

**SISTEM PAKAR UNTUK MENDIAGNOSIS
KERUSAKAN PADA MESIN SEPEDA MOTOR
SPORT DENGAN MENGGUNAKAN METODE
*FORWARD CHAINING***

SKRIPSI



**Oleh:
TOTOK ARIS SANJAYA
140210100**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS PUTERA BATAM
TAHUN 2019**

**SISTEM PAKAR UNTUK MENDIAGNOSIS
KERUSAKAN PADA MESIN SEPEDA MOTOR
SPORT DENGAN MENGGUNAKAN METODE
*FORWARD CHAINING***

SKRIPSI

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



**Oleh:
TOTOK ARIS SANJAYA
140210100**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS PUTERA BATAM
TAHUN 2019**

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 ketidak benaran 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, 05 Agustus 2019

Yang membuat pernyataan,

Totok Aris Sanjaya

NPM. 140210100

**SISTEM PAKAR UNTUK MENDIAGNOSIS KERUSAKAN
PADA MESIN SEPEDA MOTOR *SPORT* DENGAN
MENGUNAKAN METODE
*FORWARD CHAINING***

Oleh
TOTOK ARIS SANJAYA
140210100

SKRIPSI

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

**Telah disetujui oleh pembimbing pada tanggal
seperti tertera dibawah ini**

Batam, 05 Agustus 2019

Pastima Simanjuntak, S.Kom., M.SI.
Pembimbing

ABSTRAKSI

Sepeda motor adalah suatu alat transportasi yang sudah menjadi kebutuhan yang tidak dapat dielakan dari kehidupan manusia. Namun tentunya banyak kendala dalam perawatan motor. Kurangnya pengetahuan akan kerusakan mesin mengakibatkan banyaknya pengguna kendaraan sepeda motor memilih bengkel menjadi tujuan untuk mengatasi kerusakan pada mesin sepeda motor. Tetapi kenyataanya tidak semua mekanik bengkel yang bersikap jujur. Terkadang banyak juga mekanik bengkel yang melakukan tindakan kurang terpuji seperti penipuan demi mendapatkan keuntungan yang lebih banyak. Adanya sebuah sistem pakar untuk mendiagnosis kerusakan pada mesin sepeda motor memang perlu dikembangkan, sehingga mempermudah pemilik motor mengetahui kerusakan yang terjadi pada mesin motornya. Sistem pakar untuk mendiagnosis kerusakan pada mesin sepeda motor *sport* ini merupakan suatu sistem dengan metode *forward chaining* yang berguna untuk mengidentifikasi kerusakan berdasarkan gejala yang dirasakan saat mengendarai sepeda motor. Sehingga pemilik dapat mengetahui lebih dini kerusakan pada mesin sepeda motornya dan dapat melakukan tindakan awal sebelum ditindak lanjuti oleh mekanik ataupun dapat menangani kerusakan – kerusakan ringan pada sepeda motornya. Selain itu pemilik sepeda motor juga terhindar dari penipuan oleh pihak yang tidak bertanggung jawab. Dari hasil pengujian yang dilakukan menunjukkan bahwa sistem pakar yang dirancang ini dapat berfungsi dengan baik dan layak untuk digunakan. Sistem pakar ini akan sangat membantu mengetahui kerusakan pada mesin sepeda motor *sport*.

Kata kunci: Sistem pakar, diagnosis kerusakan, kerusakan mesin motor, sepeda motor *sport*, *forward chaining*.

ABSTRACT

Motorcycle is a means of transportation that has become an inevitable necessity of human life. But of course there are many obstacles in motor maintenance. Lack of knowledge about engine damage has resulted in many motorcycle vehicle users choosing workshops to be the goal to overcome damage to motorcycle engines. But the fact is that not all workshop mechanics are honest. Sometimes there are also many workshop mechanics who do less commendable actions such as fraud in order to get more profits. The existence of an expert system to diagnose damage to motorcycle engines does need to be developed, making it easier for motorcycle owners to know the damage that occurs in their motorcycle engines. This expert system for diagnosing damage to motorbike sports engines is a system with a forward chaining method that is useful for identifying damage based on symptoms felt while riding a motorcycle. So that the owner can find out earlier the damage to the engine of his motorcycle and can take the initial action before being followed up by a mechanic or can deal with minor damage to the motorcycle. Besides motorcycle owners also avoid fraud by irresponsible parties. From the results of tests conducted show that the expert system that was designed can function properly and is suitable for use. This expert system will greatly help find out the damage to the engine of a sports motorcycle.

Keywords: *Expert system, damage diagnosis, motor engine damage, sport motorbike, forward chaining.*

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, yang telah melimpahkan segala rahmat dan karunia-Nya, 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 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 Ibu Dr. Nur Elfi Husda, S.Kom., M.SI.
2. Dekan Fakultas Teknik dan Komputer Bapak Amrizal, S.Kom., M.SI.
3. Ketua Program Studi Teknik Informatika Bapak Andi Maslan, S.T., M.SI.
4. Ibu Pastima Simanjuntak, S.Kom., M.SI. selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
5. Dosen dan Staff Universitas Putera Batam.
6. Kepada kedua orang tua serta seluruh keluarga yang telah memberikan dukungan baik secara materi maupun moril. Yang selalu memberikan motivasi untuk menyelesaikan kuliah dengan baik serta mendoakan keberhasilan penulis menyelesaikan skripsi ini.

7. Rekan-rekan mahasiswa mahasiswi Universitas Putera Batam terutama khususnya teman-teman fakultas teknik dan komputer Tiban yang turut memberikan doa dan dukungannya.
8. Serta pihak-pihak lain yang tidak dapat disebutkan satu persatu.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan selalu mencurahkan hidayah serta taufik-Nya, Amin.

Batam, 05 Agustus 2019

Penulis

DAFTAR ISI

HALAMAN SAMPUL DEPAN	i
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN	iii
HALAMAN PENGESAHAN	iv
ABSTRAKSI	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Penelitian	1
1.2 Identifikasi Masalah	3
1.3 Pembatasan Masalah	4
1.4 Perumusan Masalah	4
1.5 Tujuan Penelitian	5
1.6 Manfaat Penelitian	5
BAB 2 KAJIAN PUSTAKA	7
2.1 Teori Dasar	7
2.1.1 <i>Artificial Intelligence</i> (Kecerdasan buatan)	7
2.1.2 Jaringan Saraf Tiruan (JST)	8
2.1.3 <i>Fuzzy Logic</i> (Logika Fuzzy)	11
2.1.4 Sistem Pakar (<i>Expert System</i>)	13
2.1.4.1 Sistem Pakar (<i>Expert System</i>)	14
2.2 Variabel Penelitian	21
2.2.1 Penggolongan Mesin Sepeda Motor	21
2.3 <i>Software</i> Pendukung	28
2.3.1 UML (<i>Unified Modeling Language</i>).....	28
2.3.2 HTML	33
2.3.3 PHP	34

2.3.4	MySQL.....	38
2.3.5	XAMPP.....	40
2.3.6	<i>phpMyAdmin</i>	41
2.3.7	<i>Sublime Text</i>	41
2.3.8	<i>StarUML</i>	42
2.4	Penelitian Terdahulu	42
2.5	Kerangka Pemikiran.....	46
BAB 3 METODE PENELITIAN.....		47
3.1	Desain Penelitian.....	47
3.2	Teknik Pengumpulan Data.....	49
3.3	Operasional Variabel	50
3.4	Perancangan Sistem	51
3.4.1	Basis Pengetahuan.....	51
3.4.2	UML (<i>Unified Modeling Language</i>).....	58
3.4.3	Desain <i>Database</i>	83
3.4.4	Desain Antarmuka.....	88
3.5	Lokasi dan Jadwal Penelitian	103
BAB 4 HASIL DAN PEMBAHASAN.....		104
4.1	Hasil Penelitian	104
4.2	Pembahasan.....	118
4.2.1	Pengujian Validasi	118
4.2.2	Pengujian Akurasi	122
BAB 5 PENUTUP.....		123
5.1	Kesimpulan	123
5.2	Saran.....	123
DAFTAR PUSTAKA.....		125
DAFTAR RIWAYAT HIDUP		126
SURAT KETERANGAN PENELITIAN		127
LAMPIRAN.....		129

DAFTAR TABEL

Tabel 2. 1 Simbol <i>Use Case Diagram</i>	29
Tabel 2. 2 Simbol <i>Activity Diagram</i>	30
Tabel 2. 3 Simbol <i>Sequence Diagram</i>	31
Tabel 2. 4 Simbol <i>Class Diagram</i>	32
Tabel 3. 1 Variabel dan Indikator	50
Tabel 3. 2 Kode Indikator Jenis Kerusakan	52
Tabel 3. 3 Kode Gejala.....	52
Tabel 3. 4 Kode Solusi	53
Tabel 3. 5 Tabel Data Aturan	55
Tabel 3. 6 Tabel Keputusan	55
Tabel 3. 7 Solusi Perbaikan.....	56
Tabel 3. 8 Tabel Keputusan	56
Tabel 3. 9 Tabel Pakar	84
Tabel 3. 10 Tabel Gejala	84
Tabel 3. 11 Tabel Kerusakan	84
Tabel 3. 12 Tabel Solusi.....	85
Tabel 3. 13 Tabel Relasi.....	85
Tabel 3. 14 Tabel Solusi.....	85
Tabel 3. 15 Tabel Analisa Hasil	86
Tabel 3. 16 Tabel Tmp <i>User</i>	86
Tabel 3. 17 Tabel Tmp Analisa.....	87
Tabel 3. 18 Tabel Temp Kerusakan	87
Tabel 3. 19 Tabel Tmp Gejala.....	87
Tabel 3. 20 Tabel Basis Pengetahuan	88
Tabel 3. 21 Tabel Jadwal Penelitian	103
Tabel 4. 1 Tabel Pengujian Validasi Halaman <i>User</i>	119
Tabel 4. 2 Tabel Pengujian Validasi Halaman <i>Admin</i>	120
Tabel 4. 3 Hasil Diagnosa Pakar dan Sistem	122

DAFTAR GAMBAR

Gambar 2. 1 Mesin Sepeda motor.....	22
Gambar 2. 2 Karburator	23
Gambar 2. 3 Injektor	23
Gambar 2. 4 Kepala Silinder	24
Gambar 2. 5 Blok Silinder	25
Gambar 2. 6 Bak Engkol (Bak Mesin).....	26
Gambar 2. 7 Piston.....	26
Gambar 2. 8 Poros Engkol	27
Gambar 2. 9 Logo HTML	33
Gambar 2. 10 Logo PHP	34
Gambar 2. 11 Logo MySQL	38
Gambar 2. 12 Logo Xampp.....	40
Gambar 2. 13 Logo <i>phpMyAdmin</i>	41
Gambar 2. 14 Logo <i>Sublime Text</i>	41
Gambar 2. 15 Logo <i>StarUML</i>	42
Gambar 2. 16 Kerangka Pemikiran.....	46
Gambar 3. 1 Desain Penelitian.....	47
Gambar 3. 2 Pohon Keputusan.....	57
Gambar 3. 3 <i>Use Case Diagram</i>	58
Gambar 3. 4 <i>Activity Diagram Login</i>	60
Gambar 3. 5 <i>Activity Diagram Ubah Password</i>	61
Gambar 3. 6 <i>Activity Diagram Laporan Kerusakan</i>	62
Gambar 3. 7 <i>Activity Diagram Relasi Gejala</i>	62
Gambar 3. 8 <i>Activity Diagram Relasi Solusi</i>	63
Gambar 3. 9 <i>Activity Diagram Data Kerusakan</i>	64
Gambar 3. 10 <i>Activity Diagram Data Gejala</i>	65
Gambar 3. 11 <i>Activity Diagram Data Solusi</i>	66
Gambar 3. 12 <i>Activity Diagram Logout</i>	67
Gambar 3. 13 <i>Activity Diagram Menu Halaman Utama</i>	68
Gambar 3. 14 <i>Activity Diagram Diagnosa Kerusakan</i>	69
Gambar 3. 15 <i>Activity Diagram Daftar Kerusakan</i>	70
Gambar 3. 16 <i>Activity Diagram Bantuan</i>	70
Gambar 3. 17 <i>Sequence Diagram Login</i>	71
Gambar 3. 18 <i>Sequence Diagram Ubah Password</i>	72
Gambar 3. 19 <i>Sequence Diagram Laporan Kerusakan</i>	73
Gambar 3. 20 <i>Sequence Diagram Relasi Gejala</i>	74
Gambar 3. 21 <i>Sequence Diagram Relasi Solusi</i>	75
Gambar 3. 22 <i>Sequence Diagram Data Kerusakan</i>	76
Gambar 3. 23 <i>Sequence Diagram Data Gejala</i>	77
Gambar 3. 24 <i>Sequence Diagram Data Solusi</i>	78
Gambar 3. 25 <i>Sequence Diagram Logout</i>	79

Gambar 3. 26 <i>Sequence Diagram</i> Halaman Utama	79
Gambar 3. 27 <i>Sequence Diagram</i> Diagnosa Kerusakan	80
Gambar 3. 28 <i>Sequence Diagram</i> Daftar Kerusakan	81
Gambar 3. 29 <i>Sequence Diagram</i> Bantuan	82
Gambar 3. 30 <i>Class diagram</i>	83
Gambar 3. 31 Tampilan Halaman Awal	89
Gambar 3. 32 Tampilan Halaman Utama	89
Gambar 3. 33 Tampilan <i>Form</i> Pendaftaran.....	90
Gambar 3. 34 Tampilan Halaman Diagnosa Kerusakan	90
Gambar 3. 35 Tampilan Hasil Diagnosa	91
Gambar 3. 36 Tampilan Halaman Daftar Kerusakan.....	92
Gambar 3. 37 Tampilan Gejala dari Kerusakan	92
Gambar 3. 38 Tampilan Solusi dari Kerusakan	93
Gambar 3. 39 Tampilan Halaman Bantuan.....	93
Gambar 3. 40 Tampilan <i>Form Login Admin</i>	94
Gambar 3. 41 Tampilan Halaman Beranda <i>Admin</i>	95
Gambar 3. 42 Tampilan Halaman Laporan Kerusakan.....	95
Gambar 3. 43 Tampilan Gejala dari Kerusakan	96
Gambar 3. 44 Tampilan Solusi dari Kerusakan	96
Gambar 3. 45 Tampilan Halaman Relasi Gejala.....	97
Gambar 3. 46 Tampilan Halaman Relasi Solusi	97
Gambar 3. 47 Tampilan Halaman Data Kerusakan	98
Gambar 3. 48 Tampilan Halaman Ubah Data Kerusakan.....	98
Gambar 3. 49 Tampilan Halaman Tambah Data Kerusakan	99
Gambar 3. 50 Tampilan Halaman Data Gejala	99
Gambar 3. 51 Tampilan Halaman Ubah Data Gejala	100
Gambar 3. 52 Tampilan Halaman Tambah Data Gejala	100
Gambar 3. 53 Tampilan Halaman Data Solusi.....	101
Gambar 3. 54 Tampilan Halaman Ubah Data Solusi.....	101
Gambar 3. 55 Tampilan Halaman Tambah Data Solusi	102
Gambar 3. 56 Tampilan Halaman <i>Form Ubah Password</i>	102
Gambar 4. 1 Halaman <i>User</i>	104
Gambar 4. 2 Halaman Utama.....	105
Gambar 4. 3 <i>Form</i> Pendaftaran	106
Gambar 4. 4 Pertanyaan Diagnosa	106
Gambar 4. 5 Hasil Diagnosa	107
Gambar 4. 6 Cetak Hasil Diagnosa	107
Gambar 4. 7 Daftar Kerusakan Motor.....	108
Gambar 4. 8 Gejala dari Kerusakan	108
Gambar 4. 9 Solusi Perbaikan dari Kerusakan	109
Gambar 4. 10 Bantuan.....	109
Gambar 4. 11 Form Login.....	110
Gambar 4. 12 Beranda <i>Admin</i>	110
Gambar 4. 13 Laporan Kerusakan	111
Gambar 4. 14 Daftar Gejala dari Kerusakan	111
Gambar 4. 15 Daftar Solusi dari Kerusakan	112

Gambar 4. 16 Relasi Gejala	112
Gambar 4. 17 Relasi Solusi	113
Gambar 4. 18 Menu <i>Edit</i> Data Kerusakan	114
Gambar 4. 19 Ubah Data Kerusakan	114
Gambar 4. 20 Tambah Data Kerusakan	114
Gambar 4. 21 Menu <i>Edit</i> Data Gejala.....	115
Gambar 4. 22 Ubah Data Gejala	115
Gambar 4. 23 Tambah Data Gejala.....	116
Gambar 4. 24 Menu <i>Edit</i> Data Solusi Perbaikan	116
Gambar 4. 25 Ubah Data Solusi.....	117
Gambar 4. 26 Tambah Data Solusi	117
Gambar 4. 27 <i>Form</i> Ubah <i>Password</i>	118

BAB 1

PENDAHULUAN

1.1 Latar Belakang Penelitian

Di zaman yang semakin maju dan terus berkembang seperti yang kita alami saat ini, sepeda motor adalah suatu alat transportasi yang sudah menjadi kebutuhan yang tidak dapat dielakan dari kehidupan manusia. Namun tentunya banyak kendala dalam perawatan sepeda motor. Oleh karena itu untuk mengatasinya kita harus mengetahui jenis kerusakan yang terjadi serta bagaimana cara mengatasi atau memperbaikinya.

Gejala-gejala kerusakan yang muncul dapat dikenali dengan mudah jika kita terlatih, karena pada umumnya gejala-gejala tersebut dapat dirasakan oleh panca indera. Kendalanya banyak pengguna sepeda motor yang tidak mengetahui tentang kerusakan yang sering terjadi pada mesin sepeda motor miliknya. Masalah yang kompleks dan berulang-ulang masih belum bisa menanamkan pemahaman tentang kerusakan yang biasa terjadi pada mesin motor. Keberadaan seorang ahli sangat dibutuhkan oleh banyak orang tetapi akan jauh lebih baik jika para pengguna bisa sedikit mengerti tentang mesin sepeda motor. Jadi seandainya ditengah perjalanan sepeda motor mengalami kerusakan, penggunanya bisa mengetahui dimana letak kerusakannya. Sehingga saat mengalami gangguan kerusakan ringan masih bisa diperbaiki sendiri. Tapi jika kerusakannya sudah parah, setidaknya pengguna sepeda motor tersebut sedikit mengetahui bagian

mana yang rusak dan bisa terhindar dari penipuan oleh mekanik bengkel yang tidak bertanggung jawab. Untuk memudahkan orang awam dalam mengenali jenis kerusakan yang terjadi pada mesin motor tanpa adanya seorang ahli, maka diperlukan adanya sebuah sistem pakar yang dibuat berdasarkan pengetahuan seorang pakar.

Sistem pakar merupakan suatu sistem yang dibangun untuk memindahkan kemampuan dari seorang atau beberapa orang pakar ke dalam komputer yang digunakan untuk memecahkan masalah yang dihadapi oleh pemakai dalam bidang tertentu (Jamhari, Kiryanto, & Anwariningsih, 2014). Sistem pakar juga dapat digunakan untuk mendiagnosis kerusakan untuk kendaraan sepeda motor. Sistem pakar sebagai kecerdasan buatan, menggabungkan pengetahuan dan fakta – fakta serta teknik penelusuran untuk memecahkan permasalahan yang secara normal memerlukan keahlian dari seorang pakar. Metode yang dipakai pada penelitian ini adalah *forward chaining*.

Menurut (Suwondo, 2014) 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 suatu hasil (Wilson, 1998). Dalam penalaran maju, aturan diuji satu demi satu dalam urutan tertentu. Urutan itu mungkin berupa urutan pemasukan aturan ke dalam basis aturan atau juga urutan lain yang ditentukan oleh pemakai. Saat tiap aturan diuji, sistem pakar akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisinya benar, maka aturan itu disimpan

kemudian aturan berikutnya diuji. Sebaliknya bila kondisi salah, aturan itu tidak disimpan dan aturan berikutnya diuji. Proses ini akan berulang (*iterative*) sampai seluruh basis aturan teruji dengan berbagai kondisi.

Berdasarkan uraian tersebut diatas, maka masalah yang akan disajikan atau diteliti dalam laporan ini dan juga sebagai data atau bahan informasi bagi penulis dalam menyusun skripsi ini, penulis merasa tertarik dalam mengambil judul Penelitian mengenai **“Sistem Pakar Untuk Mendiagnosis Kerusakan Pada Mesin Sepeda Motor *Sport* Dengan Menggunakan Metode *Forward Chaining*”**.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas maka dapat diperoleh identifikasi masalahnya sebagai berikut:

1. Kurangnya pengetahuan oleh para pengguna sepeda motor tentang kerusakan yang sering terjadi pada mesin sepeda motor.
2. Masalah yang berulang-ulang masih belum bisa menanamkan pemahaman tentang kerusakan yang biasa terjadi pada mesin motor.
3. Sering terjadinya penipuan oleh mekanik bengkel yang tidak bertanggung jawab.

1.3 Pembatasan Masalah

Penulis mencoba membatasi permasalahan dalam pelaksanaan dan pembuatan program dengan:

1. Mendiagnosis kerusakan hanya meliputi mesin (*engine*) motor *sport* 4 tak.
2. Penelitian hanya di Dealer Resmi Honda Senturi Ultra Dinamis yang bertempat di Ruko Tiban Impian Blok A2 No. 12A. Tiban Baru, Sekupang, Kota Batam. dan objek penelitian adalah motor merek Honda khususnya jenis motor *sport*.
3. Metode penalaran yang dipakai adalah metode penalaran maju (*forward chaining*).
4. Sistem pakar yang dibuat berbasis web dan menggunakan bahasa pemrograman HTML/PHP dan basis data MYSQL

1.4 Perumusan Masalah

Berdasarkan dari latar belakang diatas maka didapatkan perumusan masalahnya sebagai berikut:

1. Bagaimana pengguna kendaraan sepeda motor dapat memperoleh informasi tentang jenis/penyebab kerusakan dan solusi perbaikan dengan mudah?
2. Bagaimanakah merancang sebuah sistem pakar yang dapat menganalisis kerusakan motor secara cepat, efektif dan efisien?

1.5 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk membantu para pengguna sepeda motor khususnya yang masih awam dalam mengenali jenis atau letak kerusakan pada mesin serta menghemat waktu dan biaya dalam menyelesaikan masalah yang berulang dengan menggunakan sebuah sistem.
2. Untuk menciptakan sebuah sistem yang dirancang berdasarkan informasi dan pengetahuan yang dimiliki oleh seorang pakar guna memudahkan penggunaan sepeda motor memahami kerusakan mesin yang biasa terjadi.

1.6 Manfaat Penelitian

Penelitian yang telah dilakukan mempunyai beberapa manfaat sebagai berikut:

Manfaat bagi objek:

1. Dapat memberikan pemahaman lebih jauh tentang gejala-gejala kerusakan pada mesin sepeda motor.
2. Membantu mengenalkan para pengguna sepeda motor dengan bagian-bagian dari mesin sepeda motor itu sendiri.
3. Bagian-bagian mesin sepeda motor dan jenis-jenis kerusakan tidak lagi asing dimata para penggunanya.

Manfaat bagi peneliti:

1. Menambah pengalaman dan pengetahuan peneliti.
2. Membantu dan memudahkan peneliti dalam membuat sebuah sistem pakar yang bisa digunakan banyak orang.

BAB 2 **KAJIAN PUSTAKA**

2.1 Teori Dasar

Berdasarkan pembahasan pada jurnal (Aji & Suhartanto, 2015) dijelaskan bahwa menurut Sugiyono, deskripsi teori dalam suatu penelitian merupakan uraian sistematis tentang teori dan hasil-hasil penelitian yang relevan dengan variable yang diteliti. Deskripsi teori paling tidak berisi tentang penjelasan terhadap variable-variabel yang diteliti, melalui pendefisian dan uraian yang lengkap dan mendalam dari berbagai referensi, sehingga ruang lingkup, kedudukan dan prediksi terhadap hubungan antar variable yang akan diteliti menjadi lebih jelas dan terarah.

Ada beberapa teori dasar yang akan dijelaskan pada bab ini, diantaranya yaitu kecerdasan buatan atau *Artificial Intelligence (AI)* dan beberapa subdisiplin ilmunya seperti logika *fuzzy (fuzzy logic)*, jaringan saraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*), *web*, dan basis data.

2.1.1 *Artificial Intelligence* (Kecerdasan buatan)

Menurut (Sutojo, Mulyanto, & Suhartono, 2011: 1-3) kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat AI, yaitu

intelligence adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan.

Menurut Winston dan Prendergast (1984), tujuan dari kecerdasan buatan adalah:

1. Membuat mesin menjadi lebih pintar (tujuan utama)
2. Memahami apa itu kecerdasan (tujuan ilmiah)
3. Membuat mesin lebih bermanfaat (tujuan entrepreneurial)

Cerdas berarti memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Untuk membuat sebuah mesin menjadi cerdas (dapat bertindak seperti manusia) maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar.

2.1.2 Jaringan Saraf Tiruan (JST)

Menurut (Sutojo et al., 2011: 283-285) jaringan saraf tiruan adalah paradigma informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dasar paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu.

Kelebihan-kelebihan yang diberikan jaringan saraf tiruan antara lain:

1. Belajar *adaptive*, yaitu kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.

2. *Self-Organization*, yaitu kemampuan membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real Time Operation*, yaitu perhitungan jaringan saraf tiruan yang dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Selain mempunyai kelebihan-kelebihan tersebut, jaringan saraf tiruan juga mempunyai kelemahan-kelemahan berikut:

1. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatika, operasi logika, dan simbolis.
3. Untuk beroperasi butuh pelatihan sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.

Seperti yang telah disebutkan sebelumnya, baik tidaknya suatu model JST salah satunya ditentukan oleh hubungan antar-*neuron* atau yang biasa disebut sebagai arsitektur jaringan. *Neuron-neuron* tersebut terkumpul dalam lapisan-lapisan yang disebut *neuron layers*. Terdapat 3 bagian lapisan penyusun jaringan saraf tiruan(Sutojo et al., 2011: 292-295), yaitu:

1. Lapisan *Input (Input Layer)*

Unit-unit dalam lapisan ini disebut unit-unit *input* yang bertugas menerima pola *input*-an dari luar yang menggambarkan suatu permasalahan.

2. Lapisan Tersembunyi (*Hidden Layer*)

Unit-unit dalam lapisan ini disebut unit-unit tersembunyi, yang mana nilai *output*-nya tidak dapat diamati secara langsung.

3. Lapisan *Output* (*Output Layer*)

Unit-unit dalam lapisan ini disebut unit-unit *output*, yang merupakan solusi jaringan saraf tiruan terhadap suatu permasalahan.

Beberapa arsitektur jaringan yang sering digunakan dalam jaringan saraf tiruan antara lain:

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*. Jaringan ini menerima *input* kemudian mengolahnya menjadi *output* tanpa melewati lapisan tersembunyi. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *ADALINE*, *Hopfield*, dan *Perceptron*.

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. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *MADALINE*, *backpropagation*, dan *Neocognitron*.

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. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *Learning Vector Quantization (LVQ)*.

2.1.3 *Fuzzy Logic* (Logika Fuzzy)

Menurut (Sutojo et al., 2011: 211) logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang sesuai untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan komputer, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah mempunyai dua kemungkinan, “Ya” atau “Tidak”, “benar” atau “salah”, “baik” atau “buruk”, dan lain-lain.

Beberapa kelebihan logika *fuzzy* adalah (Sutojo et al., 2011: 212):

1. Perancangannya tidak memerlukan persamaan matematik yang rumit
2. Mudah dimengerti
3. Memiliki toleransi terhadap data-data yang tidak tepat
4. Mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan
6. Dapat bekerja sama dengan teknik-teknik kendali secara konvensional
7. Logika *fuzzy* didasarkan pada bahasa alami

Struktur elemen dasar sistem inferensi *fuzzy* sebagai berikut (Sutojo et al., 2011:231-237):

1. Basis pengetahuan *fuzzy*: kumpulan *rule-rule fuzzy* dalam bentuk pernyataan *IF...THEN*.
2. *Fuzzyfikasi*: proses untuk mengubah *input* sistem yang mempunyai nilai tegas menjadi variabel linguistik menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.
3. Mesin inferensi, yaitu proses untuk mengubah *input fuzzy* menjadi *output fuzzy* dengan cara mengikuti aturan-aturan (*IF-THEN Rules*) yang telah ditetapkan pada basis pengetahuan *fuzzy*.
4. *Defuzzyfikasi*, yaitu mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan *fuzzyfikasi*.

Beberapa metode yang digunakan dalam sistem inferensi *fuzzy* adalah:

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan berikut:

- a. *Fuzzyfikasi*
- b. Pembentukan basis pengetahuan *fuzzy* (*Rule* dalam bentuk *IF...THEN*)
- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (*Minimum*)
- d. *Defuzzyfikasi* menggunakan metode Rata-rata (*Average*)

2. Metode Mamdani

Metode ini sering digunakan karena strukturnya yang sederhana. Untuk mendapatkan *output*, diperlukan 4 tahapan sebagai berikut:

- a. *Fuzzyfikasi*
 - b. Pembentukan basis pengetahuan *fuzzy (rule dalam bentuk IF...THEN)*
 - c. Aplikasi fungsi implikasi menggunakan fungsi *MIN (Minimum)* dan komposisi antar-*rule* menggunakan fungsi *MAX(Maximum)* dengan menghasilkan himpunan *fuzzy* baru
 - d. Defuzzifikasi menggunakan metode *Centroid* (Titik Tengah)
3. Metode Sugeno

Dalam metode ini, *output* sistem berupa konstanta atau persamaan linier. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Dalam inferensinya, metode sugeno menggunakan tahapan sebagai berikut:

- a. *Fuzzyfikasi*
- b. Pembentukan basis pengetahuan *fuzzy (rule dalam bentuk IF...THEN)*
- c. Mesin inferensi menggunakan fungsi implikasi *MIN (Minimum)*
- d. *Defuzzyfikasi* menggunakan metode Rata-rata (*Average*)

2.1.4 Sistem Pakar (*Expert System*)

Sistem pakar merupakan cabang dari *Artificial Intelligence* (AI) yang cukup tua karena sistem ini dikembangkan pada pertengahan 1960 (Sutojo et al., 2011: 159). Ada 2 metode yang digunakan didalam sistem pakar yaitu:

1. *Forward Chaining*
2. *Backward Chaining*

2.1.4.1 Sistem Pakar (*Expert System*)

Menurut (Sutojo et al., 2011: 160-178) istilah pakar berasal dari istilah *knowledge-based expert system*. Istilah ini muncul karena untuk memecahkan masalah. Sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant*. Berikut adalah beberapa pengertian sistem pakar.

1. Tuban (2001, p402)

“Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia di mana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran keahlian manusia”.

2. Jackson (1999, p3)

“Sistem pakar adalah program komputer, yang mempresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memberikan saran”.

3. Luger dan Stubbelefield (1993, p308)

“Sistem pakar adalah program yang berbasis pengetahuan yang menyediakan solusi ‘kualitas pakar’ kepada masalah-masalah dalam bidang (domain) yang spesifik.”

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, di antaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar
3. Meningkatkan kualitas, dengan memberikan nasihat yang konsisten dan mengurangi kesalahan
4. Mampu menangkap pengetahuan dan kepakaran seseorang
5. Dapat beroperasi dilingkungan yang berbahaya
6. Memudahkan akses pengetahuan seorang pakar
7. Andal. Sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit
8. Meningkatkan kapabilitas sistem komputer
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti
10. Bisa digunakan sebagai media pelengkap dalam pelatihan
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

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

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

Ciri-ciri dari sistem pakar adalah sebagai berikut.

1. Terbatas pada domain keahlian tertentu
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami
4. Bekerja berdasarkan kaidah/*rule* tertentu
5. Mudah dimodifikasi
6. Basis pengetahuan dan mekanisme inferensi terpisah
7. Keluarannya bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

Biasanya aplikasi sistem pakar menyentuh berapa area permasalahan berikut.

1. Interpretasi: menghasilkan deskripsi situasi berdasarkan data-data masukan
2. Prediksi: memperkirakan akibat yang mungkin terjadi dari situasi yang ada
3. Diagnosis: menyimpulkan suatu keadaan berdasarkan gejala-gejala yang diberikan (*symptoms*)
4. Desain: melakukan perancangan berdasarkan kendala-kendala yang diberikan
5. *Planning*: merencanakan tindakan-tindakan yang akan dilakukan
6. *Monitoring*: membandingkan hasil pengamatan dengan proses perencanaan
7. *Debugging*: menentukan penyelesaian dari suatu kesalahan sistem
8. Reparasi: melaksanakan rencana perbaikan.

9. *Intruccion*: melakukan instruksi untuk diagnosis, *debugging*, dan perbaikan kinerja
10. Kontrol: melakukan kontrol terhadap hasil interpretasi, diagnosis, *debugging*, *monitoring*, dan perbaikan tingkah laku sistem.

Konsep dasar sistem pakar meliputi enam hal beikut ini:

1. Kepakaran (*Expertise*)

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman.

2. Pakar (*Expert*)

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat.

3. Pemindahan Kepakaran(*Transferring Expertise*)

Tujuan dari sistem pakar adalah untuk memindahkan kepakaran dari seorang pakar kedalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar.

4. Inferensi (*Inferencing*)

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran.

5. Aturan-aturan (*Rule*)

Kebanyakan software sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based systems*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.

6. Kemampuan Menjelaskan (*Explanation Capatibility*)

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikan.

Komponen-komponen yang penting dalam sebuah sistem pakar sebagai berikut.

1. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan).

2. Basis Pengetahuan (*Knowledge-Base*)

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

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

3. Mesin Inferensi (*Inference Engine*)

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

4. Daerah Kerja (*Blackboard*)

Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi.

5. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar.

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

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil.

7. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)

Kemampuan memperbaiki pengetahuan (*knowledge refining system*) dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang.

8. Pengguna (*User*)

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

Rule sebagai teknik representasi pengetahuan. Setiap *rule* terdiri dari dua bagian, yaitu bagian *IF* disebut *evidence* (fakta-fakta) dan bagian *THEN* disebut Hipotesis atau kesimpulan.

Pada sistem pakar berbasis *rule*, domain pengetahuan direpresentasikan dalam sebuah kumpulan *rule* berbentuk *IF-THEN*, sedangkan data direpresentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian seperti ini. Mesin inferensi

membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam *database*. Jika bagian *IF* (kondisi) dari *rule* cocok dengan fakta, maka *rule* dieksekusi dan bagian *THEN* (aksi) diletakkan dalam *database* sebagai fakta baru yang ditambahkan.

a. *Forward Chaining*

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rule 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 kedalam *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. Metode pencarian yang digunakan adalah *Depth-First Search* (DFS), *Breadth-First Search* (BFS) atau *Best First Search*.

b. *Backward Chaining*

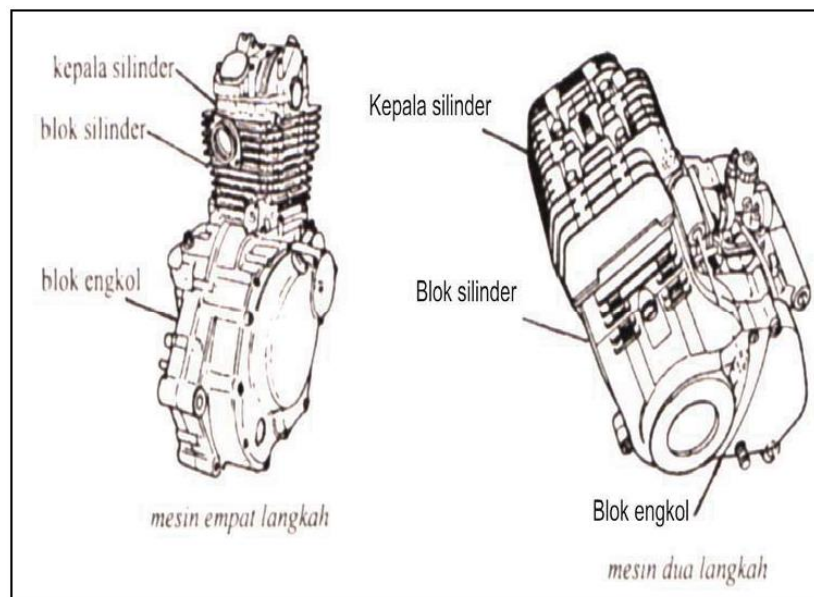
Backward Chaining adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada dibagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis dibagian *IF*. Jika cocok, *rule* dieksekusi, kemudian hipotesis dibagian *THEN* ditempatkan di basis data sebagai fakta baru. Jika tidak cocok, simpan premis dibagian *IF* ke dalam *stack* sebagai *subGoal*. Proses berakhir jika *Goal* ditemukan atau tidak ada *rule* yang bisa membuktikan kebenaran dari *subGoal* atau *Goal*.

2.2 Variabel Penelitian

Menurut (Sugiyono, 2014: 38) variabel penelitian adalah sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya. Secara teoritis variabel dapat didefinisikan sebagai atribut seseorang, atau objek, yang mempunyai “variasi” antara satu orang dengan yang lain atau satu objek dengan objek lain (Hatch dan Farhady, 1981).

2.2.1 Penggolongan Mesin Sepeda Motor

Sepeda motor dibedakan menjadi dua tipe, yaitu sepeda motor 4-tak (empat langkah) dan sepeda motor 2-tak. Perbedaan ini dilihat dari konstruksi mesinnya, sepeda motor 4 tak mempunyai katup masuk sebagai tempat masuknya bahan bakar ke dalam mesin dan katup buang untuk mengatur pembuangan gas sisa pembakaran. Pada sepeda motor 2 tak, terdapat saluran pemasukan, pembuangan, dan pembilasan bahan bakar yang diatur oleh piston dalam blok silinder. Pada sepeda motor 4-tak satu siklus kerjanya dilakukan dalam 4 langkah, yaitu langkah hisap, langkah kompresi, langkah kerja, dan langkah buang. Jadi, dalam satu kali proses kerja terjadi 4 langkah gerakan piston dalam 2 kali putaran poros engkol. Sepeda motor *sport* sendiri ada yang 2 tak dan ada juga 4 tak. Selain itu ada yang injeksi dan juga non injeksi (Supyani, Widada, & Laksito, 2013).



Gambar 2. 1 Mesin Sepeda motor
(Sumber: Data Olahan 2019)

Menurut (Nasir & Gultom, 2018) mesin sepeda motor *sport* menghasilkan daya gerak apabila terjadi proses pembakaran antara campuran udara dan bensin di dalam mesinnya. Tenaga dari hasil pembakaran itulah yang digunakan mesin untuk menggerakkan sepeda motor. Mesin berfungsi untuk mengubah tenaga dari hasil pembakaran campuran udara dan bensin di dalam suatu ruang bakar menjadi tenaga gerak atau tenaga mekanik.

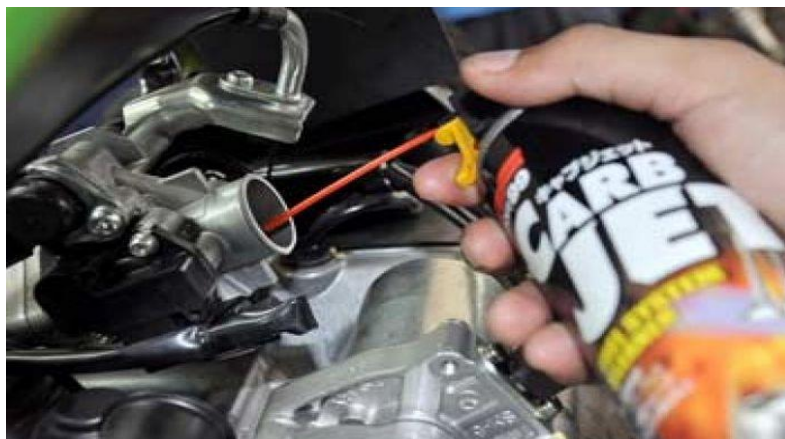
Sepeda motor memiliki beberapa bagian penting seperti kerangka/*body*, mesin, kelistrikan. Ketiga bagian itu sudah mencakup bagian-bagian kecil lainnya yang termasuk di dalamnya. Mesin sepeda motor *sport* memiliki bentuk dan bagian-bagian kecil yang jauh lebih rumit dari pada mesin sepeda motor bebek ataupun mesin sepeda motor *matic*. Penanganan bagian-bagian tertentu kerusakannya pun sering kali berbeda dengan yang lainnya khususnya pada bagian mesin.

Variabel dalam penelitian ini adalah kerusakan mesin sepeda motor *sport* yang akan dikelompokkan kedalam beberapa indikator diantaranya sebagai berikut:

1. Kerusakan Karburator/Injektor



Gambar 2. 2 Karburator
(Sumber: Data Olahan 2019)

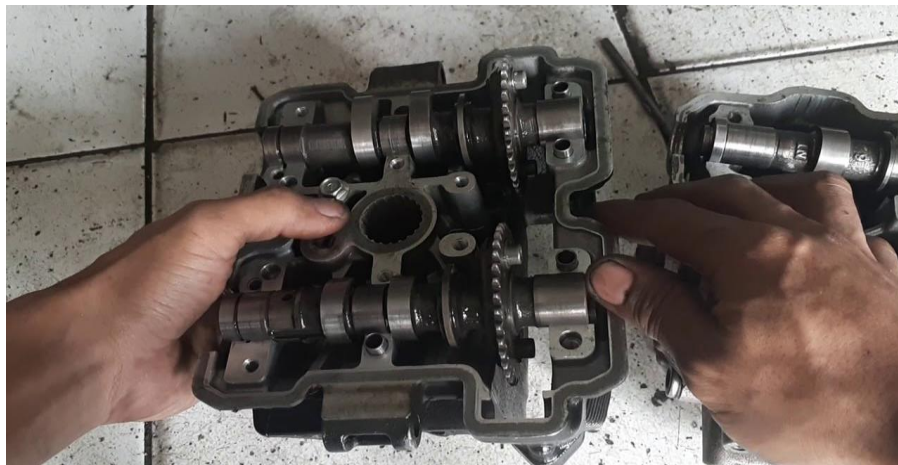


Gambar 2. 3 Injektor
(Sumber: Data Olahan 2019)

Fungsi karburator/injektor untuk mencampur bahan bakar bensin dengan udara supaya tercampur dengan halus seperti kabut. Kerusakannya meliputi:

- a. Mesin susah menyala
- b. Mesin terasa tersendat saat stasioner
- c. Tarikan sudah lemah
- d. Motor mengeluarkan asap hitam dari knalpot

2. Kerusakan Kepala Slinder (*Cylinder Head*)

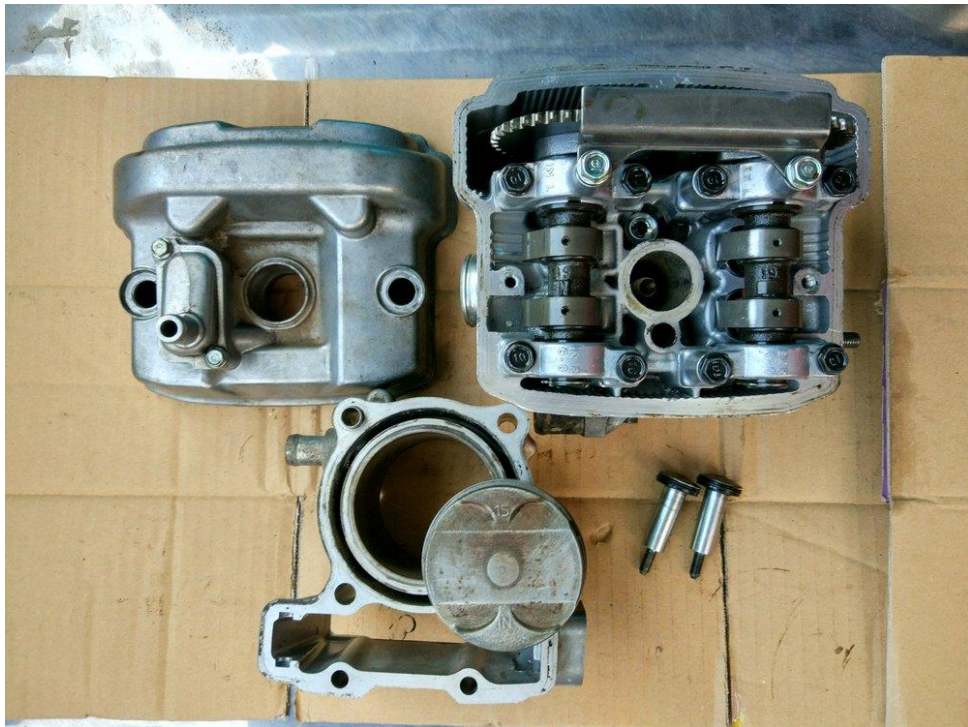


Gambar 2. 4 Kepala Silinder
(Sumber: Data Olahan 2019)

Kepala silinder (*cylinder head*) memiliki fungsi untuk menutup lubang silinder pada blok silinder dan sebagai tempat kedudukan busi. Kepala silinder ini bertumpu diatas blok silinder. Pada titik tumpunya dipasang atau disekat oleh gasket yang berguna untuk mencegah kebocoran kompresi. Kerusakannya antara lain:

- a. Mesin susah menyala
- b. Tekanan kompresi terlalu tinggi
- c. Tekanan kompresi rendah

- d. Suara mesin berisik
 - e. Mesin tidak dapat stasioner
3. Kerusakan Blok Silinder (*Cylinder Block*)



Gambar 2. 5 Blok Silinder
(Sumber: Data Olahan 2019)

Blok silinder (*cylinder block*) memiliki fungsi sebagai tempat Bergeraknya piston. Blok silinder piston terdiri dari dua komponen yang digabung menjadi satu, yaitu silinder liner dan blok silinder, keduanya saling melekat. Kerusakannya meliputi:

- a. Tekanan kompresi rendah (karena dinding silinder aus)
- b. Keluar asap putih dari knalpot

4. Kerusakan Bak Engkol (*Crankcase*)



Gambar 2. 6 Bak Engkol (Bak Mesin)
(Sumber: Data Olahan 2019)

Crankcase (bak engkol) biasanya terbuat dari aluminium *die casting* dengan sedikit campuran logam. Bak engkol biasa disebut juga bak mesin, fungsinya sebagai rumah dari komponen yang ada di bagian dalamnya. Kerusakan yang biasa terjadi adalah kebocoran oli pada sambungannya dan oli cepat habis.

5. Kerusakan Piston



Gambar 2. 7 Piston
(Sumber: Data Olahan 2019)

Piston mempunyai bentuk seperti silinder. Bekerja dan bergerak secara translasi (gerak bolak-balik) di dalam silinder. Fungsi ring piston untuk mempertahankan kerapatan antara piston dengan dinding silinder agar tidak ada kebocoran gas dari ruang bakar ke dalam bak mesin. Batang piston sering juga disebut dengan setang piston, ia berfungsi menghubungkan piston dengan poros engkol. Kerusakan yang terjadi diantaranya sebagai berikut:

- a. Keluar asap putih dari knalpot
 - b. Oli cepat berkurang atau habis
 - c. Kompresi motor menjadi bocor karena piston aus
6. Kerusakan Poros Engkol (*Crankshaft*)



Gambar 2. 8 Poros Engkol
(Sumber: Data Olahan 2019)

Fungsi poros engkol adalah mengubah gerakan piston menjadi gerakan putar (mesin) dan meneruskan gaya kopel yang dihasilkan motor ke alat pemindah tenaga sampai ke roda. Kerusakan yang biasa terjadi yaitu suara berisik dari poros engkolnya dan kompresi motor menjadi bocor.

2.3 *Software Pendukung*

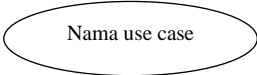


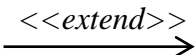
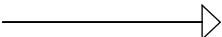
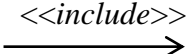
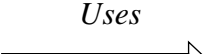
2.3.1 UML (*Unified Modeling Language*)

Unified Modeling Language (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. Dalam penelitian ini diagram yang akan digunakan untuk mendesain sistem adalah sebagai berikut (A & Shalahuddin, 2014: 137-138).

1. *Use case diagram*

Use case diagram (diagram *use case*) merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Berikut adalah simbol-simbol yang ada pada diagram *Use case* (A & Shalahuddin, 2014: 156).

Tabel 2. 1 Simbol *Use Case Diagram*

Simbol	Deskripsi
<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang
Asosiasi / <i>assosiaation</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor
Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek
Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
Menggunakan / <i>include</i> / <i>uses</i>  	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 dijalankan <i>use case</i> ini


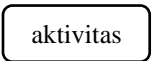
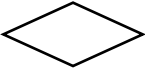


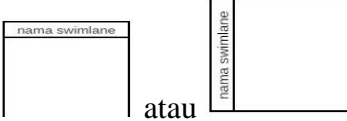
(Sumber: (A & Shalahuddin, 2014: 156-157))

2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *work-flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem buka apa yang dilakukan aktor, jadi

aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas (A & Shalahuddin, 2014: 161-163).

Tabel 2. 2 Simbol *Activity Diagram*


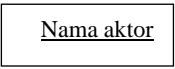

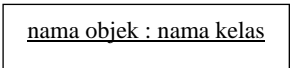

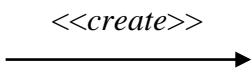
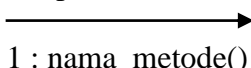
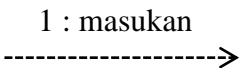
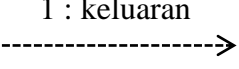
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki awal sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari 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: (A & Shalahuddin, 2014:161-163))

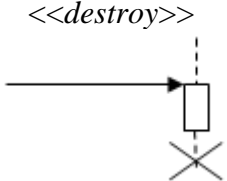
3. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim 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. Berikut adalah simbol-simbol yang ada pada diagram sekuen (A & Shalahuddin, 2014: 165-167).

Tabel 2. 3 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>atau</p>  <p>Tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata berada di awal <i>frase</i> nama aktor
<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
<p>Pesan tipe <i>create</i></p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
<p>Pesan tipe <i>call</i></p> 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri
<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
<p>Pesan tipe <i>return</i></p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian

Tabel 2. 3 Lanjutan

Simbol	Deskripsi
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

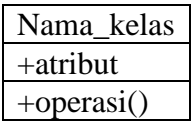


(Sumber : (A & Shalahuddin, 2014: 165-167))

4. Class Diagram


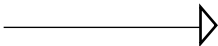
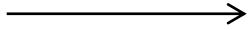

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (A & Shalahuddin, 2014: 141-147)

Berikut ini adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2. 4 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i>  Nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Tabel 2. 4 Lanjutan

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

(Sumber: (A & Shalahuddin, 2014: 141-147))

2.3.2 HTML



Gambar 2. 9 Logo HTML

(Sumber: <https://en.wikipedia.org/wiki/HTML5>)

HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman website (Saputra, 2012:1).

HTML 5 layaknya sebuah html biasa yang sering kita gunakan dalam membangun aplikasi web, hanya saja html 5 ini memiliki keunggulan dibanding versi terdahulunya (Saputra, 2012:11).

2.3.3 PHP



Gambar 2. 10 Logo PHP

(Sumber: <https://commons.wikimedia.org/wiki/File:PHP-logo.svg>)

PHP (*Hypertext Preprocessor*) adalah salah satu bahasa pemrograman yang sering digunakan dalam pemrograman web. Ada 2 macam perintah untuk menampilkan suatu kalimat atau string, yaitu menggunakan perintah **echo** dan **print** (Saputra, 2012: 91-92). *Variable* adalah tempat penyimpanan sementara didalam memori komputer . Penulisan variabel didalam pemrograman PHP, ada aturan tertentu yang harus diketahui, diantaranya:

1. Penulisan variabel harus diawali dengan simbol *dollar* (\$).
2. Karakter pertama setelah simbol *dollar*, tidak boleh menggunakan angka (harus huruf).
3. Setelah simbol *dollar* (\$) dan huruf, maka karakter selanjutnya boleh menggunakan angka.

Menurut (Saputra, 2012: 93-107) PHP memiliki 2 macam *method* yang bisa digunakan untuk mengirim data dari *client* (*browser*) ke (*server*), yaitu *POST* dan *GET*. *Method POST*, digunakan untuk mengirim data dari *client* ke *server* dimana data tersebut akan dikirim melalui *http header*. *Method GET*, yang bekerja dengan perintah fungsi `$_GET` akan mengirimkan data atau berkomunikasi melalui parameter yang akan terlihat pada URL.

Memanipulasi *file* dengan menggunakan perintah-perintah PHP, antara lain:

1. *Fopen()*, digunakan untuk membuka sebuah *file* atau URL dalam dokumen PHP.
2. *Fclose()*, digunakan menutup perintah *fopen*.
3. *FGET()*, digunakan untuk membaca baris per baris *file* yang dibuka.
4. *Feof()*, digunakan untuk memeriksa akhiran dari suatu *file*.
5. *Fgetc()*, digunakan untuk membaca karakter per karakter *file* yang dibuka.
6. *File_exists()*, digunakan untuk mengecek/memeriksa ada atau tidaknya sebuah *file*.
7. *Fputs()*, digunakan untuk menuliskan/menambah *text* pada *file* yang dibuka.
8. *File()*, digunakan untuk membaca *file* kemudian memasukkannya ke dalam bentuk array.
9. *Fstat()*, digunakan untuk mengetahui informasi dari sebuah *file*.
10. *Fread()*, digunakan untuk membaca informasi tertentu dari sebuah *file*.
11. *Copy()*, digunakan untuk meng-*copy* sebuah *file* menjadi *file* lain.
12. *Unlink()*, digunakan untuk menghapus sebuah *file*.
13. *Disk_free_space()*, digunakan untuk menampilkan informasi sisa ruang (*space*) dalam satuan *bytes* untuk *direktori* tertentu.
14. *Is_dir()*, digunakan untuk memeriksa apakah *file* tersebut merupakan suatu *direktori* atau bukan.
15. *Mkdir()*, digunakan untuk membuat *direktori*.
16. *Rename()*, digunakan untuk mengubah nama suatu *file* atau *direktori*.

Cara pemanggilan *Query SQL* dalam PHP yaitu:

1. *Mysql_query()*, digunakan untuk menjalankan *query* pada *database* MySQL.
2. *Mysql_connect()*, digunakan untuk menghubungkan *file* php dengan data MySQL.
3. *Mysql_close()*, digunakan untuk memutus hubungan dengan *database* MySQL.
4. *Mysql_select()*, digunakan untuk memilih *database* MySQL yang akan digunakan.
5. *Mysql_num_rows()*, digunakan untuk menghitung seluruh data yang terdapat pada suatu tabel dalam *database*.
6. *Mysql_affected-rows()*, digunakan untuk mendapatkan jumlah *record* yang dihasilkan dari operasi MySQL sebelumnya.
7. *Mysql_fetch_array()*, digunakan untuk mendapatkan data berdasarkan nama *field*.
8. *Mysql_fetch_row()*, digunakan untuk mendapatkan kata berdasarkan kolom *field*.
9. *Mysql_fetch_object()*, digunakan untuk mendapatkan data dan menjadikanya sebagai objek.
10. *Mysql_field_name()*, digunakan untuk mendapatkan informasi nama *field* dalam suatu tabel.
11. *Mysql_num_field()*, digunakan untuk mendapatkan informasi jumlah *field* pada suatu tabel.

Pada pemrograman PHP, dapat melakukan pengkodean dengan melakukan perintah bersyarat atau perintah *If Else* yaitu perintah yang akan dieksekusi apabila persyaratan telah terpenuhi. PHP memiliki 4 macam model/kerangka fungsi pernyataan kontrol, diantaranya sebagai berikut (Saputra, 2012: 113):

1. Fungsi *IF*, digunakan apabila kita memiliki 1 kondisi dan kondisi tersebut kan dijalankan jika pernyataan dianggap benar.
2. *If-Else*, digunakan apabila kita memiliki 2 kondisi pembanding. Jika kondisi/pernyataan bernilai *TRUE*, maka pernyataan 1 akan dijalankan, namun jika kondisi bernilai sebaliknya, maka pernyataan 2 yang akan dijalankan.
3. *If-Elseif-Else*, digunakan apabila kita memiliki banyak kondisi (minimal 3 kondisi).
4. Perintah *switch* merupakan alternatif pengganti dari *If*. Perintah *switch* akan menyelesaikan kondisi yang diberikan dan kemudian membandingkan hasilnya dengan konstanta-konstanta yang ada didalam case.

Operasi CRUD merupakan singkatan dari *Create, Read, Update, Delete*, yang merupakan suatu fungsi yang digunakan untuk mengolah suatu data secara dinamis. Untuk menggunakan operasi tersebut, kita harus terhubung dengan *database* (Saputra, 2012: 117-123).

1. *Create* (Menambah data)

Digunakan apabila kita ingin menambah suatu data kedalam tabel pada *database*.

2. *Read* (Membaca/Menampilkan Data)

Setelah data berhasil disimpan, maka langkah selanjutnya adalah menampilkan data.

3. *Update* (Mengubah Data)

Data yang telah disimpan kita simpan terkadang mengandung kesalahan *input* atau perlu dipebaharui karena itu, kita perlu proses meng*update* data.

4. *Delete* (Menghapus Data)

Ketika data sudah tidak dibutuhkan lagi, kita bisa menghapusnya dari *database*.

2.3.4 MySQL



Gambar 2. 11 Logo MySQL

(Sumber: <https://www.mysql.com/about/legal/logos.html>)

Menurut (Saputra, 2012:77-78) *MySQL* merupakan salah satu *database* kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. *MySQL* bekerja menggunakan bahasa *SQL* (*Structure Query Language*) yang merupakan bahasa standar yang digunakan untuk manipulasi *database*.

Pada umumnya, perintah yang sering digunakan dalam *MySQL* adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan

DELETE (menghapus). Selain itu, *SQL* juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data.

Ada beberapa alasan yang menjadikan *database MySQL* sangat diminati oleh para *programmer*, diantaranya:

1. Bersifat *open source*
2. Menggunakan bahasa *SQL (Structure Query Language)*, yang merupakan standar bahasa dalam pengolahan data.
3. *Performance* dan *reliable*, pemrosesan *database*-nya sangat cepat dan stabil
4. Sangat mudah dipelajari (*easy of use*)
5. Memiliki dukungan (*group*) pengguna *MySQL*
6. Lintas *Platform*, dapat digunakan pada berbagai Sistem Operasi berbeda
7. *Multiuser*, dimana *MySQL* dapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik
8. Dan masih banyak lagi.

Database server sangatlah penting karena sifatnya yang aktif sehingga akan meringankan kerja aplikasi yang kita bangun. Dengan adanya *database server*, beban kerja akan terbagi menjadi dua yaitu *database server*nya dan aplikasi. Selain itu, penggunaan *database server* juga meningkatkan keamanan aplikasi secara keseluruhan karena kita bisa memisahkan *database* dan aplikasi secara fisik kedalam *server* yang berbeda (Saputra, 2012:80).

2.3.5 XAMPP



Gambar 2. 12 Logo Xampp

(Sumber: https://commons.wikimedia.org/wiki/File:Xampp_logo.svg)

XAMPP merupakan paket PHP berbasis *open source*. Informasinya dapat diperoleh diwebsite resminya: <http://www.apachefriends.com>. XAMPP membantu memudahkan dalam mengembangkan aplikasi berbasis PHP. XAMPP mengkombinasikan beberapa paket *software* berbeda kedalam satu paket. Adapun lisensi masing-masing paket *software* tersebut dapat ditemukan didirektori `\xampp\licence`. XAMPP menyediakan antar muka control panel tersendiri yang dapat digunakan untuk menjalankan semua *service* (paket *software* pendukung) yang telah terinstal. Pada sistem operasi *windows*, *control panel* dapat diakses melalui menu [Start]? [Program] ? [Apachefriends] ? [xampp] ? [control xampp server panel]. Pada *web server* (lokal komputer, tidak di *server* internet sesungguhnya) pada XAMPP, akan menyediakan satu *folder* kerja yang bernama *htdocs*. Pada paket ini, *folder* kerja tersebut dapat ditemukan pada *subfolder* `C:\..\XAMPP` (sesuai lokasi dimana menyimpan hasil instalasinya) (Maudi, Nugraha, & Sasmito, 2014).

2.3.6 *phpMyAdmin*



Gambar 2. 13 Logo *phpMyAdmin*
(Sumber: <https://www.phpmyadmin.net/>)

PhpMyAdmin adalah perangkat lunak yang bebas ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring jagat Jembar (World Wide Web). PHPMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengolah basis data, tabel-tabel, bidang (*fields*), relasi (*relation*), *indeks*, pengguna (*users*), perjanjian (*permissions*), dan lain-lain (Susilo, Kurniati, & Kasmawi, 2018)

2.3.7 *Sublime Text*



Gambar 2. 14 Logo *Sublime Text*
(Sumber: <https://www.sublimetext.com/>)

Sublime Text adalah *editor* kode sumber lintas-*platform* bermilik dengan antarmuka pemrograman aplikasi *Python*.

2.3.8 *StarUML*



Gambar 2. 15 Logo *StarUML*
(Sumber: Aplikasi *StarUML*)

StarUML adalah *Software* pemodelan yang mendukung UML yang digunakan untuk perancangan desain UML.

2.4 Penelitian Terdahulu

Beberapa penelitian mengenai sistem pakar terkait yang didokumentasikan ke dalam sebuah jurnal dan menjadi acuan bagi penulis untuk menyusun laporan penelitian ini adalah sebagai berikut:

1. Menurut (Susilo et al., 2018) pada jurnal *e-ISSN: 2540-7600, p-ISSN: 2540-7597* vol: 2, no: 2 yang berjudul **“RANCANG BANGUN WEBSITE TOKO ONLINE MENGGUNAKAN METODE WATERFALL”**. *World wide web* atau sering di kenal sebagai web adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang

memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui internet). Keistimewaan inilah yang telah menjadikan web sebagai *service* yang paling cepat pertumbuhannya. Web mengizinkan pemberian highlight (penyorotan atau penggaris bawah) pada kata-kata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti dokumen, *frase*, *movie clip*, atau *file* suara. Web dapat menghubungkan dari sembarang tempat dalam sebuah dokumen atau gambar ke sembarang tempat di dokumen lain. Dengan sebuah *browser* yang memiliki *Graphical User Interface* (GUI), *link-link* dapat di hubungkan ke tujuannya dengan menunjuk *link* tersebut dengan *mouse* dan menekannya.

2. Menurut (Maudi et al., 2014) pada jurnal ISSN: 2337-845X vol: 3, no: 2 yang berjudul **“DESAIN APLIKASI SISTEM INFORMASI PELANGGAN PDAM BERBASIS WebGIS (STUDI KASUS : KOTA DEMAK)”**. Seiring berkembangnya zaman, Sistem Informasi dituntut agar berkembang sesuai kebutuhan. Informasi yang akurat dan handal membutuhkan sebuah sistem yang dapat mengelola data atau informasi dengan baik, dalam artian data atau informasi yang diperoleh dapat dipanggil kembali dan diperbarui menurut kebutuhan secara konsisten dan berkelanjutan.
3. Menurut (Jamhari et al., 2014) pada jurnal ISSN: 2337-4349 yang berjudul **“SISTEM PAKAR DIAGNOSIS KERUSAKAN SEPEDA MOTOR NON MATIC”**. Banyak pengendara sepeda motor yang tidak mengetahui

kendala kerusakan sepeda motor. Kebanyakan pengendara cenderung menyerahkan kendala kerusakan sepeda motor kepada mekanik tanpa mengetahui bahwa sebenarnya kerusakan tersebut merupakan kerusakan sederhana atau terlalu rumit untuk diperbaiki. Sebenarnya menyerahkan penanganan kerusakan sepeda motor kepada mekanik dapat menjadi salah satu solusi. Akan tetapi jika pengendara memiliki pengetahuan tentang perawatan kerusakan sepeda motor maka penanganan kerusakan motor dapat dikerjakan sendiri oleh pengendara sehingga penanganan kerusakan dapat segera ditangani tanpa harus menunggu sepeda motor diperbaiki di bengkel.

4. Menurut (Suwondo, 2014) pada jurnal ISSN: 2354-869X yang berjudul **“SISTEM PAKAR SEBAGAI ALAT BANTU MENGATASI MASALAH (STUDI KASUS KERUSAKAN SEPEDA MOTOR)”**. Permasalahan kerusakan sepeda motor menjadi tidak henti-hentinya dibahas dan ditemukan solusinya guna pemakaian yang tahan lama dan bernilai ekonomis bagi penggunaannya, berbagai macam teknik dan cara untuk dapat dengan cepat mengetahui kerusakan yang ada. Pakarpun dilibatkan dalam teknik penyelesaian dengan berbagai macam penelitian dan uji coba, hasilnya pakar dapat menemukan permasalahan-permasalahan baru teknik mekanik sepeda motor.
5. Menurut (Aji & Suhartanto, 2015) pada jurnal ISSN: 2302-3805 yang berjudul **“RANCANG BANGUN SISTEM PAKAR KERUSAKAN SEPEDA MOTOR HONDA SUPRA SEBAGAI MEDIA PENUNJANG**

PEMBELAJARAN STUDY KASUS: SMK NEGERI 1 GEGER–KAB.

MADIUN". Dikaitkan dengan tuntutan masa depan yang bukan hanya bersifat kompetitif tapi juga sangat terkait dengan berbagai kemajuan teknologi dan informasi maka sistem pembelajaran yang dikembangkan di Teknik Sepeda Motor SMKN 1 Geger harus mampu secara cepat memperbaiki berbagai kekurangan sarana penunjang yang ada. Tingkat kesulitan pendampingan pada kegiatan pembelajaran praktek di bengkel masih membebani pada guru dan asisten pengajar, karena terkendala kuantitas dan kualitas tenaga pengajar.

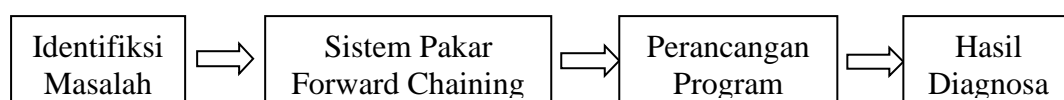
6. Menurut (Boonkanit & Charoenkid, 2016) pada jurnal ISSN: 1662-7482 vol: 848, pp: 259-262 yang berjudul "***EXPERT SYSTEM SHOFTWARE FOR PRODUCTION PLANNING AND QUALITY CONTROL IN BLEACHING DYEING AND FINISHING PROCESS OF TEXTILES INDUSTRY***". Menjelaskan bahwa saat ini banyak sistem pakar yang digunakan sebagai metode yang diterapkan dalam berbagai bidang. Sistem pakar memiliki kemampuan melakukan fungsi-fungsi yang sama dengan para ahli manusia yang memberikan saran serta pengetahuan dan berbagi pengalaman yang ditransfer ke sistem secara permanen.
7. Menurut (Minarni & Fadhillah, 2017) pada jurnal e-ISSN: 2502-0692 vol: 2 yang berjudul "***EXPERT SYSTEM IN DETECTING RICE PLANT DISEASES USING CERTAINTY FACTOR***". Menjelaskan bahwa perlu adanya sistem pakar untuk mendeteksi penyakit tanaman padi agar bisa mengatasi masalah keterlambatan mengontrol tanaman padi yang terserang

penyakit dan menyebabkan berkurangnya hasil panen ataupun gagal panen. Sistem pakar disini menggunakan metode faktor kepastian (*certainty factor*).

2.5 Kerangka Pemikiran

Uma Sekaran dalam bukunya *Business Research* (1992) mengemukakan bahwa, kerangka berpikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting (Sugiyono, 2014:60)

Dari uraian diatas, maka penulis dapat membuat suatu kerangka pemikiran sebagai berikut:

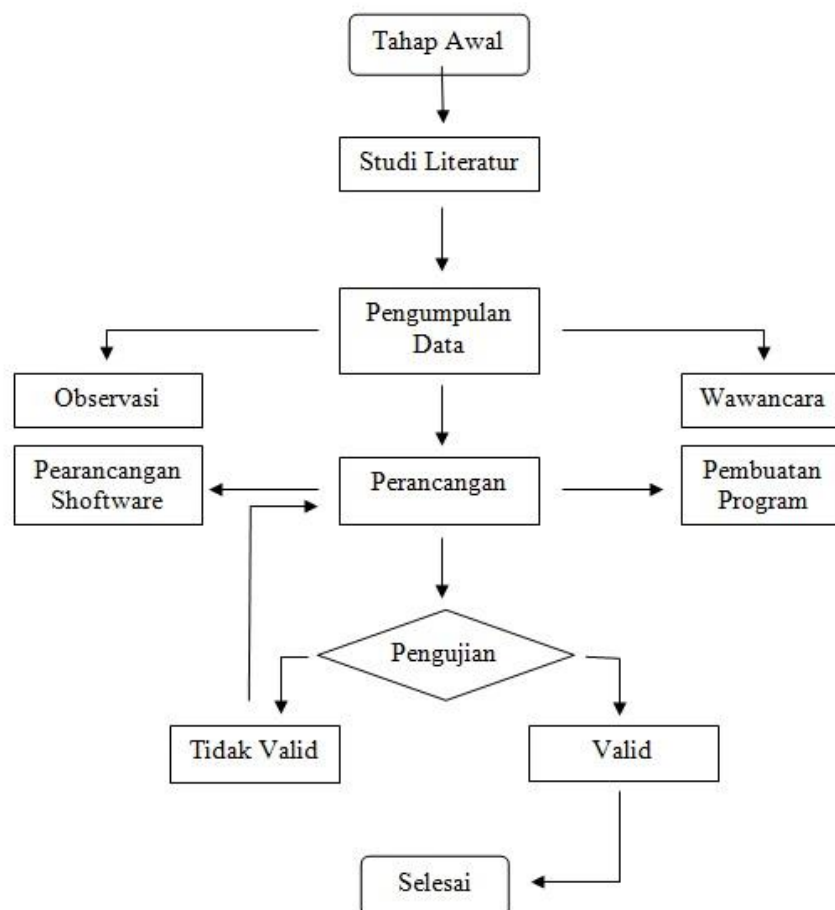


Gambar 2. 16 Kerangka Pemikiran
(Sumber: Data Penelitian 2019)

BAB 3 METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian yaitu tahapan yang akan dilakukan peneliti untuk mempermudah dalam melakukan penelitian. Langkah-langkah dalam penyusunan pada penelitian ini adalah sebagai berikut:



Gambar 3. 1 Desain Penelitian
(Sumber: Data Penelitian 2019)

Penjelasan dari Desain Penelitian di atas adalah sebagai berikut:

1. Tahap awal penelitian

Penelitian dimulai dengan menentukan kebutuhan data penelitian diantaranya mencari gejala-gejala kerusakan pada mesin sepeda motor dan menyiapkan alat dan bahan penelitian.

2. Studi literatur

Mempelajari aspek-aspek yang berkaitan dengan penelitian ini guna untuk merancang sistem pakar yang akan dibuat yaitu didapat dari referensi buku, jurnal, penelusuran internet dan lain sebagainya.

3. Pengumpulan dan analisis data

Pada tahap ini dilakukan pengumpulan data yang dibutuhkan dalam penelitian dengan cara observasi dan juga wawancara. Observasi adalah pengamatan dan pencatatan secara sistematis terhadap unsur-unsur yang tampak dalam suatu gejala atau gejala-gejala dalam objek penelitian. Pengumpulan data ini dengan melakukan pengamatan secara langsung terhadap komponen mesin sepeda motor. Sedangkan wawancara dilakukan dengan mengadakan tanya jawab dengan pakar yang berkaitan dengan kerusakan sepeda motor. Metode dilakukan dengan lisan sehingga dijawab dengan lisan oleh mekanik. Pertanyaan yang diajukan berupa pertanyaan mengenai mesin sepeda motor, gejala kerusakan sepeda motor, beserta solusinya.

4. Perancangan sistem

Perancangan dari sistem pakar yang akan dibuat meliputi kebutuhan sistem, pemodelan sistem, basis data, antarmuka sistem dan sebagainya. Setelah dirancang akan diterapkan pada bahasa pemrograman dengan menggunakan pemrograman HTML/PHP dan MySQL.

5. Pengujian

Setelah sistem pakar selesai dibuat maka dilakukan pengujian untuk mengetahui apakah sistem pakar tersebut telah bekerja dengan benar dan sesuai dengan harapan. Jika *valid* maka pembuatan sistem telah selesai, tapi jika tidak *valid* maka akan kembali ke tahap perancangan ulang.

6. Penyusunan laporan

Pada tahap ini akan dilakukan penyusunan laporan hasil akhir penelitian dan akan ditarik kesimpulan dari penelitian ini.

3.2 Teknik Pengumpulan Data

Bila dilihat dari segi cara atau teknik pengumpulan data, maka teknik pengumpulan data dapat dilakukan dengan interview (wawancara), kuesioner (angket), observasi (pengamatan), dan gabungan ketiganya.

1. Observasi dilakukan secara langsung di lokasi tempat dimana penelitian dilakukan, untuk mendapatkan informasi-informasi yang dibutuhkan untuk melanjutkan suatu penelitian.

2. Sedangkan wawancara dilakukan dengan mengadakan tanya jawab dengan pakar yang berkaitan dengan penelitian yang akan dilakukan. Hasil dari wawancara tersebut yang akan di olah peneliti secara lebih lanjut.

Selain observasi dan juga wawancara penulis juga melakukan studi literatur dalam proses pengumpulan data. Yaitu dengan cara mencari bahan yang mendukung dan erat kaitannya dengan objek permasalahannya dari sumber-sumber referensi buku, penelusuran internet, serta jurnal-jurnal penelitian.

3.3 Operasional Variabel

Variabel dari penelitian ini adalah kerusakan mesin sepeda motor *sport* dan indikatornya yaitu meliputi jenis- jenis kerusakan pada bagian yang ada. Berikut penjelasan dari Operasional Variabel Penelitian melalui tabel berikut ini.

Tabel 3. 1 Variabel dan Indikator

Variabel	Indikator
Kerusakan Mesin Sepeda Motor <i>Sport</i>	Kerusakan Karburator/Injektor
	Kerusakan Kepala Silinder
	Kerusakan Blok Silinder
	Kerusakan Bak Engkol
	Kerusakan Piston
	Kerusakan Poros Engkol

(**Sumber:** Data Penelitian 2019)

Dari tabel di dijelaskan variabelnya yaitu kerusakan mesin sepeda motor *sport* dan diperoleh beberapa indikator diantaranya yaitu kerusakan karburator/injektor, kerusakan kepala silinder, kerusakan blok silinder, kerusakan bak engkol, kerusakan piston, dan kerusakan poros engkol. Dari masing-masing indikator akan didapatkan lagi berbagai macam gejala kerusakannya. Sampai akhirnya nanti akan ditemukan suatu solusi untuk mengatasi kerusakan-kerusakan tersebut.

3.4 Perancangan Sistem

Desain atau perancangan dalam pembangunan perangkat lunak merupakan upaya untuk mengonstruksi sebuah sistem yang memberikan kepuasan (mungkin *informal*) akan spesifikasi kebutuhan fungsional, meliputi target, memenuhi kebutuhan secara *implisit* atau *eksplisit* dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat (A & Shalahuddin, 2014:23).

3.4.1 Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan (*rule*). Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang

cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui (Jamhari et al., 2014).

Fakta dan aturan disimpan dalam bentuk database. Database ini berisi rangkaian informasi tentang status masalah yang sudah dipecah-pecah. Fakta direpresentasikan dengan menetapkan kesesuaian antara representasi internal fakta dengan representasi bahasa alami. Aturan ini berisi tentang bagaimana menggunakan pengetahuan untuk memecahkan masalah khusus pada setiap domain. Aturan pada basis pengetahuan direpresentasikan sebagai perintah berpasangan atau sebagai IF kondisi THEN aksi. Bagian IF mendeskripsikan representasi situasi pasti berupa kumpulan dari pernyataan (Aji & Suhartanto, 2015).

Tabel 3. 2 Kode Indikator Jenis Kerusakan

No.	Kode	Indikator
1	K001	Kerusakan Karburator/Injektor
2	K002	Kerusakan Kepala Silinder
3	K003	Kerusakan Blok Silinder
4	K004	Kerusakan Bak Engkol
5	K005	Kerusakan Piston
6	K006	Kerusakan Poros Engkol

(Sumber: Data Penelitian 2019)

Tabel 3. 3 Kode Gejala

No	Kode	Gejala
1	G01	Mesin susah menyala
2	G02	Mesin terasa tersendat saat stasioner
3	G03	Tarikan sudah lemah

Tabel 3. 3 Lanjutan

No	Kode	Gejala
4	G04	Motor mengeluarkan asap hitam dari knalpot
5	G05	Tekanan kompresi terlalu tinggi
6	G06	Tekanan kompresi rendah
7	G07	Suara mesin berisik
8	G08	Mesin tidak dapat stasioner
9	G09	Keluar asap putih dari knalpot
10	G10	Kebocoran oli pada sambungannya
11	G11	Oli cepat berkurang atau habis
12	G12	Kompresi motor menjadi bocor

(Sumber: Data Penelitian 2019)

Tabel 3. 4 Kode Solusi

No	Kode	Solusi
1	S01	Bersihkan injektornya dengan cairan khusus pembersih injektor yang dituangkan ke tangki bahan bakar (Motor Injeksi)
2	S02	Buka saluran pembuangan bahan bakar dengan obeng untuk mengeluarkan air yang masuk didalam karburator. Kemudian kita bisa putar sekrup setelan udara searah jarum jam atau arah menutup untuk memperbanyak campuran udara dengan bahan bakar (Non Injeksi)
3	S03	Ganti main jet dengan ukuran yang lebih besar, ini terjadi karna asupan bahan bakar tidak sesuai dengan standarnya (Non Injeksi)
4	S04	Pertama putar sekrup setelan udara keluar atau kearah melepas hal ini dimaksudkan agar memperbanyak udara di lubang venturi, selanjutnya ganti spuyer pilot jet dengan

Tabel 3. 4 Lanjutan

No	Kode	Solusi
		ukuran yang lebih kecil dan ganti jarum skep dengan jarum yang ukuran lebih besar (Non Injeksi)
5	S05	Cek dan bersihkan busi, kalau perlu diganti dengan yang baru
6	S06	Buka kepala silinder dan bersihkan kepala silinder dari endapan / kotoran
7	S07	Buka tutup katup masuk dan katup buang lalu setel katup dengan benar
8	S08	Perbaiki <i>rocker arm</i> dan <i>cam shaft</i> atau ganti dengan yang baru
9	S09	Buka sistem katup dan diskir katup hingga menutup
10	S10	Ganti ring piston dengan yang baru, jika piston aus maka ganti juga dengan piston yang baru
11	S11	Perbaiki dengan menambah <i>oversize</i> blok silinder, ring dan piston ganti yang baru sesuai <i>oversize</i> (hanya jika dinding silinder aus)
12	S12	Periksalah baut bila benar-benar keras maka yang perlu di ganti adalah <i>paking-pakingnya</i>
13	S13	Ganti ring piston dengan yang baru, jika piston aus maka ganti juga dengan piston yang baru
14	S14	Jika menambah <i>oversize</i> blok silinder, ring dan piston perlu diganti yang baru sesuai <i>oversize</i>
15	S15	Periksa jika bantalan poros engkol yang aus maka gantilah bantalan poros engkolnya
16	S16	Jika pena piston aus, gantilah pena piston dengan yang baru
17	S17	Jika bantalan <i>connecting rod</i> yang aus, gantilah bantalan <i>connecting rod</i> dengan yang baru

(Sumber: Data Penelitian 2019)

Aturan antara gejala kerusakan dan jenis kerusakan adalah sebagai berikut :

Tabel 3. 5 Tabel Data Aturan

No.	Aturan (Rule)	Gejala	Kerusakan
1	R1	IF G01, G02, G03, G04	THEN K001
2	R2	IF G01, G05, G06, G07, G08	THEN K002
3	R3	IF G06, G09	THEN K003
4	R4	IF G10, G11	THEN K004
5	R5	IF G09, G11, G12	THEN K005
6	R6	IF G07, G12	THEN K006

(Sumber: Data Penelitian 2019)

Tabel 3. 6 Tabel Keputusan

No	Gejala	Kerusakan					
		K001	K002	K003	K004	K005	K006
1	G01	√	√	-	-	-	-
2	G02	√	-	-	-	-	-
3	G03	√	-	-	-	-	-
4	G04	√	-	-	-	-	-
5	G05	-	√	-	-	-	-
6	G06	-	√	√	-	-	-
7	G07	-	√	-	-	-	√
8	G08	-	√	-	-	-	-
9	G09	-	-	√	-	√	-
10	G10	-	-	-	√	-	-
11	G11	-	-	-	√	√	-
12	G12	-	-	-	-	√	√

(Sumber: Data Penelitian 2019)

Berikut adalah tabel dari solusi perbaikan:

Tabel 3. 7 Solusi Perbaikan

No	Kerusakan	Solusi
1	IF K001	THEN S01, S02, S03, S04
2	IF K002	THEN S05, S06, S07, S08, S09
3	IF K003	THEN S10, S11
4	IF K004	THEN S12
5	IF K005	THEN S13, S14
6	IF K006	THEN S15, S16, S17

(Sumber: Data Penelitian 2019)

Tabel 3. 8 Tabel Keputusan

No	Solusi	Kerusakan					
		K001	K002	K003	K004	K005	K006
1	S01	√	-	-	-	-	-
2	S02	√	-	-	-	-	-
3	S03	√	-	-	-	-	-
4	S04	√	-	-	-	-	-
5	S05	-	√	-	-	-	-
6	S06	-	√	-	-	-	-
7	S07	-	√	-	-	-	-
8	S08	-	√	-	-	-	-
9	S09	-	√	-	-	-	-
10	S10	-	-	√	-	-	-
11	S11	-	-	√	-	-	-
12	S12	-	-	-	√	-	-
13	S13	-	-	-	-	√	-
14	S14	-	-	-	-	√	-

Tabel 3. 8 Lanjutan

No	Solusi	Kerusakan					
		K001	K002	K003	K004	K005	K006
15	S15	-	-	-	-	-	√
16	S16	-	-	-	-	-	√
17	S17	-	-	-	-	-	√

(Sumber: Data Penelitian 2019)

Berdasarkan tabel keputusan pada tabel 3.6 maka pohon keputusannya adalah sebagai berikut:



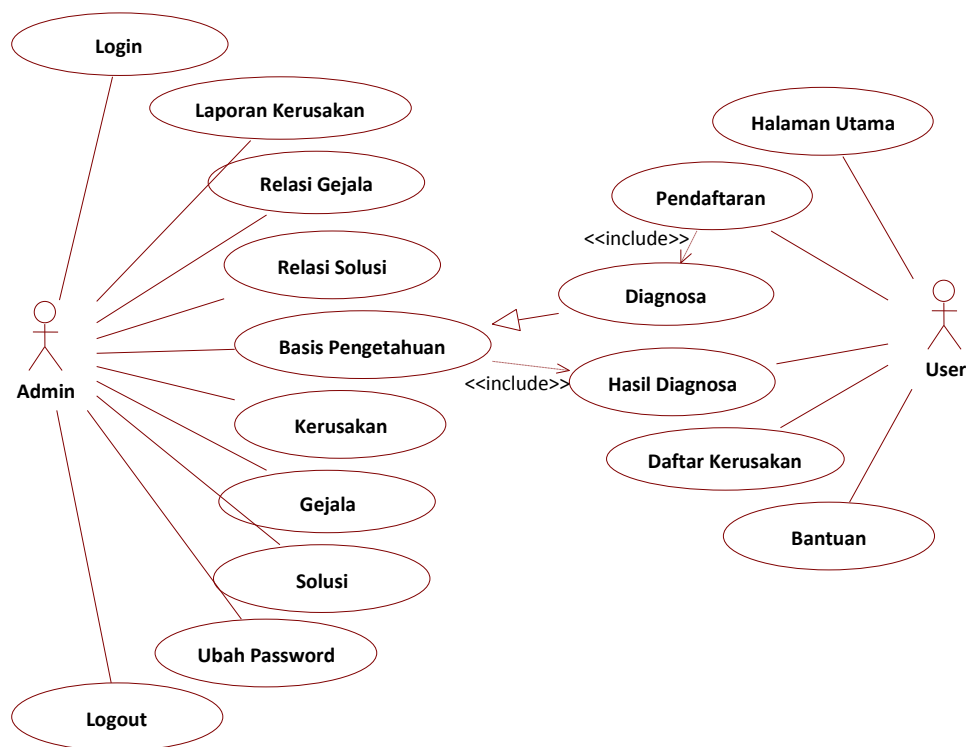
Gambar 3. 2 Pohon Keputusan
(Sumber: Data Penelitian 2019)

3.4.2 UML (*Unified Modeling Language*)

Menurut (A & Shalahuddin, 2014: 137-138) UML (*Unified Modelling Language*) adalah salah standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisi dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

1. *Use case Diagram*

Use Case Diagram yang akan digunakan pada sistem diagnosa kerusakan mesin sepeda motor *sport* seperti pada gambar dibawah ini.



Gambar 3. 3 *Use Case Diagram*
(Sumber: Data Penelitian 2019)

Aktor disini ada 2 yaitu *admin* dan *user*. *Admin* sendiri dapat berinteraksi dengan lebih leluasa jika dibandingkan dengan *user*, yaitu *login admin*, *update*

data, melihat laporan kerusakan atau hasil *output* dari konsultasi *user* dan masih banyak lainnya. Khusus untuk *update* data *admin* harus *login* terlebih dahulu. Sedangkan *user* berinteraksi dengan sistem yaitu menu utama, melihat daftar gejala kerusakan, konsultasi atau diagnosa, dan melihat *output* hasil diagnosa.

2. *Activity Diagram*

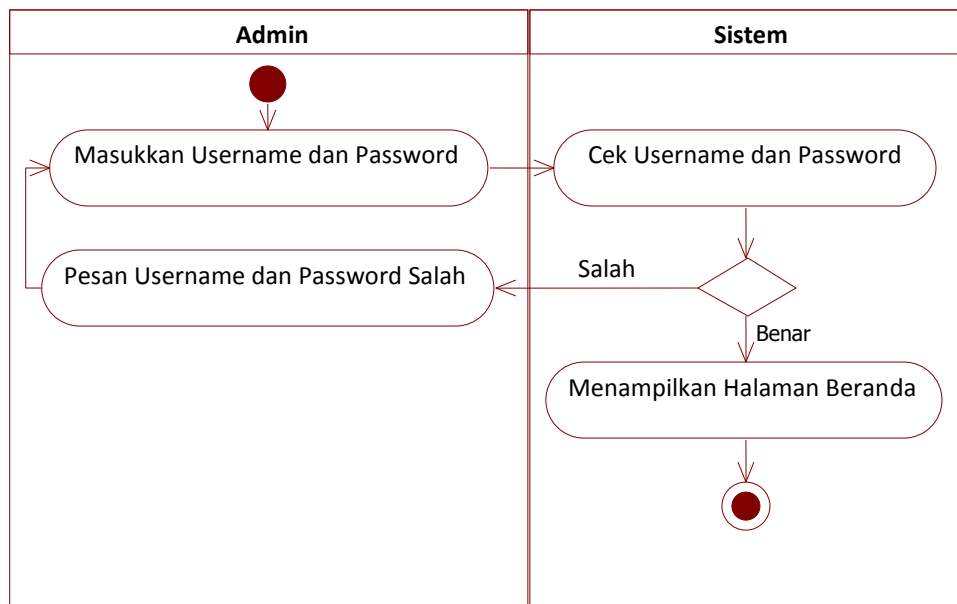
Berikut ini adalah *activity diagram* yang dirancang pada penelitian sistem pakar ini:

1. *Activity Diagram Admin*

Activity diagram admin adalah alur dari aktivitas sistem yang dapat diakses oleh *admin* seperti *form login*, ubah *password*, laporan kerusakan, relasi gejala, relasi solusi, data kerusakan, data gejala, data solusi, dan *logout*. Untuk lebih jelasnya bisa dilihat pada *activity diagram* dibawah ini:

a. *Activity Diagram Login*

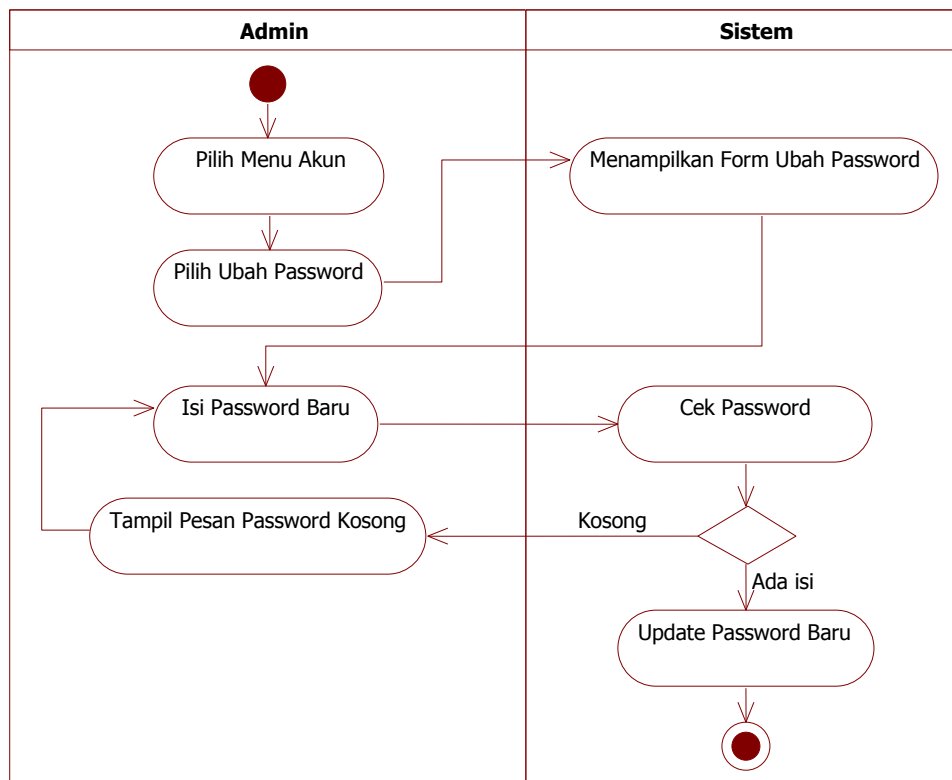
Activity diagram login digunakan untuk mengakses dan mengelola aplikasi sistem pakar. Dapat dijelaskan dari aktivitas *form login* yaitu pertama admin memasukkan *username* dan *passwordnya* lalu sistem melakukan pengecekan apakah *username* dan *passwordnya* terdaftar, jika dinyatakan salah atau tidak terdaftar maka akan menampilkan pesan *username* dan *password* salah, jika benar maka sistem akan menampilkan halaman beranda admin, untuk lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 3. 4 *Activity Diagram Login*
(Sumber: Data Penelitian 2019)

b. *Activity Diagram Ubah Password*

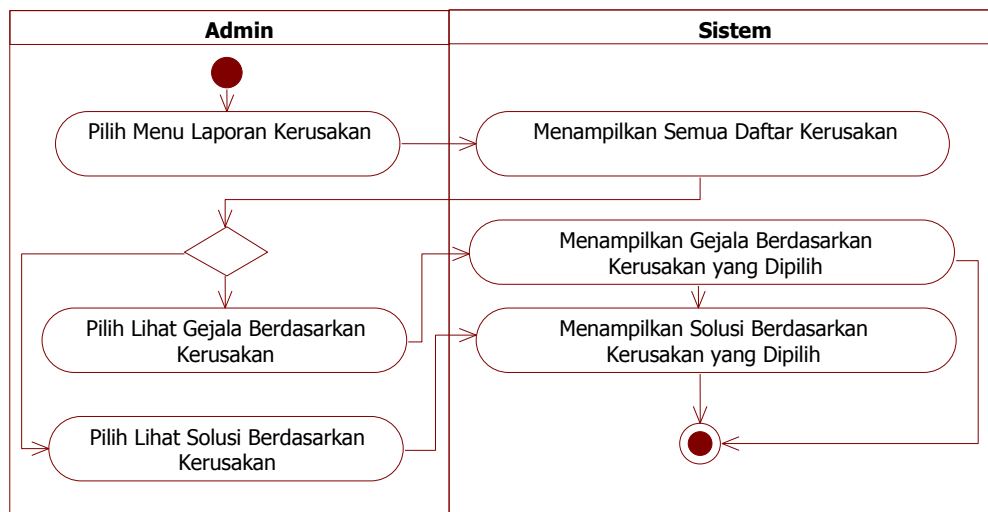
Activity diagram ubah password digunakan untuk mengganti *password admin*. Dapat dijelaskan dari aktivitas *ubah password* yaitu pertama saat *admin* masih login pilih menu *akun* kemudian pilih *ubah password*, akan tampil *username admin* dan *field password* kosong, jika *admin* ingin mengubah *password* isi *field* dengan *password* baru kemudian konfirmasi *password* setelah itu tekan *simpan*, *password* akan di *update* dengan yang baru, untuk lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 3. 5 *Activity Diagram Ubah Password*
(Sumber: Data Penelitian 2019)

c. *Activity Diagram Laporan Kerusakan*

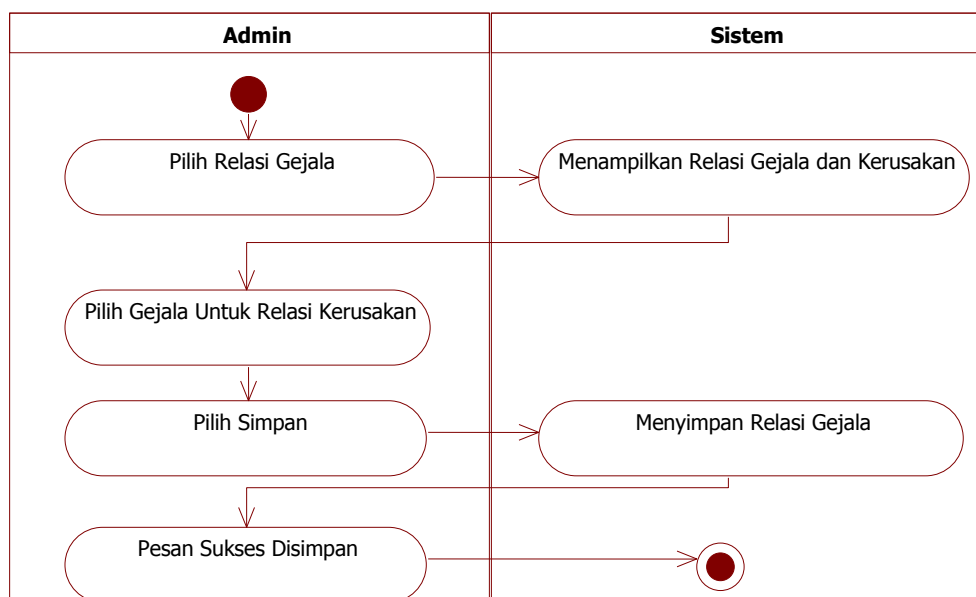
Activity diagram laporan kerusakan menampilkan keseluruhan daftar kerusakan beserta gejala dan rusaknya. Pertama pilih laporan kerusakan, kemudian bisa pilih lihat gejala berdasarkan kerusakan atau pilih solusi berdasarkan kerusakan. Dapat dilihat pada gambar berikut.



Gambar 3. 6 Activity Diagram Laporan Kerusakan
(Sumber: Data Penelitian 2019)

d. Activity Diagram Relasi Gejala

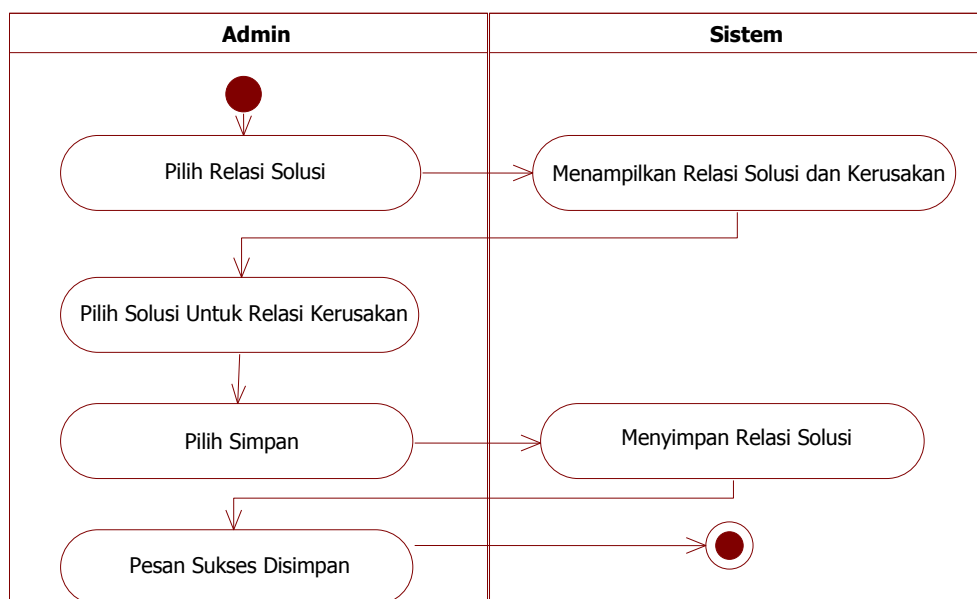
Activity diagram relasi gejala akan menampilkan relasi antara gejala dan kerusakan, *admin* dapat mengatur, mengubah ataupun menambah gejala dari suatu kerusakan, *admin* dapat mengatur, mengubah ataupun menambah gejala dari suatu kerusakan. Langkah pertama pilih relasi gejala kemudian *admin* bisa mengubah jika ada yang ingin diubah, setelah itu *klik* simpan dan data akan tersimpan.



Gambar 3. 7 Activity Diagram Relasi Gejala
(Sumber: Data Penelitian 2019)

e. *Activity Diagram* Relasi Solusi

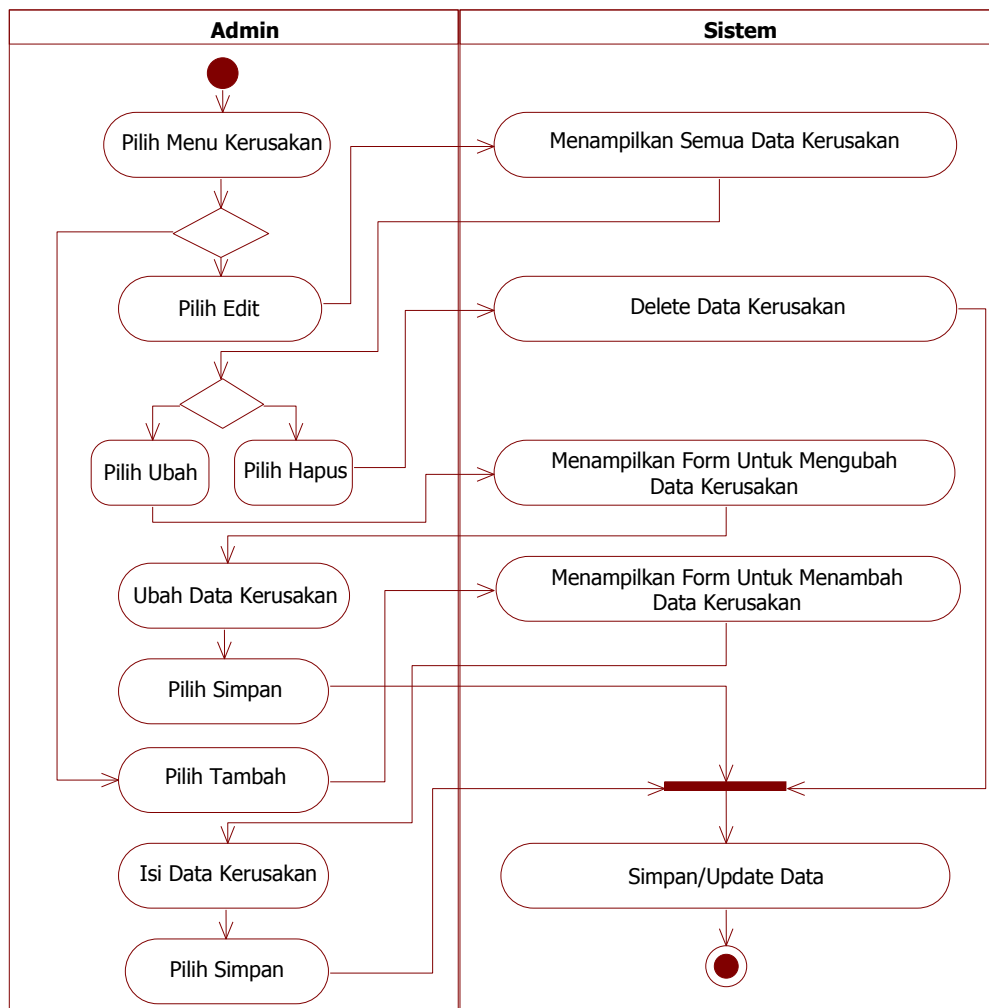
Activity diagram relasi solusi akan menampilkan relasi antara solusi dan kerusakan, *admin* dapat mengatur, mengubah ataupun menambah solusi perbaikan dari kerusakan. Langkah pertama pilih relasi solusi kemudian *admin* bisa mengubah jika ada yang ingin diubah, setelah itu *klik* simpan dan data akan tersimpan.



Gambar 3. 8 *Activity Diagram* Relasi Solusi
(Sumber: Data Penelitian 2019)

f. *Activity Diagram* Data Kerusakan

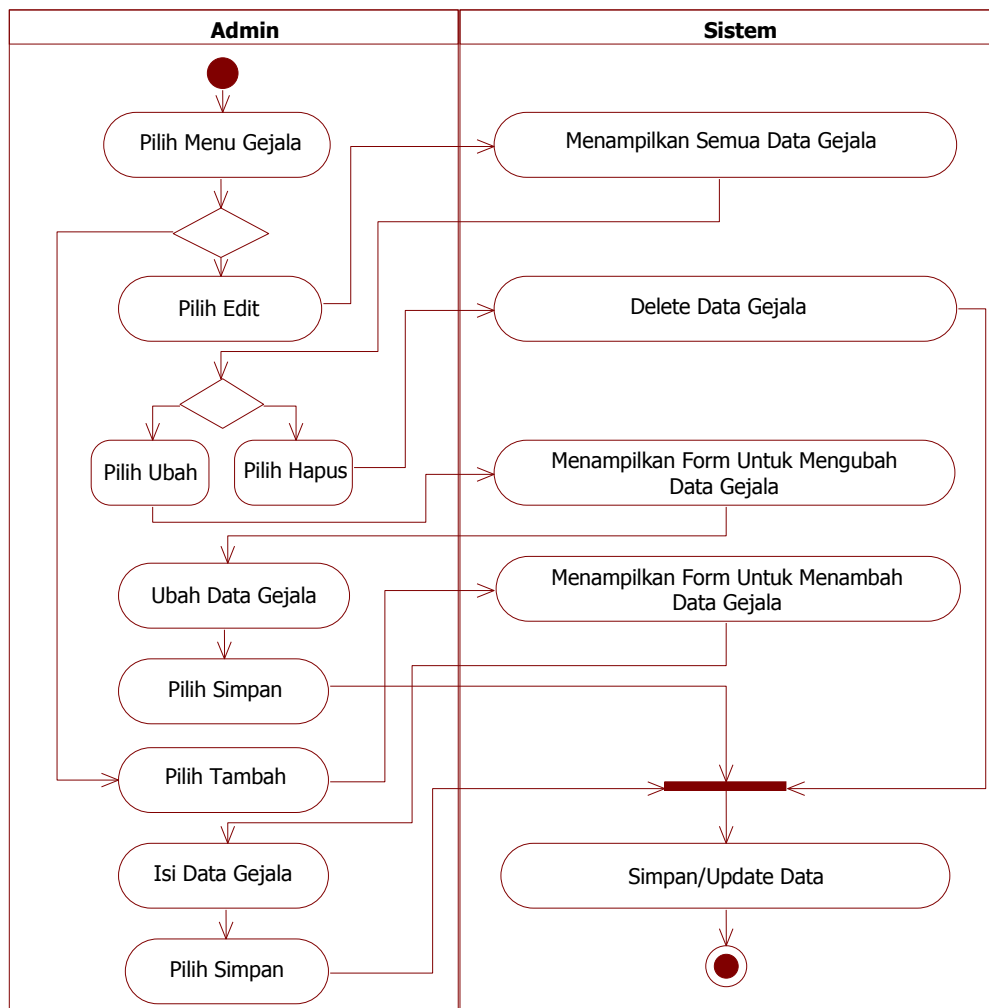
Activity diagram data kerusakan memungkinkan *admin* untuk melakukan *update* data kerusakan, mengubah, menambah, maupun menghapus data. Pertama pilih menu kerusakan kemudian pilih *edit/ubah* jika ingin mengubah, disini bisa juga memilih hapus untuk menghapus atau pilih tambah jika ingin menambah data, selanjutnya isi data terus pilih simpan maka perubahan akan tersimpan.



Gambar 3. 9 Activity Diagram Data Kerusakan
(Sumber: Data Penelitian 2019)

g. Activity Diagram Data Gejala

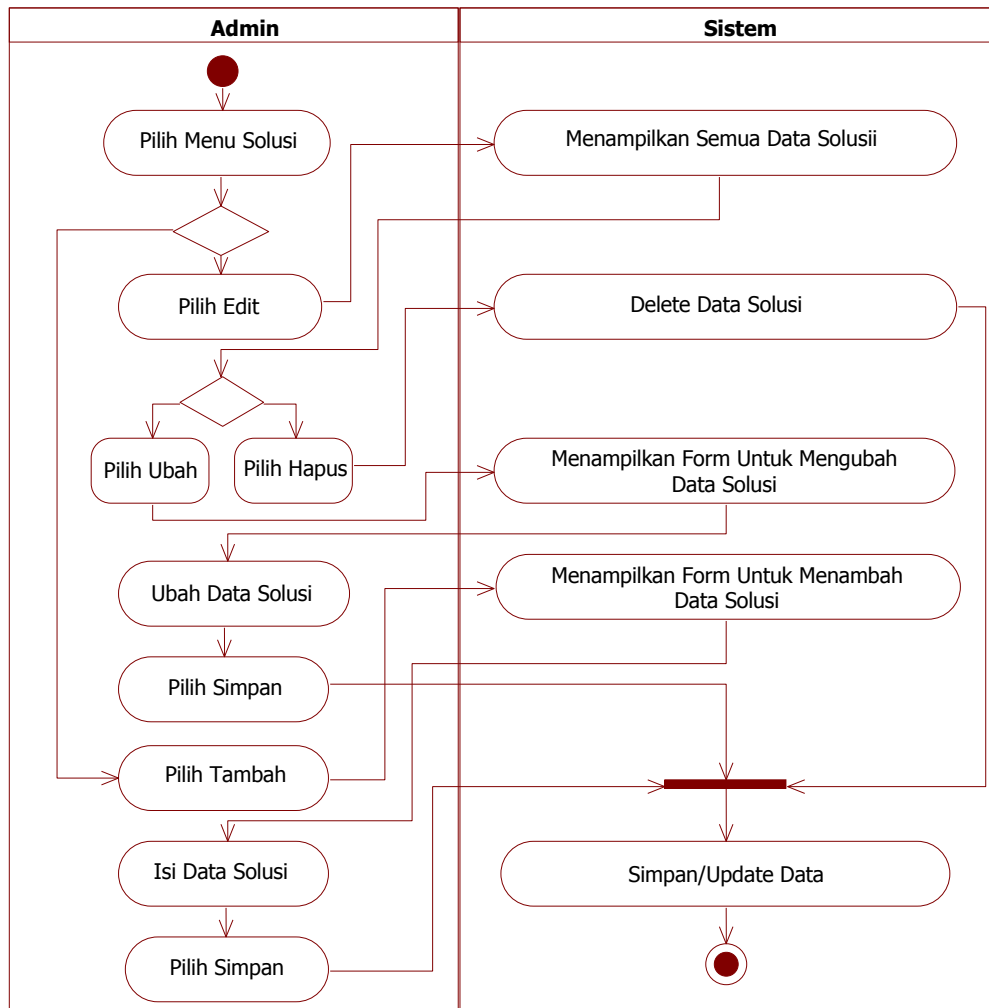
Activity diagram data gejala memungkinkan *admin* untuk melakukan *update* data gejala, mengubah, menambah, maupun menghapus data. Pertama pilih menu gejala kemudian pilih *edit/ubah* jika ingin mengubah, disini bisa juga memilih hapus untuk menghapus atau pilih tambah jika ingin menambah data, selanjutnya isi data terus pilih simpan maka perubahan akan tersimpan.



Gambar 3. 10 *Activity Diagram* Data Gejala
(Sumber: Data Penelitian 2019)

h. *Activity Diagram* Data Solusi

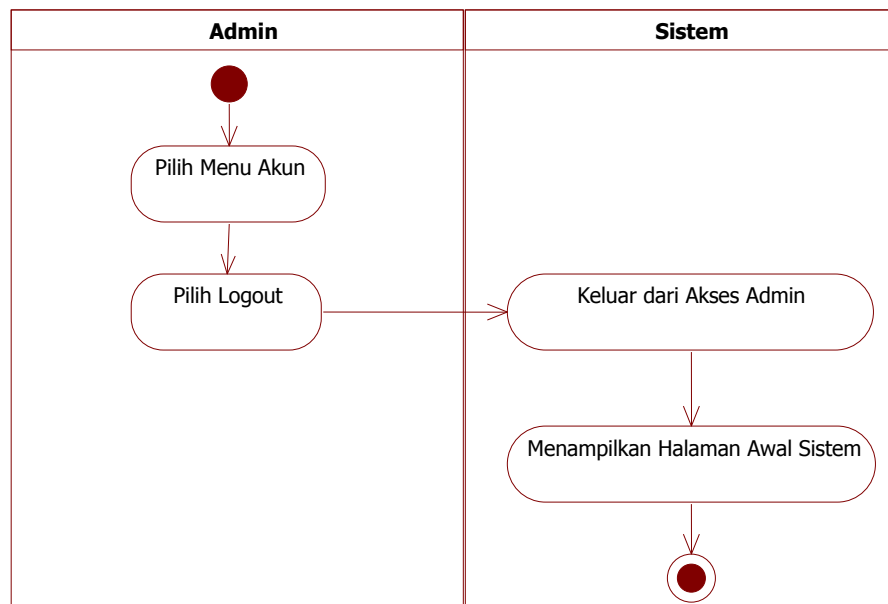
Activity diagram data solusi memungkinkan *admin* untuk melakukan *update* data solusi perbaikan, mengubah, menambah, maupun menghapus data. Pertama pilih menu solusi kemudian pilih *edit*/ubah jika ingin mengubah, disini bisa juga memilih hapus untuk menghapus atau pilih tambah jika ingin menambah data, selanjutnya isi data terus pilih simpan maka perubahan akan tersimpan.



Gambar 3. 11 *Activity Diagram Data Solusi*
(Sumber: Data Penelitian 2019)

i. *Activity Diagram Logout*

Activity diagram logout digunakan *admin* untuk keluar dari halaman *admin* hanya dengan *klik* menu *logout admin* akan kembali ke halaman awal dari sistem.



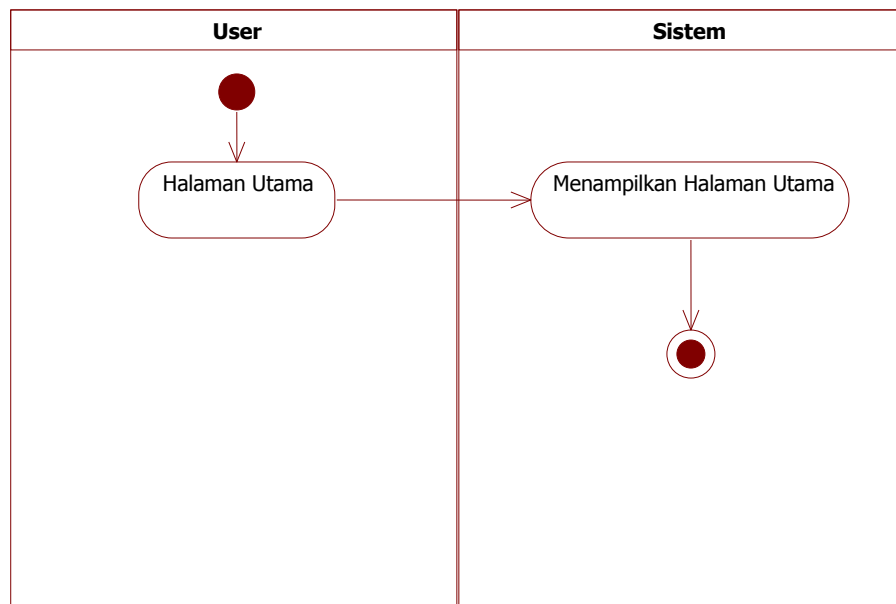
Gambar 3. 12 *Activity Diagram Logout*
(Sumber: Data Penelitian 2019)

2. *Activity Diagram User*

Activity diagram user adalah alur dari aktivitas sistem yang dapat diakses oleh *user* seperti halaman utama, diagnosa kerusakan, daftar kerusakan dan bantuan. Untuk lebih jelasnya bisa dilihat pada *activity diagram* dibawah ini:

a. *Activity Diagram Menu Halaman Utama*

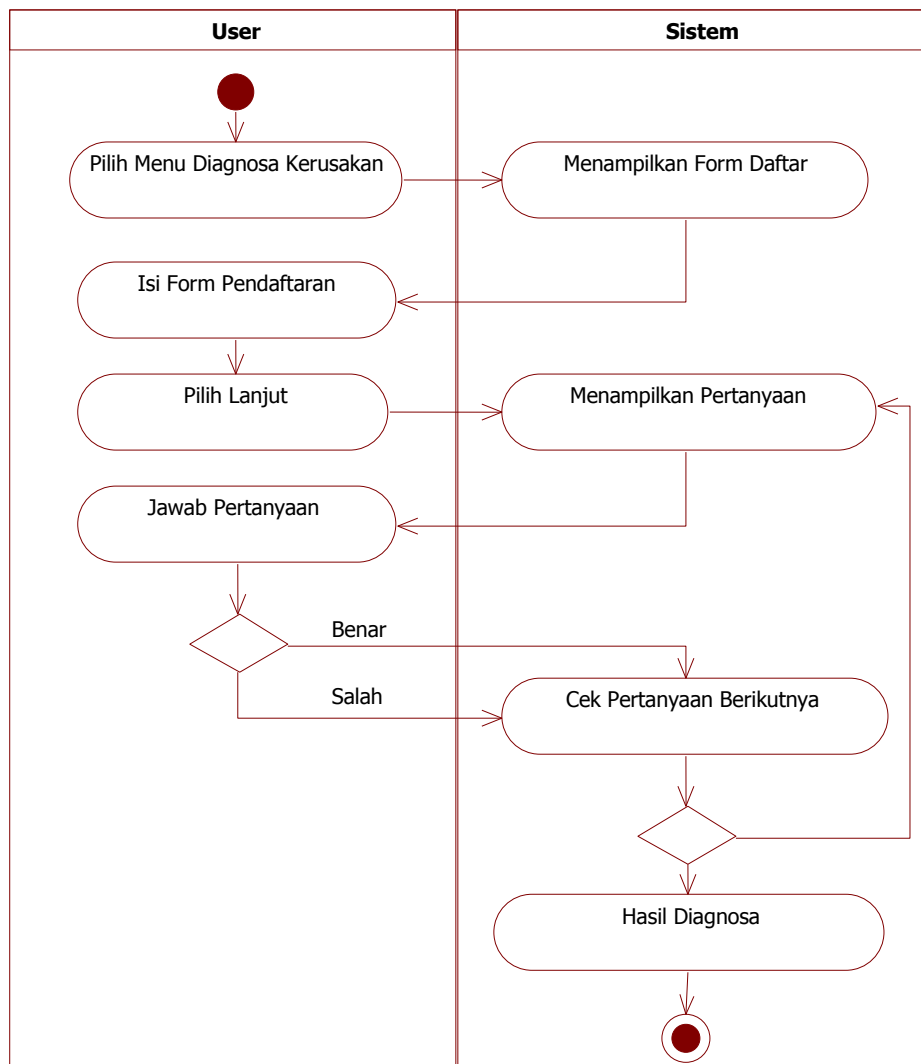
Activity diagram menu halaman utama menampilkan halaman yang memuat sekilas pengetahuan dasar tentang mesin sepeda motor. Hanya tinggal *klik* menu halaman utama akan langsung ditampilkan isi yang ada dihalaman utama.



Gambar 3. 13 *Activity Diagram* Menu Halaman Utama
(Sumber: Data Penelitian 2019)

b. *Activity Diagram* Diagnosa Kerusakan

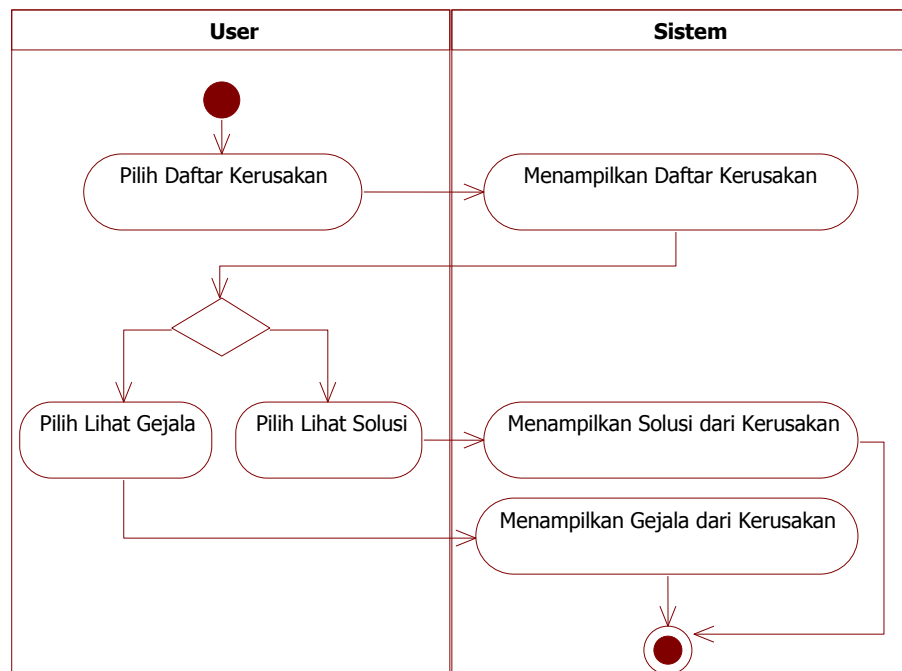
Activity diagram diagnosa kerusakan digunakan oleh *user* untuk melakukan diagnosa kerusakan. Pertama pilih menu diagnosa kerusakan dan akan muncul *form* pendaftaran untuk diisi data *user* yaitu nama, jenis kelamin, alamat, dan email, kemudian *user* menjawab pertanyaan sesuai masalah yang dialami, setelah selesai akan keluar hasil dari diagnosa kerusakannya.



Gambar 3. 14 *Activity Diagram Diagnosa Kerusakan*
(Sumber: Data Penelitian 2019)

c. *Activity Diagram* Daftar Kerusakan

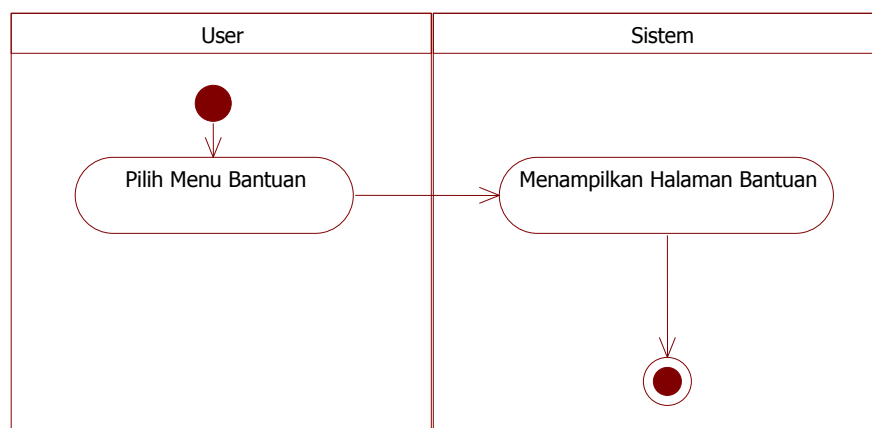
Activity diagram daftar kerusakan menampilkan semua data kerusakan beserta gejala dan solusi perbaikannya. Pertama pilih menu daftar kerusakan dan akan muncul semua data kerusakan disertai gejala dan solusi.



Gambar 3. 15 *Activity Diagram* Daftar Kerusakan
(Sumber: Data Penelitian 2019)

d. *Activity Diagram* Bantuan

Activity diagram bantuan menampilkan halaman bantuan yang berisi kontak yang bisa dihubungi. Hanya dengan *klik* menu bantuan akan langsung muncul kontak yang bisa dihubungi.



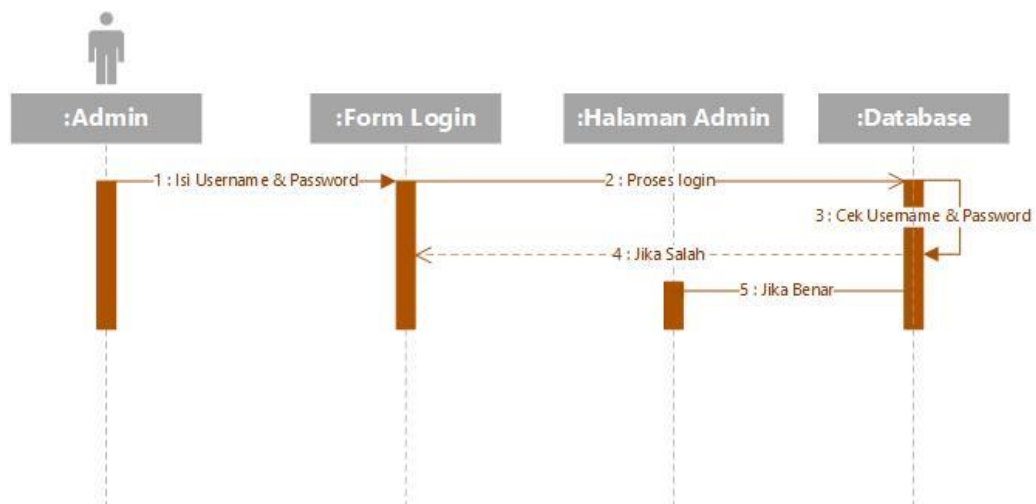
Gambar 3. 16 *Activity Diagram* Bantuan
(Sumber: Data Penelitian 2019)

3. *Sequence Diagram*

Sequence diagram (diagram sekuen) menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (A & Shalahuddin, 2014: 165). Diagram sekuen yang akan rancang dalam penelitian ini adalah sebagai berikut:

a. *Sequence Diagram Login*

Adapun *sequence diagram* data login dapat dilihat pada gambar berikut.



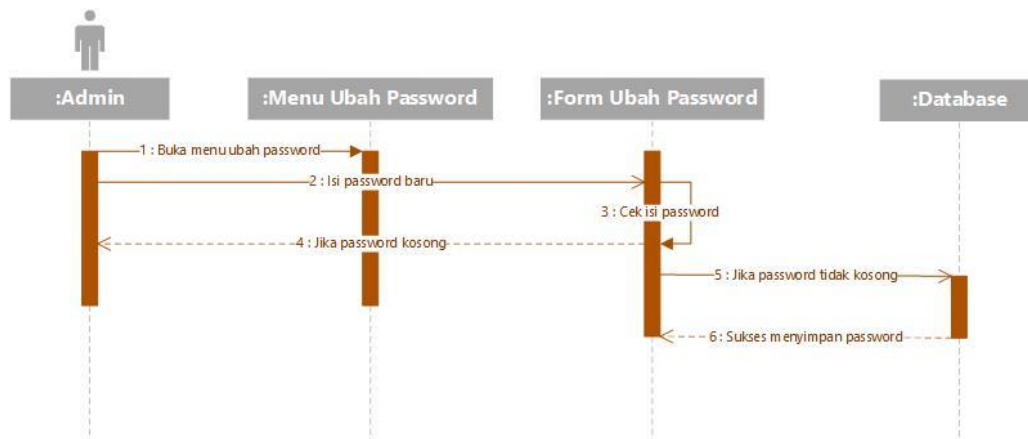
Gambar 3. 17 *Sequence Diagram Login*
(Sumber: Data Penelitian 2019)

Dari gambar 3. 17 *sequence diagram login* dapat dijelaskan bahwa:

- Admin* mengisi *username* dan *password* pada *form login*,
- Kemudian melakukan verifikasi *login*,
- Jika dinyatakan salah maka kembali ke halaman *login*
- Jika dinyatakan benar maka *admin* masuk ke menu utama dan bisa melakukan pengolahan data pada aplikasi sistem pakar ini.

2. *Sequence Diagram* Ubah Password

Adapun *sequence diagram* ubah password dapat dilihat pada gambar berikut.

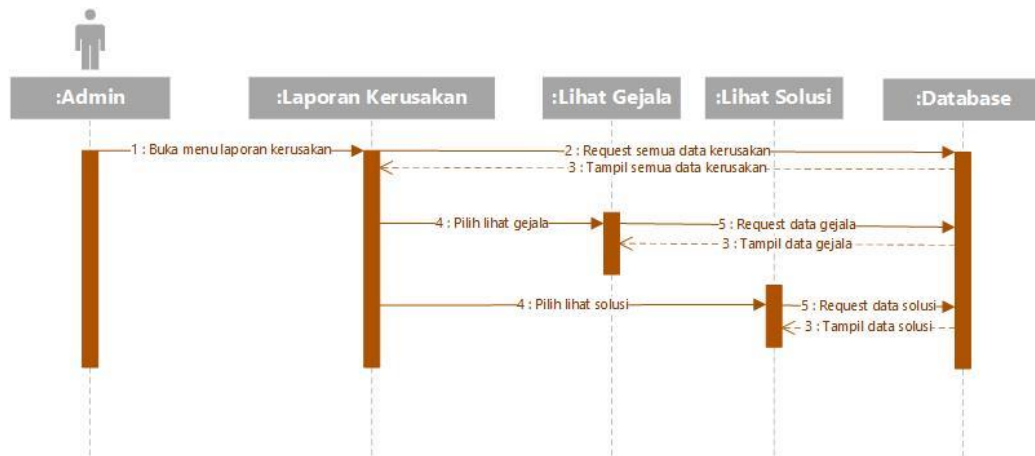


Gambar 3. 18 *Sequence Diagram* Ubah Password
(Sumber: Data Penelitian 2019)

Dari gambar 3.18 *sequence diagram* ubah password dapat dijelaskan bahwa:

- a. *Admin* membuka menu ubah password dan sistem akan menampilkan form ubah password
 - b. Kemudian *admin* mengubah password jika diperlukan
 - c. Jika diklik simpan maka data password akan diproses dan menyimpannya ke database
 - d. Jika proses simpan selesai dilakukan maka akan menampilkan kembali form ubah password.
- ## 3. *Sequence Diagram* Laporan Kerusakan

Adapun *sequence diagram* laporan kerusakan dapat dilihat pada gambar berikut.



Gambar 3. 19 *Sequence Diagram* Laporan Kerusakan
(Sumber: Data Penelitian 2019)

Dari gambar 3. 19 *sequence diagram laporan kerusakan* dapat dijelaskan bahwa:

- a. *Admin* membuka menu laporan kerusakan maka akan menampilkan halaman laporan kerusakan
 - b. Jika *admin* ingin melihat gejala yang berhubungan dengan kerusakan maka *user* harus mengklik tombol lihat dan menampilkan halaman data gejala
 - c. Jika *admin* ingin melihat solusi yang berhubungan dengan kerusakan maka *user* harus mengklik tombol lihat dan menampilkan halaman data solusi.
4. *Sequence Diagram* Relasi Gejala

Adapun *sequence diagram* relasi gejala dapat dilihat pada gambar berikut.



Gambar 3. 20 *Sequence Diagram* Relasi Gejala
(Sumber: Data Penelitian 2019)

Dari gambar 3. 20 *sequence diagram* relasi gejala dapat dijelaskan bahwa:

- a. *Admin* membuka menu relasi gejala kemudian menampilkan halaman relasi gejala
 - b. Kemudian jika diperlukan *admin* bisa mengubah relasi antara data kerusakan dan data gejala
 - c. Jika diklik simpan maka hasil perubahan akan diproses dan menyimpannya ke *database*.
5. *Sequence Diagram* Relasi Solusi

Adapun *sequence diagram* relasi solusi dapat dilihat pada gambar berikut.

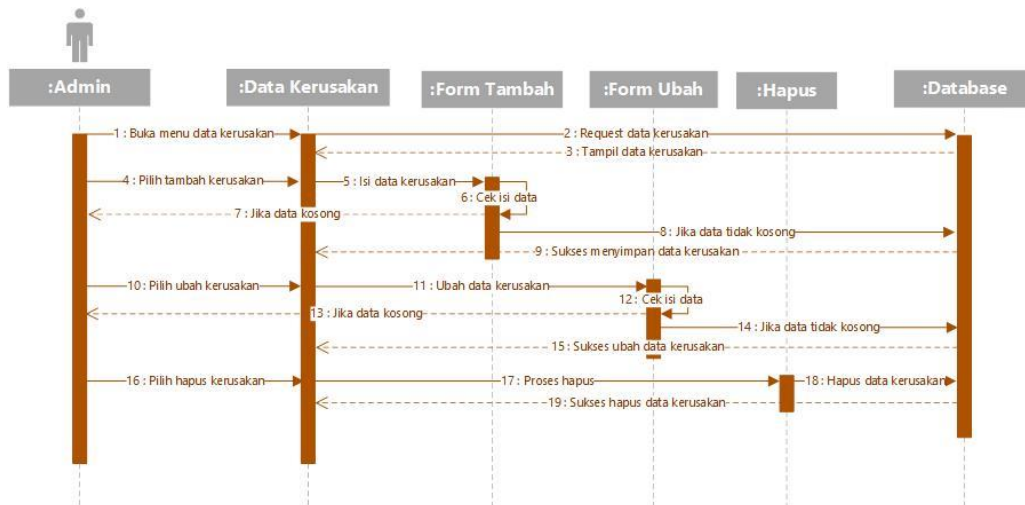


Gambar 3. 21 *Sequence Diagram* Relasi Solusi
(Sumber: Data Penelitian 2019)

Dari gambar 3. 21 *sequence diagram* relasi solusi dapat dijelaskan bahwa:

- a. *Admin* membuka menu relasi solusi kemudian menampilkan halaman relasi solusi
 - b. Kemudian jika diperlukan *admin* bisa mengubah relasi antara data kerusakan dan data solusi
 - c. Jika diklik simpan maka hasil perubahan akan diproses dan menyimpannya ke *database*.
6. *Sequence Diagram* Data Kerusakan

Adapun *sequence diagram* data kerusakan dapat dilihat pada gambar berikut.



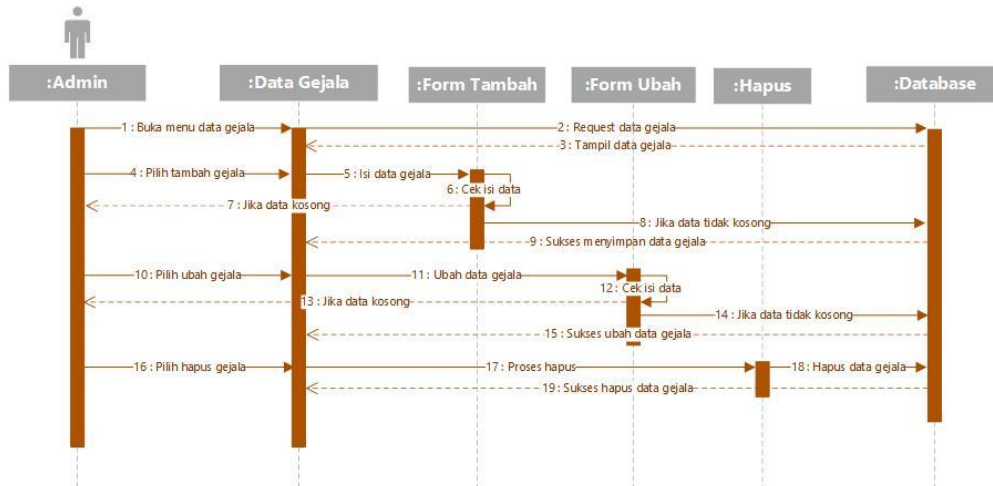
Gambar 3. 22 *Sequence Diagram* Data Kerusakan
(Sumber: Data Penelitian 2019)

Dari gambar 3. 22 *sequence diagram* data kerusakan dapat dijelaskan bahwa:

- Admin* membuka menu kerusakan kemudian menampilkan menu pilihan antara *edit* dan tambah
- Jika *admin* ingin menambahkan data kerusakan baru maka pilih menu tambah dan menampilkan *form* tambah data kerusakan
- Jika sudah di *input* kerusakannya maka otomatis sistem melakukan proses simpan data ke *database*
- Jika *admin* ingin mengedit atau mengubah data maka pilih menu *edit/ubah* dan akan menampilkan *form* ubah data kerusakan
- Jika sudah di ubah data kerusakannya maka otomatis sistem melakukan proses *update* data ke *database*
- Jika *admin* menghapus data kerusakan maka otomatis sistem menghapus data dari *database*.

7. Sequence Diagram Data Gejala

Adapun *sequence diagram* data gejala dapat dilihat pada gambar berikut.



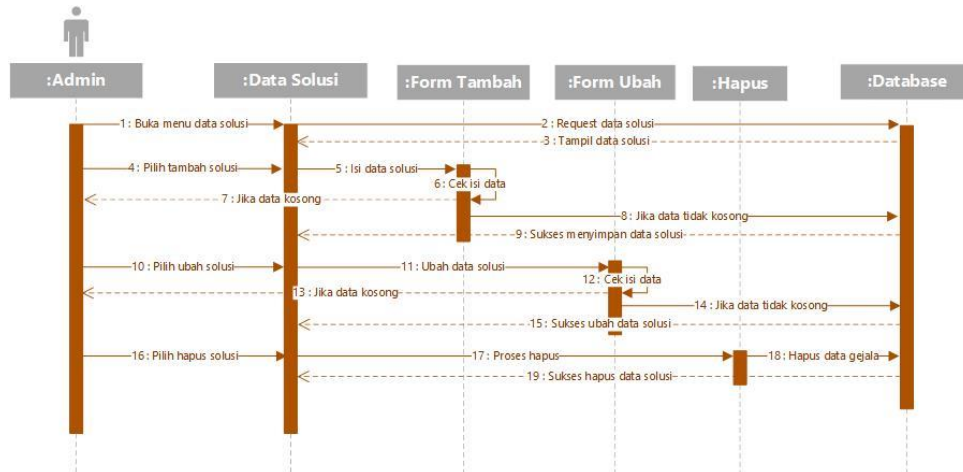
Gambar 3. 23 *Sequence Diagram* Data Gejala
(Sumber: Data Penelitian 2019)

Dari gambar 3. 23 *sequence diagram* data gejala dapat dijelaskan bahwa:

- Admin* membuka menu gejala kemudian menampilkan pilihan menu antara *edit* dan tambah
- Jika *admin* ingin menambahkan data gejala baru maka pilih menu tambah dan menampilkan *form* tambah data gejala
- Jika sudah di *input* gelajanya maka otomatis sistem melakukan proses simpan data ke *database*
- Jika *admin* ingin mengedit atau mengubah data maka pilih menu *edit/ubah* dan akan menampilkan *form* ubah data gejala
- Jika sudah di ubah data gejalanya maka otomatis sistem melakukan proses *update* data ke *database*
- Jika *admin* menghapus data gejala maka otomatis sistem menghapus data dari *database*.

8. Sequence Diagram Data Solusi

Adapun *sequence diagram* data solusi dapat dilihat pada gambar berikut.



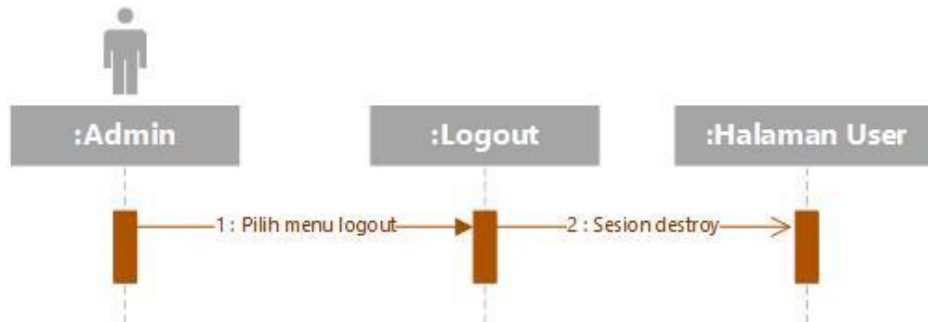
Gambar 3. 24 *Sequence Diagram* Data Solusi
(Sumber: Data Penelitian 2019)

Dari gambar 3. 24 *sequence diagram* data solusi dapat dijelaskan bahwa:

- Admin* membuka menu solusi kemudian menampilkan pilihan menu antara *edit* dan tambah
- Jika *admin* ingin menambahkan data solusi baru maka pilih menu tambah dan menampilkan *form* tambah data solusi
- Jika sudah di *input* solusinya maka otomatis sistem melakukan proses simpan data ke *database*
- Jika *admin* ingin mengedit atau mengubah data maka pilih menu *edit/ubah* dan akan menampilkan *form* ubah data solusi
- Jika sudah di ubah data solusinya maka otomatis sistem melakukan proses *update* data ke *database*
- Jika *admin* menghapus data solusi maka otomatis sistem menghapus data dari *database*.

9. *Sequence Diagram Logout*

Adapun *sequence diagram* data solusi dapat dilihat pada gambar berikut.



Gambar 3. 25 *Sequence Diagram Logout*
(Sumber: Data Penelitian 2019)

Dari gambar 3. 25 *sequence diagram logout* dapat dijelaskan bahwa:

- a. Jika *admin* klik menu *logout* maka sistem akan memprosesnya
 - b. Jika selesai diproses maka otomatis keluar dari *akun admin* dan dialihkan ke halaman awal dari pertama kali masuk ke sistem.
10. *Sequence Diagram Halaman Utama*

Adapun *sequence diagram* halaman utama dapat dilihat pada gambar berikut.

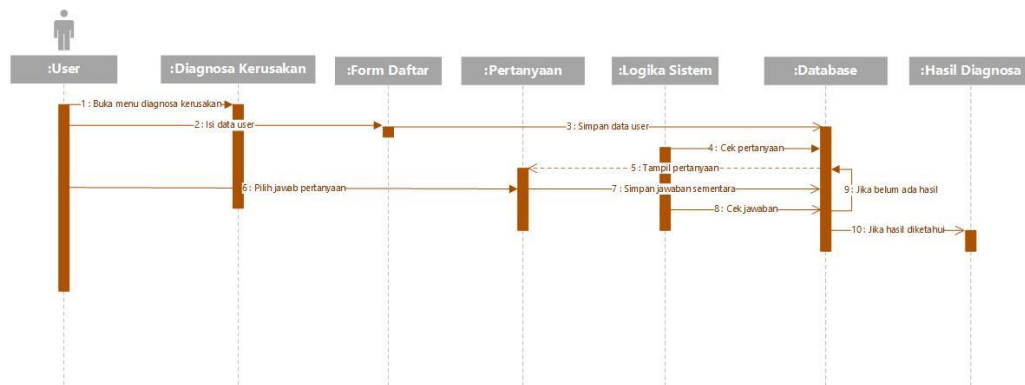


Gambar 3. 26 *Sequence Diagram Halaman Utama*
(Sumber: Data Penelitian 2019)

Dari gambar 3.26 *sequence diagram* halaman utama dapat dijelaskan bahwa:

- a. Jika *user klik* menu halaman utama maka sistem akan memprosesnya
 - b. Sistem akan menampilkan halaman utama.
11. *Sequence Diagram* Diagnosa Kerusakan

Adapun *sequence diagram* diagnosa kerusakan dapat dilihat pada gambar berikut.



Gambar 3. 27 *Sequence Diagram* Diagnosa Kerusakan
(Sumber: Data Penelitian 2019)

Dari gambar 3. 27 *sequence diagram* diagnosa kerusakan dapat dijelaskan bahwa:

- a. *User* membuka menu diagnosa kerusakan
- b. *User* mengisi data pada *form* pendaftaran.
- c. *User* melakukan diagnosa dan sistem akan menampilkan pertanyaan seputar jenis-jenis gejala yang mungkin dialami dengan menjawab “Ya atau Tidak”
- d. Jawaban yang dilakukan *user* akan disesuaikan dengan basis data pengetahuan yang tersimpan di *database*
- e. Sistem akan meneruskan peranyaan selanjutnya sampai hasil ditemukan

- f. Selama diagnosa interaksi yang dilakukan *user* akan selalu disimpan sementara ke *database* untuk menentukan dan menyesuaikan basis data pengetahuan dengan jawaban dari *user*
 - g. Sistem menampilkan hasil diagnosa berdasarkan basis pengetahuan dan pertanyaan yang dijawab *user*
 - h. *User* bisa melihat hasil diagnosa.
12. *Sequence Diagram* Daftar Kerusakan

Adapun *sequence diagram* daftar kerusakan dapat dilihat pada gambar berikut.



Gambar 3. 28 *Sequence Diagram* Daftar Kerusakan
(Sumber: Data Penelitian 2019)

Dari gambar 3. 28 *sequence diagram* daftar kerusakan dapat dijelaskan bahwa:

- a. *User* membuka menu daftar kerusakan maka akan menampilkan halaman daftar kerusakan
- b. Jika *user* ingin melihat gejala atau solusi dari kerusakan yang dipilih maka *user* harus mengklik tombol lihat dibagian gejala atau dibagian solusi dan sistem menampilkan halaman gejala atau solusi.

13. *Sequence Diagram* Bantuan

Adapun *sequence diagram* bantuan dapat dilihat pada gambar berikut.



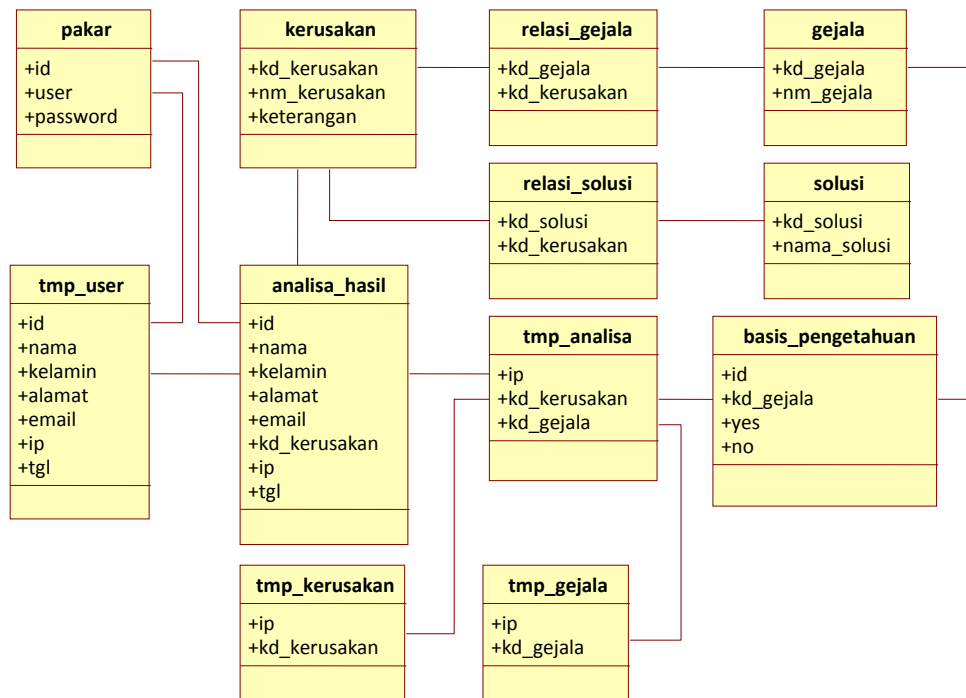
Gambar 3. 29 *Sequence Diagram* Bantuan
(Sumber: Data Penelitian 2019)

Dari gambar 3.29 *sequence diagram* bantuan dapat dijelaskan bahwa:

- a. Jika *user klik* menu bantuan maka sistem akan memprosesnya
- b. Sistem akan menampilkan halaman bantuan.

4. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.



Gambar 3. 30 *Class diagram*
(Sumber: Data Penelitian 2019)

3.4.3 Desain *Database*

Sistem basis data (*database*) adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat (A & Shalahuddin, 2014: 43). Adapun desain *databasenya* adalah sebagai berikut.

1. Tabel Pakar

Tabel pakar berguna untuk menyimpan data *username* dan *password* agar *admin* pakar dapat masuk ke menu utama *admin* dan dapat melakukan manipulasi data.

Tabel 3. 9 Tabel Pakar

Field	Tipe	Panjang	Kunci
id	Int	11	PK
user	varchar	20	
pass	varchar	100	

(**Sumber:** Data Penelitian 2019)

2. Tabel Gejala

Tabel ini berguna untuk menyimpan semua daftar gejala.

Tabel 3. 10 Tabel Gejala

Field	Tipe	Panjang	Kunci
kd_gejala	varchar	4	PK
nm_gejala	varchar	100	

(**Sumber:** Data Penelitian 2019)

3. Tabel Kerusakan

Tabel ini berguna untuk menyimpan semua daftar kerusakan.

Tabel 3. 11 Tabel Kerusakan

Field	Tipe	Panjang	Kunci
kd_kerusakan	varchar	5	PK
nm_kerusakan	varchar	100	
keterangan	mediumtext		

(**Sumber:** Data Penelitian 2019)

4. Tabel Solusi

Tabel ini berguna untuk menyimpan semua daftar solusi.

Tabel 3. 12 Tabel Solusi

Field	Tipe	Panjang	Kunci
kd_solusi	varchar	3	PK
nama_solusi	varchar	10	

(**Sumber:** Data Penelitian 2019)

5. Tabel Relasi

Tabel ini berguna untuk membuat relasi antara gejala dengan kerusakan.

Tabel 3. 13 Tabel Relasi

Field	Tipe	Panjang	Kunci
kd_gejala	varchar	4	
kd_kerusakan	varchar	5	

(**Sumber:** Data Penelitian 2019)

6. Tabel Relasi Solusi

Tabel ini berguna untuk membuat relasi antara solusi dengan kerusakan.

Tabel 3. 14 Tabel Solusi

Field	Tipe	Panjang	Kunci
kd_kerusakan	varchar	4	
kd_solusi	varchar	3	

(**Sumber:** Data Penelitian 2019)

7. Tabel Analisa Hasil

Tabel ini berguna untuk menyimpan hasil dari diagnosa.

Tabel 3. 15 Tabel Analisa Hasil

Field	Tipe	Panjang	Kunci
id	int	11	PK
nama	varchar	30	
kelamin	enum(L,P)		
alamat	varchar	100	
email	varchar	30	
kd_kerusakan	varchar	5	
ip	varchar	20	
tgl	datetime		

(Sumber: Data Penelitian 2019)

8. Tabel Tmp *User*

Tabel ini berguna untuk penyimpanan sementara data *user*.

Tabel 3. 16 Tabel Tmp *User*

Field	Tipe	Panjang	Kunci
id	int	11	PK
nama	varchar	30	
kelamin	enum(L,P)		
alamat	varchar	100	
email	varchar	30	
ip	varchar	20	
tgl	datetime		

(Sumber: Data Penelitian 2019)

9. Tabel Tmp Analisa

Tabel ini berfungsi untuk penyimpanan sementara hasil analisa diagnosa.

Tabel 3. 17 Tabel Tmp Analisa

Field	Tipe	Panjang	Kunci
ip	varchar	20	
kd_kerusakan	varchar	5	
kd_gejala	varchar	4	

(Sumber: Data Penelitian 2019)

10. Tabel Tmp Kerusakan

Tabel ini berfungsi untuk penyimpanan sementara data kerusakan.

Tabel 3. 18 Tabel Temp Kerusakan

Field	Tipe	Panjang	Kunci
ip	varchar	20	
kd_kerusakan	varchar	5	

(Sumber: Data Penelitian 2019)

11. Tabel Tmp Gejala

Tabel ini berfungsi untuk penyimpanan sementara data gejala.

Tabel 3. 19 Tabel Tmp Gejala

Field	Tipe	Panjang	Kunci
ip	varchar	20	
kd_gejala	varchar	4	

(Sumber: Data Penelitian 2019)

12. Tabel Basis Pengetahuan

Tabel ini sebagai data kecerdasan untuk menyimpan daftar kemungkinan potensi kecerdasan pada saat menjawab pertanyaan yang diajukan.

Tabel 3. 20 Tabel Basis Pengetahuan

Field	Tipe	Panjang	Kunci
id	int	11	PK
kd_gejala	varchar	5	
yes	varchar	5	
no	varchar	5	

(Sumber: Data Penelitian 2019)

3.4.4 Desain Antarmuka

Desain antarmuka merupakan rancangan antarmuka yang akan digunakan untuk mendeskripsikan rencana tampilan dari setiap *form* yang akan digunakan pada sistem pakar yang sebenarnya. Berikut adalah tampilan antarmuka pada sistem pakar diagnosis kerusakan mesin.

1. Tampilan Halaman Awal

Halaman yang pertama kali ditampilkan saat user masuk ke *web* sistem pakar diagnosis kerusakan mesin sepeda motor. Berikut ini adalah tampilan halamannya.

SISTEM PAKAR	Halaman Utama	Diagnosa Kerusakan	Daftar Kerusakan	Bantuan	PAKAR
FOOTER					

Gambar 3. 31 Tampilan Halaman Awal
(Sumber: Data Penelitian 2019)

2. Tampilan Halaman Utama

Halaman ini akan menampilkan konten yang berisi sekilas pengetahuan dasar tentang mesin sepeda motor. Tampilannya sebagai berikut.

SISTEM PAKAR	<u>Halaman Utama</u>	Diagnosa Kerusakan	Daftar Kerusakan	Bantuan	PAKAR
CONTENT					
FOOTER					

Gambar 3. 32 Tampilan Halaman Utama
(Sumber: Data Penelitian 2019)

3. Tampilan Halaman Diagnosa Kerusakan

Halaman ini akan menampilkan halaman untuk *user* melakukan diagnosa kerusakan, tetapi *user* diharuskan untuk mendaftar dulu sebelum melakukan diagnosa kerusakannya. Lebih jelasnya bisa dilihat pada gambar berikut.

SISTEM PAKAR	Halaman Utama	<u>Diagnosa Kerusakan</u>	Daftar Kerusakan	Bantuan	PAKAR
---------------------	---------------	---------------------------	------------------	---------	-------

Masukkan data anda untuk mulai Diagnosa

Masukkan Nama

Laki-Laki Perempuan

Alamat

Email

FOOTER

Gambar 3. 33 Tampilan *Form* Pendaftaran
(Sumber: Data Penelitian 2019)

SISTEM PAKAR	Halaman Utama	<u>Diagnosa Kerusakan</u>	Daftar Kerusakan	Bantuan	PAKAR
---------------------	---------------	---------------------------	------------------	---------	-------

JAWABLAH PERTANYAAN BERIKUT :

PERTANYAAN?

Benar (YA) Salah (TIDAK)

FOOTER

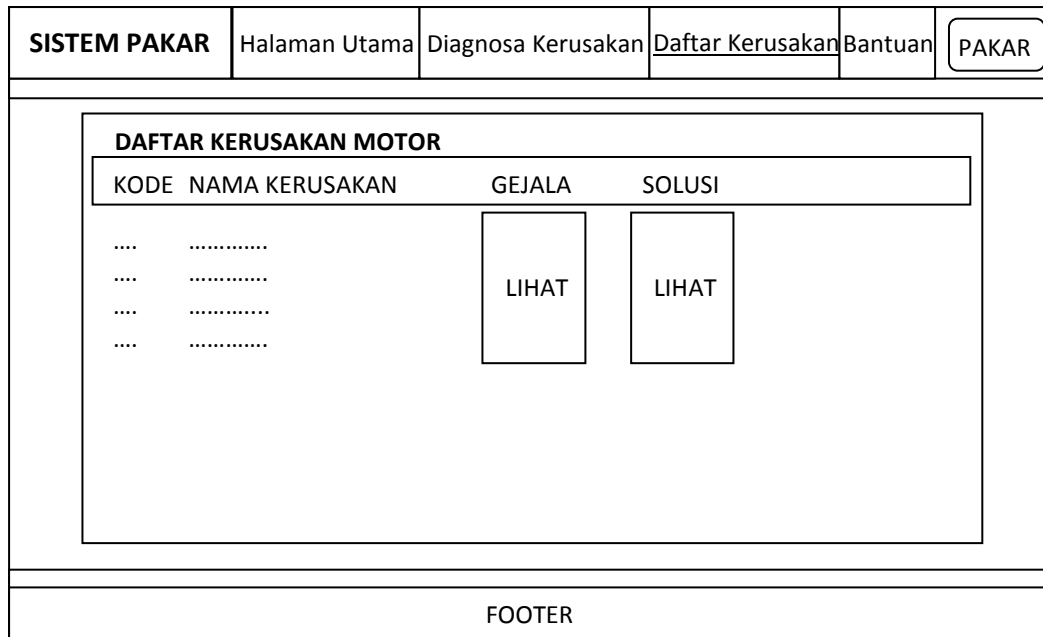
Gambar 3. 34 Tampilan Halaman Diagnosa Kerusakan
(Sumber: Data Penelitian 2019)

SISTEM PAKAR	Halaman Utama	<u>Diagnosa Kerusakan</u>	Daftar Kerusakan	Bantuan	PAKAR
HASIL KEMUNGKINAN DIAGNOSA KERUSAKAN					
DATA USER :					
Nama					
Jenis Kelamin					
Alamat					
Email					
HISTORI JAWABAN :					
HASIL ANALISA TERAKHIR :					
SOLUSI :					
					CETAK
FOOTER					

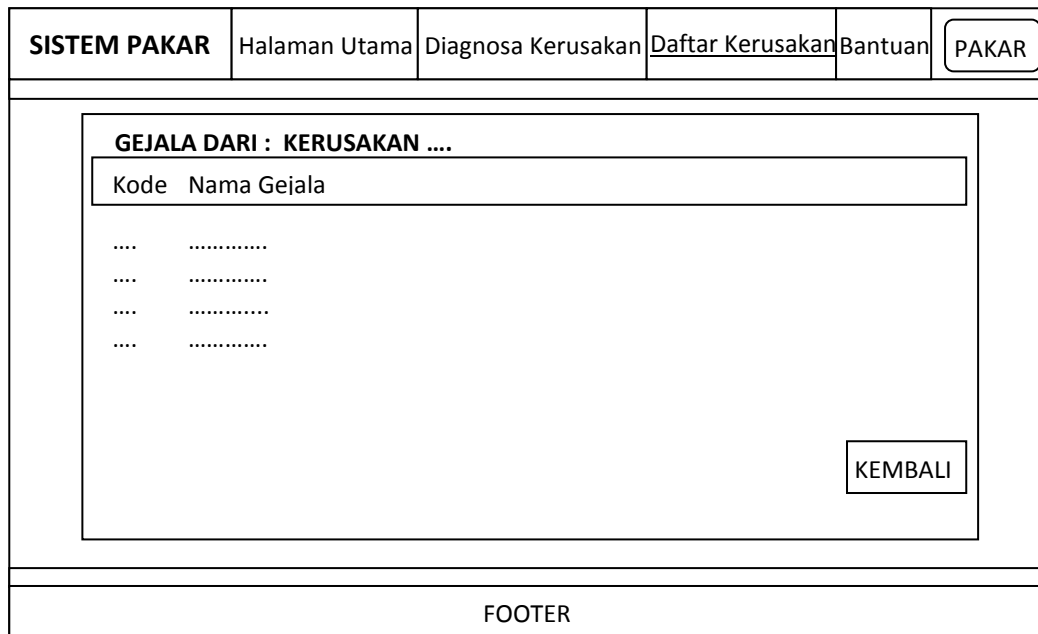
Gambar 3. 35 Tampilan Hasil Diagnosa
(Sumber: Data Penelitian 2019)

4. Tampilan Halaman Daftar Kerusakan

Halaman ini menampilkan jenis-jenis kerusakan mesin motor yang disertai dengan gejala dan solusi perbaikannya. Untuk lebih jelasnya sebagai berikut.



Gambar 3. 36 Tampilan Halaman Daftar Kerusakan
(Sumber: Data Penelitian 2019)



Gambar 3. 37 Tampilan Gejala dari Kerusakan
(Sumber: Data Penelitian 2019)

SISTEM PAKAR	Halaman Utama	Diagnosa Kerusakan	Daftar Kerusakan	Bantuan	PAKAR														
<table border="1"> <tr> <td colspan="2">SOLUSI DARI : KERUSAKAN</td> </tr> <tr> <td>Kode</td> <td>Nama Solusi</td> </tr> <tr> <td>....</td> <td>.....</td> </tr> <tr> <td>....</td> <td>.....</td> </tr> <tr> <td>....</td> <td>.....</td> </tr> <tr> <td>....</td> <td>.....</td> </tr> <tr> <td colspan="2" style="text-align: right;">KEMBALI</td> </tr> </table>						SOLUSI DARI : KERUSAKAN		Kode	Nama Solusi	KEMBALI	
SOLUSI DARI : KERUSAKAN																			
Kode	Nama Solusi																		
....																		
....																		
....																		
....																		
KEMBALI																			
FOOTER																			

Gambar 3. 38 Tampilan Solusi dari Kerusakan
(Sumber: Data Penelitian 2019)

5. Tampilan Halaman Bantuan

Halaman ini akan menampilkan informasi tentang kontak yang bisa dihubungi jika *user* mengalami masalah saat mengakses sistem pakar.

SISTEM PAKAR	Halaman Utama	Diagnosa Kerusakan	Daftar Kerusakan	Bantuan	PAKAR	
<table border="1"> <tr> <td style="height: 150px; vertical-align: top;"> <div style="border: 1px solid black; width: 100%; height: 100%;"></div> </td> </tr> </table>						<div style="border: 1px solid black; width: 100%; height: 100%;"></div>
<div style="border: 1px solid black; width: 100%; height: 100%;"></div>						
FOOTER						

Gambar 3. 39 Tampilan Halaman Bantuan
(Sumber: Data Penelitian 2019)

6. Tampilan *Form Login Admin*

Menampilkan *form login* untuk masuk dan mengakses halaman *admin* sehingga bisa melakukan *update* data. Tampilannya sebagai berikut.

SISTEM PAKAR	<input type="text" value="Username"/>	<input type="text" value="Password"/>	<input type="button" value="LOGIN"/>	<input type="button" value="PAKAR"/>
FOOTER				

Gambar 3. 40 Tampilan *Form Login Admin*
(Sumber: Data Penelitian 2019)

7. Tampilan Halaman Beranda *Admin*

Halaman ini yang pertama kali akan muncul setelah *admin* melakukan *login* dengan benar. Terdapat banyak pilihan menu yang bisa diakses oleh *admin* untuk meng*update* data. Untuk lebih jelasnya bisa dilihat pada gambar-gambar berikut.

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td><u>Beranda</u></td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	<u>Beranda</u>	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<p>SELAMAT DATANG ADMIN !</p> <p>.....</p>	
<u>Beranda</u>									
Laporan Kerusakan									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 41 Tampilan Halaman Beranda *Admin*
(Sumber: Data Penelitian 2019)

8. Tampilan Halaman Laporan Kerusakan

Halaman ini akan menampilkan semua data kerusakan beserta gejala dan solusi perbaikannya. Berikut adalah tampilan-tampilannya.

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td>Beranda</td></tr> <tr><td><u>Laporan Kerusakan</u></td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	<u>Laporan Kerusakan</u>	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<p>DAFTAR SEMUA KERUSAKAN</p> <p>Kode Nama Kerusakan Gejala <input type="button" value="Lihat Gejala"/> Solusi <input type="button" value="Lihat Solusi"/></p> <p style="text-align: right;"><input type="button" value="Selanjutnya"/></p>	
Beranda									
<u>Laporan Kerusakan</u>									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 42 Tampilan Halaman Laporan Kerusakan
(Sumber: Data Penelitian 2019)

Administrator / Pakar		AKUN ▾																			
<table border="1"> <tr><td>Beranda</td></tr> <tr><td><u>Laporan Kerusakan</u></td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	<u>Laporan Kerusakan</u>	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<p>DAFTAR GEJALA : KERUSAKAN</p> <table border="1"> <thead> <tr> <th>No</th> <th>Kode</th> <th>Nama Gejala</th> </tr> </thead> <tbody> <tr><td>1</td><td>....</td><td>.....</td></tr> <tr><td>2</td><td>....</td><td>.....</td></tr> <tr><td>3</td><td>....</td><td>.....</td></tr> </tbody> </table> <p style="text-align: right;"><input type="button" value="Kembali"/></p>		No	Kode	Nama Gejala	1	2	3
Beranda																					
<u>Laporan Kerusakan</u>																					
Relasi Gejala																					
Relasi Solusi																					
Kerusakan ▾																					
Gejala ▾																					
Solusi ▾																					
No	Kode	Nama Gejala																			
1																			
2																			
3																			

Gambar 3. 43 Tampilan Gejala dari Kerusakan
(Sumber: Data Penelitian 2019)

Administrator / Pakar		AKUN ▾																			
<table border="1"> <tr><td>Beranda</td></tr> <tr><td><u>Laporan Kerusakan</u></td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	<u>Laporan Kerusakan</u>	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<p>DAFTAR SOLUSI : KERUSAKAN</p> <table border="1"> <thead> <tr> <th>No</th> <th>Kode</th> <th>Nama Solusi</th> </tr> </thead> <tbody> <tr><td>1</td><td>....</td><td>.....</td></tr> <tr><td>2</td><td>....</td><td>.....</td></tr> <tr><td>3</td><td>....</td><td>.....</td></tr> </tbody> </table> <p style="text-align: right;"><input type="button" value="Kembali"/></p>		No	Kode	Nama Solusi	1	2	3
Beranda																					
<u>Laporan Kerusakan</u>																					
Relasi Gejala																					
Relasi Solusi																					
Kerusakan ▾																					
Gejala ▾																					
Solusi ▾																					
No	Kode	Nama Solusi																			
1																			
2																			
3																			

Gambar 3. 44 Tampilan Solusi dari Kerusakan
(Sumber: Data Penelitian 2019)

9. Tampilan Halaman Relasi Gejala

Halaman ini akan menampilkan relasi antara gejala dan kerusakan. *Admin* bisa melakukan perubahan jika diperlukan. Tampilannya adalah sebagai berikut.

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td>Beranda</td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<div style="border: 1px solid black; padding: 5px;"> <p>RELASI GEJALA DAN KERUSAKAN</p> <p>Nama Kerusakan : <input style="width: 100%;" type="text"/></p> <p>[Daftar Kerusakan] ▾</p> <p>Daftar Gejala :</p> <p>1 <input type="checkbox"/> 2 <input type="checkbox"/> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> SIMPAN RELASI NORMALKAN </div> </p></div>	
Beranda									
Laporan Kerusakan									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 45 Tampilan Halaman Relasi Gejala
(Sumber: Data Penelitian 2019)

10. Tampilan Halaman Relasi Solusi

Halaman ini akan menampilkan relasi antara solusi dan kerusakan. *Admin* bisa melakukan perubahan jika diperlukan. Tampilannya adalah sebagai berikut.

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td>Beranda</td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<div style="border: 1px solid black; padding: 5px;"> <p>RELASI SOLUSI DAN KERUSAKAN</p> <p>Nama Kerusakan : <input style="width: 100%;" type="text"/></p> <p>[Daftar Kerusakan] ▾</p> <p>Daftar Solusi :</p> <p>1 <input type="checkbox"/> 2 <input type="checkbox"/> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> SIMPAN RELASI NORMALKAN </div> </p></div>	
Beranda									
Laporan Kerusakan									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 46 Tampilan Halaman Relasi Solusi
(Sumber: Data Penelitian 2019)

11. Tampilan Halaman Data Kerusakan

Halaman ini akan menampilkan data kerusakan yang memungkinkan *admin* untuk melakukan *update* data seperti ubah data, hapus data, dan tambah data. Untuk lebih jelasnya bisa dilihat pada gambar-gambar berikut.

Gambar 3. 47 Tampilan Halaman Data Kerusakan
(Sumber: Data Penelitian 2019)

Gambar 3. 48 Tampilan Halaman Ubah Data Kerusakan
(Sumber: Data Penelitian 2019)

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td>Beranda</td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<div style="border: 1px solid black; padding: 5px;"> <p>TAMBAH DATA KERUSAKAN</p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">TAMBAH DATA KERUSAKAN</div> <p>Kode <input style="width: 50px;" type="text"/></p> <p>Nama Kerusakan <input style="width: 150px;" type="text"/></p> <p>Keterangan <input style="width: 150px; height: 20px;" type="text"/></p> <div style="display: flex; justify-content: flex-end; gap: 10px;"> <input type="button" value="SIMPAN"/> <input type="button" value="RESET"/> </div> </div>	
Beranda									
Laporan Kerusakan									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 49 Tampilan Halaman Tambah Data Kerusakan
(Sumber: Data Penelitian 2019)

12. Tampilan Halaman Data Gejala

Halaman ini akan menampilkan data gejala yang memungkinkan *admin* untuk melakukan *update* data seperti ubah data, hapus data, dan tambah data. Untuk lebih jelasnya bisa dilihat pada gambar-gambar berikut.

Administrator / Pakar		AKUN ▾																											
<table border="1"> <tr><td>Beranda</td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<div style="border: 1px solid black; padding: 5px;"> <p>DATA GEJALA</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Kode</th> <th>Nama Gejala</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>....</td> <td>.....</td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>....</td> <td>.....</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>....</td> <td>.....</td> <td></td> <td></td> </tr> </tbody> </table> <div style="display: flex; justify-content: flex-end; gap: 10px; margin-top: 10px;"> <input type="button" value="UBAH"/> <input type="button" value="HAPUS"/> </div> </div>		No	Kode	Nama Gejala			1			2			3		
Beranda																													
Laporan Kerusakan																													
Relasi Gejala																													
Relasi Solusi																													
Kerusakan ▾																													
Gejala ▾																													
Solusi ▾																													
No	Kode	Nama Gejala																											
1																											
2																											
3																											

Gambar 3. 50 Tampilan Halaman Data Gejala
(Sumber: Data Penelitian 2019)

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td>Beranda</td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<h3>UBAH DATA GEJALA</h3> <p>Kode <input type="text"/></p> <p>Nama Gejala <input type="text"/></p> <p style="text-align: center;"> <input type="button" value="SIMPAN"/> <input type="button" value="RESET"/> </p>	
Beranda									
Laporan Kerusakan									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 51 Tampilan Halaman Ubah Data Gejala
(Sumber: Data Penelitian 2019)

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td>Beranda</td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<h3>TAMBAH DATA GEJALA</h3> <p>Kode <input type="text"/></p> <p>Nama Gejala <input type="text"/></p> <p style="text-align: center;"> <input type="button" value="SIMPAN"/> <input type="button" value="RESET"/> </p>	
Beranda									
Laporan Kerusakan									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 52 Tampilan Halaman Tambah Data Gejala
(Sumber: Data Penelitian 2019)

13. Tampilan Halaman Data Solusi

Halaman ini akan menampilkan data solusi yang memungkinkan *admin* untuk melakukan *update* data seperti ubah data, hapus data, dan tambah data. Untuk lebih jelasnya bisa dilihat pada gambar-gambar berikut.

Administrator / Pakar		AKUN ▾												
<ul style="list-style-type: none"> Beranda Laporan Kerusakan Relasi Gejala Relasi Solusi Kerusakan ▾ Gejala ▾ Solusi ▾ 	<p>DATA SOLUSI</p> <table border="1"> <thead> <tr> <th>No</th> <th>Kode</th> <th>Nama Solusi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>....</td> <td>....</td> </tr> <tr> <td>2</td> <td>....</td> <td>....</td> </tr> <tr> <td>3</td> <td>....</td> <td>....</td> </tr> </tbody> </table> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="UBAH"/> <input type="button" value="HAPUS"/> </div>		No	Kode	Nama Solusi	1	2	3
No	Kode	Nama Solusi												
1												
2												
3												

Gambar 3. 53 Tampilan Halaman Data Solusi
(Sumber: Data Penelitian 2019)

Administrator / Pakar		AKUN ▾
<ul style="list-style-type: none"> Beranda Laporan Kerusakan Relasi Gejala Relasi Solusi Kerusakan ▾ Gejala ▾ Solusi ▾ 	<p>UBAH DATA SOLUSI</p> <p>Kode <input type="text"/></p> <p>Nama Solusi <input type="text"/></p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="SIMPAN"/> <input type="button" value="RESET"/> </div>	

Gambar 3. 54 Tampilan Halaman Ubah Data Solusi
(Sumber: Data Penelitian 2019)

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td>Beranda</td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<div style="border: 1px solid black; padding: 10px;"> <p>TAMBAH DATA SOLUSI</p> <p>Kode <input type="text"/></p> <p>Nama Solusi <input type="text"/></p> <p style="text-align: right;"> <input type="button" value="SIMPAN"/> <input type="button" value="RESET"/> </p> </div>	
Beranda									
Laporan Kerusakan									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 55 Tampilan Halaman Tambah Data Solusi
(Sumber: Data Penelitian 2019)

14. Tampilan Halaman Ubah *Password*

Halaman ini akan menampilkan *form* yang harus diisi *admin* jika ingin mengubah *username* dan *password admin* untuk digunakan saat *login*. Berikut ini adalah tampilannya.

Administrator / Pakar		AKUN ▾							
<table border="1"> <tr><td>Beranda</td></tr> <tr><td>Laporan Kerusakan</td></tr> <tr><td>Relasi Gejala</td></tr> <tr><td>Relasi Solusi</td></tr> <tr><td>Kerusakan ▾</td></tr> <tr><td>Gejala ▾</td></tr> <tr><td>Solusi ▾</td></tr> </table>	Beranda	Laporan Kerusakan	Relasi Gejala	Relasi Solusi	Kerusakan ▾	Gejala ▾	Solusi ▾	<div style="border: 1px solid black; padding: 10px;"> <p>UBAH PASSWORD</p> <p>Masukkan Username dan Password baru</p> <p><input type="text" value="admin"/></p> <p><input type="text" value="Password Baru"/></p> <p><input type="text" value="Konfirmasi Password Baru"/></p> <p style="text-align: right;"> <input type="button" value="SIMPAN"/> <input type="button" value="RESET"/> </p> </div>	
Beranda									
Laporan Kerusakan									
Relasi Gejala									
Relasi Solusi									
Kerusakan ▾									
Gejala ▾									
Solusi ▾									

Gambar 3. 56 Tampilan Halaman *Form* Ubah *Password*
(Sumber: Data Penelitian 2019)

3.5 Lokasi dan Jadwal Penelitian

Penelitian ini dilakukan di Dealer Resmi Honda Senturi Ultra Dinamis yang bertempat di Ruko Tiban Impian Blok A2 No. 12A. Tiban Baru, Sekupang, Kota Batam. Pelaksanaan Penelitian pada semester genap 2018/2019 yaitu antara bulan April 2019 sampai dengan bulan Mei 2019 dengan alokasi waktu sebagai berikut:

Tabel 3. 21 Tabel Jadwal Penelitian

Kegiatan	Tahun 2019					
	Maret 2019	April 2019	Mei 2019	Juni 2019	Juli 2019	Agustus 2019
Studi Literatur						
Pengumpulan Data dan Analisis Data						
Perancangan Sistem						
Pembuatan Program						
Pengujian Sistem						
Penyusunan Laporan						

(Sumber: Data Penelitian 2019)