelistrikan. Bagian kelistrikan terbagi menjadi (Jama, dkk, 2008: 19):

1. Kelompok pengapian
2. Kelompok pengisian
3. Kelompok beban (lampu-lampu dan alat-alat pemakai kelistrikan lainnya).

Menurut Rickieno (2008: 59) untuk mengatasi masalah kelistrikan ini yaitu dengan memperhatikan beberapa hal berikut:

1. Menghindari atau perbaiki jika ada saluran kelistrikan yang kendur.
2. Menghindari atau perbaiki saluran kabel kelistrikan yang terjepit.
3. Melakukan penempatan atau pengaturan yang terlalu kencang.
4. Melapisi atau tidak membiarkan saluran kelistrikan yang terbuka
5. Menjauhkan susunan kabel dari perangkat mesin yang panas, seperti knalpot atau blok mesin.

Sistem kelistrikan pada sepeda motor matic merupakan bagian yang sangat penting karena sistem ini menyediakan arus listrik untuk keperluan pembakaran pada mesin dan untuk menyuplai arus listrik pada kelengkapan-kelengkapan pendukung sepeda motor matic agar dapat dikendarai dan berfungsi sebagai alat

transportasi. Karena dengan adanya sistem kelistrikan tersebut, maka fungsi mekanik lainnya bisa bersinergi untuk melakukan proses kerja (Marsudi, 2016: 171).

2.3 Software Pendukung

Software pendukung merupakan beberapa perangkat lunak yang digunakan untuk mendukung pembuatan sistem pakar dalam penelitian ini. Perangkat lunak tersebut antara lain: *StarUML*, *XAMPP*, *Sublime Text*, *phpMyAdmin*, dan *Mozilla Firefox*.

2.3.1 StarUML



Gambar 2. 27 Logo *StarUML*
Sumber: Triandini dan Suardika, 2012

StarUML merupakan proyek *open source* untuk mengembangkan *platform Unified Modeling Language (UML)* atau *Model Driven Architecture (MDA)* yang cepat, fleksibel, dapat diperluas, memiliki banyak fitur, dan tidak dipungut biaya. Tujuan dari proyek ini dalam untuk membangun sebuah perangkat lunak pemodelan dan sekaligus *platform* yang dapat menggantikan perangkat *UML* berbayar lain, seperti *Rational Rose*, *Together*, dan sebagainya. *StarUML* dikembangkan dalam bahasa pemrograman *Delphi*. Walaupun begitu, *StarUML* merupakan proyek yang *multi-lingual* dan tidak bergantung pada bahasa

pemrograman yang spesifik, sehingga bahasa pemrograman apapun dapat digunakan untuk mengembangkan *StarUML*, seperti *C/C++*, *Java*, *VisualBasic*, *Delphi*, *Jscript*, *VBScript*, *C#*, *VB.NET*, dan sebagainya (Iswari, 2015: 73).

StarUML adalah *platform* pemodelan perangkat lunak yang mendukung *UML(Unified Modeling Language)*. *StarUML* yang berbasis pada *UML* versi 1.4, menyediakan sebelas jenis diagram yang berbeda dan mendukung notasi *UML 2.0* (Triandini dan Suardika, 2012: 1).

Triandini dan Suardika (2012: 2-3) fitur-fitur utama dalam *StarUML* versi 5.0 adalah sebagai berikut:

1. Dukungan terhadap Diagram *UML 2.0*:
 - a. *Use Case Diagram*
 - b. *Clas Diagram*
 - c. *Sequence Diagram*
 - d. *Collaboration Diagram*
 - e. *Statechart Diagram*
 - f. *Activity Diagram*
 - g. *Componen Diagram*
 - h. *DeploymentDiagram*
 - i. *Composite Structure Diagram (UML 2.0)*
2. Dukungan terhadap beberapa bahasa pemrograman:
 - a. *Java Profile, Code Generator, dan Reverse Engineer*
 - b. *C++ Profile, Code Generator, dan Reverse Engineer*
 - c. *C# Profile, Code Generator, dan Reverse Engineer*

- d. *Microsoft Office Document Generator*
- e. *Microsoft Word document template and generation*
- f. *Microsoft Excel document template and generation*
- g. *Microsoft PowerPoint document template and generation*
3. *Customizable CodeGeneration*
4. Mendukung teknologi *MDA*
5. Diagram yang dapat diperluas
6. *Extensibility*
7. Kompabilitas yang tinggi
8. *Editing*
9. *UserInterface*
10. *Modelverification*
11. *PatternSupport*

2.3.2 XAMPP



Gambar 2. 28 Logo XAMPP

Sumber: Wicaksono dan Community, 2008

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis *PHP* dan menggunakan pengolahan data *MYSQL* di komputer lokal. *XAMPP* berperan sebagai *server web* pada komputer. *XAMPP* juga dapat

disebut sebuah *CPanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet* (Wicaksono dan Community, 2008: 7).

2.3.3 Sublime Text



Gambar 2. 29 Logo Sublime Text
Sumber: <https://www.sublimetext.com/>

Sublime Text adalah perangkat *text editor* canggih yang dapat digunakan untuk menulis bahasa pemrograman. *Sublime Text* mendukung sejumlah bahasa pemrograman diantaranya C, C++, C#, PHP, CSS, HTML, ASP dan lain-lainnya.

Ada beberapa kelebihan yang ada pada *Sublime Text* diantaranya (<https://www.sublimetext.com/>):

1. *Goto Anything*, digunakan untuk membuka *file* diawali dengan menarik satu *project file* yang sedang dikerjakan dengan menekan CTRL+P maka dapat dicari *file* apa yang akan dibuka dengan menuliskan nama *file* tersebut.
2. *Multiple selection*, berfungsi untuk membuat perubahan pada kode program pada saat yang sama dalam beberapa baris yang berbeda.
3. *Command Pallete*, dengan menekan CTRL+SHIFT+P untuk ,menutup semua *file*, *convert case: lower caser*, *remove tag* dan lain-lain.
4. *Split editing*, memperbolehkan mengedit *file* berdampingan, atau mengedit dua lokasi pada satu *file* dengan beberapa baris dan kolom yang diinginkan.

5. *Cross Platform*, *Sublime Text* tersedia untuk platform windows, Linux, OS X dan satu lisensi untuk semua sistem operasi yang digunakan.

2.3.4 phpMyAdmin



Gambar 2. 30 Logo *phpMyAdmin*
Sumber: <https://www.phpmyadmin.net/>

PhpMyadmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman *PHP* yang digunakan untuk menangani administrasi *MySQL* melalui *WWW (World Wide Web)*. *phpMyAdmin* mendukung berbagai operasi *MySQL*. Operasi pengolahan dalam *phpMyAdmin* yaitu mengelola basis data (*database*), tabel-tabel, kolom (*fields*), relasi (*relations*), *indeks*, pengguna (*users*), perijinan (*permissions*), dan lain-lain. Pada dasarnya, mengelola basis data dengan *MySQL* harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*commandline*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data (Barri, Lumenta, & Anneke Wowor, 2015: 25).

2.3.5 Mozilla Firefox

Manzur (2010: 2) *Mozilla Firefox* merupakan aplikasi *web browser* gratis yang dikembangkan oleh Yayasan *Mozilla* dan beberapa *developer*

pendukungnya. Pada saat pertama kali dirilis tanggal 9 November 2004 (versi 1.0) *browser* ini bernama *Phoenix*. Kemudian berganti nama *Mozilla Firebird*, hingga akhirnya berganti nama menjadi *Mozilla Firefox*. Sedangkan versi 2.0 untuk pemakaian pertamakalinya dirilis pada tanggal 24 Oktober 2006 dan versi 3.0 dirilis pada tanggal 17 Juni 2008. Berikut ini adalah logo *Mozilla Firefox*:



Gambar 2. 31 Logo *Mozilla Firefox*
Sumber:Manzur, 2010

Saat ini *Firefox* tercatat sebagai perangkat lunak yang banyak digunakan para pengguna rumahan karena sifatnya yang gratis dan *open source* (sumber terbuka). Proyek pembuatan *Firefox* ditujukan untuk mengembangkan sebuah *webbrowser* yang kecil, cepat, *simple* dan masih terbuka untuk dikembangkan karena sudah terpisah dari *MozillaSuite* yang lebih besar (Manzur, 2010: 2).

2.3.7 Mendeley

Mendeley merupakan salah satu perangkat lunak manajemen referensi berbasis *open source* yang dapat diperoleh secara gratis dan mendukung berbagai *platform* seperti Microsoft Windows, Apple MacOS, maupun Linux. Versi terbaru dari Mendeley bahkan sudah mendukung sistem operasi Android, sehingga perangkat ini dapat digunakan pada ponsel. Mendeley merupakan kombinasi dari aplikasi *desktop* dan situs web yang dapat digunakan untuk mengelola, berbagi, dan mencari referensi maupun kontak. Berikut fitur utama dari Mendeley *Desktop*:

- a. Ekstraksi detil dokumen (judul, penulis, nama jurnal, dll.) dari paper ke dalam database daftar pustaka.
- b. Pengelolaan referensi yang efisien dengan menyediakan “live” *full-text search* terhadap seluruh paper dalam database.
- c. Berbagi pakai sebagian atau semua dokumen pada database dengan pengguna lain serta sinkronisasi pustaka referensi dengan data yang tersimpan di *serverweb* Mendeley (Isa, Soewito, & Gunawan, 2016: 245-246).



Gambar 2. 32 Logo Mendeley
Sumber: Isa, dkk., 2016

Pada penelitian ini penulis menggunakan *software* mendeley untuk melakukan pengolahan sumber referensi dan citasi serta pembuatan daftar pustaka. *Software* ini memang tidak diperlukan dalam melakukan perancangan sistem dalam penelitian ini, akan tetapi penulis melihat adanya kepentingan untuk mencantumkan *software* mendeley didalam sub-bab *software* pendukung karena selain memang sudah diwajibkan untuk menggunakannya, *software* ini juga sangat membantu dalam melakukan pengolahan sumber referensi, citasi, dan daftar pustaka.

2.4 Penelitian Terdahulu

1. Nama Jurnal : *International Journal of System Application, Engineering & Development*

- Judul Jurnal : *Expert System design for Fault Diagnosis in Diesel Engines*
- Nama Peneliti : Ileana Concho, Mary Vergara, Francklin Rivas-Echeverria, Fernando Chica and Nestor Rivera
- ISSN/Tahun/Volume : 2074-1308/2016/10
- Kesimpulan : Berdasarkan penelitian yang dilakukan, maka dapat disimpulkan bahwa diagnosa Expert System(ES) atau sistem pakar menggunakan CBM (Condition-Base Management System) sebagai pemandu yang maksimal dan telah ditanamkan dengan CMMS (Computerized Maintenance Management System) untuk mesin disel telah dibuat. Sistem pakar ini dikembangkan untuk mendiagnosa kerusakan pada fasilitas yang bergerak pada pengawasan dan diagnosa pada mesin disel Hyundai CRDi 2.0, dan sebagai pemandu untuk pengambilan keputusan untuk perawatan mereka. Sistem pakar sebagai penghubung teknik CBM dan CMMS untuk mesin disel, menggunakan manajemen pemeliharaan dan untuk mengetahui riwayat kondisi mesin disel yang akan berguna untuk kajian tingkat lanjut (Concho, Vergara, Rivas-echeverría, Chica, & Rivera, 2016).

2. Nama Jurnal : *Citec Journal*
- Judul Jurnal : Pembuatan Aplikasi Diagnosa Kerusakan Mesin Sepeda Motor Matic dengan Case-Based Reasoning
- Nama Peneliti : Sandy Kosasi
- ISSN/Tahun/Volume : 2354-5771/2015/2
- Kesimpulan : Melalui pembuatan aplikasi diagnosa kerusakan mesin sepeda motor matic berbasis web dapat memberikan informasi dan solusi penyelesaian terhadap kerusakan yang dialami oleh pengguna. Setiap hasil solusi yang diberikan sistem kepada user dapat diberikan rating yang menandakan seringnya kerusakan yang terjadi dan solusi yang diberikan aplikasi tersebut menjadi penyelesaian, dimana rating ini akan menjadi bahan pertimbangan bagi admin dalam melakukan perbaikan dan memberikan pengembangan pada database. Pengguna sepeda motor matic dapat menambah edukasi mengenai informasi seputar perawatan dan pemeliharannya dalam memperpanjang umur kendaraan tanpa batasan waktu dan tempat. Hasil aplikasi ini sementara hanya dapat menangani komponen-komponen kerusakan mesin seperti starter assy,

CDI, koil, karburator, secondary, V-belt, WAP, gigi reduksi, dan valve (Kosasi, 2015).

3. Nama Jurnal : *International Journal of Engineering and Applied Sciences*

Judul Jurnal : *An Expert System for Diagnosing Faults in Motorcycle*

Nama Peneliti : Olanloye, Dauda Odunayo

ISSN/Tahun/Volume : 2305-8269/2014/5

Kesimpulan : Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa sama seperti mesin lainnya, mesin sepeda motor juga bisa dikaji untuk satu kerusakan atau lebih. Oleh karenanya bisa ditangani pengendara atau pemeliliknya. Sebuah sistem pakar untuk mendiagnosis kerusakan motor telah dipaparkan dalam penelitian ini yang berfungsi sebagai pemandu bagi pemilik atau pengendara dan akan sangat berguna bagi mereka. Meskipun sepeda motor lebih murah dibandingkan kendaraan roda 4, sistem yang dikembangkan akan mengurangi biaya perawatan karena pengendara atau pemiliknya dapat melakukan pengembangan lebih lanjut dengan bantuan sistem yang dikembangkan. Hal ini akan

mendorong lebih banyak orang untuk membeli sepeda motor secara pribadi atau untuk penggunaan komersial yang kemudian akan mendorong perekonomian negara (Olanloye, 2014).

4. Nama Jurnal : *International Journal of Computer Science Issues*
- Judul Jurnal : *Implementing an Expert Diagnostic Assistance System for Car Failure and Malfunction*
- Nama Peneliti : Salama A. Mostafa, Mohd Sharifuddin Ahmad, Mazin Abed Mohammed, dan Omar Ibrahim Obaid
- ISSN/Tahun/Volume : 1694-0814/2012/9
- Kesimpulan : Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa aplikasi yang dibangun ini mengindikasikan hasil yang signifikan dimana sistem ini meningkatkan produktifitas dengan mempersingkat waktu diagnosa kesalahan dan meningkatkan pemahaman kepada pengguna. Lebih dari itu, sistem ini dapat dipertimbangkan sebagai alternatif yang baik dari skill mekanik yang tinggi (Mostafa, Ahmad, Mohammed, & Obaid, 2012).
5. Nama Jurnal : Jurnal PPKM II
- Judul Jurnal : Sistem Pakar Sebagai Alat Bantu Mengatasi Masalah (Studi Kasus Kerusakan Sepeda Motor)

Nama Peneliti : Adi Suwondo

ISSN/Tahun/Volume : 2354-869X/2014/

Kesimpulan : Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa sistem pakar dalam penggunaan sehari-hari dirasa sangat perlu untuk menjawab kebutuhan pengguna dalam menyelesaikan permasalahan, tentunya dengan masalah yang spesifik dan memiliki tingkat keakuratan yang tinggi, sehingga kebutuhan seorang pakar dalam setiap permasalahan dapat diwakilkan oleh sistem pakar berbasis komputer. Sistem pakar ini juga diharapkan dapat membantu para mekanik dalam memecahkan permasalahan tentang kerusakan sepeda motor. *Rule* yang dibuat memiliki tingkat akurasi yang cukup, sehingga dalam sesi konsultasi, kemungkinan seorang pengguna dengan masalahnya dapat didiagnosis dan dapat diberikan saran untuk penyelesaiannya (Adi Suwondo, 2014).

6. Nama Jurnal : Jurnal Algoritma

Judul Jurnal : Pengembangan Sistem Pakar Diagnosis Kerusakan Sepeda Motor *Automatic* Non Injeksi Berbasis *Android*.

Nama Peneliti : Ahmad Hilmi A dan Dini Destiani

ISSN/Tahun/Volume : 2302-7339/2015/12

Kesimpulan : Berdasarkan penelitian tersebut maka didapat kesimpulan bahwa penelitian tersebut telah berhasil merancang sistem pakar diagnosis kerusakan sepeda motor automatic non injeksi sesuai dengan tujuan dan dengan adanya aplikasi tersebut dapat membantu pengguna yang ingin menambah wawasan dan pengetahuan mengenai kerusakan sepeda motor *automatic* non injeksi sebelum diperbaiki di bengkel. Dengan aplikasi ini pengguna juga dapat dengan mudah menggunakannya karena bisa digunakan dimana saja (A & Destiani, 2015).

7. Nama Jurnal : *SMARTICS Journal*

Judul Jurnal : Sistem Pakar Diagnosa Kerusakan Sepeda Motor Matic Injeksi Menggunakan Metode *Forward Chaining* Berbasis Android

Nama Peneliti : Anggri Sartika Wiguna, Isman Harianto

ISSN/Tahun/Volume : 2476-9754/2017/3

Kesimpulan : Berdasarkan permasalahan yang dibangun pada aplikasi sistem pakar untuk diagnosa kerusakan dan perawatan pada motor matic injeksi berbasis android, maka dapat diambil beberapa kesimpulan

adalah sebagai berikut (Anggri Sartika Wiguna, 2017) :

1. Aplikasi yang dibangun ini membantu para pengguna motor matic injeksi untuk mengetahui kerusakan motor matic injeksi berdasarkan gejala-gejala yang ada.
2. Berdasarkan dari hasil kuisisioner aplikasi ini menggunakan metode forward chaining di dapatkan kesimpulan yang sesuai dari pengguna motor matic injeksi untuk mengetahui kerusakan motornya

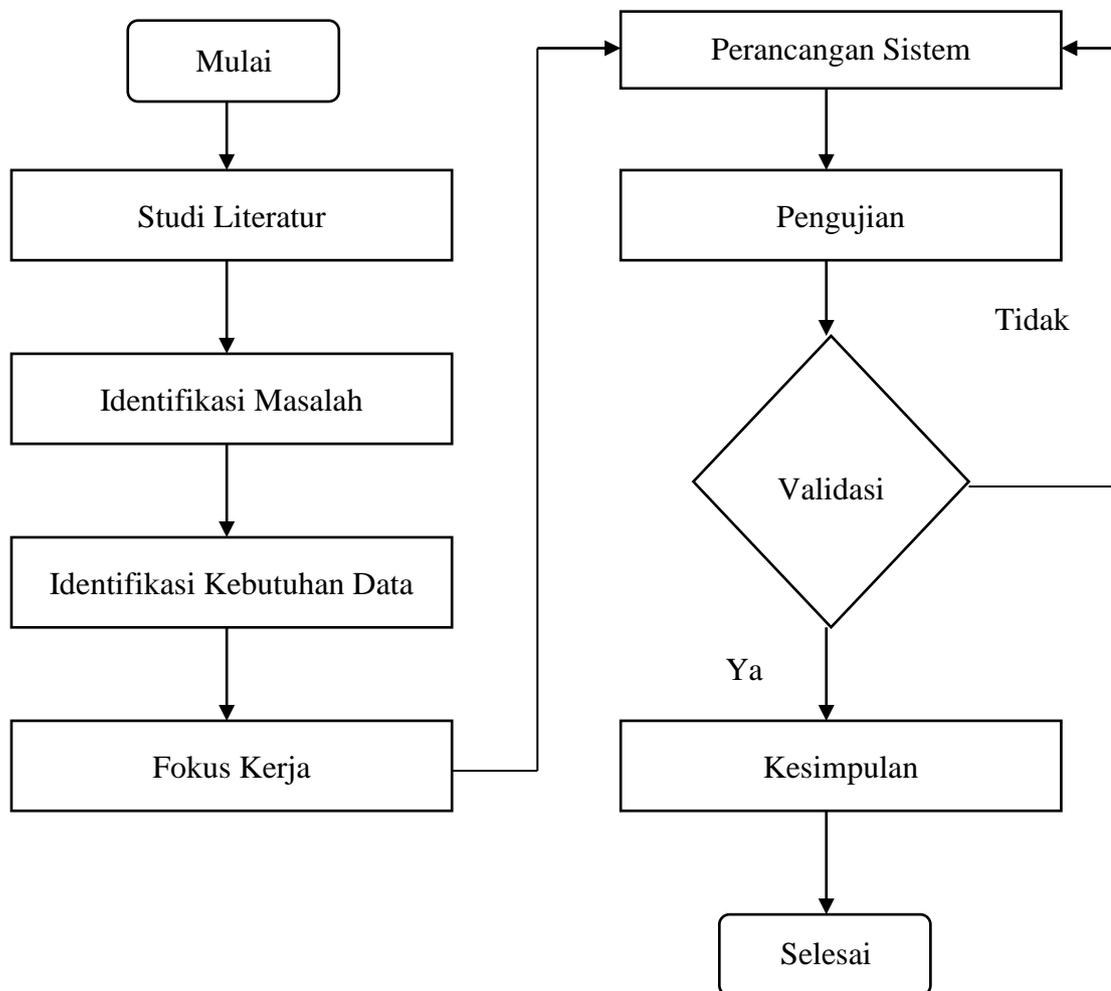
2.5 Kerangka Pemikiran

Kerangka pemikiran memuat pemikiran terhadap alur yang dipahami sebagai acuan dalam pemecahan masalah yang diteliti secara logis dan sistematis. Kerangka berfikir yang baik akan menjelaskan secara teoritis pertautan antar variabel yang diteliti (Sugiyono, 2014: 60).

Kerusakan-kerusakan yang terjadi pada sepeda motor Yamaha Nmax nantinya menjadi sumber data penelitian. Kemudian data tadi akan diolah menggunakan metode *forward chaining* menjadi sebuah sistem pakar berbasis *web*. Aplikasi tersebut mempunyai kecerdasan layaknya seorang pakar yang bisa mendeteksi kerusakan sepeda motor Yamaha Nmax sehingga didapatkan solusi berdasarkan permasalahan yang ada pada sepeda motor Yamaha Nmax.

Berikut ini adalah kerangka pemikiran yang menjadi dasar dalam penelitian

ini:



Gambar 2. 33 Kerangka Pemikiran
Sumber: Data penelitian, 2018

BAB III

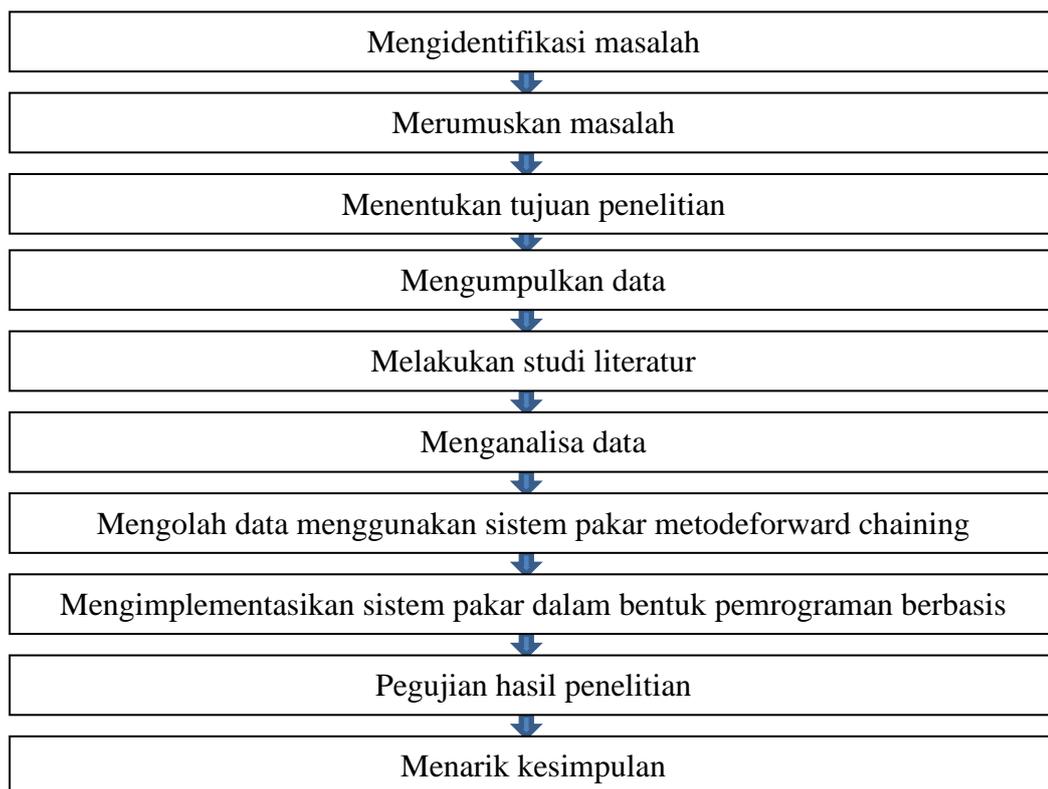
METODE PENELITIAN

Pada bab ini akan dibahas tentang perancangan sistem dari aplikasi sistem pakar yang akan dibuat. Desain aplikasi itu sendiri digunakan untuk penggambaran umum terhadap aplikasi yang akan dibuat sehingga kebutuhan akan konsep aplikasi dapat diketahui sebelum pembuatan aplikasi. Dengan desain aplikasi inilah akan mempermudah untuk pembangunan lebih lanjut terhadap aplikasi yang akan dibuat. Metode penelitian diperlukan untuk menjawab masalah dan menguji hipotesa. Untuk itu, dibagian ini perlu ditetapkan metode penelitian apa yang digunakan, apakah metode survei, metode eksperimen, metode kasus, metode penelitian dan pengembangan, atau kaji tindakan (*action research*), (Sudaryono, 2015: 157).

Penelitian ini dilakukan dengan cara mendiagnosis beberapa penyebab kerusakan pada sepeda motor Yamaha Nmax. Dari beberapa analisa yang dilakukan, maka data akan diolah dan dibuat sistem pakar untuk mengatasi kerusakan pada sepeda motor Yamaha Nmax. Data-data yang didapat digunakan untuk membuat *rule* sebuah aplikasi sistem pakar *forward chaining* untuk menduplikasikan kemampuan pakar, sehingga bisa dimanfaatkan teknisi atau mekanik di bengkel Marfen Utama Motor dan juga masyarakat awam.

3.1 Desain Penelitian

Desain penelitian menggambarkan apa yang akan dilakukan oleh peneliti dalam terminologi teknis. Dalam hal ini, desain penelitian harus mencakup antara lain tahapan yang akan dilakukan, informasi mengenai cara penarikan *sampel* bila diperlukan survei primer, besarnya *sampel*, metode pengumpulan data, instrumen penelitian, prosedur pengujian validitas, dan reliabilitas instrumen (Sudaryono, 2015: 157). Berikut ini adalah gambar konsep dasar desain penelitian yang digunakan dalam penelitian ini (Gambar 3.1):



Gambar 3. 1 Desain Penelitian
Sumber: Data penelitian, 2018

Berikut ini adalah langkah-langkah yang dilakukan dalam penelitian yaitu sebagai berikut:

1. Mengidentifikasi masalah

Tahap pertama dalam melakukan penelitian adalah mengidentifikasi masalah. Pada tahapan identifikasi masalah, dilakukan analisa terhadap faktor-faktor penyebab kerusakan sepeda motor Yamaha Nmax di bengkel Marfen Utama Motor Batam. Setelah identifikasi masalah selesai, kemudian dilakukan perumusan masalah.

2. Merumuskan masalah

Perumusan masalah bertujuan untuk menspesifikasikan masalah yang ada sehingga dapat dijawab dengan baik melalui penelitian. Setelah masalah dirumuskan, maka tahapan berikutnya adalah menentukan tujuan penelitian.

3. Menentukan tujuan penelitian

Tujuan penelitian yaitu mengetahui bagaimana sistem pakar *forwardchaining* berbasis *web* mendeteksi kerusakan sepeda motor Yamaha Nmax di bengkel Marfen Utama Motor, Batam. Setelah tujuan penelitian ditentukan, maka langkah selanjutnya adalah mengumpulkan data.

4. Mengumpulkan data

Dalam pengumpulan data, peneliti mengambil data dengan melakukan wawancara dengan seorang pakar dan studi *literature* yaitu mengumpulkan buku yang berkaitan dengan kecerdasan buatan, sistem pakar, pemrograman *web*, *UML*, buku referensi tentang sepeda motor, dan buku yang berkaitan

dengan penelitian. Setelah bahan-bahan yang mendukung dalam penelitian didapatkan maka tahapan selanjutnya adalah melakukan studi literatur.

5. Melakukan studi literatur

Mempelajari dan mengkaji buku-buku yang berhubungan dengan penelitian sehingga menambah wawasan dan pengetahuan tentang objek yang akan diteliti. Dalam hal ini peneliti mempelajari tentang kecerdasan buatan, sistem pakar, *UML*, membangun pemrograman *web*, buku referensi tentang sepeda motor, dan buku yang berkaitan dengan penelitian. Setelah studi literatur dilakukan, maka tahapan selanjutnya adalah menganalisa data yang didapat.

6. Menganalisa data

Setelah data-data yang berhubungan dengan penelitian didapatkan, maka langkah selanjutnya dengan menganalisa data-data yang diperoleh. Pada tahapan menganalisa data, ditentukan terlebih dahulu operasional variabel dan indikator yang akan digunakan supaya mempermudah dalam proses pengolahan data. Jika operasional variable dan indikator sudah ditentukan maka tahapan berikutnya adalah mengolah data.

7. Mengolah data menggunakan sistem pakar metode *forward chaining*

Data-data yang sudah dianalisa kemudian diolah menggunakan sistem pakar *forward chaining* untuk membuat *rule-rule* yang akan digunakan saat sistem pakar melakukan penelusuran berdasarkan gejala yang ada. Setelah *rule-rule* sebagai *knowlege base* sistem pakar dibuat maka tahapan selanjutnya adalah mengimplementasikan sistem pakar kedalam pemrograman berbasis *web*.

8. Mengimplementasikan sistem pakar dalam bentuk pemrograman berbasis *web*
Setelah data yang diperoleh diolah menggunakan sistem pakar metode *forward chaining*, data-data tersebut disusun dan dibuat kode-kode untuk mempermudah pembuatan program. Pembuatan aplikasi sistem pakar dilakukan mulai dari perancangan sistem yaitu: desain *UML*, desain antarmuka, desain *database*, dan desain basis pengetahuan (aturan). Setelah itu proses pengodean menggunakan bahasa *HTML*, *PHP*, *CSS*, *JavaScript*, dan *jQuery* dengan editor *Sublime Text*. Langkah terakhir dalam pembuatan aplikasi sistem pakar adalah membuat *database* dengan menggunakan *phpMyAdmin* yang ada pada program *XAMPP*. Dalam pembuatan program, peneliti menggunakan referensi program dari buku karya Bunafit Nugroho yang berjudul *Aplikasi Sistem Pakar dengan PHP&Editor Dreamweaver*. Sistem pakar yang sudah dibuat kemudian diuji keakuratannya pada tahapan pengujian hasil penelitian. Setelah program dibuat, maka aplikasi sistem pakar tersebut diimplementasikan pada salah satu komputer yang sudah terinstal aplikasi *XAMPP* ada di bengkel Marfen Utama Motor, Batam.
9. Pegujian hasil penelitian
Tahapan ini adalah pengujian aplikasi yang telah dibuat. Pengujian dilakukan oleh pakar yaitu sorang mekanik yang telah berpengalaman lebih dari 8 tahun dibidang perbengkelan dan permesinan sepeda motor. Pengujian dilakukan dengan 2 sistem yaitu *black box testing* dan pengujian keakuratan dengan pakar. Pengujian *black box testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian

dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Setelah lolos pengujian aplikasi maka tahapan berikutnya adalah dengan pengujian tingkat keakuratan sistem pakar. Pengujian dengan pakar dilakukan dengan merujuk pada data diagnosa gejala kerusakan unit yang telah disusun sebanyak 6 studi kasus. Kemudian hasil pengujian dihitung tingkat persentasi keakuratannya. Setelah pengujian dilakukan maka tahapan berikutnya adalah menarik kesimpulan.

10. Menarik kesimpulan

Tahapan terakhir dalam penelitian ini yaitu menyimpulkan hasil penelitian yang berisi jawaban singkat terhadap rumusan masalah berdasarkan data-data yang ada. Dalam tahap ini, peneliti juga memberikan saran yang penting untuk membantu dalam memecahkan permasalahan yang ada.

3.2 Teknik Pengumpulan Data

Teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data (Sugiyono, 2012: 224). Langkah-langkah yang dilakukan untuk pengumpulan data dalam penelitian ini adalah sebagai berikut:

1. Studi literatur

Studi literatur yaitu mengumpulkan data dan menambah wawasan dengan belajar dari mekanik yang bekerja di bengkel Yamaha Nmax, mempelajari buku yang berisikan materi pengetahuan tentang mesin dan anatomi sepeda

motor, mempelajari buku tentang pembuatan aplikasi sistem pakar berbasis *web*, dan membaca jurnal penelitian yang berkaitan dengan topik bahasan penelitian.

2. Wawancara

Melakukan wawancara dengan nara sumber yang bekerja sebagai mekanik sepeda motor sekaligus pemilik bengkel Yamaha Nmax dan berpengalaman lebih dari 8 tahun dalam menganalisa dan menangani kerusakan sepeda motor Yamaha Nmax di bengkel Marfen Utama Motor. Wawancara dilakukan dengan memberikan beberapa pertanyaan yang sudah disusun sebelumnya, kemudian jawaban tersebut ditulis oleh peneliti pada *form* wawancara.

3.3 Operasional Variabel

Variabel penelitian adalah suatu atribut atau sifat atau nilai dari orang, objek atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya (Sugiyono, 2014:38).

Berikut ini tabel hubungan variabel dan indikator di dalam penelitian ini:

Tabel 3. 1 Tabel hubungan Variabel dan Indikator

Variabel	Indikator
Kerusakan sepeda motor Yamaha Nmax	Masalah ruang bakar dan pengapian
	Masalah pada CVT
	Masalah klep dan <i>filter</i>
	Masalah kelistrikan

Sumber: Data penelitian, 2018

Dalam Tabel 3.1 diatas menjelaskan hubungan antara variabel dan indikator. Variabelnya adalah kerusakan sepeda motor Yamaha Nmax, sedangkan indikatornya adalah 4 bagian kerusakan yaitu: kerusakan pada ruangbakar dan pengapian, kerusakan pada CVT, kerusakan pada klep dan filter dan kerusakan pada kelistrikan. Pada Tabel 3.2 menjelaskan indikator, kerusakan, dan solusi kerusakan sepeda motor Yamaha Nmax.

Tabel 3. 2 Tabel Kerusakan dan Solusi Kerusakan sepeda motor Yamaha Nmax

Indikator	Kerusakan	Solusi
Masalah pada ruang bakar dan pengapian	Busi kotor dan berkerak	Lepas busi dari dudukannya, bersihkan <i>insert</i> dan <i>tapered tip</i> dan juga pada ujung terminalnya, jika terdapat kerak, maka harus diampelas hingga bersih. Apabila busi terlihat gosong dan sudah dapat berfungsi maka harus diganti dengan yang baru
	Keausan pada <i>ring piston</i> dan terjadi lecet/beret pada silinder	Mengganti satu set blok silinder, karena motot tipe ini harus selalu standart
	Komponen pada <i>throttle body</i> sepeda motor tersumbat oleh kotoran.	Bersihkan komponen pada <i>throttle body</i> sepeda motor

Tabel 3.2 Lanjutan		
<i>Masalah Pada CVT</i>	Terdapat kotoran atau kemasukan air pada <i>V-Belt</i> sepeda motor anda atau disebabkan terlalu kerasnya pemasangan spring yang menyebabkan <i>V-Belt</i> mendapatkan beban kerja yang lebih berat dan menjadi jauh lebih cepat aus atau putus	Periksa kondisi <i>V-Belt</i> apakah ada keretakan atau tidak dengan cara membaliknya atau menekuknya. Kalau ada keretakan, segera diganti. Jika masih ada kemungkinan terpakai lagi gunakan larutan <i>V-Belt cleaner</i> untuk menghilangkan suara berdecit
Masalah pada klep dan filter	Klep <i>IN</i> bocor, sehingga bensin(bahan bakar) lebih banyak masuk ke ruang bakar, menyebabkan pembakaran basah yang mengakibatkan bensin boros.	Melakukan Skir (skur) klep, untuk mengembalikan posisi klep dan rumah pada posisi ideal sekaligus untuk membersihkan kotoran pada klep yang mengakibatkan klep bocor.
	Filter udara sudah kotor sehingga ada penyumbatan udara dan tidak bisa menyaring udara bersih, akibatnya pasokan udara masuk menjadi sangat kurang.	Mengganti filter udara, karena filter yang dipakai adalah filter tipe basah

Tabel 3.2 Lanjutan		
Masalah pada kelistrikan	Saluran kelistrikan ada yang terputus, ini mempengaruhi pasokan arus ke ECU dan coil yang diperlukan untuk memercikan api pada busi.	Mencari saluran elektrik yang putus atau sikring yang putus dari bagian elektrik starter, batteray, ECU dan kelistrikan seputar .
	Poros dinamo yang aus karena tergerus rumah arang, dan dua magnet di dalam rumah motor starter lepas dan ikut berputar.	Memasang kedua magnet dengan membuat tanda pada garis pada bodi motor starter untuk menentukan bagian bagian depan dan belakang, dan jangan lupa untuk melakukan pengeleman menggunakan <i>loctite</i> sebelum menutup rumah bodi starter.
	<i>Brush</i> pada dinamo starter sudah aus (habis)	Mengganti <i>brush</i> yang lama dengan yang baru
	Terdapat saluran kelistrikan yang terputus dari tombol elektrik starter ke <i>batteray</i> , bisa juga <i>batteray</i> sudah habis atau lemah.	Periksa jalur kelistrikan dari tombol elektrik starter ke arah <i>batteray</i> dan dari <i>batteray</i> ke motor starter. Masih bisa menggunakan <i>Kickstarter</i> atau <i>starter manual</i> .

Sumber: Data penelitian, 2018

Pada Tabel 3.2 diatas, masing-masing indikator memiliki beberapa kerusakan. Dari kerusakan tersebut, juga disertakan solusi dari setiap kerusakan.

3.4 Perancangan Sistem

Menurut Rosa A.S dan M. Shalahuddin (2011: 21) perancangan sistem merupakan upaya 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.

3.4.1 Desain Basis Pengetahuan

Peneliti melakukan proses akuisisi pengetahuan dengan mengumpulkan pengetahuan dan fakta dari sumber-sumber yang tersedia. Sumber pengetahuan dan fakta diperoleh melalui wawancara dengan seorang *mekanik* yang berpengalaman, selain itu peneliti melakukan studi literatur tentang materi yang berkaitan dengan sepeda motor Yamaha Nmax dan penyebab-penyebab kerusakannya.

Sumber pengetahuan yang telah didapatkan dari seorang pakar ditampilkan dalam table bagian kerusakan atau dalam penelitian ini sebagai *indikator* sepeda motor (Tabel 3.3), tabel kerusakan dan solusi kerusakan sepeda motor (Tabel 3.4), table gejala kerusakan sepeda motor (Tabel 3.5) dan table aturan (Tabel 3.6) berikut ini:

Tabel 3. 3 Tabel nama indikator kerusakan sepeda motor Yamaha Nmax

Kode	Indikator
BK01	Masalah pada ruang bakar dan pengapian
BK02	Masalah Pada CVT
BK03	Masalah pada klep dan <i>filter</i>
BK04	Masalah pada kelistrikan

Sumber: Data penelitian, 2018

Pada Tabel 3.3 diatas, masing-masing indikator diberikan kode untuk membedakan satu dengan yang lainnya. Kode BK01 untuk masalah sepeda motor pada ruang bakar dan pengapian, kode BK02 untuk masalah sepeda motor pada bagian CVT, Kode BK03 untuk masalah sepeda motor pada klep dan filter, kode BK04 untuk masalah sepeda motor pada kelistrikan.

Tabel 3. 4 Tabel kerusakan dan solusi kerusakan sepeda motor Yamaha Nmax

Kode	Kerusakan	Solusi
K01	Saluran kelistrikan ada yang terputus, ini mempengaruhi pasokan arus ke ECU dan coil yang diperlukan untuk memercikan api pada busi.	Mencari saluran elektrik yang putus atau sikring yang putus dari bagian elektrik starter, batteray, dan kelistrikan seputar ECU.
K02	Keausan pada ring piston dan terjadi lecet/beret pada silinder	Mengganti satu set blok silinder, karena motot tipe ini harus selalu standart

Tabel 3.4 Lanjutan

K03	Busi kotor dan berkerak	Lepas busi dari dudukannya, bersihkan insert dan tapered tip dan juga pada ujung terminalnya, jika terdapat kerak, maka harus diampelas hingga bersih. Apabila busi terlihat gosong dan sudah dapat berfungsi maka harus diganti dengan yang baru
K04	Terdapat kotoran atau kemasukan air pada V-Belt sepeda motor anda atau disebabkan terlalu kerasnya pemasangan spring yang menyebabkan V-Belt mendapatkan beban kerja yang lebih berat dan menjadi jauh lebih cepat aus atau putus	Periksa kondisi V-Belt apakah ada keretakan atau tidak dengan cara membaliknya atau menekuknya. Kalau ada keretakan, segera diganti. Jika masih ada kemungkinan terpakai lagi gunakan larutan V-Belt cleaner untuk menghilangkan suara berdecit
K05	Klep <i>IV</i> bocor, sehingga bensin (bahan bakar) lebih banyak masuk ke ruang bakar, menyebabkan pembakaran basah yang mengakibatkan bensin boros.	Melakukan Skir (skur) klep, untuk mengembalikan posisi klep dan rumah magnet pada posisi ideal sekaligus untuk membersihkan kotoran pada klep yang mengakibatkan klep bocor.

Tabel 3.4 Lanjutan		
K06	Filter udara sudah kotor sehingga ada penyumbatan udara dan tidak bisa menyaring udara bersih, akibatnya pasokan udara masuk menjadi sangat kurang.	Mengganti filter udara, karena filter yang dipakai adalah filter tipe basah
K07	Poros dinamo yang aus karena tergerus rumah arang, dan dua magnet di dalam rumah motor starter lepas dan ikut berputar.	Memasang kedua magnet dengan membuat tanda pada garis pada bodi motor starter untuk menentukan bagian bagian depan dan belakang, dan jangan lupa untuk melakukan pengeleman menggunakan <i>loctite</i> sebelum menutup rumah bodi starter.
K08	<i>Brush</i> pada dinamo starter sudah aus (habis)	Mengganti <i>brush</i> yang lama dengan yang baru
K09	Terdapat saluran kelistrikan yang terputus dari tombol elektrik starter ke batteray, bisa juga batteray sudah habis atau lemah.	Periksa jalur kelistrikan dari tombol elektrik starter ke arah batteray dan dari batteray ke motor starter. Masih bisa menggunakan Kicstarter atau starter manual.
K10	Komponen pada throttle body sepeda motor tersumbat oleh kotoran.	Bersihkan komponen pada throttle body sepeda motor

Sumber: Data penelitian, 2018

Pada Tabel 3.4, menunjukkan bahwa masing-masing diberikan kode untuk membedakan antara satu kerusakan dengan yang lainnya. Dari masing-masing kerusakan diberikan solusi untuk mengatasi kerusakan sepeda motor Yamaha Nmax.

Tabel 3. 5 Tabel gejala kerusakan sepeda motor Yamaha Nmax

Kode Gejala	Nama Gejala
G01	Motor susah dihidupkan baik dengan elektrik starter maupun secara manual
G02	Tenaga yang dihasilkan lemah
G03	Mesin cepat panas
G04	Keluar asap putih dari knalpot
G05	Timbul suara berisik / berdecit
G06	Keluar asap hitam pada knalpot
G07	Bahan bakar boros
G08	Saat dihidupkan dengan dengan elektrik starter, tidak ada bunyi atau reaksi sama sekali
G09	Dinamo starter panas
G10	Suara kasar pada dinamo starter

Sumber: Data penelitian, 2018

Pada Tabel 3.5 diatas, menunjukkan pengkodean dari masing-masing gejala kerusakan sepeda motor Yamaha Nmax supaya membedakan gejala kerusakan satu dengan yang lainnya.

Data aturan berisi relasi antara data-data nama bagian kerusakan (*indikator*) sepeda motor Yamaha Nmax, kerusakan sepeda motor Yamaha Nmax,

dan gejala kerusakan sepeda motor Yamaha Nmax, yang telah diberi kode sebelumnya. Data-data yang didapat kemudian dibuat relasi antar data sehingga menghasilkan *rule-rule* dalam sistem pakar yang memudahkan penyusunan basis pengetahuan. Berikut ini adalah tabel data aturan (Tabel 3.6):

Tabel 3. 6 Tabel Data Aturan

Kode Indikator	Kode Kerusakan	Kode Gejala
BK04	K01	G01
BK01	K02	G01, G02, G03, G04
BK01	K03	G01, G02, G03
BK02	K04	G05
BK03	K05	G01, G06, G07
BK03	K06	G01, G06
BK04	K07	G08, G09, G10
BK04	K08	G08, G09
BK04	K09	G08
BK01	K10	G01, G02

Sumber: Data penelitian, 2018

Pada Tabel 3.6 diatas, indikator, kerusakan dan gejala dibuat kode yang berbeda-beda. Pengkoden ini dibuat untuk memudahkan dalam penyusunan kaidah produksi yang akan dibuat. Setiap kerusakan mempunyai gejala yang berbeda, tetapi ada beberapa kerusakan mempunyai salah satu ciri gejala sama dengan kerusakan lainnya. Urutan pengkodean kerusakan disesuaikan atau dikelompokkan sesuai dengan kode bagian kerusakan sepeda motor Yamaha Nmax (kode *indikator*).

3.4.2 Pembentukan Aturan

Sutojo, dkk. (2011: 170) setiap rule terdiri dari dua bagian, yaitu bagian *IF* disebut *evidence* (fakta-fakta) dan bagian *THEN* disebut hipotesis atau kesimpulan. Representasi pengetahuan pada dasarnya berupa aturan *IF – THEN* dalam sebuah sistem pakar. Data-data yang sudah disusun dalam tabel aturan (Tabel 3.7), dirangkai menjadi suatu kaidah aturan dalam sistem pakar. Berikut ini adalah tabel aturan *inference* dalam sistem pakar ini:

Tabel 3. 7 Aturan Inference

Aturan	Kaidah
Rule 1	IF G01 THEN K01
Rule 2	IF G01 AND G02 AND G03 AND G04 THEN K02
Rule 3	IF G01 AND G02 AND G03 THEN K03
Rule 4	IF G05 THEN K04
Rule 5	IF G01 AND G06 AND G07 THEN K05
Rule 6	IF G01 AND G06 THEN K06
Rule 7	IF G08 AND G09 AND G10 THEN K07
Rule 8	IF G08 AND G09 THEN K08
Rule 9	IF G08 THEN K09
Rule 10	IF G01 AND G02 THEN K10

Sumber: Data penelitian, 2018

Berdasarkan Tabel 3.7 di atas, dapat disimpulkan bahwa di dalam sistem pakar ini terdapat 10 aturan (*rule*). Berikut ini adalah pembahasannya:

RULE 1

Jika motor susah dihidupkan baik dengan elektrik starter maupun secara manual (G01) maka masalah pada kelistrikan. Saluran kelistrikan ada yang terputus, sehingga ini mempengaruhi pasokan arus ke ECU untuk dan *coil* yang diperlukan untuk memercikan api pada busi. (K01).

RULE 2

Jika motor susah dihidupkan baik dengan elektrik starter maupun secara manual (G01) dan tenaga yang dihasilkan lemah (G02) dan mesin cepat panas (G03) dan keluar asap putih dari knalpot (G04) maka masalah pada ruang bakar dan pengapian. Keausan pada *ring piston* terdapat beret pada silinder ruang bakar (K02).

RULE 3

Jika motor susah dihidupkan baik dengan elektrik starter maupun secara manual (G01) dan tenaga yang dihasilkan lemah (G02) dan mesin cepat panas (G03) maka masalah pada ruang bakar dan pengapian. Busi kotor atau sudah gosong karena usia pemakaian. (K03).

RULE 4

Jika Timbul suara berisik / berdecit (G05) maka terdapat kotoran atau kemasukan air pada *V-Belt* sepeda motor anda atau disebabkan terlalu kerasnya pemasangan spring yang menyebabkan *V-Belt* mendapatkan beban kerja yang lebih berat dan menjadi jauh lebih cepat aus atau putus (K04).

RULE 5

Jika motor susah dihidupkan baik dengan elektrik starter maupun secara manual (G01) dan keluar asap hitam pada knalpot (G06) dan bahan bakar boros (G07) maka masalah pada klep dan filter. Klep *IN* bocor, sehingga bensin (bahan bakar) lebih banyak masuk ke ruang bakar, menyebabkan pembakaran yang basah yang mengakibatkan bensin boros (K05).

RULE 6

Jika motor susah dihidupkan baik dengan elektrik starter maupun secara manual (G01) dan keluar asap hitam pada knalpot (G06) maka masalah pada klep dan filter. Filter udara sudah kotor sehingga ada penyumbatan udara dan tidak bisa menyaring udara bersih, akibatnya pasokan udara masuk menjadi sangat kurang (K06).

RULE 7

Jika saat dihidupkan dengan dengan elektrik starter, tidak ada bunyi atau reaksi sama sekali (G08) dan dinamo starter panas (G09) dan suara kasar pada dinamo srarter (G10) maka Masalah pada elektrik starter, poros dinamo yang aus karena tergerus rumah arang, dan dua magnet di dalam rumah motor starter lepas dan ikut berputar(K07).

RULE 8

Jika saat dihidupkan dengan dengan elektrik starter, tidak ada bunyi atau reaksi sama sekali (G08) dan dinamo starter panas (G09) maka Masalah pada elektrik starter, brush pada dinamo starter sudah aus (habis),(K08).

RULE 9

Jika saat dihidupkan dengan dengan elektrik starter, tidak ada bunyi atau reaksi sama sekali (G08) maka terdapat saluran kelistrikan yang terputus dari tombol eletrik starter ke bateray, bisa juga bateray sudah habis atau lemah(K09).

RULE 10

Jika motor susah dihidukan baik dengan elektrik starter maupun secara manual (G01) dan tenaga yang dihasilkan lemah (G02) maka Komponen pada karburator sepeda motor tersumbat oleh kotoran. (K10).

Berikut ini adalah tabel relasi gejala, kerusakan dan bagian kerusakan (Tabel 3.8) dari sistem pakar yang akan dibuat:

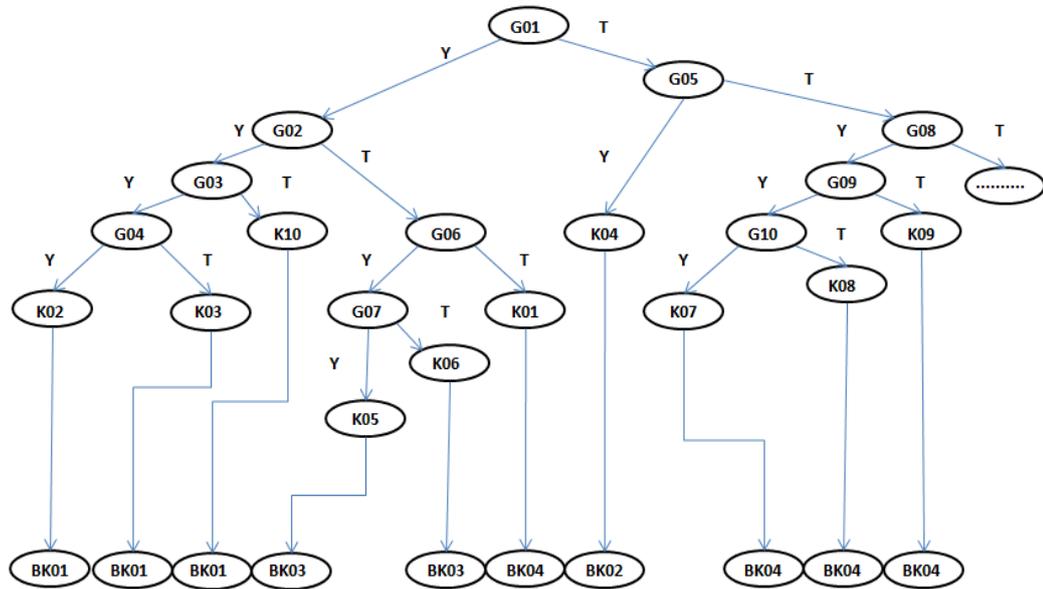
Tabel 3. 8 Tabel relasi gejala, kerusakan dan bagian kerusakan

Gejala	BK01			BK02	BK03		BK04			
	K10	K02	K03	K04	K05	K06	K07	K08	K09	K01
G01	√	√	√		√	√				√
G02	√	√	√							
G03		√	√							
G04		√								
G05				√						
G06					√	√				
G07					√					
G08							√	√	√	
G09							√	√		
G10							√			

Sumber: Data penelitian, 2018

Pada Tabel 3.8 diatas, kolom bagian indikator kerusakan (MK) dikelompokkan dan diurutkan berdasarkan kode kerusakan sepeda motor Yamaha Nmax (K), setelah itu diberi tanda *checklist* untuk baris kode gejala (G) yang memenuhi aturan dari masing-masing penyebab. Hal ini dibuat untuk memudahkan dalam menyusun aturan kaidah produksi sistem pakar yang akan dibuat.

Setelah disusun tabel relasi gejala, kerusakan dan bagian kerusakan (Tabel 3.8) diatas maka dapat dibuat pohon keputusan (Gambar 3.2) sebagai berikut:



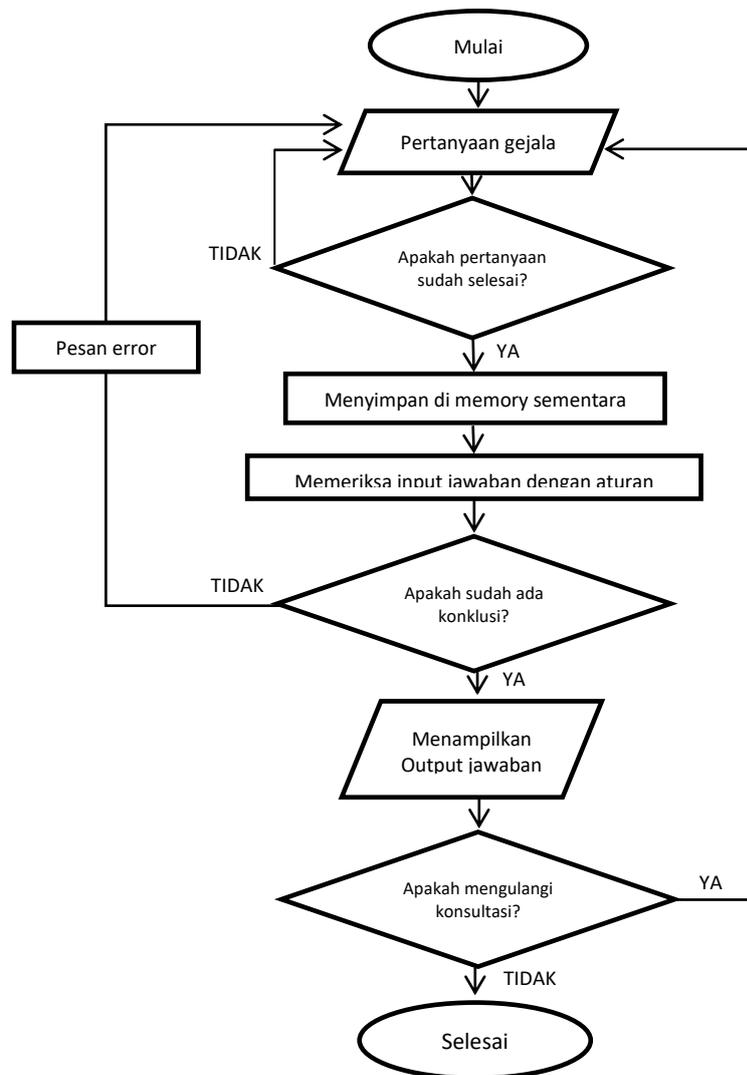
Gambar 3. 2 Pohon Keputusan Sistem Pakar
Sumber: Data penelitian, 2018

Pada Gambar 3.2 menunjukkan pohon keputusan yang memperlihatkan hubungan antara gejala dan kerusakan. Penelusuran menggunakan sistem *dept first search*. *Depth first search* merupakan metode pencarian yang dilakukan pada suatu simpul dalam setiap level yang dimulai dari kiri. Jika pada level yang terdalam solusi belum ditemukan, maka pencarian dilanjutkan pada simpul sebelah kanan dan simpul yang kiri dapat dihapus dari memori.

3.4.3 Struktur Kontrol (Mesin Inferensi)

Mesin inferensi dalam sistem pakar ini menggunakan metode penelusuran *forward chaining*, yaitu analisa penelusuran maju mulai dari fakta-fakta yang berupa ciri-ciri/gejala yang ditimbulkan pada kerusakan sepeda motor sehingga dapat menguji kebenaran hipotesis yaitu penyebab kerusakan, sehingga

menghasilkan solusi dari penyebab kerusakan. Berikut ini *flowchart* mesin inferensi sistem pakar *forward chining* digambarkan pada Gambar 3.3 berikut ini:



Gambar 3. 3 *Flowchart* Mesin Inferensi Sistem Pakar
Sumber: Data penelitian, 2018

Langkah-langkah yang digunakan dalam proses penelusuran masalah adalah sebagai berikut:

1. Mulai mengakses sistem.

2. Mengajukan pertanyaan tentang gejala-gejala kerusakan sepeda motor Yamaha Nmax.
3. Setelah jawaban pertanyaan dijawab oleh *user*, maka jawaban akan disimpan kedalam tabel gejala kerusakan sementara.
4. Sistem akan memeriksa jawaban tersebut dan membandingkannya dengan *rule* yang sudah dibuat sehingga menghasilkan jawaban, jika semua pertanyaan yang diajukan kepada *user* tetapi belum memenuhi konklusi pada sistem, maka akan keluar pesan pengulangan diagnosa.
5. Jika jawaban pertanyaan dari *user* sesuai dengan *rule* yang ada pada *database*, maka aplikasi akan menampilkan diagnosa berupa gejala kerusakan, penyebab kerusakan, dan solusi untuk memperbaiki kerusakan sepeda motor Yamaha Nmax.
6. Ada pertanyaan untuk melakukan pengulangan diagnosa, jika menjawab “Ya” maka akan kembali ke menu diagnosa awal dan jika menjawab “Tidak” maka diagnosa berhenti. Selesai.

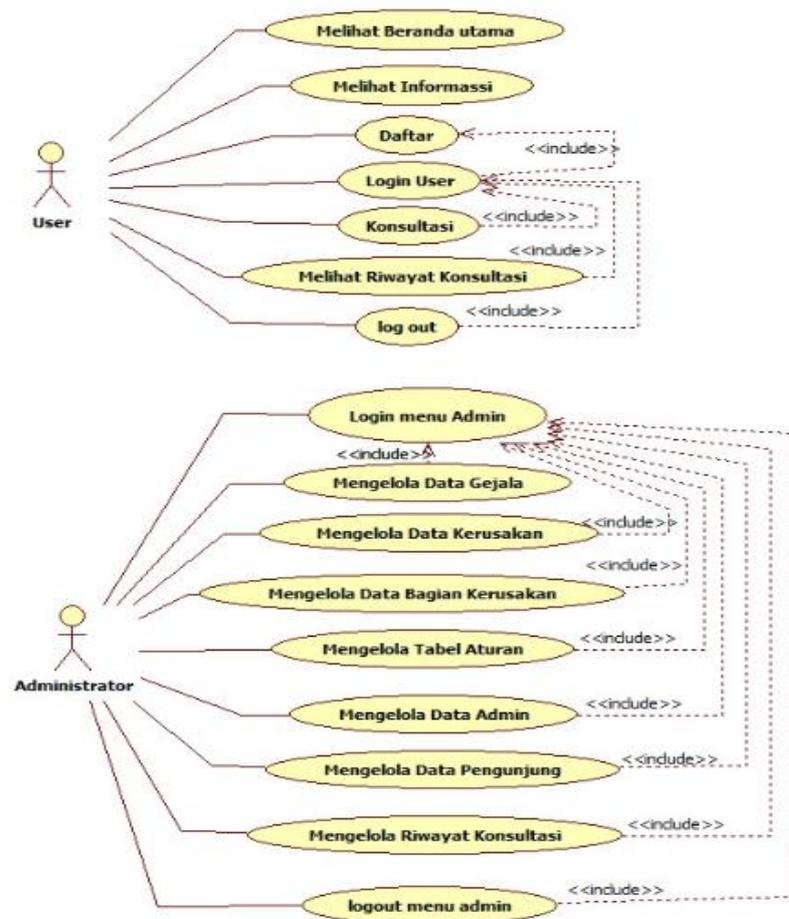
3.4.4 Desain UML (*Unified Modeling Language*)

UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa A.S dan Shalahuddin. M, 2011: 113). *Design UML* dibuat untuk memudahkan dalam pembuatan program. Pemodelan *UML*

menggunakan alat bantu *software StarUML* versi 2.5.1. Berikut ini adalah diagram UML yang digunakan dalam perancangan program:

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 A.S dan Shalahuddin. M, 2011:130-131). Pada Gambar 3.4 berikut ini menggambarkan *use case diagram* yang dirancang dalam penelitian ini:



Gambar 3. 4 Use Case Diagram
Sumber: Data penelitian, 2018

Terdapat 2 aktor yaitu *user* dan *Administrator* yang melakukan interaksi dengan sistem. *User* berinteraksi dengan sistem yaitu melihat beranda utama, melihat informasi, melakukan pendaftaran, melakukan *login*, melakukan konsultasi, melihat riwayat konsultasi, dan *logout*. Sebelum konsultasi dilakukan, *user* diminta untuk memasukkan data identitas diri pada *form* pendaftaran. Diagnosa dilakukan dengan cara menjawab pertanyaan yang diajukan oleh sistem, setelah semua jawaban sesuai *rule*, maka sistem akan menampilkan hasil dari konsultasi berupa gejala yang dialami, kesimpulan kerusakan yang dialami, bagian kerusakan dan cara penanganan atau solusi permasalahan. Sedangkan untuk *Administrator*, melakukan *login* pada menu *admin*, mengelola data gejala, mengelola data kerusakan, mengelola data bagian kerusakan, mengelola data aturan, mengelola data admin, mengelola data pengunjung, mengelola data riwayat konsultasi, dan *logout* menu *admin*.

2. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. *Activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor (A.S dan Shalahuddin: 2011: 134). Berikut ini adalah *activity diagram* yang dirancang dalam penelitian ini. *Activity diagram* untuk menu *user* disajikan melalui gambar-gambar berikut:

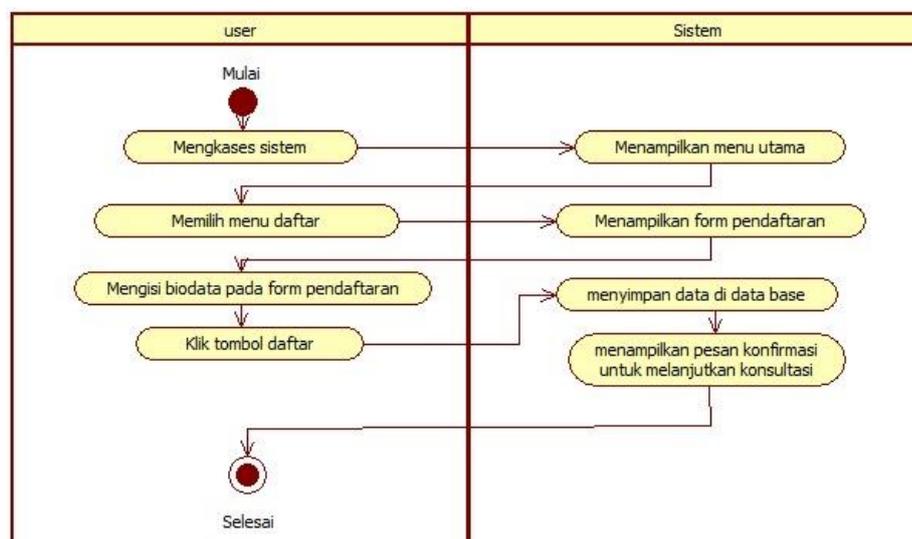
a. Activity diagram melihat beranda utama

Activity diagram melihat beranda merupakan *UML* yang menggambarkan kegiatan sistem saat menerima perintah pengguna saat melihat halaman utama atau beranda utama. Seperti pada gambar 3.5 berikut:

Pada Gambar 3.6 diatas, *user* mulai mengakses beranda utama, kemudian memilih menu petunjuk, maka sistem akan menampilkan halaman petunjuk yang diperlukan oleh *user*. *User* disuguhkan dua pilihan informasi, pertama petunjuk tentang cara penggunaan aplikasi, maka sistem akan menampilkan halaman cara penggunaan sistem ini. Kedua informasi detail tentang kami, jika *user* memilih menu ini maka sistem akan menampilkan halaman detail tentang kami (perancang/penulis). Selesai.

c. Activity diagram daftar

Activity diagram daftar merupakan diagram *UML* yang menggambarkan kegiatan sistem saat menerima perintah *user* saat melakukan pendaftaran sebelum melakukan konsultasi. Berikut ini adalah gambar *activity diagram* Daftar (Gambar 3.7):

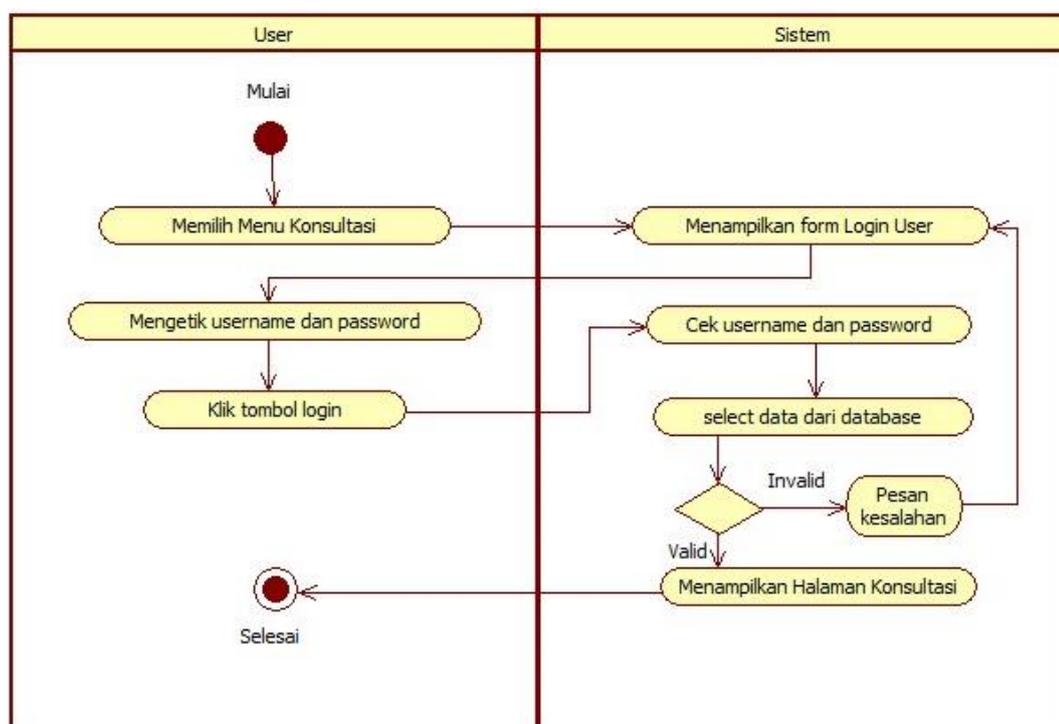


Gambar 3. 7 Activity Diagram Daftar
Sumber: Data penelitian, 2018

Pada Gambar 3.7 diatas,*user* memulai dengan mengakses sistem, kemudian sistem akan menampilkan menu utama. Selanjutnya *user* memilih menu daftar, sistem

akan menampilkan *form* pendaftaran. Dalam *form* tersebut *user* diminta untuk mengisi data identitas diri kemudian menekan tombol daftar. Sistem akan menyimpan data *user* di dalam *database*. Kemudian sistem akan menampilkan pesan konfirmasi apakah *user* akan melakukan melanjutkan konsultasinya. Selesai.

d. *Activity diagram login user*



Gambar 3. 8 *Activity Diagram Login User*

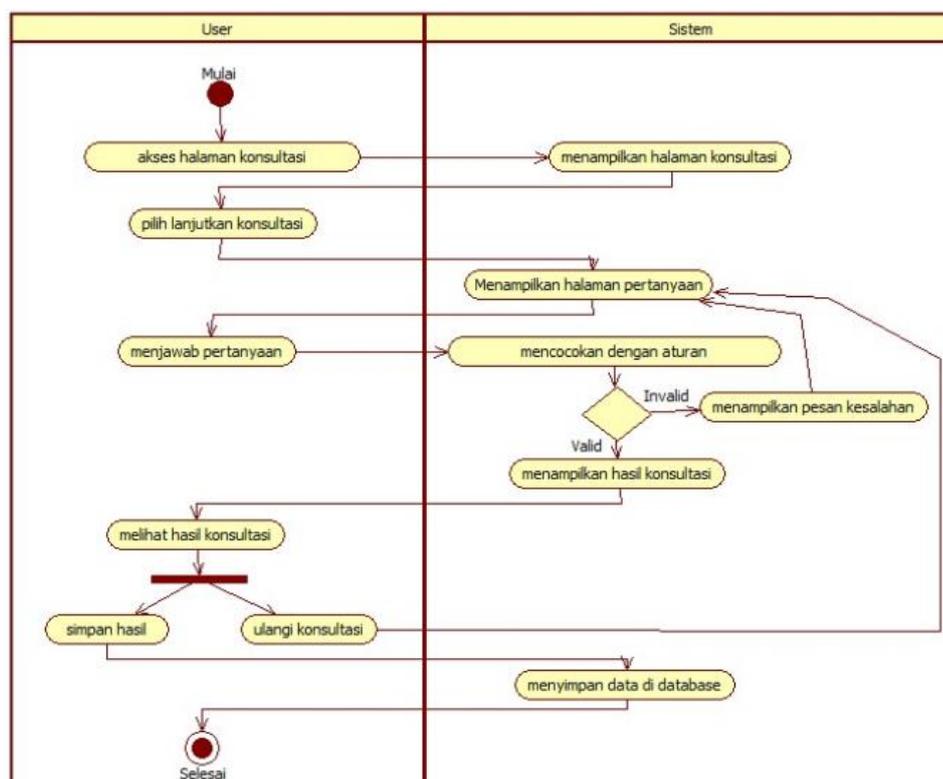
Sumber: Data penelitian, 2018

Pada Gambar 3.8 diatas proses *login user* adalah *user* memilih menu konsultasi, maka sistem akan menampilkan *form* menu *login user*. *User* mengetik *username* dan *password* kemudian mengklik tombol *login*. Maka sistem akan mengecek *username* dan *password* kemudian dicocokkan dengan data yang ada di *database*. Jika *username* dan *password* tidak sesuai dengan yang ada di *database* maka

sistem akan menampilkan pesan kesalahan dan sistem kembali menampilkan *form* menu *loginuser*, apabila sesuai dengan *database*, maka sistem akan menampilkan halaman konsultasi. Selesai.

e. *Activity diagram* konsultasi

Activity diagram konsultasi merupakan diagram *UML* yang menggambarkan kegiatan sistem saat menerima perintah pengguna (*user*) dalam menggunakan menu konsultasi. Berikut ini gambar *activity diagram* konsultasi (Gambar 3.9):



Gambar 3.9 *Activity Diagram* Konsultasi

Sumber: Data penelitian, 2018

Pada Gambar 3.9 diatas, dimulai pengguna telah mengakses halaman konsultasi, maka sistem akan menampilkan halaman konsultasi yang terdapat tombol lanjutkan konsultasi sebagai aksi konfirmasi, kemudian *user* memilih tombol lanjutkan konsultasi, maka sistem mengambil data dari *database* dan

menampilkan pertanyaan pada *form* konsultasi. Pengguna menjawab pertanyaan yang berupa *multiple choice* ya atau tidak. Kemudian sistem akan mengambil data dari *database* dan membandingkan jawaban dengan data aturan di *database*. Kemudian sistem akan memasukkan data hasil ke *database* sementara dan mengambil data dari *database* untuk ditampilkan. Pengguna akan melihat hasil diagnosa. Jika pengguna menekan tombol diagnosa ulang maka sistem akan mengulangi diagnosa dan menampilkan kembali pertanyaan pada *form* diagnosa. Jika pengguna tidak mengulangi dengan memilih tombol simpan hasil diagnosa maka sistem akan menyimpan di *database* hasil diagnosa tersebut dan proses selesai.

f. *Activity diagram* melihat riwayat konsultasi

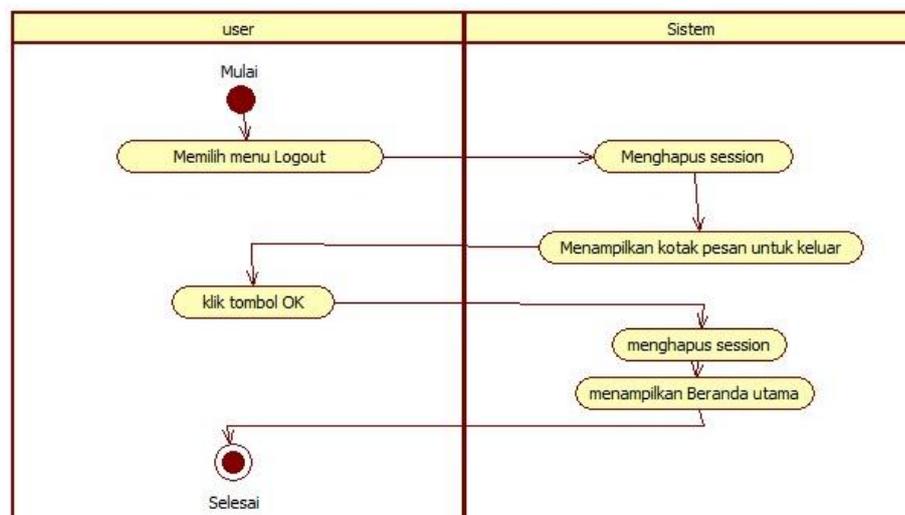
Activity diagram melihat riwayat konsultasi merupakan diagram *UML* yang menggambarkan kegiatan sistem saat melakukan perintah pengguna (*user*) saat ingin melihat hasil dari konsultasinya pada menu riwayat konsultasi. Berikut ini gambar *activity diagram* melihat riwayat konsultasi (Gambar 3.10):



Gambar 3. 10 *Activity Diagram* Melihat Riwayat Konsultasi
Sumber: Data penelitian, 2018

Pada Gambar 3.10 diatas, *user* mulai mengakses berandautama konsultasi kemudian sistem menampilkan beranda utama konsultasi. *User* memilih menu riwayat konsultasi. Sistem akan menampilkan halaman riwayat konsultasi. Selesai.

g. Activity diagram logout user



Gambar 3. 11 *Activity Diagram Logout User*

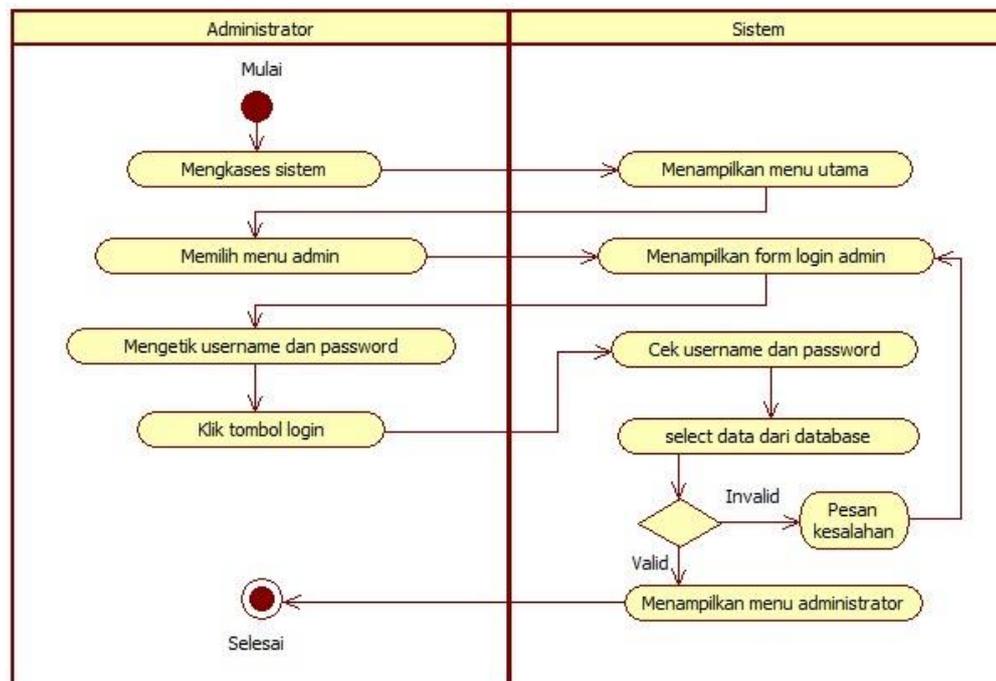
Sumber: Data penelitian, 2018

Pada Gambar 3.11 diatas, *user* mulai dengan memilih menu *logout* kemudian sistem akan menghapus *session* dan menampilkan kotak pesan konfirmasi untuk keluar. *User* memilih tombol “*ok*”. Sistem akan menampilkan halaman beranda utama. Selesai

Selanjutnya *Activity diagram* untuk menu *Administrator* disajikan dalam gambar-gambar berikut:

a. Activity diagram login menu admin

Activity diagram login administrator merupakan UML yang menggambarkan kegiatan pengguna pada halaman login administrator. Dapat dilihat pada Gambar 3.12 berikut:



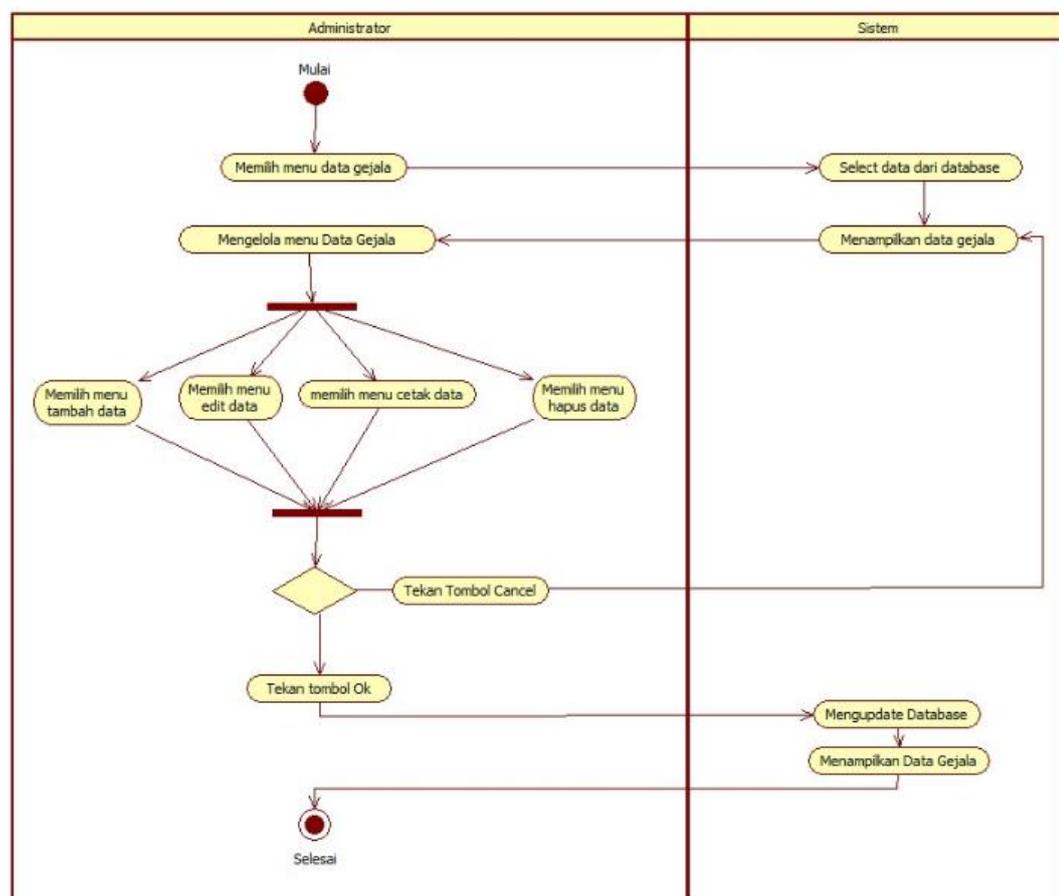
Gambar 3. 12 Activity Diagram Login Menu Admin
Sumber: Data penelitian, 2018

Pada Gambar 3.12 diatas, proses login administrator adalah administrator mengakses sistem, kemudian sistem akan menampilkan halaman menu utama. Selanjutnya administrator memilih menu admin, maka sistem akan menampilkan form menu login admin. Administrator mengetik username dan password kemudian klik tombol login. Maka sistem akan mengecek username dan password kemudian dicocokkan dengan data yang ada di database. Jika username dan password tidak sesuai dengan yang ada di database maka sistem akan

menampilkan pesan kesalahan dan sistem kembali menampilkan *form* menu *login* admin, apabila benar maka sistem akan menampilkan menu *administrator*.

b. *Activity diagram* mengelola data gejala

Activity diagram mengelola data gejala merupakan diagram *UML* yang menggambarkan kegiatan *administrator* dalam mengelola data gejala. Berikut ini gambar *activity diagram* mengelola data gejala (Gambar 3.13):



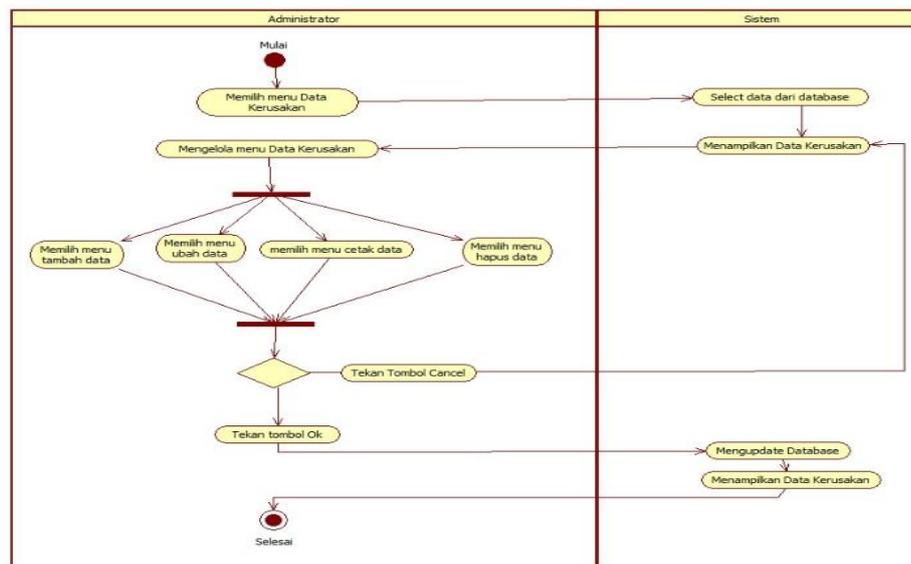
Gambar 3. 13 *Activity Diagram* Mengelola Data Gejala

Sumber: Data penelitian, 2018

Pada Gambar 3.13 diatas, *administrator* mulai memilih menu data gejala, kemudian sistem melakukan select data dari database dan menampilkan data gejala. Dalam mengeloladata gejala *administrator* melihat ada 4 pilihan yaitu

tambah, cetak data, ubah dan hapus. Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* gejala. *Administrator* akan mengisi *form tambah* gejala, kemudian menekan tombol “ok”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang ketiga *administrator* mengisi *form* ubah gejala, setelah itu menekan tombol “ok” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

c. *Activity diagram* mengelola data kerusakan

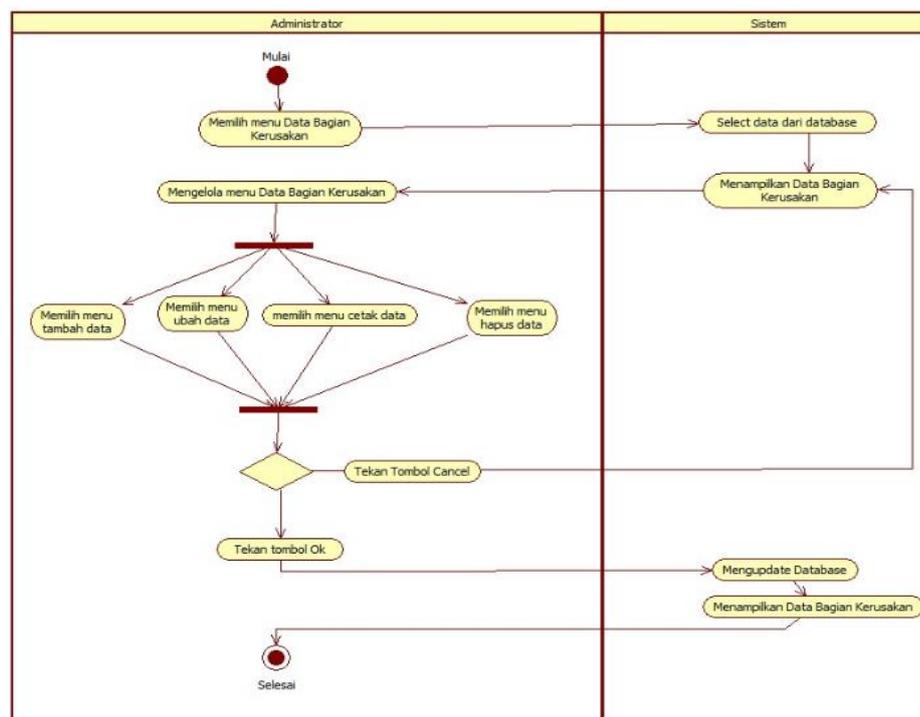


Gambar 3. 14 *Activity Diagram* Mengelola Data Kerusakan
Sumber: Data penelitian, 2018

Pada Gambar 3.14 diatas, *administrator* mulai memilih menu data kerusakan, kemudian sistem melakukan *select* data dari *database* dan menampilkan data kerusakan. Dalam mengelola data kerusakan, *administrator* melihat ada 4 pilihan, yaitu tambah, cetak data, ubah dan hapus. Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* kerusakan. *Administrator* akan mengisi *form tambah* kerusakan, kemudian menekan tombol “ok”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang ketiga *administrator* mengisi *form* ubah kerusakan, setelah itu menekan tombol “ok” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

d. *Activity diagram* mengelola data bagian kerusakan

Activity diagram mengelola data bagian kerusakan merupakan diagram *UML* yang menggambarkan kegiatan sistem dalam menerima perintah *administrator* dalam mengelola data bagian dari kerusakan. Berikut ini adalah gambar *activity diagram* kelola data bagian kerusakan (Gambar 3.15):

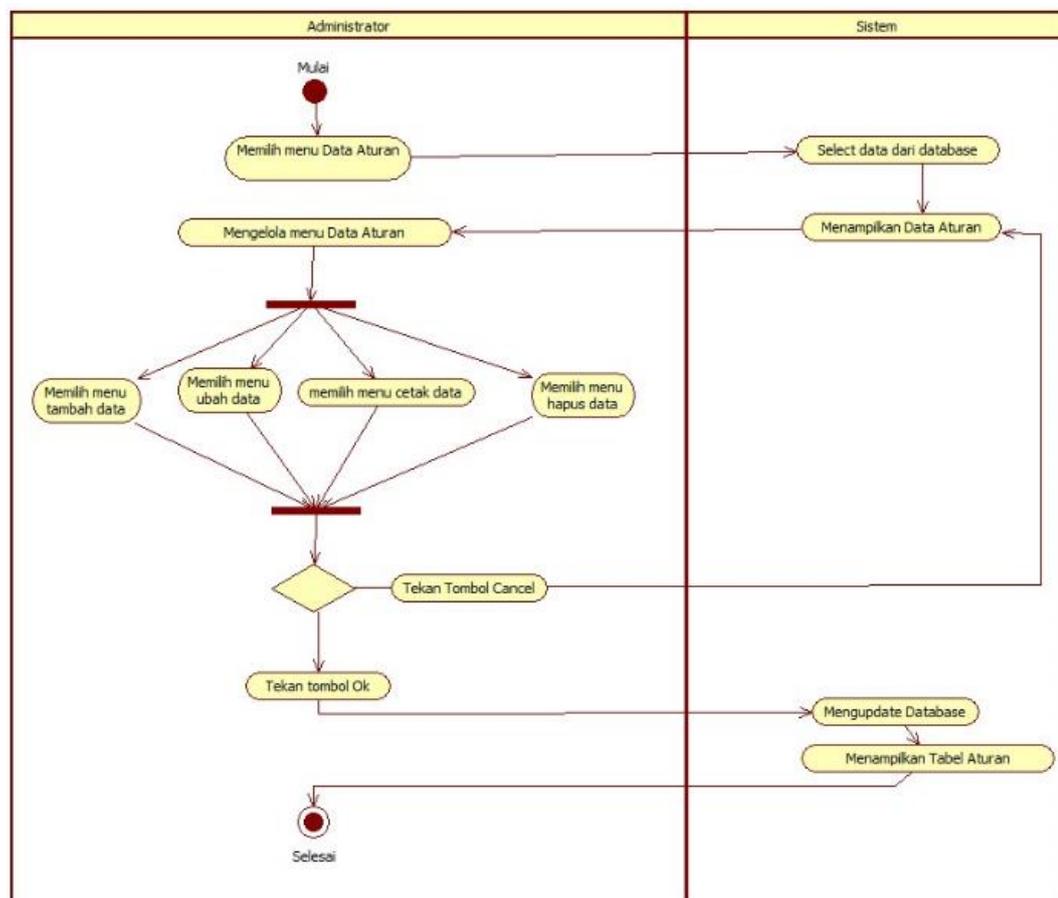


Gambar 3. 15 Activity Diagram Mengelola Data Bagian Kerusakan
Sumber: Data Penelitian, 2018

Pada Gambar 3.15 diatas, *administrator* mulai memilih menu data bagian kerusakan, kemudian sistem melakukan *select* data dari *database* dan menampilkan data bagian kerusakan. Dalam mengeloladata bagian kerusakan, *administrator* melihat ada 4 pilihan, yaitu tambah, cetak data, ubah dan hapus. Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* kerusakan. *Administrator* akan mengisi *form tambah* bagian kerusakan, kemudian menekan tombol “ok”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang

ketiga *administrator* mengisi *form*ubah kerusakan, setelah itu menekan tombol “*ok*” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

e. *Activity diagram* mengelola data aturan



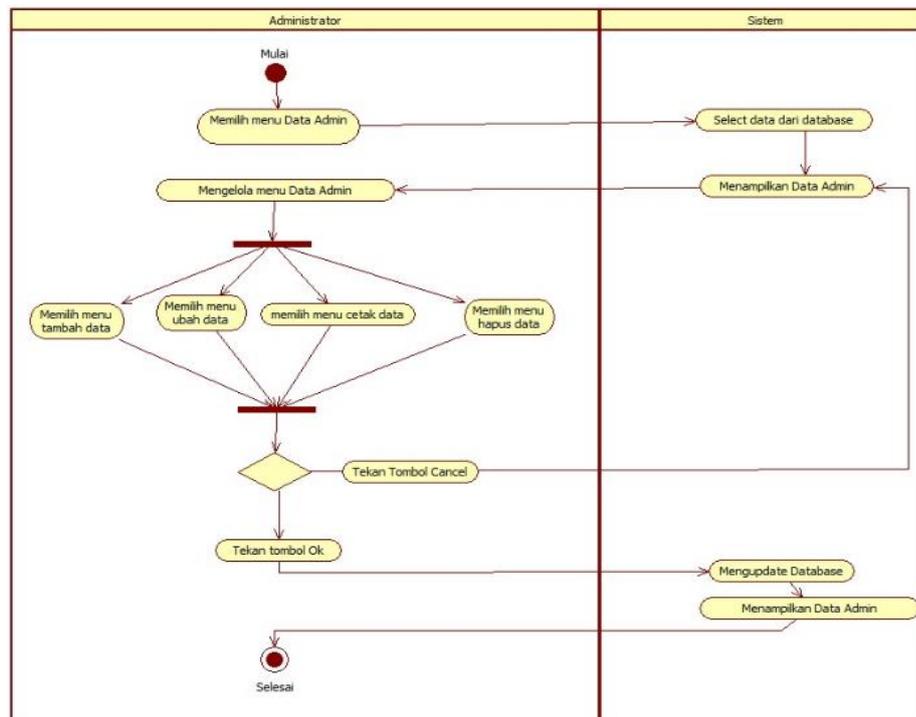
Gambar 3. 16 *Activity Diagram* Mengelola Data Aturan
Sumber: Data penelitian, 2018

Pada Gambar 3.16 diatas, *administrator* mulai memilih menu data aturan, kemudian sistem melakukan select data dari *database* dan menampilkan data aturan. Dalam mengeloladata aturan, *administrator* melihat ada 4 pilihan, yaitu

tambah, cetak data, ubah dan hapus. Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* aturan. *Administrator* akan mengisi *form tambah* aturan, kemudian menekan tombol “*ok*”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “*ok*”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang ketiga administrator mengisi *form* ubah kerusakan, setelah itu menekan tombol “*ok*” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

f. *Activity diagram* mengelola data admin

Activity diagram mengelola data admin merupakan diagram *UML* yang menggambarkan kegiatan *administrator* dalam mengelola data admin. Berikut ini gambar *activity diagram* mengelola data *admin* (Gambar 3.17):



Gambar 3. 17 Activity Diagram Mengelola Data Admin
Sumber: Data penelitian, 2018

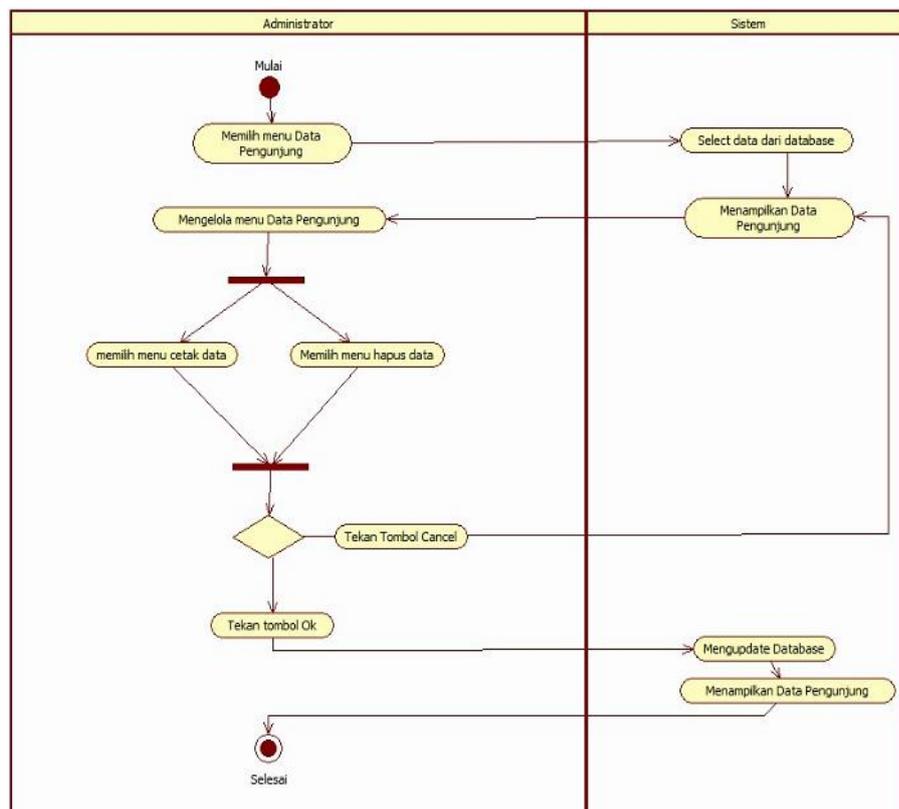
Pada Gambar 3.17 diatas, *administrator* mulai memilih menu data admin, kemudian sistem melakukan *select* data dari *database* dan menampilkan data admin. Dalam mengeloladata admin, *administrator* melihat ada 4 pilihan, yaitu tambah, cetak data, ubah dan hapus. Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* admin. *Administrator* akan mengisi *form tambah* admin, kemudian menekan tombol “ok”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang ketiga administrator mengisi *form*ubah admin, setelah itu menekan tombol “ok”

maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

g. *Activity diagram* mengelola data riwayat pengunjung

Activity diagram mengelola data pengunjung merupakan diagram *UML* yang menggambarkan kegiatan *administrator* dalam mengelola data pengunjung.

Berikut ini gambar *activity diagram* mengelola data pengunjung (Gambar 3.18):



Gambar 3. 18 *Activity Diagram* Mengelola Data Riwayat Pengunjung
Sumber: Data penelitian, 2018

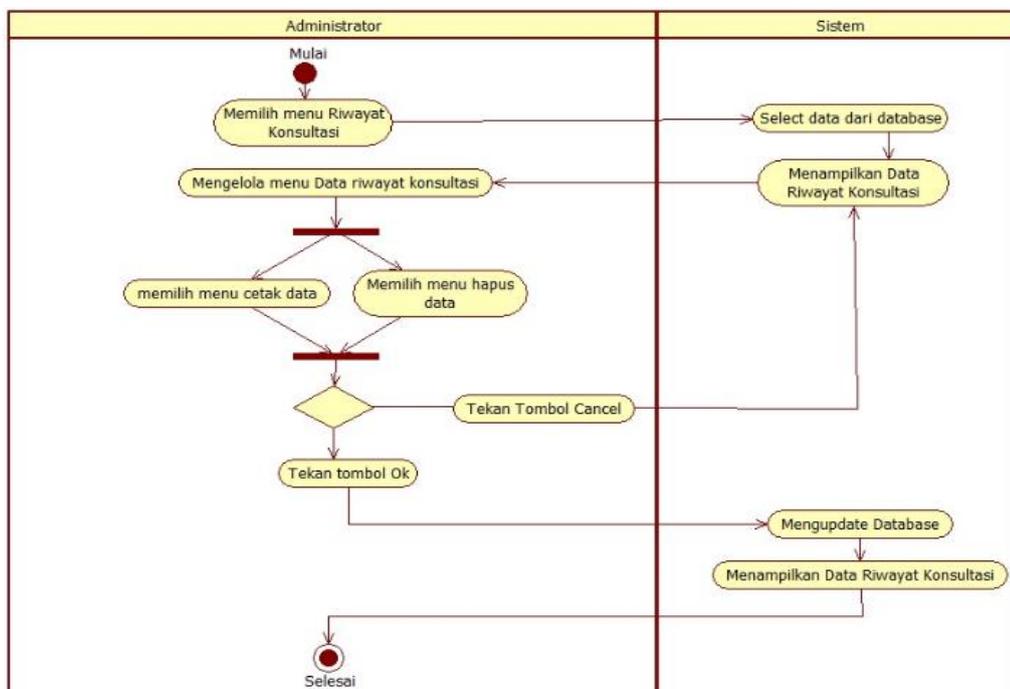
Pada Gambar 3.18 diatas, *administrator* mulai memilih menu data pengunjung, kemudian sistem mengambil data dari *database* dan menampilkan halamandata

pengunjung. Dalam mengelola data pengunjung, *administrator* melihat ada 2 pilihan yaitu cetak data, dan hapus. Pertama, jika *administrator* menekan tombol cetak data maka sistem akan mengambil data dari *database* kemudian menampilkan halaman *Print*. *administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang kedua, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

h. *Activity diagram* mengelola data riwayat konsultasi

Activity diagram riwayat konsultasi merupakan diagram *UML* yang menggambarkan kegiatan *administrator* dalam mengelola riwayat konsultasi.

Berikut ini gambar *activity diagram* mengelola riwayat konsultasi (Gambar 3.19):

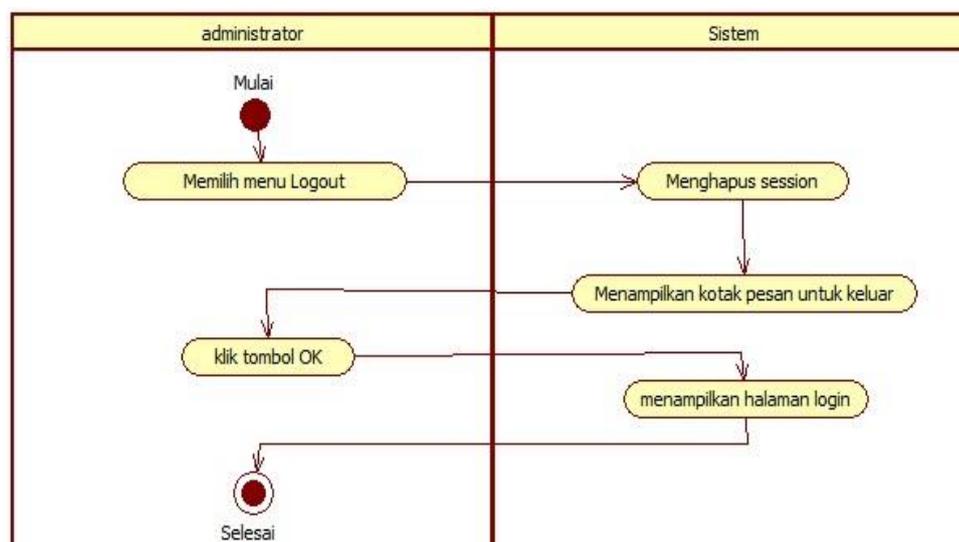


Gambar 3. 19 *Activity Diagram* Mengelola Data Riwayat Konsultasi

Sumber: Data penelitian, 2018

Pada Gambar 3.19 diatas, *administrator* mulai memilih menu data riwayat konsultasi sehingga sistem mengambil data dari *database* dan menampilkan data riwayat konsultasi. Dalam mengelola data riwayat konsultasi *administrator* melihat ada 2 pilihan yaitu cetak data, dan hapus. Pertama, jika *administrator* menekan tombol cetak data maka sistem akan mengambil data dari *database* kemudian menampilkan halaman *Print*. Administrator menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang kedua, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

i. *Activity diagram logout admin*



Gambar 3. 20 *Activity Diagram Logout Admin*
Sumber: Data penelitian, 2018

Pada gambar 3.20 menunjukkan *activity diagram* menu *logout*. Pada saat administrator memilih menu *logout*, sistem akan menghapus *session* dan

menampilkan kotak pesan untuk keluar. Saat *administrator* mengklik tombol “OK”, maka sistem akan mengarahkan ke halaman *login*.

3. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (A.S dan Shalahuddin: 2011: 137). Berikut ini adalah *sequence diagram* yang menggambarkan aktifitas yang dilakukan oleh *user* dalam penelitian ini:

a. Sequence diagram melihat beranda utama

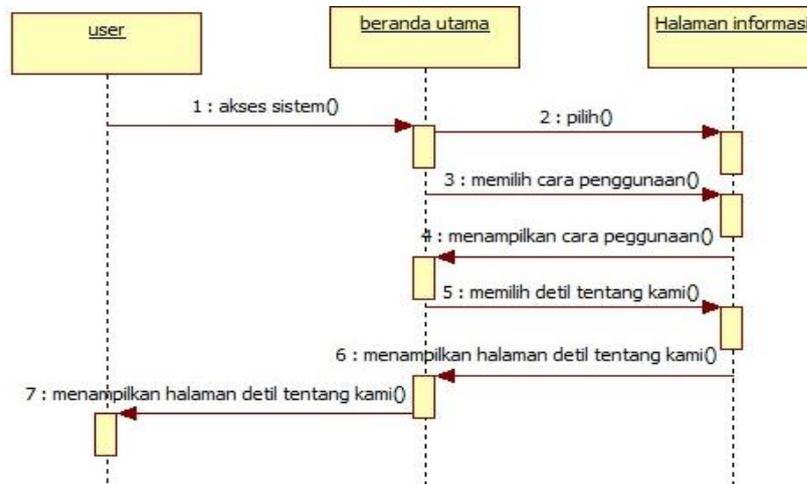


Gambar 3. 21 *Sequence Diagram* Melihat Beranda Utama

Sumber: Data penelitian, 2018

Pada gambar 3.21 disajikan *sequence diagram* melihat beranda utama. Pada saat *user* memilih menu beranda, sistem akan menampilkan halaman beranda

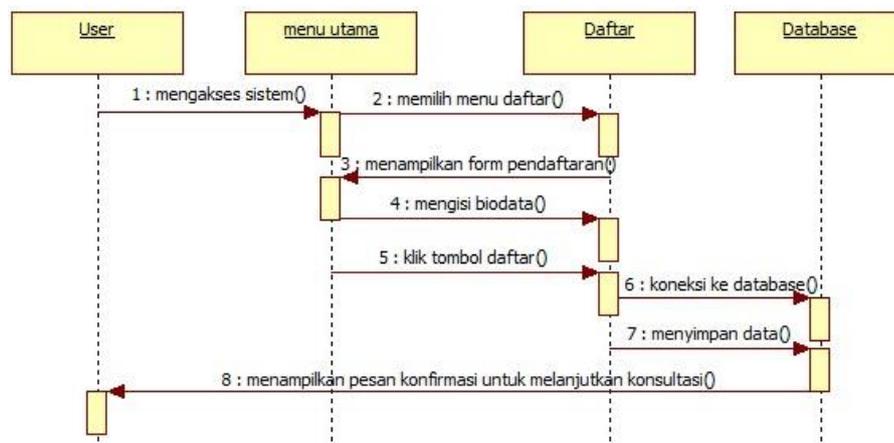
b. *Sequence diagram* melihat petunjuk



Gambar 3. 22 *Sequence Diagram* Melihat Petunjuk
Sumber: Data penelitian, 2018

Pada gambar 3.22 di atas menunjukkan pada saat user mengakses sistem, maka akan di arahkan kehalaman beranda utama, kemudian dipilih halaman petunjuk, saat *user* memilih cara penggunaan, maka sistem akan menampilkan halaman cara penggunaan, begitu juga saat *user* memilih menu detail tentang kami maka sistem akan menampilkan halaman detail tentang kami. Proses selesai.

c. *Sequence diagram* daftar

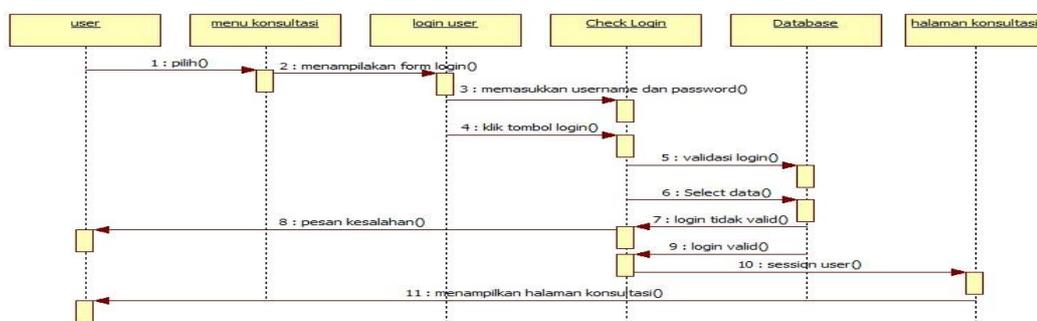


Gambar 3. 23 *Sequence Diagram* Daftar
Sumber: Data penelitian, 2018

Pada gambar 3.23 diatas, menunjukkan saat user mengakses sistem pada menu utama dan memilih menu daftar, maka sistem akan menampilkan form pendaftaran, kemudian *user* akan melakukan pengisian biodata, setelah selesai mengisi biodata maka selanjutnya *user* akan menekan tombol daftar, sistem akan melakukan koneksi ke *database* dan menyimpan data tersebut. Selanjutnya sistem akan menampilkan pesan konfirmasi untuk melanjutkan konsultasi. Proses daftar selesai.

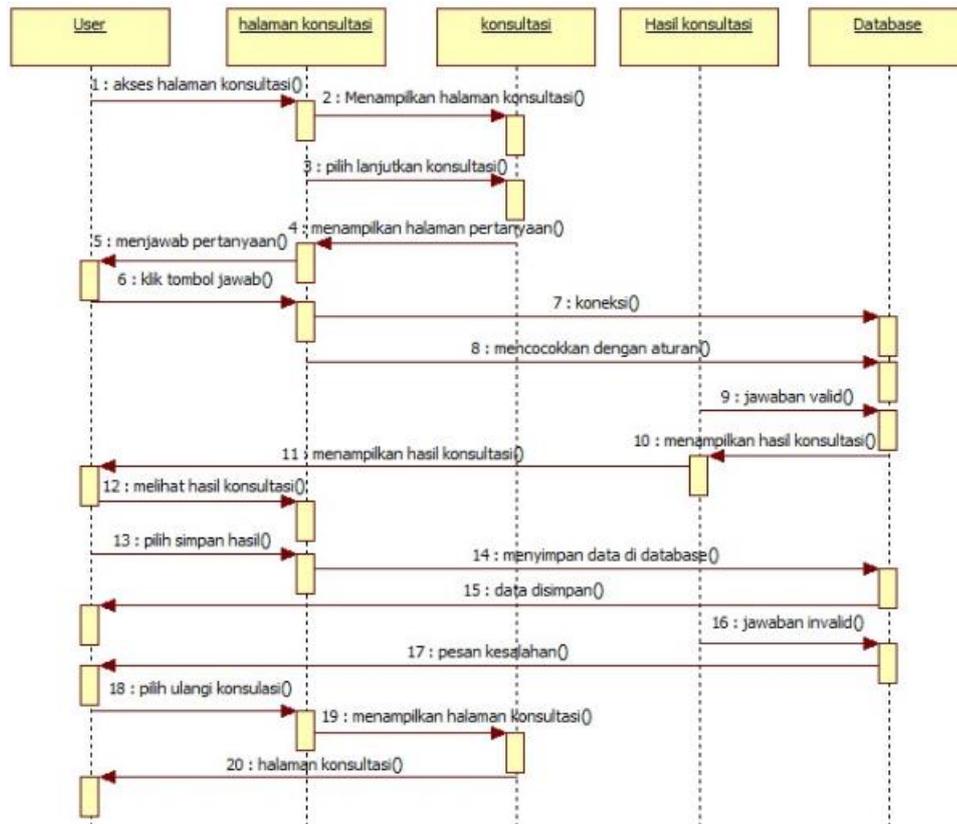
d. Sequence diagram login user

Pada gambar 3.24 dibawah ini akan menjelaskan kegiatan *user* saat melakukan *login* disajikan dalam *diagram sequence*, berikut penjelasannya: dimulai saat *user* memilih menu konsultasi, kemudian sistem akan menampilkan *form login*, *user* akan melakukan pengisian *form* tersebut dengan memasukkan *username* dan *password* kemudian menekan tombol *login*. Selanjutnya sistem akan melakukan koneksi dengan *database* untuk melakukan validasi *username* dan *password*, jika data yang dimasukkan tidak *valid* maka sistem akan menampilkan pesan kesalahan, jika data *valid* maka sistem akan memuat *session user* dan menampilkan halaman konsultasi.



Gambar 3. 24 Sequence Diagram Login User
Sumber: Data penelitian, 2018

e. *Sequence diagram* konsultasi



Gambar 3. 25 *Sequence Diagram* Konsultasi

Sumber: Data penelitian, 2018

Pada gambar 3.25 di atas disajikan *sequence diagram* konsultasi. Pada saat mengakses halaman konsultasi, maka sistem akan menampilkan halaman konsultasi dan *user* akan mengklik tombol lanjutkan konsultasi, kemudian sistem akan menampilkan halaman pertanyaan yang akan di jawab oleh *user* dengan memilih salah satu pilihan jawaban (“ya” atau “tidak”) dan mengklik tombol jawab. Selanjutnya sistem akan melakukan *check* jawaban dengan aturan, jika jawaban *valid* maka sistem akan menampilkan hasil konsultasi. Jika *user* memilih tombol simpan hasil, maka sistem akan menyimpan hasil konsultasi tersebut ke dalam *database*. Dan jika *user* masih ingin melakukan konsultasi

ulang, maka *user* akan memilih tombol ulangi konsultasi, maka sistem tidak akan menyimpan hasil konsultasi tersebut ke dalam *database* dan menampilkan halaman konsultasi. Kemudian jika jawaban dari *user* tidak *valid* (tidak sesuai dengan aturan) maka sistem akan menampilkan pesan kesalahan dan akan menampilkan pilihan tombol ulangi konsultasi, maka sistem akan menampilkan halaman konsultasi.

f. *Sequence diagram* melihat riwayat konsultasi

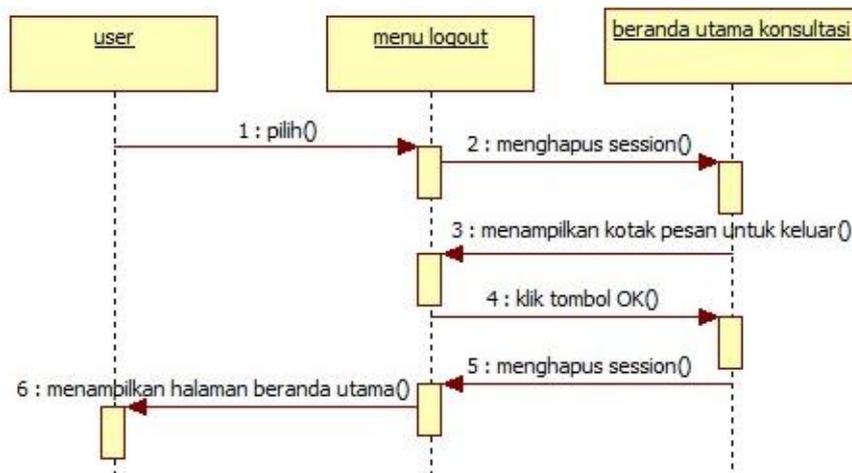


Gambar 3. 26 *Sequence Diagram* Melihat Riwayat Konsultasi

Sumber: Data penelitian, 2018

Pada gambar 3.26 di atas menunjukkan pada saat *user* mengakses sistem, maka akan di arahkan kehalaman beranda utama konsultasi, kemudian dipilih menu riwayat konsultasi, maka sistem akan menampilkan halaman riwayat konsultasi dan hasil konsultasi. Proses selesai.

g. *Sequence diagram logout user*



Gambar 3. 27 *Sequence Diagram Logout User*
Sumber: Data penelitian, 2018

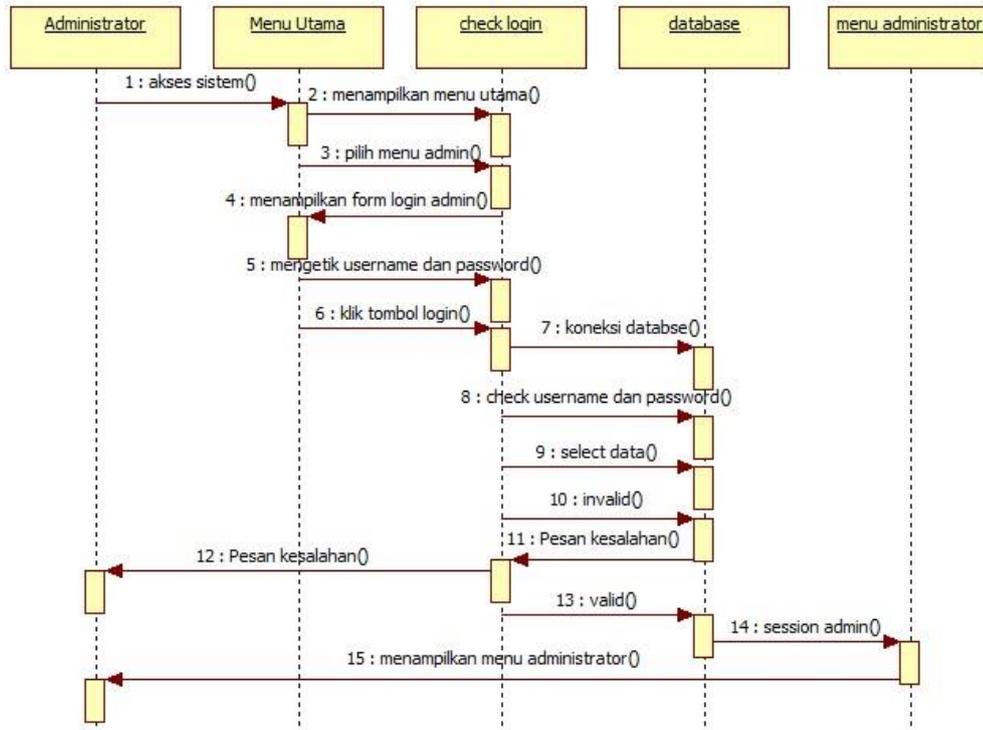
Pada gambar 3.27 disajikan diagram *sequence logout*. Pada saat *user* memilih menu *logout*, sistem akan menampilkan kotak pesan untuk keluar dan memilih tombol “ok”, maka sistem akan menghapus *session* pada halaman beranda utama konsultasi dan mengarahkan *user* ke halaman beranda utama.

Adapun *sequence diagram* untuk menggambarkan aksi yang dilakukan oleh *administrator* adalah sebagai berikut:

a. *Sequence diagram login*

Pada Gambar 3.28 dibawah ini, proses *login administrator* adalah *administrator* mengakses sistem, kemudian sistem akan menampilkan halaman menu utama. Selanjutnya *administrator* memilih menu admin, maka sistem akan menampilkan *form* menu *login* admin. *Administrator* mengetik *username* dan *password* kemudian klik tombol *login*. Maka sistem akan mengecek *username* dan *password* kemudian dicocokkan dengan data yang ada di *database*. Jika *username* dan *password* tidak sesuai (*invalid*) dengan yang ada di *database* maka

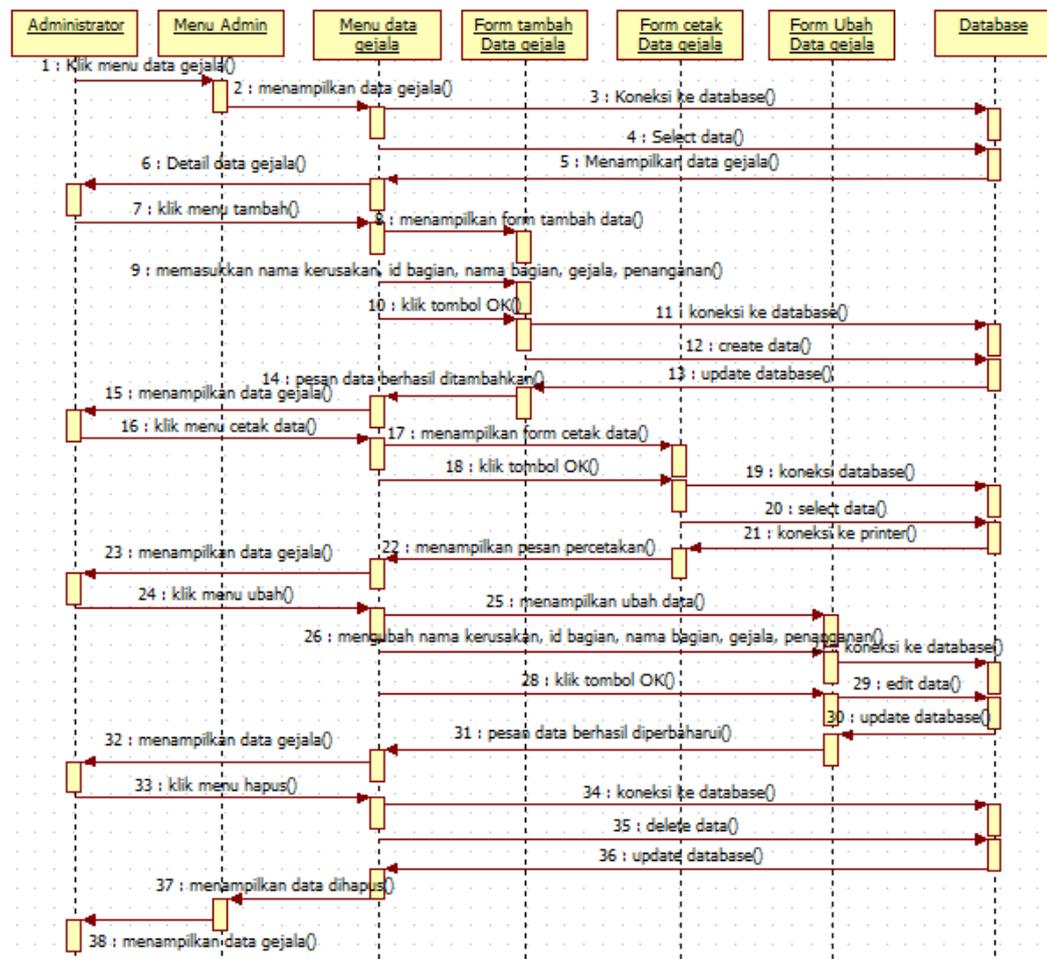
sistem akan menampilkan pesan kesalahan, apabila benar maka sistem akan memuat *session* admin dan menampilkan menu *administrator*.



Gambar 3. 28 *Sequence Diagram Login*
Sumber: Data penelitian, 2018

b. *Sequence diagram* mengelola data gejala

Pada Gambar 3.29 dibawah ini, *administrator* mulai memilih menu data gejala, kemudian sistem melakukan *select* data dari *database* dan menampilkan data gejala. Dalam mengeloladata gejala *administrator* melihat ada 4 pilihan yaitu tambah, cetak data, ubah dan hapus.

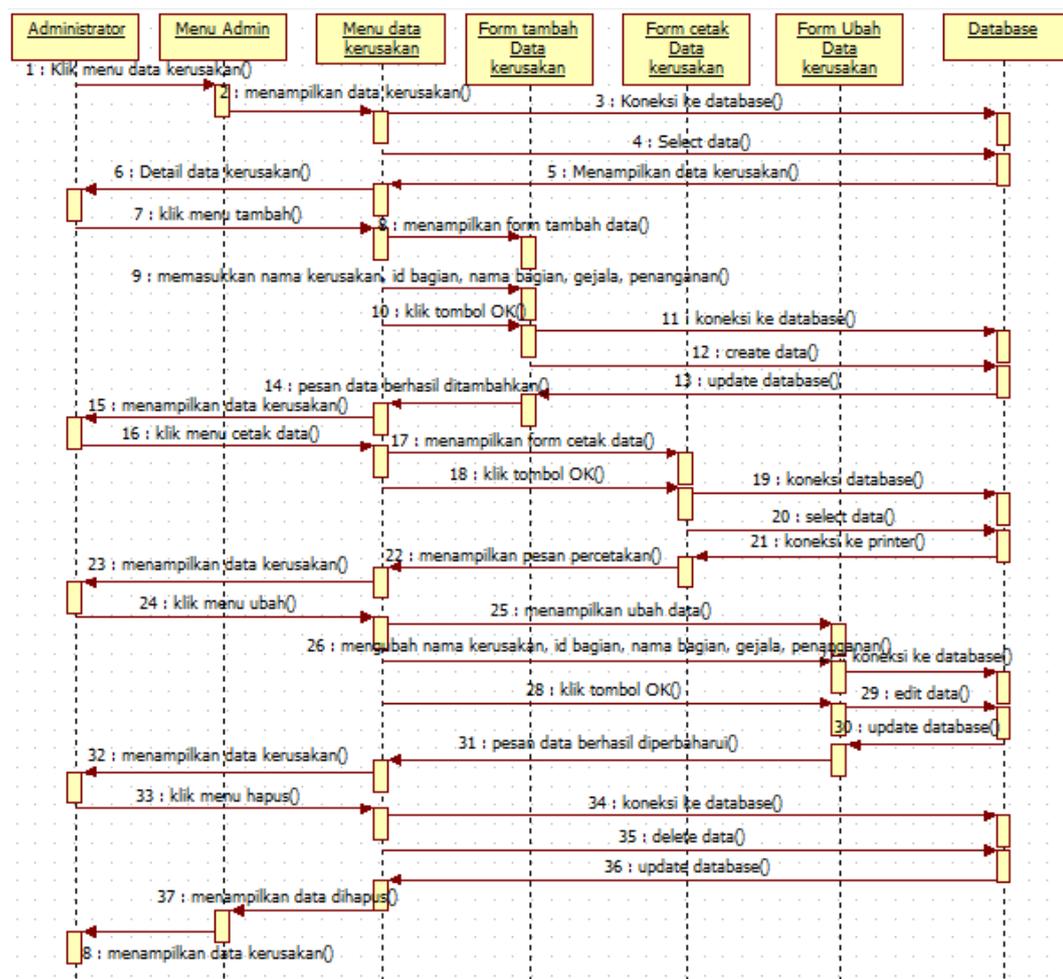


Gambar 3. 29 Sequence Diagram Mengelola Data Gejala
Sumber: Data penelitian, 2018

Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* gejala. *Administrator* akan mengisi *form tambah* gejala, kemudian menekan tombol “ok”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang ketiga *administrator*

mengisi *form* ubah gejala, setelah itu menekan tombol “ok” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

c. *Sequence diagram* mengelola data kerusakan



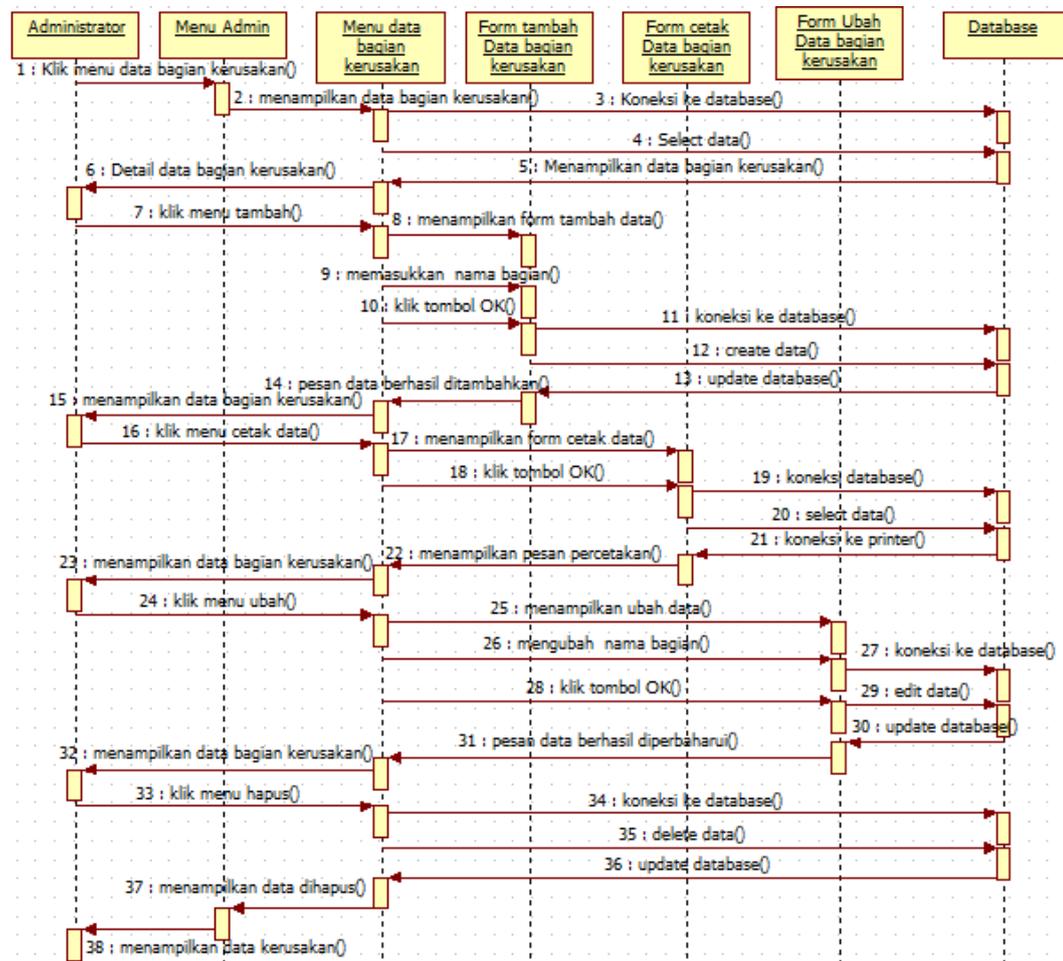
Gambar 3. 30 *Sequence Diagram* Mengelola Data Kerusakan
Sumber: Data Penelitian, 2018

Pada Gambar 3.30 diatas, *administrator* mulai memilih menu data kerusakan, kemudian sistem melakukan *select* data dari *database* dan menampilkan data

kerusakan. Dalam mengeloladata kerusakan, *administrator* melihat ada 4 pilihan, yaitu tambah, cetak data, ubah dan hapus. Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* kerusakan. *Administrator* akan mengisi *form tambah* kerusakan, kemudian menekan tombol “ok”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang ketiga *administrator* mengisi *form* ubah kerusakan, setelah itu menekan tombol “ok” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

d. *Sequence diagram* mengelola data bagian kerusakan

Pada Gambar 3.31, *administrator* mulai memilih menu data bagian kerusakan, kemudian sistem melakukan *select* data dari *database* dan menampilkan data bagian kerusakan. Dalam mengeloladata bagian kerusakan, *administrator* melihat ada 4 pilihan, yaitu tambah, cetak data, ubah dan hapus.

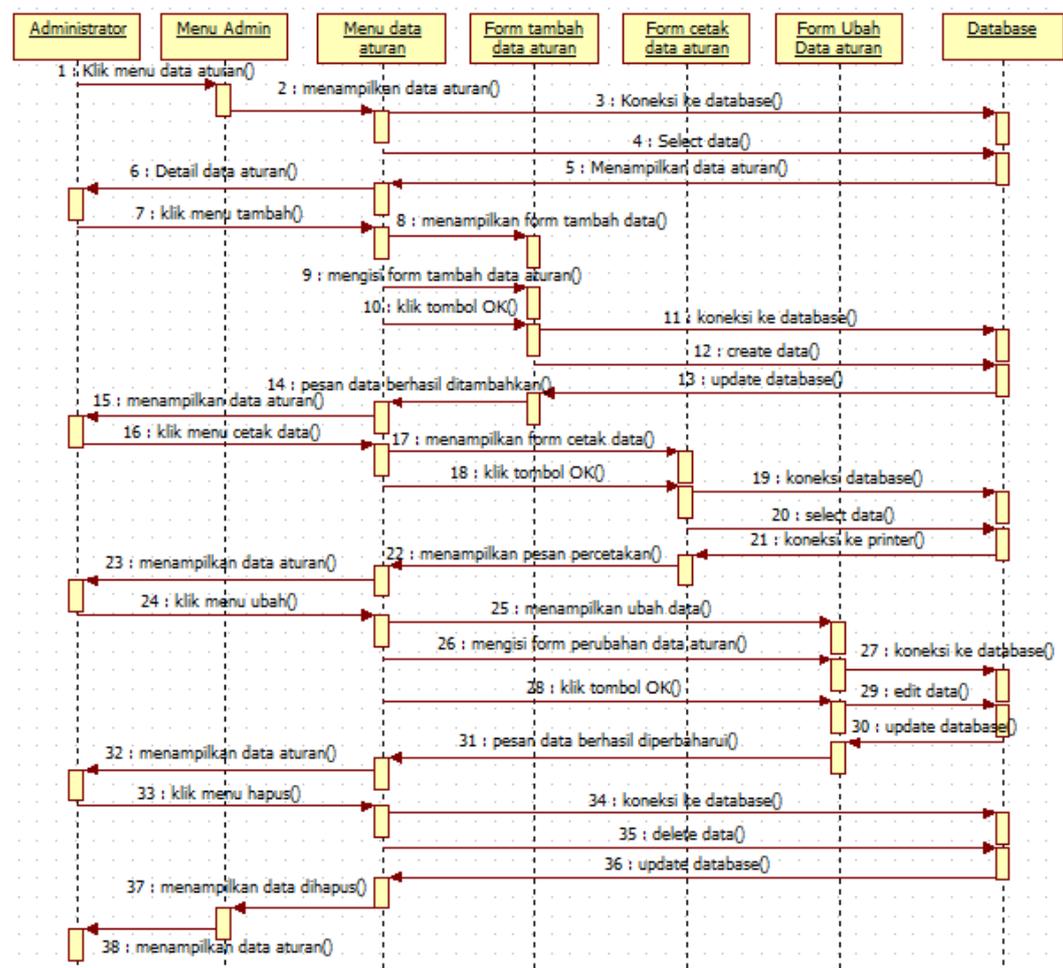


Gambar 3. 31 *Sequence Diagram* Mengelola Data Bagian Kerusakan
Sumber: Data penelitian, 2018

Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* kerusakan. *Administrator* akan mengisi *form tambah* bagian kerusakan, kemudian menekan tombol “ok”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang

ketiga *administrator* mengisi *form* ubah kerusakan, setelah itu menekan tombol “ok” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

e. *Sequence diagram* mengelola data aturan



Gambar 3. 32 *Sequence Diagram* Mengelola Data Aturan

Sumber: Data penelitian, 2018

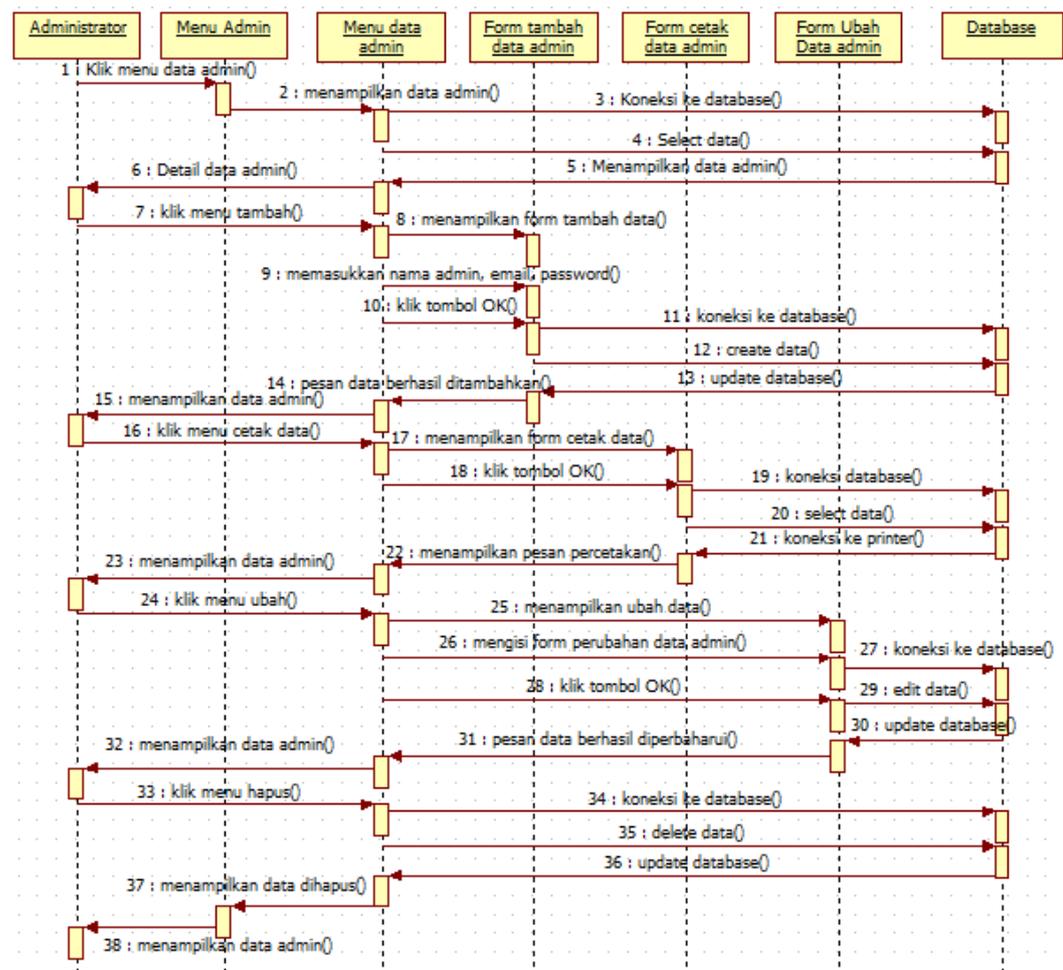
Pada Gambar 3.32 diatas, *administrator* mulai memilih menu data aturan, kemudian sistem melakukan *select* data dari *database* dan menampilkan data

aturan. Dalam mengelola data aturan, *administrator* melihat ada 4 pilihan, yaitu tambah, cetak data, ubah dan hapus. Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form* tambah aturan. *Administrator* akan mengisi *form tambah* aturan, kemudian menekan tombol “*ok*”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “*ok*”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang ketiga *administrator* mengisi *form* ubah kerusakan, setelah itu menekan tombol “*ok*” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

f. *Sequence diagram* mengelola data admin

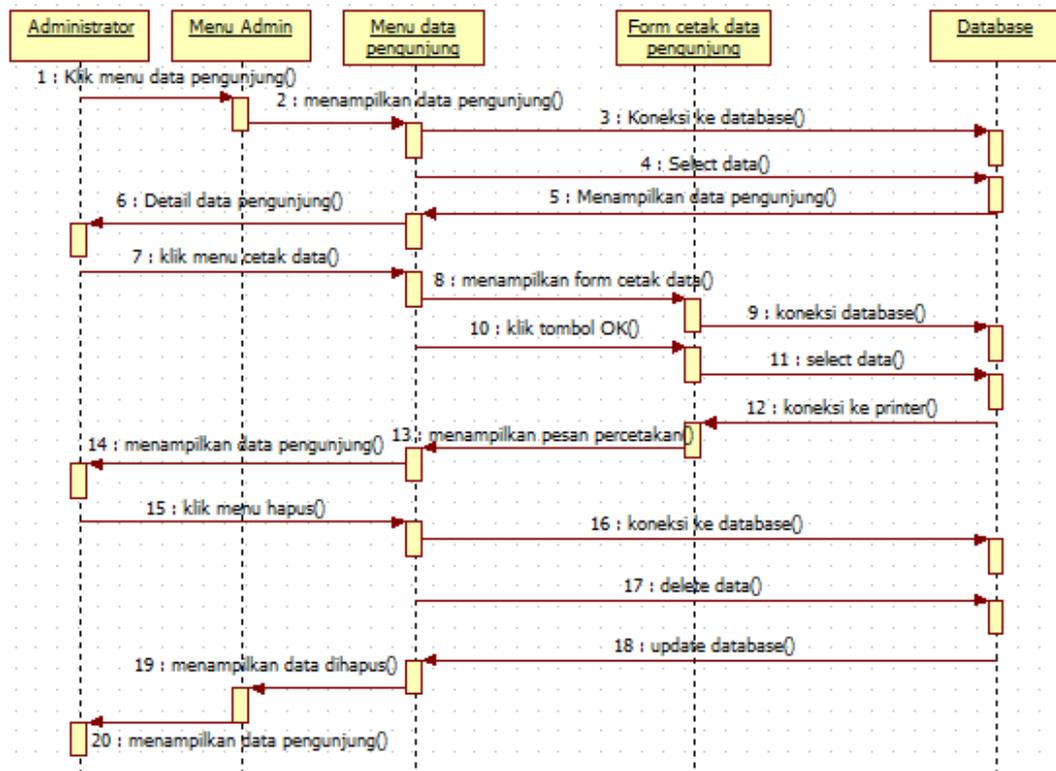
Pada Gambar 3.33 dibawah, *administrator* mulai memilih menu data admin, kemudian sistem melakukan select data dari *database* dan menampilkan data admin. Dalam mengeloladata admin, *administrator* melihat ada 4 pilihan, yaitu tambah, cetak data, ubah dan hapus. Pertama, jika *administrator* menekan tombol tambah maka sistem akan menampilkan *form tambah* admin. *Administrator* akan mengisi *form tambah* admin, kemudian menekan tombol “*ok*”, maka sistem akan memasukkan data baru tersebut ke *database* kemudian menampilkan pesan data berhasil ditambahkan. Kedua, jika *administrator* menekan tombol cetak data maka

sistem akan memilih data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang ketiga *administrator* mengisi *form* ubah admin, setelah itu menekan tombol “ok” maka sistem akan melakukan *update* data ke *database* dan menampilkan pesan data berhasil diperbaharui. Terakhir, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.



Gambar 3. 33 *Sequence Diagram* Mengelola Data Admin
Sumber: Data penelitian, 2018

g. *Sequence diagram* mengelola data riwayat pengunjung

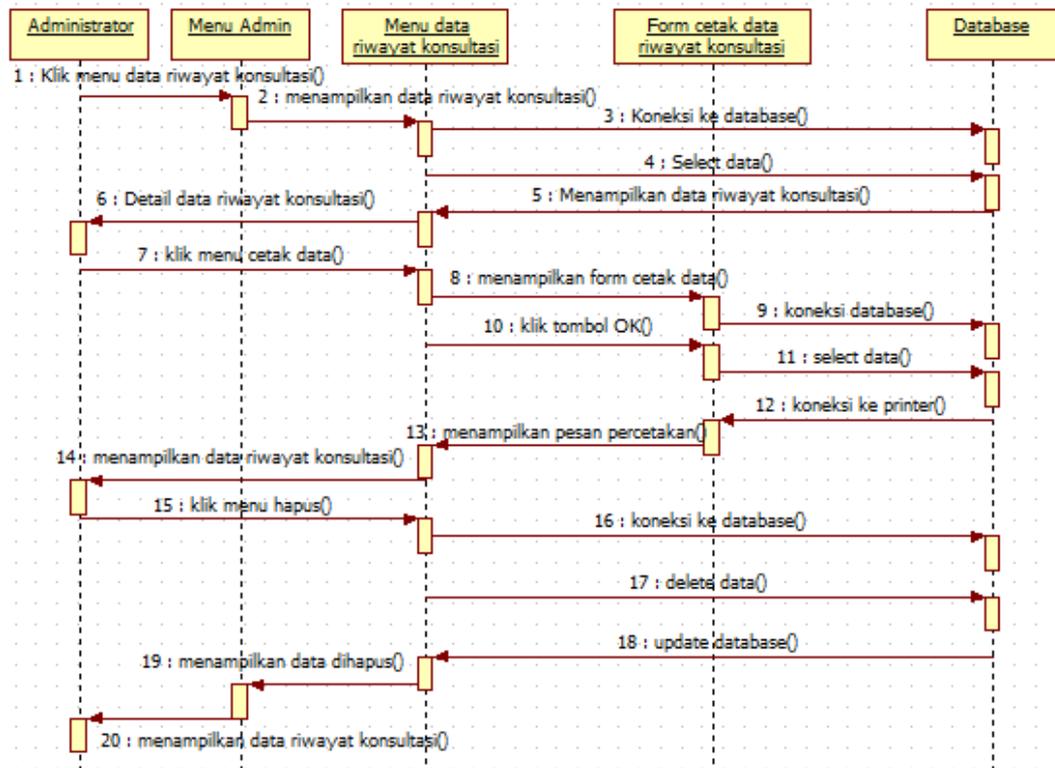


Gambar 3. 34 *Sequence Diagram* Mengelola Data Riwayat Pengunjung

Sumber: Data penelitian, 2018

Pada Gambar 3.34 diatas, *administrator* mulai memilih menu data pengunjung, kemudian sistem mengambil data dari *database* dan menampilkan halamandata pengunjung. Dalam mengelola data pengunjung, *administrator* melihat ada 2 pilihan yaitu cetak data, dan hapus. Pertama, jika *administrator* menekan tombol cetak data maka sistem akan mengambil data dari *database* kemudian menampilkan halaman *Print*. *administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang kedua, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

h. *Sequence diagram* mengelola riwayat konsultasi

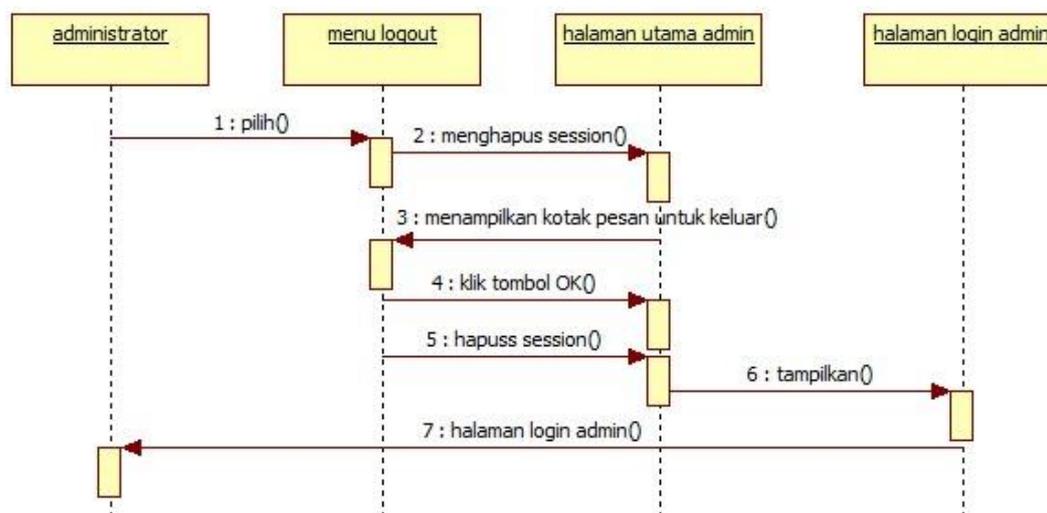


Gambar 3. 35 *Sequence Diagram* Mengelola Riwayat Konsultasi

Sumber: Data penelitian, 2018

Pada Gambar 3.35 diatas, *administrator* mulai memilih menu data riwayat konsultasi sehingga sistem mengambil data dari *database* dan menampilkan data riwayat konsultasi. Dalam mengelola data riwayat konsultasi *administrator* melihat ada 2 pilihan yaitu cetak data, dan hapus. Pertama, jika *administrator* menekan tombol cetak data maka sistem akan mengambil data dari *database* kemudian menampilkan halaman *Print*. *Administrator* menekan tombol “ok”, maka sistem akan mengirim perintah ke perangkat keras *printer* untuk mencetak data yang dimaksud. Kemudian yang kedua, jika *administrator* menekan tombol hapus maka sistem akan menghapus data dari *database* dan menampilkan pesan data dihapus. Proses selesai.

i. *Sequence diagram logout admin*

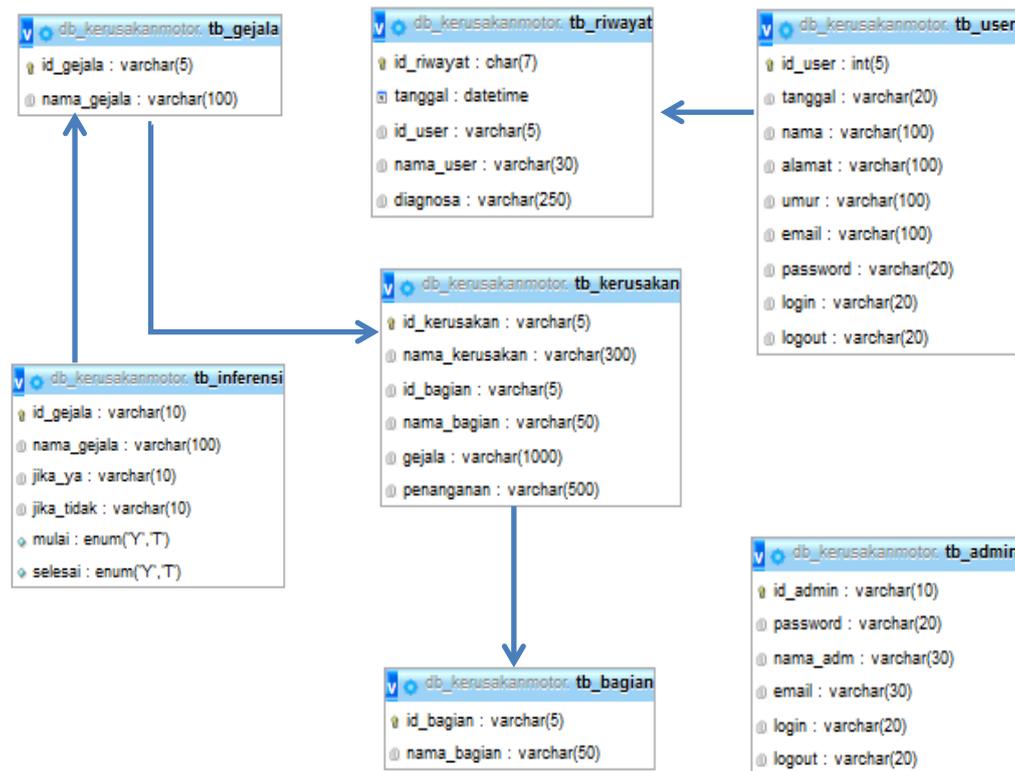


Gambar 3. 36 *Sequence Diagram Logout Admin*
Sumber: Data penelitian, 2018

Pada gambar 3.36 menunjukkan *activity diagram* menu *logout*. Pada saat *administrator* memilih menu *logout*, sistem akan menghapus *session* dan menampilkan kotak pesan untuk keluar. Saat *administrator* mengklik tombol “OK”, maka sistem akan mengarahkan ke halaman *login*.

3.4.5 Desain Database

A.S. dan Shalahuddin (2011: 57) *PDM (Physical Data Model)* adalah model yang menggunakan sejumlah table untuk menggambarkan data serta hubungan antar data-data tersebut. *PDM* merupakan konsep yang menerangkan detail bagaimana data disimpan di dalam *database*. Berikut ini adalah konsep *database* yang dibuat dalam penelitian ini:



Gambar 3. 37 Desain *Physical Data Model*
Sumber: Data penelitian, 2018

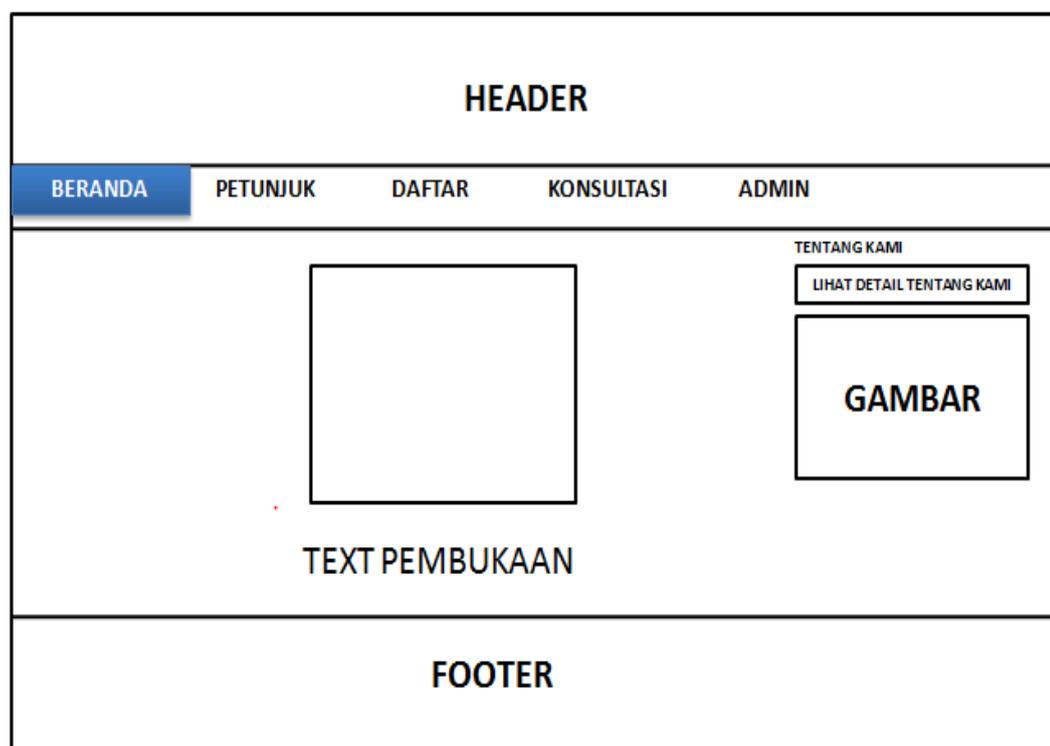
Pada Gambar 3.37 diatas, terdapat 7 tabel *database* yang terdiri dari: tabel gejala, tabel inferensi, tabel riwayat, tabel kerusakan, tabel bagian, tabel *user*, dan tabel *admin*. Pada tabel inferensi adalah tempat untuk penyimpanan data sementara dari pengolahan dari tabel gejala. Pada riwayat, user dapat melihat riwayat hasil konsultasinya. Hasil diagnosa akan disimpan di dalam tabel bagian kerusakan yang sebelumnya di proses di tabel kerusakan setelah data dari tabel gejala yang telah dicocokkan dengan *rule* yang berada di tabel inferensi selesai dilakukan. Dari data gejala yang didapatkan saat diagnosa akan dicocokkan dengan tabel aturan yang sudah dibuat, maka akan diketahui kerusakan yang akan dicocokkan dengan data pada tabel inferensi. Tabel daftar *administrator* berisi nama, *id* admin, *email* dan *password* pengguna.

3.4.6 Desain Antarmuka

Berikut ini adalah desain antarmuka yang akan dibuat pada aplikasi sistem pakar kerusakan sepeda motor Yamaha Nmax. Pada *design* antarmuka ini peneliti hanya membuat *design* untuk tampilan *web*.

4. Rancangan halaman beranda

Pada gambar 3.38 menunjukkan halaman beranda, yaitu berisi tampilan awal sistem pakar untuk mendeteksi kerusakan sepeda motor.



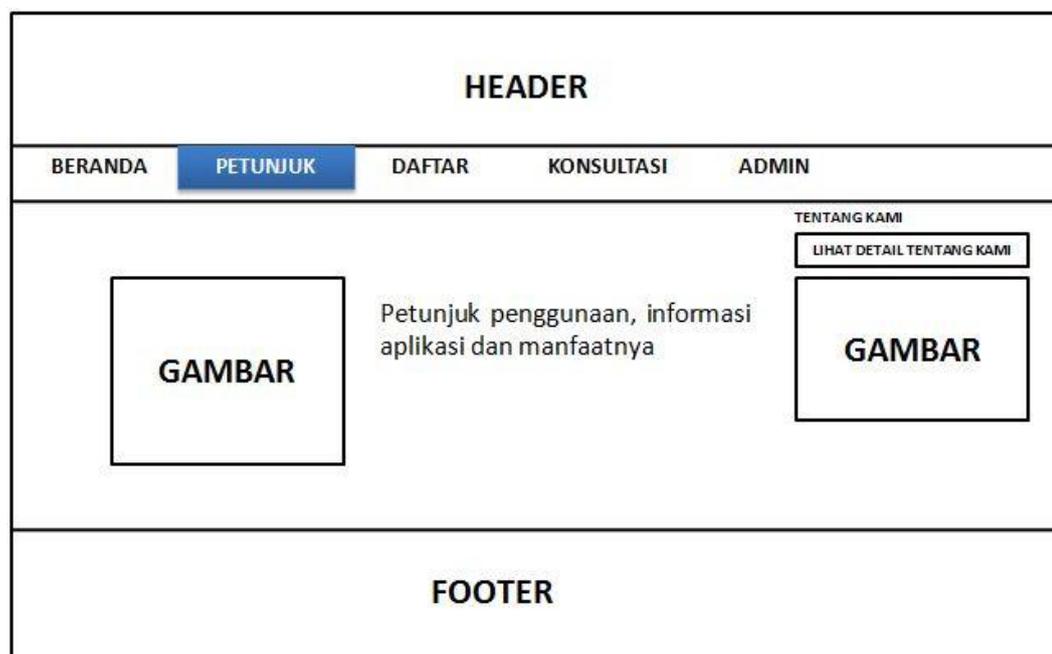
Gambar 3. 38 Rancangan Halaman Beranda

Sumber: Data penelitian, 2018

5. Rancangan halaman petunjuk

Pada gambar 3.39 menunjukkan halaman petunjuk. Pada halaman ini akan ditampilkan kata sambutan dan dijelaskan tentang informasi sistem pakar yang

sedang digunakan berikut manfaat penelitian akan ditampilkan pada halaman ini. Pada halaman ini juga petunjuk penggunaan sistem pakar ini dan juga informasi seputar penulis jika meng-klik tombol “lihat detail tentang kami”.



Gambar 3. 39 Rancangan Halaman Petunjuk
Sumber: Data penelitian, 2018

6. Rancangan halaman pendaftaran

Pada gambar 3.40 menunjukkan halaman pendaftaran, halaman ini untuk *user* yang baru pertama kali melakukan konsultasi, maka *user* akan diminta melakukan pengisian *form* pendaftaran dengan data yang diminta oleh sistem.

HEADER		
BERANDA	PETUNJUK	DAFTAR
FORM PENDAFTARAN		TENTANG KAMI
Nama	<input type="text"/>	<input type="button" value="LIHAT DETAIL TENTANG KAMI"/>
Alamat	<input type="text"/>	
Umur	<input type="text"/>	
Email	<input type="text"/>	
Password	<input type="text"/>	
<input type="button" value="Daftar"/>		
FOOTER		

Gambar 3. 40 Rancangan Halaman Pendaftaran
Sumber: Data penelitian, 2018

7. Rancangan halaman beranda konsultasi.

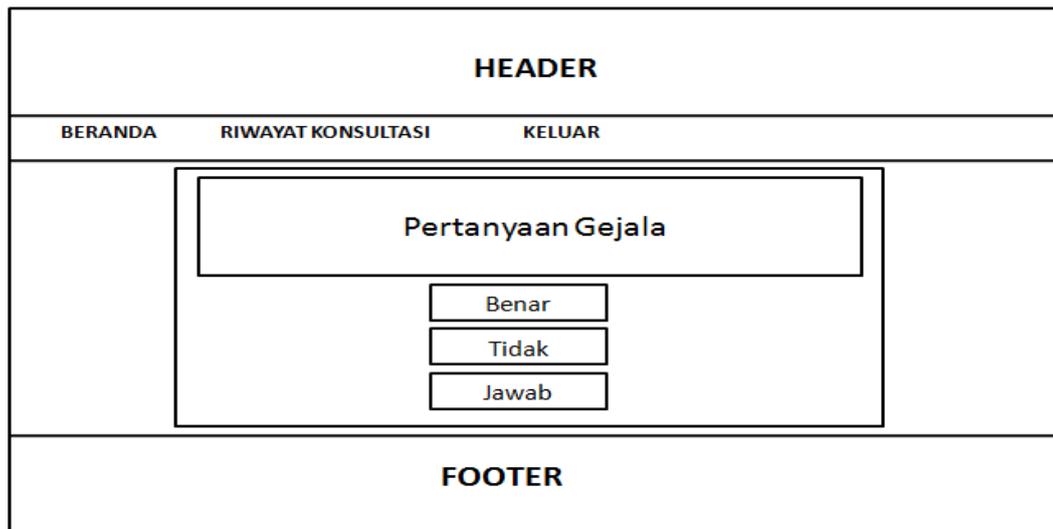
Pada gambar 3.41 menunjukkan halaman beranda konsultasi. Pada halaman ini terdapat tombol untuk melanjutkan ke halaman konsultasi yang sebenarnya. Dengan mengklik tombol lanjutkan konsultasi, maka *user* akan dibawa ke halaman konsultasi dan dapat memulai melakukan konsultasi.

HEADER		
BERANDA	RIWAYAT KONSULTASI	KELUAR
<p>Hai (User)</p> <p>Selamat Datang di Halaman Konsultasi Sistem Pakar Diagnosa Kerusakan Sepeda Motor Untuk melanjutkan konsultasi silahkan tekan tombol dibawah ini.</p> <p><input type="button" value="Lanjutkan Konsultasi"/></p>		
FOOTER		

Gambar 3. 41 Rancangan Halaman Beranda Konsultasi
Sumber: Data penelitian, 2018

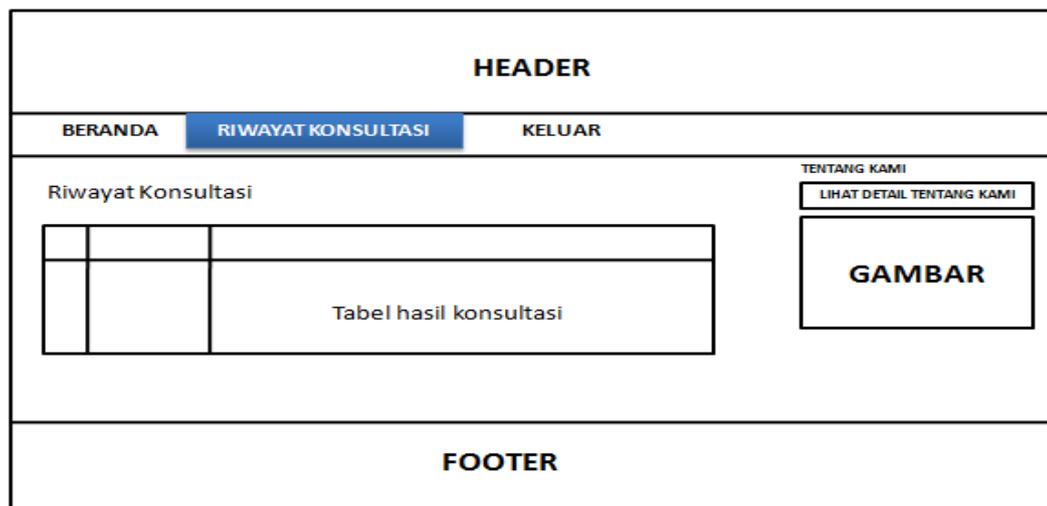
5. Rancangan halaman konsultasi

Pada gambar 3.42 menunjukkan halaman konsultasi. Pada halaman ini tersedia tombol untuk mengakses riwayat konsultasi yang telah dilakukan oleh *user* atau bisa juga langsung keluar untuk meninggalkan halaman konsultasi.



Gambar 3. 42 Rancangan Halaman Konsultasi
Sumber: Data penelitian, 2018

6. Rancangan halaman riwayat konsultasi

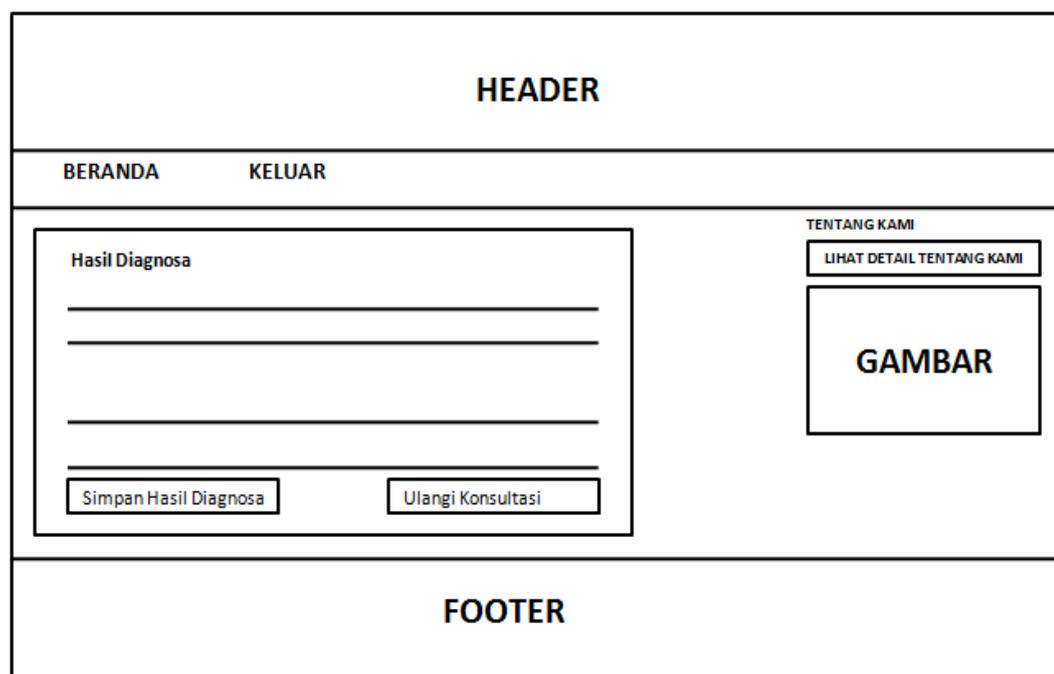


Gambar 3. 43 Rancangan Halaman Riwayat Konsultasi
Sumber: Data penelitian, 2018

Pada gambar 3.43 menunjukkan halaman riwayat konsultasi. Halaman ini menampilkan riwayat konsultasi yang telah dilakukan oleh *user*.

7. Rancangan halaman hasil diagnosa

Pada gambar 3.44 menunjukkan halaman hasil diagnosa. Halaman ini akan muncul setelah *user* menjawab semua pertanyaan. Jika *user* merasa hasilnya sudah tepat, maka *user* dapat menerimanya dengan mengklik tombol “simpan hasil diagnosa” yang kemudian akan disimpan di *database*. Jika *user* masih ragu dengan hasilnya, maka *user* dapat melakukan konsultasi ulang dengan mengklik tombol “ulangi konsultasi”.



Gambar 3. 44 Rancangan Halaman Hasil Diagnosa
Sumber: Data penelitian, 2018

8. Rancangan halaman data gejala

HEADER																																																	
BERANDA		KELUAR																																															
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Gambar</div> <ul style="list-style-type: none"> <li style="background-color: #4a86e8; color: white; padding: 2px 5px; margin-bottom: 2px;">Data Gejala <li style="padding: 2px 5px; margin-bottom: 2px;">Data Kerusakan <li style="padding: 2px 5px; margin-bottom: 2px;">Data Bagian Kerusakan <li style="padding: 2px 5px; margin-bottom: 2px;">Data Aturan <li style="padding: 2px 5px; margin-bottom: 2px;">Data Admin <li style="padding: 2px 5px; margin-bottom: 2px;">Data Riwayat Pengunjung <li style="padding: 2px 5px;">Data Riwayat Konsultasi 	<div style="display: flex; justify-content: space-between; align-items: center;"> Data Gejala <div style="border: 1px solid black; padding: 2px 5px; margin-left: 10px;">Tambah</div> <div style="border: 1px solid black; padding: 2px 5px; margin-left: 10px;">Cetak Data</div> </div>	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 5%;">No</th> <th style="width: 15%;">ID Gejala</th> <th style="width: 50%;">Nama Gejala</th> <th colspan="2" style="width: 25%;">Aksi</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td>Ubah</td><td>Hapus</td></tr> </tbody> </table>			No	ID Gejala	Nama Gejala	Aksi					Ubah	Hapus																																			
No	ID Gejala	Nama Gejala	Aksi																																														
			Ubah	Hapus																																													
			Ubah	Hapus																																													
			Ubah	Hapus																																													
			Ubah	Hapus																																													
			Ubah	Hapus																																													
			Ubah	Hapus																																													
			Ubah	Hapus																																													
			Ubah	Hapus																																													
<div style="display: flex; justify-content: center; gap: 5px;"> <div style="border: 1px solid black; padding: 2px 5px;">1</div> <div style="border: 1px solid black; padding: 2px 5px;">2</div> <div style="border: 1px solid black; padding: 2px 5px;">3</div> <div style="border: 1px solid black; padding: 2px 5px;">4</div> <div style="border: 1px solid black; padding: 2px 5px;">5</div> </div>																																																	
FOOTER																																																	

Gambar 3. 45 Rancangan Halaman Data Gejala
Sumber: Data penelitian, 2018

Pada gambar 3.45 menunjukkan halaman data gejala. Pada halaman ini *admin* dapat mengelola data kerusakan yakni menambah data, mencetak data, merubah data, dan menghapus data

9. Rancangan halaman data kerusakan

Pada gambar 3.46 menunjukkan halaman data kerusakan. Pada halaman ini *admin* dapat mengelola data kerusakan yakni menambah data, mencetak data, merubah data, dan menghapus data.

HEADER																																																																																	
BERANDA		KELUAR																																																																															
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Gambar</div> <ul style="list-style-type: none"> Data Gejala <li style="background-color: #e0e0e0;">Data Kerusakan Data Bagian Kerusakan Data Aturan Data Admin Data Riwayat Pengunjung Data Riwayat Konsultasi 	<div style="display: flex; justify-content: space-between; align-items: center;"> Data Kerusakan Tambah Cetak Data </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th>No</th> <th>Id Ker</th> <th>Kerusakan</th> <th>Id bag</th> <th>Bagian Kerusakan</th> <th>Gejala</th> <th>Penanganan</th> <th colspan="2">Aksi</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Ubah</td><td>Hapus</td></tr> </tbody> </table> <div style="text-align: center; margin-top: 5px;"> 1 2 3 4 5 </div>									No	Id Ker	Kerusakan	Id bag	Bagian Kerusakan	Gejala	Penanganan	Aksi									Ubah	Hapus								Ubah	Hapus								Ubah	Hapus								Ubah	Hapus								Ubah	Hapus								Ubah	Hapus								Ubah	Hapus
No	Id Ker	Kerusakan	Id bag	Bagian Kerusakan	Gejala	Penanganan	Aksi																																																																										
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
FOOTER																																																																																	

Gambar 3. 46 Rancangan Halaman Data Kerusakan
Sumber: Data penelitian, 2018

10. Rancangan halaman data bagian kerusakan

Pada gambar 3.47 menunjukkan halaman data bagian kerusakan. Pada halaman ini *admin* dapat mengelola data kerusakan yakni menambah data, mencetak data, merubah data, dan menghapus data.

HEADER																																												
BERANDA		KELUAR																																										
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Gambar</div> <ul style="list-style-type: none"> Data Gejala Data Kerusakan <li style="background-color: #e0e0e0;">Data Bagian Kerusakan Data Aturan Data Admin Data Riwayat Pengunjung Data Riwayat Konsultasi 	<div style="display: flex; justify-content: space-between; align-items: center;"> Data Bagian Kerusakan Tambah Cetak Data </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th>No</th> <th>Id Bagian</th> <th>Nama Bagian</th> <th colspan="2">Aksi</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td>Ubah</td><td>Hapus</td></tr> </tbody> </table> <div style="text-align: center; margin-top: 5px;"> 1 2 3 4 5 </div>				No	Id Bagian	Nama Bagian	Aksi					Ubah	Hapus																														
No	Id Bagian	Nama Bagian	Aksi																																									
			Ubah	Hapus																																								
			Ubah	Hapus																																								
			Ubah	Hapus																																								
			Ubah	Hapus																																								
			Ubah	Hapus																																								
			Ubah	Hapus																																								
			Ubah	Hapus																																								
FOOTER																																												

Gambar 3. 47 Rancangan Halaman Data Bagian Kerusakan
Sumber: Data penelitian, 2018

11. Rancangan halaman data aturan

HEADER																																																																																	
BERANDA		KELUAR																																																																															
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Gambar</div> <ul style="list-style-type: none"> Data Gejala Data Kerusakan Data Bagian Kerusakan <li style="background-color: #4a86e8; color: white; padding: 2px;">Data Aturan Data Admin Data Riwayat Pengunjung Data Riwayat Konsultasi 	<div style="display: flex; justify-content: space-between; align-items: center;"> Data Aturan <div style="display: flex; gap: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">Tambah</div> <div style="border: 1px solid black; padding: 2px 10px;">Cetak Data</div> </div> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th>No</th> <th>Id</th> <th>Gejala</th> <th>Ya</th> <th>Tidak</th> <th>Mulai</th> <th>Selesai</th> <th colspan="2">Aksi</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Ubah</td><td>Hapus</td></tr> </tbody> </table> <div style="text-align: center; margin-top: 5px;"> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">3</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">4</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">5</div> </div>									No	Id	Gejala	Ya	Tidak	Mulai	Selesai	Aksi									Ubah	Hapus								Ubah	Hapus								Ubah	Hapus								Ubah	Hapus								Ubah	Hapus								Ubah	Hapus								Ubah	Hapus
No	Id	Gejala	Ya	Tidak	Mulai	Selesai	Aksi																																																																										
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
							Ubah	Hapus																																																																									
FOOTER																																																																																	

Gambar 3. 48 Rancangan Halaman Data Aturan

Sumber: Data penelitian, 2018

Pada gambar 3.48 menunjukkan halaman data aturan. Pada halaman ini *admin* dapat mengelola data kerusakan yakni menambah data, mencetak data, merubah data, dan menghapus data.

12. Rancangan halaman data *admin*

Pada gambar 3.49 menunjukkan halaman data admin. Pada halaman ini *admin* dapat mengelola data kerusakan yakni menambah data, mencetak data, merubah data, dan menghapus data.

HEADER																																																																	
BERANDA		KELUAR																																																															
<div style="border: 1px solid black; padding: 5px; text-align: center;">Gambar</div> <ul style="list-style-type: none"> Data Gejala Data Kerusakan Data Bagian Kerusakan Data Aturan <li style="background-color: #e0e0e0;">Data Admin Data Riwayat Pengunjung Data Riwayat Konsultasi 		<div style="display: flex; justify-content: space-between;"> Data Admin Tambah Cetak Data </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Nama</th> <th>Email</th> <th>Password</th> <th>Log in terakhir</th> <th>Log out terakhir</th> <th colspan="2">Aksi</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Ubah</td><td>Hapus</td></tr> </tbody> </table> <div style="text-align: center; margin-top: 5px;"> 1 2 3 4 5 </div>								No	Nama	Email	Password	Log in terakhir	Log out terakhir	Aksi								Ubah	Hapus							Ubah	Hapus							Ubah	Hapus							Ubah	Hapus							Ubah	Hapus							Ubah	Hapus
No	Nama	Email	Password	Log in terakhir	Log out terakhir	Aksi																																																											
						Ubah	Hapus																																																										
						Ubah	Hapus																																																										
						Ubah	Hapus																																																										
						Ubah	Hapus																																																										
						Ubah	Hapus																																																										
						Ubah	Hapus																																																										
FOOTER																																																																	

Gambar 3. 49 Rancangan Halaman Data Admin
Sumber: Data penelitian, 2018

13. Rancangan halaman data riwayat pengunjung

Pada halaman ini *admin* dapat mengelola data riwayat pengunjung yakni hanya bisa melakukan percetakan dan penghapusan data. Hal ini dimaksudkan untuk menjaga keaslian dari data *user* yang tersimpan di *database*. Pada gambar 3.50 menunjukkan halaman data riwayat pengunjung.

HEADER																																																																															
BERANDA		KELUAR																																																																													
<div style="border: 1px solid black; padding: 5px; text-align: center;">Gambar</div> <ul style="list-style-type: none"> Data Gejala Data Kerusakan Data Bagian Kerusakan Data Aturan Data Admin <li style="background-color: #e0e0e0;">Data Riwayat Pengunjung Data Riwayat Konsultasi 		<div style="display: flex; justify-content: space-between;"> Data Riwayat Pengunjung Cetak Data </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Id</th> <th>Nama user</th> <th>Alamat</th> <th>Umur</th> <th>Email</th> <th>Password</th> <th>Log in terakhir</th> <th>Log out terakhir</th> <th>Aksi</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>Hapus</td></tr> </tbody> </table> <div style="text-align: center; margin-top: 5px;"> 1 2 3 4 5 </div>								No	Id	Nama user	Alamat	Umur	Email	Password	Log in terakhir	Log out terakhir	Aksi										Hapus										Hapus										Hapus										Hapus										Hapus										Hapus
No	Id	Nama user	Alamat	Umur	Email	Password	Log in terakhir	Log out terakhir	Aksi																																																																						
									Hapus																																																																						
									Hapus																																																																						
									Hapus																																																																						
									Hapus																																																																						
									Hapus																																																																						
									Hapus																																																																						
FOOTER																																																																															

Gambar 3. 50 Rancangan Halaman Data Riwayat Pengunjung
Sumber: Data penelitian, 2018

14. Rancangan halaman data riwayat konsultasi

Pada halaman ini *admin* dapat mengelola Data Riwayat Konsultasi yakni hanya bisa melakukan percetakan dan penghapusan data. Hal Ini dimaksudkan untuk menjaga keaslian dari data *user* yang tersimpan di *database*. Pada gambar 3.51 menunjukkan halaman data riwayat konsultasi.

HEADER																																																													
BERANDA		KELUAR																																																											
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Gambar</div> <ul style="list-style-type: none"> Data Gejala Data Kerusakan Data Bagian Kerusakan Data Aturan Data Admin Data Riwayat Pengunjung <li style="background-color: #4a86e8; color: white; padding: 2px;">Data Riwayat Konsultasi 	<div style="display: flex; justify-content: space-between;"> Data Riwayat Konsultasi Cetak Data </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 5%;">No</th> <th style="width: 10%;">Id Riwayat</th> <th style="width: 10%;">Tanggal</th> <th style="width: 10%;">Id Kunjung</th> <th style="width: 15%;">Nama</th> <th style="width: 30%;">Diagnosa</th> <th style="width: 10%;">Aksi</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td style="text-align: center;">Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td style="text-align: center;">Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td style="text-align: center;">Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td style="text-align: center;">Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td style="text-align: center;">Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td style="text-align: center;">Hapus</td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td style="text-align: center;">Hapus</td></tr> </tbody> </table> <div style="text-align: center; margin-top: 5px;"> 1 2 3 4 5 </div>					No	Id Riwayat	Tanggal	Id Kunjung	Nama	Diagnosa	Aksi							Hapus																																										
No	Id Riwayat	Tanggal	Id Kunjung	Nama	Diagnosa	Aksi																																																							
						Hapus																																																							
						Hapus																																																							
						Hapus																																																							
						Hapus																																																							
						Hapus																																																							
						Hapus																																																							
						Hapus																																																							
FOOTER																																																													

Gambar 3. 51 Rancangan Halaman Data Riwayat Konsultasi

Sumber: Data penelitian, 2018

3.5 Lokasi dan Jadwal Penelitian

3.5.1 Lokasi Penelitian

Penelitian ini dilakukan di bengkel Marfen Utama Motor yang beralamat di Ruko Bukit Sentosa C 9-10, Jalan Jendral S. Parman, Tanjung Piayu, Kecamatan Sei Beduk, Kota Batam, Kepulauan Riau 29433. Alasan peneliti memilih

perusahaan ini sebagai lokasi penelitian adalah kemudahan mendapatkan data serta efisiensi biaya dan waktu.

Address: Komplek Bukit Sentosa, Jalan Jendral S. Parman, Mangsang, Sei Beduk, Mangsang, Batam, Kota Batam, Kepulauan Riau 29433

Phone: 0812-6801-9718

Province: Riau Islands

[Suggest an edit](#) · [Own this business?](#)

Add missing information

[Add business hours](#)

[Add website](#)

Questions & answers

[Be the first to ask a question](#)

[Ask a question](#)



Gambar 3. 52 Tampilan Denah Lokasi Penelitian
Sumber: Data penelitian, 2018

3.5.2 Jadwal Penelitian

Jadwal penelitian perlu dibuat untuk menggambarkan kapan dan berapa lama waktu yang diperlukan untuk melakukan setiap langkah dalam penelitian. Selain itu, jadwal penelitian juga merupakan tengat (*deadline*) bagi peneliti yang

bersangkutan untuk dapat melaksanakan dan menyelesaikan penelitian. Berikut ini adalah tabel jadwal kegiatan yang dilakukan selama penelitian berlangsung:

Tabel 3. 9 Tabel Jadwal Penelitian

No	Kegiatan	Jadwal																					
		September 2018				Oktober 2018				November 2018				Desember 2018				Januari 2018				Februari 2018	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
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