

**PERANGKAT LUNAK DIAGNOSA KERUSAKAN SEPEDA MOTOR
BERBASIS SISTEM PAKAR**

(Skripsi)

Oleh

RONI YANUAR NAINGGOLAN



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2015**

ABSTRACT

**MOTORCYCLE BROCKEN DIAGNOSTIC SOFTWARE BASED ON
EXPERT SYSTEM**

By

RONI YANUAR NAINGGOLAN

The system of brocken diagnose expert can be accessed through Personal Computer (PC) or Netbook so that making it easier for people to diagnose the brocken of Honda motorcycle. This expert system made with programming language Amzi Prolog. This thesis will discuss about how the mechanic who adopted the overall expert brocked system, the basic theory, analysist stage, design, coding, testing and maintenace the software on the use of this system. This system can be run well and according to the system user this is needed. According to user, the result which showed by this system is accurate enough.

Key words : Prolog (Program Logic) Language, Expert System

ABSTRAK

PERANGKAT LUNAK DIAGNOSA KERUSAKAN SEPEDA MOTOR BERBASIS SISTEM PAKAR

Oleh

RONI YANUAR NAINGGOLAN

Sistem pakar pendiagnosa kerusakan ini dapat diakses melalui *Personal Computer* (PC) maupun *Netbook* sehingga sangat memudahkan orang dalam mendiagnosa kerusakan sepeda motor Honda. Sistem pakar ini dibuat dengan menggunakan bahasa pemrograman Amzi Prolog. Skripsi ini akan membahas tentang cara kerja mekanik yang diadopsi pada sistem pakar kerusakan secara keseluruhan, dasar teori yang digunakan, tahap analisa, perancangan (desain), pengodean, pengujian, dan *maintenance* (pemeliharaan) perangkat lunak (*Software*) pada penggunaan sistem ini. Sistem ini dapat dijalankan dengan baik dan menurut tanggapan pengguna sistem ini diperlukan. Berdasarkan tanggapan pengguna, hasil diagnosa yang diberikan oleh sistem ini sudah cukup akurat.

Kata kunci: Bahasa Prolog (*Program Logic*), berbasis sistem pakar

**PERANGKAT LUNAK DIAGNOSA KERUSAKAN SEPEDA MOTOR
BERBASIS SISTEM PAKAR**

(Skripsi)

Oleh

RONI YANUAR NAINGGOLAN

Skripsi

**Sebagai salah satu syarat untuk mencapai gelar
SARJANA TEKNIK**

Pada

Jurusan Teknik Elektro



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2015**

**Judul Skripsi : PERANGKAT LUNAK DIAGNOSA
KERUSAKAN SEPEDA MOTOR
BERBASIS SISTEM PAKAR**

Nama Mahasiswa : Roni Yanuar Nainggolan

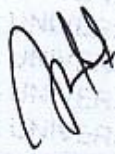
Nomor Pokok Mahasiswa : 0815031098

Program Studi : Teknik Elektro

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing



Yessi Mulyani, S.T., M.T.
NIP 19731226 200012 2 001



Ing. Hery Dian Septama, S.T.
NIP 19850915 200812 1 001

2. Ketua Jurusan Teknik Elektro

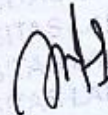


Dr. Ardian Ulvan, S.T., M.Sc.
NIP 19731128 199903 1 005

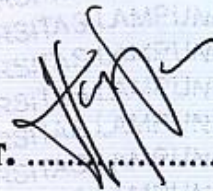
MENGESAHKAN

1. Tim Penguji

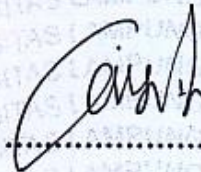
Ketua : Yessi Mulyani, S.T., M.T.



Sekretaris : Ing. Hery Dian Septama, S.T.



Penguji Bukan Pembimbing : Gigh Forda Nama, S.T., M.T.



2. Dekan Fakultas Teknik



Prof. Suharno, M.Sc., Ph.D.
NIP 19620717 198703 1 002

Tanggal Lulus Ujian Skripsi : 23 Desember 2015


SURAT PERNYATAAN

Dengan ini menyatakan bahwa skripsi ini dibuat oleh Saya sendiri. Adapun karya orang lain yang terdapat dalam skripsi ini telah dicantumkan sumbernya pada daftar pustaka.

Apabila pernyataan saya tidak benar maka saya bersedia dikenakan sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, 10 Januari 2016




Roni Yanuar Nainggolan
NPM.0815031098

RIWAYAT HIDUP



Penulis dilahirkan di Bandar Lampung Provinsi Lampung pada tanggal 10 Januari 1989, sebagai anak keempat dari empat saudara, dari Bapak A. Ranto Nainggolan dan Ibu Khatijah Br. Manihuruk.

Riwayat pendidikan penulis dimulai dari sekolah dasar diselesaikan di SD Negeri 3

Labuhan Dalam Kecamatan Tanjung Senang Kota Bandar Lampung pada tahun 2001, Sekolah Menengah Pertama di SLTP Negeri 20 Bandar Lampung diselesaikan pada tahun 2004, dan Sekolah Menengah Kejuruan di SMK Negeri 2 Bandar Lampung diselesaikan pada tahun 2007.

Pada tahun 2008, Penulis terdaftar sebagai mahasiswa Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung melalui jalur Ujian SPMB 2008. Pada tahun 2014, penulis melaksanakan kerja praktik di PT. Sampoerna Telekomunikasi Indonesia (Ceria) area Lampung Jl. Sriwijaya Enggal Bandar Lampung pada 15 September 2014 sampai dengan 15 Oktober 2014.

MOTO

"Kuliah dan menyelesaikan Skripsi itu ibadah, gak boleh ngeluh."

" Untuk melakukan segala sesuatu, semangat aja tidak cukup, harus punya nyali. *Keep Rock n Roll* "

"Kebahagiaan itu dekat dengan rasa syukur. Maka semakin jauh syukurnya, semakin jauh pula bahagiannya."

quotes:

-RONI YANUAR NAINGGOLAN, S.T-

-----Bissmillahirohmannirohim-----

DENGAN BANGGA
KU PERSEMBAHKAN KARYA
SEDERHANA INI
UNTUK

AYAHANDA dan IBUNDA TERCINTA
A. RANTO NAINGGOLAN dan
KHATIJAH Br. MANIHURUK

Serta

ABANG dan KAKAK-KAKAK KU
TERCINTA
RANTO HAMDANI NAINGGOLAN,
YULIANA NAINGGOLAN, dan
RINAWATI NAINGGOLAN.

SANWACANA

Alhamdulillahirobbil'alamin, segala puji bagi Allah SWT atas limpahan nikmat kesehatan, kesempatan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini. Sholawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW sang penutup para Nabi dan Rasul, kepada keluarga, sahabat, dan pengikutnya yang setia sampai akhir zaman.

Skripsi dengan judul "**Perangkat Lunak Diagnosa Kerusakan Sepeda Motor Berbasis Sistem Pakar**" sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.

Dalam kesempatan ini penulis mengucapkan terima kasih yang tulus kepada :

1. Bapak Prof. Suharno, M.Sc., Ph.D., selaku dekan Fakultas Teknik Universitas Lampung.
2. Bapak Dr. Ardian Ulvan, S.T, M. Sc., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Bapak Dr. Herman H Sinaga , selaku Sekretaris Jurusan Teknik Elektro Universitas Lampung.
4. Bapak Dr. Eng. Lukmanul Hakim, S.T., M.T., selaku Pembimbing Akademik.

5. Ibu Yessi Mulyani, S.T., M.T., sebagai Dosen Pembimbing Utama atas kesediaannya membimbing, membantu, meluangkan waktu serta memberikan saran dan kritik kepada penulis dalam penyelesaian penulisan skripsi ini.
6. Bapak Ing. Hery Dian Septama, S.T., sebagai Dosen Pembimbing Pendamping atas kesediaannya membimbing, membantu, meluangkan waktu serta memberikan saran dan kritik kepada penulis dalam penyelesaian penulisan skripsi ini.
7. Bapak Gigih Forda Nama, S.T., M.T., selaku Dosen Penguji yang telah memberikan kritikan, dan saran dalam penyelesaian tugas akhir ini.
8. Segenap Dosen dan Pegawai di Jurusan Teknik Elektro yang telah memberikan ilmu dan wawasan yang tak terlupakan oleh penulis.
9. Bapak dan Emak tercinta, A. Ranto Nainggolan dan Khatijah Br. Manihuruk
10. Abang dan kakak-kakakku Ranto Hamdani Nainggolan, Yuliana Nainggolan dan Rinawati Nainggolan yang selama ini telah memberikan kasih sayang , semangat, doa, nasihat serta dukungan.
11. Kakak ipar Bang Rusdiansyah dan Mas Pandu, serta Ibu Dewi Hartati yang telah memberikan bantuan doa, semangat dan suportnya.
12. Orang terdekat yang udah setia menemani dari kuliah sampai skripsi selesai dikerjakan Nur Khotimah.
13. Rekan 2007 Bang Agung Tri Ilhami terima kasih atas segala bantuan, semangat, suport dan kebersamaannya.

14. Rekan Elektro 2008 Dedi Kurniawan dan Taufik Rangkuti Munandar, yang telah bersama hingga penulis dapat selesai sampai titik penghabisan walaupun kalian lebih dulu meninggalkan Elektro Unila karena memang sudah menjadi pilihannya. Kalian Luar Biasa.
15. Rekan Elektro 2008 Hasron, Endi Azrofata, Riadi Amarta dan Andre Dwi Setiawan yang telah duluan Wisuda, terima kasih atas segala masukan, doa, dan suportnya.
16. Para Pejuang Skripsi Teknik Elektro angkatan 2008 : Adam Hussein, Aferdi Siswa, Yustinus Sinaga, Dapot Tua Malau, M. Cahyadi, M. Zainal Abidin, Wan Novri S, Herdiawan Yudistira, Felix Manahan Abet Nego Nora Aditian, Mipdeka, Dimas Angga, Rahmatulloh, Kholil Arifuddin, Arya Wiguna Bangun, Aris Susilo, Ujang Faturrohman, Ayub Wisesa, dan M. Syuhada atas kebersamaan yang selama ini kita lalui. Semoga kebersamaan ini akan tetap terjaga selamanya. Kita Luar Biasa...!!
17. Rekan angkatan 2009 Dedi Irawan (Botoy), M.Rifqi (Mbew) dan adik tingkat 2011 Frisky Volino Andreas atas kebersamaan yang selama ini kita lalui. Semoga kebersamaan ini akan tetap terjaga selamanya. Kita Luar Biasa...!!
18. Seluruh penghuni Laboratorium Terpadu Teknik Elektro khususnya Laboratorium Teknik Kendali, Midia Haki, Teguh Firman Ardiansyah, Khairul Anwar dan lain-lain atas bantuannya baik langsung ataupun tidak langsung.
19. Para Yuniior Elektro Universitas Lampung yang telah memberikan doa dan suportnya.

20. Teman-teman dekat Tirta Ramadhani Akbar Syarif, Septiawan Wagenata, Febri Teja Dikatar, Meilando Rahmat Hidayat, Zilvania Zuita, dan Rindang Andam Suri yang telah memberikan semangat, doa dan suportnya, kalian semua juga Luar Biasa.
21. Para penghuni rumah klasik Arin Susanto, Beni Yuliansyah dan Bang Suman Jaya, yang selalu bersama di basecamp rumah klasik memberikan semangat, dan doa, kalian luar biasa.
22. Bengkel AMS GRAGAZZ yang di pegang oleh kak Septa Jenia dan Bengkel CHICKEN RACING TEAM yang dipegang oleh Didi Wahyu Prasetyo yang telah bersedia untuk penelitiannya skripsi ini hingga selesai.
23. Semua pihak yang tidak dapat penulis sebutkan satu per satu atas bantuan dan dukungannya dalam menyelesaikan penyusunan skripsi ini.

Penulis meminta maaf atas segala kesalahan dan ketidaksempurnaan dalam penyusunan tugas akhir ini. Saran dan kritik membangun sangat diharapkan penulis demi kebaikan dimasa yang akan datang. Sekali lagi penulis ucapkan terimakasih dan semoga Allah SWT membalas kebaikan Anda semua dan diberi kemudahan dalam segala urusannya. Amin.

Bandar Lampung, 10 Januari 2015

Penulis,

Roni Yanuar Nainggolan

DAFTAR ISI

	Halaman
DAFTAR ISI	i
DAFTAR TABEL	iii
DAFTAR GAMBAR	iv
I. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Penelitian	4
1.3. Manfaat Penelitian	5
1.4. Rumusan Masalah	5
1.5. Batasan Masalah	6
1.6. Sistematika Penulisan	6
II. TINJAUAN PUSTAKA	6
2.1. Sistem Diagnosa Kerusakan berbasis Perangkat Lunak	8
2.2. Cara Diagnosa Mekanik	9
2.2.1 Pengecekan Fisik	10
2.2.2 Pengecekan Penunjang	10
2.3. Ilmu Kecerdasan Buatan	10
2.3.1 Sistem Pakar	11

2.3.2	Pencarian Heuristik	13
2.4.	Rekayasa Perangkat Lunak	14
2.4.1	Model rekayasa Perangkat Lunak (<i>Modified Waterfall Model</i>)	14
2.4.2	<i>Data Flow Diagram</i>	16
2.5.	Aplikasi Pendukung	17
2.5.1	Sekilas Mengenai Turbo Prolog.....	17
2.5.2	Sejarah Singkat Prolog.....	18
2.5.3	Pemrograman Logika.....	19
2.5.4	Ciri Bahasa Prolog	21
2.5.5	Bahasa Deklaratif.....	27
2.5.6	Prolog dan Proyek Komputer Generasi Kelima.....	28
2.5.7	Turbo Prolog dan Prolog Tradisional.....	29
2.5.8	Penerapan Prolog	30
III.	METODE PENELITIAN	34
3.1.	Bagan Alir dan Penelitian	34
3.2.	Jadwal Kegiatan	35
3.2.	Waktu dan Tempat	36
3.3.	Alat dan Bahan	36
3.4.	Studi Literatur.....	36
3.5.	Gambaran Umum	44
IV.	HASIL DAN PEMBAHASAN	45
4.1.	Hasil Implementasi Data Ahli Sistem	48
4.2.	Menu Diagnosa	47

1. Pembuka.....	47
2. Pengolah Pertanyaan dan Basis Data	49
3. Pengambil Keputusan.....	51
4.3. Halaman Antar Muka	53
4.4. Hasil Pengujian Perangkat Lunak Diagnosa Kerusakan	56
4.4.1. Pengujian Kerusakan Busi	56
4.4.2. Pengujian Kerusakan Karburator	56
4.4.3. Pengujian Kerusakan Klep	57
4.5.4. Pengujian Kerusakan Sentrik	57
4.5. <i>Functionality Test</i>	58
4.6. <i>User Acceptance Test</i>	59
V. SIMPULAN DAN SARAN	61
5.1. Simpulan.....	61
5.2. Saran	62
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR TABEL

Tabel	Halaman
3.3 Daftar <i>attribute</i> beserta kuncinya	41
4.1 Data Kerusakan Sepeda Motor Bagian Pelumasan	45
4.2 <i>Functionality Test</i>	58

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 Bagian Sistem Pakar	13
Gambar 2.2 <i>Modified Waterfall Model</i>	15
Gambar 2.3 <i>Simbol Data Flow Diagram</i>	16
Gambar 3.1 Alur Penelitian	34
Gambar 3.2 <i>Modified Waterfall Model</i>	37
Gambar 3.3 <i>Data Flow Diagram</i>	38
Gambar 3.4 DFD Level 1	39
Gambar 3.5 Diagram ER	42
Gambar 3.6 Gambaran Umum Sistem	44
Gambar 4.1 Kode Program Basis Data Kerusakan Bagian Pelumasan	46
Gambar 4.2 Kode Program Menu Pembuka	48
Gambar 4.3 Kode Program Pengolah Pertanyaan dari Basis Data	50
Gambar 4.4 Kode Program Pengambil Keputusan	52
Gambar 4.5 Halaman Antar Muka Sistem Pakar Amzy Prolog	53
Gambar 4.6 Halaman Pembuka Sistem Pakar	54
Gambar 4.7 Sistem Pakar Memulai Diagnosa Kerusakan	55
Gambar 4.8 Sistem Pakar Berhasil Menemukan Kerusakan	55
Gambar 4.9 Sistem Pakar Tidak Menemukan Kerusakan	55

Gambar 4.10 Pengujian Kerusakan Busi	56
Gambar 4.11 Pengujian Kerusakan Karburator	56
Gambar 4.12 Pengujian Kerusakan Klep	57
Gambar 4.13 Pengujian Kerusakan Sentrik	57

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Sepeda motor adalah salah satu kendaraan transportasi darat beroda dua yang banyak disukai oleh kalangan menengah kebawah. *Distributor* sepeda motor berkembang sangat pesat di Indonesia, salah satu *distributor* sepeda motor di Indonesia adalah Honda. Sepeda motor Honda adalah salah satu yang digemari oleh masyarakat Indonesia, karena sepeda motor Honda penggunaanya yang mudah, harga terjangkau dan hemat bahan bakar.

Jumlah permintaan sepeda motor Honda di Indonesia sangat berkembang pesat hingga 40 juta unit pada tahun 2013. Perusahaan motor Honda yang di naungi oleh PT Astra Honda Motor (AHM) mempunyai 1.800 *showroom* penjualan, 3.600 layanan *service* atau bengkel AHASS (Astra Honda *Authorized Service Station*), serta 7.550 gerai suku cadang, yang siap melayani jutaan penggunaan sepeda motor Honda di seluruh Indonesia ^[1].

Banyaknya pembelian sepeda motor Honda di Indonesia yang meningkat tiap tahun tidak seimbang dengan adanya layanan *service* atau bengkel resmi Honda. Sedangkan pelanggan sedikit tidak percaya terhadap layanan bengkel biasa (tidak

resmi) dikarenakan penggantian suku cadang yang bukan asli dari Honda, dan mekanik/teknisi jarang memberi informasi detail tentang kerusakan sepeda motor, jenis kerusakan dan cara memperbaikinya. Dengan banyaknya pengguna sepeda motor, tetapi layanan *service* atau bengkel resmi yang tidak banyak di Indonesia, sehingga di butuhnya teknologi informasi agar dapat memudahkan para pengguna motor Honda mengetahui tentang kerusakan pada sepeda motor mereka. Salah satu produk teknologi informasi adalah sistem pakar, sistem yang dapat membantu mekanik atau pengguna motor dalam menyelesaikan masalah kerusakan motor.

Setiap orang selalu mendambakan kendaraan sepeda motor yang sehat tanpa ada kerusakan. Permasalahan kerusakan adalah hal yang vital, karena merupakan modal utama dalam berkendara sehari-hari, misalnya berangkat kerja, berangkat kuliah, dan keperluan lainnya. Disisi lain, perkembangan di dunia teknologi juga telah berkembang secara pesat dalam beberapa tahun terakhir. Untuk itu diperlukannya Perangkat Lunak Diagnosa Kerusakan Sepeda Motor Berbasis Sistem Pakar.

Seiring dengan kemajuan dan berkembangnya teknologi informasi serta komunikasi pada saat ini, maka kebutuhan akan informasi yang cepat dan akurat sudah menjadi kebutuhan yang utama bagi setiap insan. Keadaan yang seperti inilah yang telah mendorong para ahli untuk mengembangkan teknologi dan mengadopsi proses serta cara berfikir komputer seperti manusia.

Hal ini dapat diwujudkan dengan cara menerapkan ilmu *Artificial Intelligence* (kecerdasan buatan) dengan membuat *Expert System* (sistem pakar) yang di dalamnya memuat informasi tentang kerusakan kendaraan roda dua pada sepeda motor. Semakin majunya peradaban maka himpitan kesibukan setiap manusia akan semakin bertambah. Oleh sebab itu diperlukan suatu sistem yang praktis bagi setiap orang, sehingga mereka yang ingin menggunakan suatu layanan dapat dengan mudah melakukannya. Suatu perangkat lunak dapat mengatasi kendala ini karena menawarkan kepraktisan yang dapat memanjakan penggunanya.

Salah satu pelayanan yang mungkin dapat ditawarkan kepada masyarakat adalah sebuah pelayanan perbaikan sepeda motor berbasis sistem informasi. Hal ini tentu akan mempermudah masyarakat untuk melakukan konsultasi kerusakan, dan pastinya hal ini juga akan meringankan biaya yang perlu dikeluarkan apabila ingin melakukan konsultasi perbaikan kerusakan sepeda motor. Terobosan ini juga dapat menjadi pemecah masalah dalam menghadapi tantangan wilayah Indonesia yang luas dengan topografi yang sulit dijangkau. Sehingga dengan adanya pelayanan kerusakan berbasis sistem informasi ini dapat menghadirkan layanan kerusakan untuk masyarakat diseluruh Nusantara. Walaupun demikian dapat diketahui bahwa belum seluruh rakyat Indonesia yang sadar dan memanfaatkan teknologi informasi berbasis sistem informasi, tetapi kiranya terobosan ini dapat dimanfaatkan pada seluruh lapisan masyarakat Indonesia yang belum mampu menguasai dan menggunakan teknologi tersebut.

Pelayanan kerusakan berbasis sistem informasi ini akan mengadopsi cara kerja seorang mekanik dalam mendiagnosa kerusakan kendaraan roda dua pada sepeda motor. Perangkat ini akan memberikan penjelasan kerusakan beserta solusi untuk memperbaiki kerusakan. Sistem ini akan membandingkan gejala atau *Symptom* yang dialami oleh sepeda motor dengan pusat data berupa basis data tentang kerusakan beserta gejalanya. Kemudian hasil dari perbandingan tersebut akan diambil yang memiliki tingkat ketepatan yang paling tinggi. Sehingga diketahui diagnosa tentang kerusakan sepeda motor, maka kemudian dapat diketahui pula solusi yang cocok untuk memperbaiki kerusakan sepeda motor tersebut.

Namun demikian, tentunya sistem ini tidak akan serta merta menggantikan peran serta kerja seorang mekanik secara keseluruhan. Sistem ini dapat berfungsi dengan optimal untuk kasus-kasus gangguan kerusakan ringan, dimana tidak memerlukan interaksi antara mekanik dan pengguna sepeda motor secara langsung.

1.2 Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan untuk:

1. Menghasilkan sistem pakar untuk mendiagnosa kerusakan sepeda motor dengan menggunakan metode *Interview* dengan ahli kendaraan sepeda motor untuk mempermudah *user*.
2. Membangun aplikasi yang dapat membantu pengambilan keputusan dalam menentukan jenis kerusakan yang dialami dari beberapa alternatif gejala-gejala atau permasalahan yang dimasukkan.

1.3 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Berguna bagi pengguna sepeda motor sebagai alat bantu pengguna sepeda motor untuk mengetahui kerusakan pada sepeda motor saat tidak ada seorang pakar.
2. Dapat membantu orang awam atau para ahli untuk mengetahui dan memastikan jenis kerusakan dan gangguan kerusakan ringan, dengan cara memasukkan gejala-gejala atau permasalahan yang sering dialami pengguna kendaraan sepeda motor.

1.4 Rumusan Masalah

Untuk mengaplikasikan rancangan sistem pakar pendiagnosa kerusakan berbasis *Software* yang dirancang dengan bahasa pemrograman *Turbo Prolog* yang diintegrasikan dalam bentuk *Database*. Sistem ini mengadopsi cara kerja seorang mekanik untuk mendiagnosa kerusakan sepeda motor dengan metode wawancara. Sistem ini memungkinkan pengguna mengetahui kerusakan sepeda motor dengan memasukan gejala atau permasalahan yang dialami oleh pengguna sepeda motor ke dalam sistem. Kemudian gejala yang dialami oleh pengguna sepeda motor tersebut akan diolah dan dibandingkan dengan pusat data. Hasilnya akan ditampilkan berupa jenis kerusakan, dan anjuran perbaikan bagi sepeda motor tersebut. Sehingga dapat mempermudah *user* dalam perawatan sepeda motor dan dapat memperbaikinya dengan cara yang tidak perlu mengeluarkan biaya besar untuk datang ke bengkel resmi atau bengkel-bengkel setempat.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem pakar ini dibatasi sampai sepeda motor dengan mesin 4 langkah (*Four Stroke Engine*) yang menggunakan karburator.
2. Metode pendiagnosaan yang akan digunakan pada skripsi ini merupakan metode wawancara dan perancangan yang digunakan dalam penelitian ini menggunakan metode *Modified Waterfall*.
3. Kerusakan yang terdapat dibasis data sangat terbatas, hal ini dimaksudkan hanya untuk mencoba apakah sistem yang dibuat dapat berjalan dengan baik atau tidak.

1.6 Sistematika Penulisan

Dalam rangka penulisan skripsi ini, disusun suatu sistematika penulisan dengan membaginya menjadi beberapa bab. Susunan sistematika tersebut adalah:

BAB I. PENDAHULUAN

Dalam bab ini diuraikan tentang latar belakang, tujuan penulisan, manfaat penelitian, permasalahan, batasan masalah, dan sistematika penulisan.

BAB II. TINJAUAN PUSTAKA

Bab ini membahas tentang teori-teori dasar tentang cara pendiagnosaan mekanik yang diadopsi oleh sistem ini, sistem pakar, dan teknologi yang digunakan pada sistem ini.

BAB III. METODE PENELITIAN

Bab ini membahas mengenai tahapan-tahapan yang dilakukan dalam merancang sistem pakar pendiagnosa kerusakan berbasis sistem informasi menggunakan metode *Modified Waterfall*.

BAB IV. HASIL DAN PEMBAHASAN

Bab ini berisi analisis simulasi, uji coba, dan analisa untuk kinerja perangkat lunak saat menjalankan sistem pakar pendiagnosa kerusakan berbasis sistem informasi.

BAB V. SIMPULAN DAN SARAN

Bab ini berisi simpulan dan saran yang diperoleh berdasarkan hasil pembahasan.

BAB II DASAR TEORI

2.1 Sistem Diagnosa Kerusakan berbasis Perangkat Lunak

Kendaraan merupakan hal yang begitu penting bagi manusia. Ironisnya banyak sekali kerusakan-kerusakan yang pada akhirnya terlambat didiagnosa sehingga mencapai tahap kerusakan cukup parah yang membuatnya memerlukan biaya yang cukup besar untuk diperbaiki. Padahal setiap kerusakan sebelum mencapai tahap kerusakan cukup parah umumnya menunjukkan gejala-gejala kerusakan yang telah terjadi pada kendaraan tetapi masih dalam tahap yang ringan misalnya seperti performa motor menurun, dan suara mesin kasar. Sayangnya karena ketidaktahuan pada kerusakan tersebut maka biasanya pengguna kendaraan motor akan mengabaikan gejala-gejala atau permasalahan ringan tersebut yang berpotensi untuk menjadi fatal jika tidak diberi tindakan perbaikan.

Kemajuan teknologi mendorong setiap orang untuk terus memperbaharui sistem pelayanan kepada seorang pengguna jasa. Sistem pelayanan yang dimaksud disini akan membahas tentang sistem pelayanan kerusakan sepeda motor. Sistem ini harus dapat memberikan kenyamanan kepada pengguna sepeda motor dalam segala bidang. Hal yang perlu menjadi pertimbangan adalah bahwa pelayanan kerusakan konvensional akan banyak menyita waktu pengguna kendaraan sepeda motor. Meskipun pengguna kendaraan sepeda motor hanya ditanyai beberapa

pertanyaan oleh seorang mekanik mengenai gejala kerusakannya. Akan tetapi pengguna tersebut tetap saja diharuskan untuk meluangkan waktunya untuk datang ke bengkel resmi, mengantri dan bertemu mekanik. Belum lagi seorang pengguna kendaraan sepeda motor diharuskan untuk membayar biaya yang cukup mahal untuk ke bengkel resmi.

Sistem pelayanan kerusakan seperti ini sebenarnya dapat diganti dengan sistem baru yang berbasiskan perangkat lunak asalkan diketahui pola kerja seorang mekanik dalam mendiagnosa suatu kerusakan. Sehingga sistem ini akan sangat mudah dijangkau oleh setiap orang dalam pengaksesannya.

2.2 Cara Diagnosa Mekanik

Seorang mekanik dalam mendiagnosa kerusakan sepeda motor akan melalui dua tahapan yang akan dilakukannya :

1. Pengecekan Fisik
2. Pengecekan Penunjang

Kedua tahapan ini diurutkan berdasar seberapa tingkat keakuratan seorang mekanik dalam mendiagnosa suatu kerusakan.

Dari data diatas dapat diketahui bahwa tahapan pemeriksaan fisik yang dilakukan oleh seorang mekanik kepada pengguna sepeda motor sudah cukup untuk mendapatkan gambaran tentang kemungkinan kerusakan pada sepeda motor. Pada prakteknya, seorang mekanik kemungkinan hanya melakukan tahapan pemeriksaan fisik untuk mendiagnosa kerusakan pada sepeda motor. Karena keakuratan pada tahapan ini yang cukup besar, sehingga tahapan lainnya hanya

dilakukan untuk menambah keyakinan dari seorang mekanik untuk memvonis kerusakan sepeda motor tentang kerusakan yang dialami.

2.2.1 Pengecekan Fisik

Tahapan ini merupakan tahapan dimana seorang mekanik melihat tanda-tanda fisik yang ada pada kendaraan sepeda motor. Pengecekan ini dilakukan dikarenakan ada beberapa kerusakan yang memiliki ciri-ciri yang dapat dilihat pada tanda-tanda fisik dari kendaraan sepeda motor tersebut.

2.2.2 Pengecekan Penunjang

Pengecekan penunjang merupakan tahapan dimana seorang mekanik memerlukan pembongkaran mesin untuk memeriksa sepeda motor tentang suatu kerusakan yang dialami. Pengecekan ini meliputi : Pembongkaran mesin. Pengecekan penunjang hanya dilakukan pada jenis-jenis kerusakan tertentu yang sangat rumit dalam penentuan jenis kerusakan jika hanya mengandalkan tahapan pengecekan fisik.

2.3 Ilmu Kecerdasan Buatan

Artificial Intelligence atau kecerdasan buatan merupakan bagian dari ilmu pengetahuan komputer yang khusus ditunjukkan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer. Adapun definisi kecerdasan buatan yang mempelajari bagaimana membuat suatu mesin seolah-olah memiliki kecerdasan dalam memecahkan suatu masalah yang diberikan kepadanya.

Ada beberapa definisi kecerdasan buatan yaitu:

1. Kecerdasan buatan adalah ilmu komputer yang yang berhubungan dengan studi dan kreasi sitem komputer yang mempertunjukkan beberapa bentuk kecerdasan.
2. Sistem yang mempelajari konsep-konsep baru dan tugas-tugas.
3. Sistem yang dapat berpikir dan menarik kesimpulan yang berguna bagi dunia di sekitar kita.
4. Sistem yang dapat mengerti bahasa dan memahami pemandangan visual.
5. Sistem yang melakukan tipe-tipe yang lain seperti prestasi yang membutuhkan kecerdasan manusia.

Berikut adalah beberapa ilmu kecerdasan butan yang dipakai pada penulisan tugas akhir Perangkat Lunak Diagnosa Kerusakan Sepeda Motor Berbasis Sistem Pakar.

2.3.1 Sistem Pakar

Akhir abad ini, aplikasi tunggal teknik kecerdasan buatan yang terbesar adalah sistem pakar (*expert system*). Sudah banyak dana dan waktu yang diinvestasikan dalam pengembangan sistem pakar ini, dibanding cabang kecerdasan buatan lainnya. Sistem pakar adalah program kecerdasan buatan yang menggabungkan basis pengetahuan (*knowledge base*) dengan sistem inferensi. Inferensi berasal dari bahasa inggris yaitu *inference*. Inferensi adalah suatu proses memperoleh pengetahuan berdasarkan pengalaman yang terjadi.

Hal ini merupakan *software* spesialisasi tingkat tinggi yang berusaha menduplikasi fungsi seorang pakar dalam suatu bidang keahlian. Program ini bertindak sebagai seorang konsultan yang cerdas atau penasehat dalam suatu lingkungan keahlian tertentu, sebagai hasil himpunan pengetahuan yang telah dikumpulkan dari beberapa pakar. Dengan demikian seorang yang awam sekalipun bisa menggunakan sistem pakar tersebut untuk memecahkan berbagai persoalan yang dihadapi.

Sistem pakar merupakan suatu perkembangan inovasi baru yang sangat inovatif dalam menghimpun ilmu pengetahuan. Keunggulan yang paling utama terletak pada kemampuan dan penggunaan secara praktis, yang dapat diterapkan ditempat dimana tidak ada seorangpun ahli atau pakar dalam bidang ilmu tertentu ditempat tersebut.

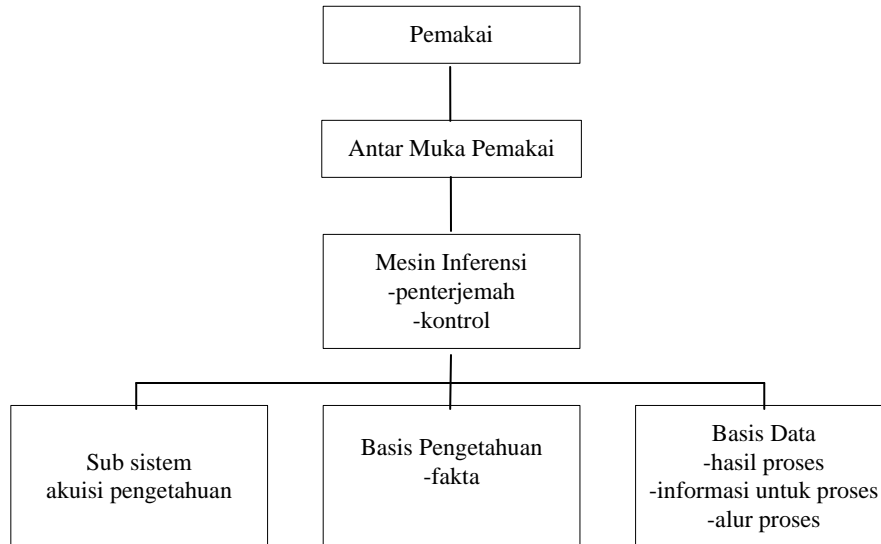
Oleh karena itu sistem pakar akan mengubah peta keahlian. Ilmu pengetahuan yang selama ini hanya dimonopoli oleh pakar-pakar yang langka dan hanya ada di universitas tertentu, pada saat ini sudah tersebar luas, sehingga akan membantu dalam mengatasi berbagai kesulitan .

Sistem pakar mempunyai beberapa karakteristik yang biasanya diterapkan dalam pembuatan program secara praktis. Karakteristik- karakteristik dari sistem pakar tersebut adalah sebagai berikut :

1. Basis pengetahuan mudah diperbaharui
2. Mudah dipakai oleh siapa saja (*user friendly*)

3. Mampu menjelaskan proses pemikiran dan langkah-langkah yang dilakukan untuk mencapai kesimpulan.

Sedangkan bagian-bagian dari sistem pakar dapat dilihat dari bagan berikut ini :



Gambar 2.1 Bagian Sistem Pakar

Dari bagian sistem pakar tersebut di atas dapat dijelaskan bahwa pada bagian *Inference Engine* terdapat pada Pemakai, Antar Muka Pemakai, dan Mesin Inferensi. Untuk bagian *Knowledge Base* nya dapat dilihat pada bagian Sub sistem akuisi pengetahuan dan Basis Pengetahuan. Sedangkan pada *Working Memori* terdapat pada bagian Basis Data. ^[2]

2.3.2 Pencarian Heuristik

Heuristik adalah teknik yang digunakan untuk meningkatkan efisiensi dari proses pencarian. Teknik ini baik diterapkan dalam tujuan yang umum, tapi tidak untuk tujuan yang khusus. Fungsi heuristik yang dirancang dengan baik dapat berperan

dalam sebuah bagian yang penting untuk memandu secara efisien proses pencarian menuju sebuah solusi.^[2]

Sistem pakar telah menegaskan pentingnya heuristik sebagai komponen pokok dalam penyelesaian permasalahan. Bila seorang pakar menyelesaikan suatu permasalahan-permasalahan, pakar tersebut akan memeriksa keterangan yang ada dan mengambil keputusan. Aturan yang baik yang digunakan untuk menyelesaikan masalah dengan cepat guna adalah sifat dari heuristik.

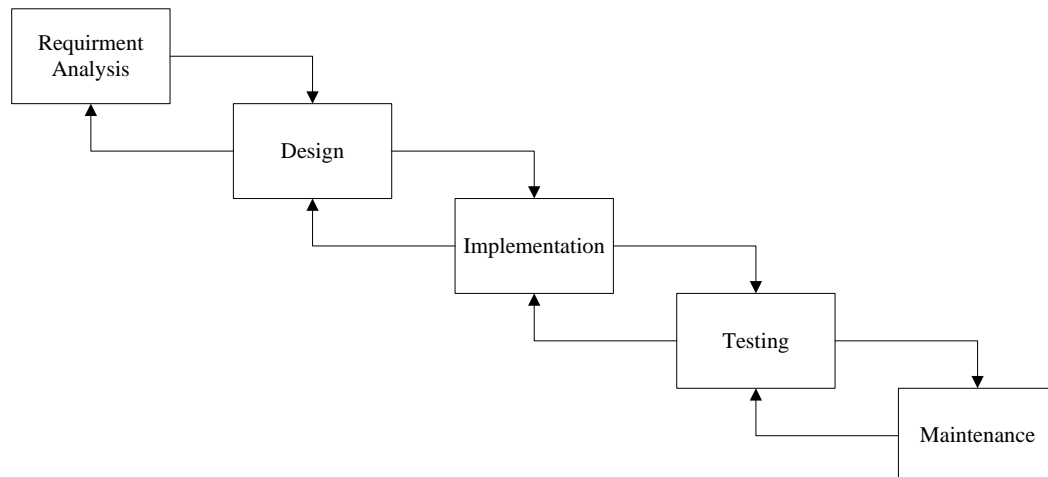
Algoritma Heuristik terdiri dari dua bagian ukuran dari Heuristik yang berhubungan dengan manfaatnya serta algoritma yang digunakan untuk mencari *state space*.^[2]

2.4 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang dipercaya dan bekerja secara efisien.^[3]

2.4.1. Model Rekayasa Perangkat Lunak “*Modified Waterfall Model*”

Modified Waterfall merupakan salah satu cara dalam pemodelan rekayasa perangkat lunak. Pada pemodelan *modified waterfall* memiliki tahapan-tahapan yang meliputi analisis, desain, implementasi, pengujian dan perawatan. Berikut ini merupakan bentuk dari diagram *waterfall* :



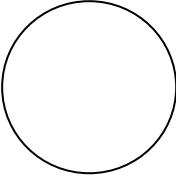



Gambar 2.2. *Modified Waterfall Model*

Pada model *waterfall* terdapat fungsi-fungsi dari tiap tahap, yaitu ^[4]:

1. *Requrment Analysis*. Yaitu proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pda tahap ini perlu untuk didokumentasikan.
2. *Design*. Yaitu melakukan proses dalam mendesain sistem dengan mengalokasikan persyaratan yang telah ada dengan membentuk arsitektur secara keseluruhan.
3. *Implementation*. Yaitu desain yang ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
4. *Testing*. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.
5. *Maintenance*. Ini merupakan tahap terakhir dalam model waterfall. Software yang sudah jadi dijalankan serta dilakukan pemeliharaan.

2.4.2. Data Flow Diagram

Data Flow Diagram (DFD) termasuk dalam *Structured System Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah sistem. Berikut adalah notasi notasi pada DFD ^[3] :

Notasi	Keterangan
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi ilmiah inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p>
	<p>File atau basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi table-table basisdata yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabe pada basisdata <i>Entity Relationship Diagram</i> (ERD).</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau user yang berinteraksi dengan perangkat lunak yang dimodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan.</p>
	<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p>

Gambar 2.3. Simbol Data Flow Diagram

Data Flow Diagram (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

DFD digunakan dan sangat cocok sebagai model analisis sistem perangkat lunak untuk sistem perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.^[4]

2.5 Aplikasi Pendukung.

Untuk dapat menjalankan sistem diagnosa kerusakan sepeda motor berbasis perangkat lunak dibutuhkan beberapa aplikasi pendukung seperti Amzi Prolog. Untuk bahasa pemrograman yang dipakai adalah bahasa pemrograman *Turbo Prolog*. Untuk lebih jelasnya, berikut ini adalah fungsi dari masing-masing komponen.^[5]

2.5.1 Sekilas Mengenai Turbo Prolog

Bahasa Prolog merupakan bahasa generasi kelima yang mendorong pemrograman kedalam dimensi baru. Bahasa Prolog dibangun atas dasar pemrograman alamiah dan logika. Untuk itulah lahir nama Prolog, yaitu singkatan dari ***Programming in Logic***.

Prolog juga merupakan bahasa **deklaratif**. Artinya, jika kita memberi fakta dan aturan, prolog akan menyelesaikan problem secara deduktif; atau dari banyak fakta dan aturan kemudian diturunkan kesimpulan sebagai jawaban. Hal ini berbeda sekali dengan bahasa prosedural seperti *Pascal* atau *Fortran*. Dalam

bahasa prosedural, pemrogram harus memberi perintah untuk memecahkan persoalan, langkah demi langkah; dengan kata lain, pemrogram harus tahu terlebih dahulu bagaimana memecahkan masalah itu. Bandingkanlah dengan pemrograman Prolog yang hanya perlu memberikan penjelasan masalah (fakta) dan aturan dasar untuk memecahkannya. Disini Prolog dibiarkan untuk menentukan sendiri bagaimana mencari jawaban.

Dengan demikian program aplikasi penunjang kecerdasan buatan seperti basis pengetahuan, sistem pakar, antarmuka kebahasa alami, pemrograman simbolik serta sistem manajemen informasi yang cerdas, dan pengenalan citra, akan lebih mudah diwujudkan.

2.5.2 Sejarah Singkat Prolog

Kelahiran Prolog diawali ketika Alain Colmeaurer dengan sekelompok peneliti menghadapi masalah penterjemahan bahasa dengan komputer di Montreal. Hal ini terutama akibat hasil karya Chomsky dalam ilmu bahasa. Dia dan koleganya menelusuri cara mempertemukan bahasa alami dengan bahasa komputer.

Usaha itu disusul dengan dikembangkannya sejumlah bahasa percobaan di Montreal dan Marseilles, diantaranya Tarzan yang merupakan bahasa untuk memanipulasi *tree*. di antara rekan Col Meraurer, Roussel tertarik pada segi deduksi pada sistem percobaannya dan menentukan kemungkinan penggunaan logika.

Dari segi teori, pembangunan gagasan pemrograman dengan logika (*programming in logic*) yang pertama adalah Kowalski di Edinburgh. Kemudian didemonstrasikan secara eksperimental oleh Maarten Van Emden juga di

Edinburgh, dan diwujudkan oleh Alain Colmerauer di Universitas Marseilles, dengan berhasil membuat interpreter Prolog yang pertama bersama kelompoknya pada tahun 1972.

Yang paling berpengaruh dalam implementasi Prolog ini adalah dihasilkannya kompilator Prolog pertama untuk komputer DEC-10 karya Warren dan rekan-rekannya pada tahun 1979. Secara implisit sistem ini memperlihatkan sintaks prolog baku (*standard*) dan efisiensinya membuktikan bahwa Prolog, disamping bahasa tingkat tinggi yang menarik, juga bahasa yang berdaya guna ^[6]. Karya David Warren juga memperlihatkan daya saingnya terhadap bahasa kecerdasan buatan (AI) lainnya (Lisp) dalam hal kecepatan ^[7].

Akhirnya, dalam konferensi tentang komputer generasi kelima di London yang diselenggarakan PL International, disajikan berbagai keberhasilan penggunaan Prolog yang memberi kesan bahwa Prolog adalah bahasa kecerdasan buatan yang lebih baik dari Lisp ^[8].

2.5.3 Pemrograman Logika

Untuk mengenal pemrograman logika, terlebih dahulu kita singgung logika predikat orde satu (*first order predicat logic*), yang selanjutnya kita sebut logika predikat (kalkulus predikat). Logika predikat merupakan bagian dari komputasi logika yang juga mencakup aljabar *Boole* (logika proposisional). Dalam logika predikat, fakta dan aturan dinyatakan melalui predikat, seperti:

lelaki(Beck)	(fakta)
menikah(Beck, Ewon)	(fakta)
$\forall x \forall y [\text{menikah}(x,y) \wedge \text{lelaki}(x) \rightarrow \sim \text{lelaki}(y)]$	(aturan)
$\forall x \exists y [\text{orang}(x) \rightarrow \text{punyaibu}(x,y)]$	(aturan)

Kalimat pertama menyatakan Beck seorang lelaki. Yang kedua menyatakan bahwa Beck menikah dengan Ewon. Kalimat ketiga mengatakan untuk setiap x dan untuk semua y , jika x menikah dengan y dan x lelaki, maka y bukan lelaki. Berikutnya, yang terakhir menyatakan bahwa untuk setiap x , ada y , jika x orang, maka y adalah ibu dari x .

Simbol predikat yang digunakan dalam contoh diatas adalah: *lelaki*, *menikah*, *orang punya ibu*. Simbol konstanta dalam contoh tadi adalah Beck dan Ewon, dengan operator \wedge (dan), \rightarrow (implikasi), \sim (negasi), \forall (untuk setiap), \exists (ada).

Perhatikan bahwa ada dua simbol terakhir merupakan besaran logika (kwantor).

Fakta dan aturan dalam prolog tidak tertulis tepat seperti cara diatas tapi dalam bentuk yang terkadang disebut bentuk klausa Kowalski.

Bentuk klausa tersebut sebenarnya diturunkan dari bentuk klausa Horn. Yang dimaksud klausa disini adalah suatu hubungan disjungsi beberapa literal (fakta dan aturan), dan klausa Horn adalah klausa yang hanya terdiri dari (paling banyak) satu literal positif. Contoh klausa Horn dan penurunan menjadi klausa Prolog adalah (tanda \vee berarti atau /disjungsi):

$$\sim \text{lelaki}(X) \vee \sim \text{menikah}(X,Y) \vee \text{perempuan}(Y) \quad (\text{klausa Horn})$$

(*perempuan* merupakan satu-satunya literal positif) dengan menggunakan hukum

De Morgan dan rumus:

$$\sim(A \vee B) = \sim A \wedge \sim B$$

diperoleh:

$$\sim(\sim \text{lelaki}(X) \vee \sim \text{menikah}(X,Y) \vee \text{perempuan}(Y)) =$$

$$\text{lelaki}(X) \wedge \text{menikah}(X,Y) \wedge \sim \text{perempuan}(Y)$$

Yang dalam prolog ditulis sebagai:

Perempuan(Y) if lelaki(X) and menikah(X,Y)

Gagasan yang melatar-belakangi pemrograman logika adalah menggantikan pemrograman dan komputasi dengan deskripsi logika suatu masalah dan mekanisme pembuktian untuk mendeduksi jawabannya. Pemikiran dasar dalam pemrograman logika membawa kita satu langkah kedepan, yakni komputasi adalah deduksi. Sebagai contoh, fungsi factorial biasa didefinisikan sebagai berikut:

```
faktorial(n) := if n = 0 then 1
              else n* faktorial(n-1);
```

jika sebagai pengganti program diatas kita tulis predikat dalam Prolog yang ditulis:

```
faktorial (n,m)
```

yang secara logika benar jika m adalah faktorial n, dan dengan klausa:

```
faktorial(0,1).
faktorial(n,m) if n1 = n-1 and faktorial(n1,m1) and
                m = n*m1.
```

Maka dengan klausa logika diatas kita dapat menghitung nilai faktorial. Inilah yang merupakan pokok pemrograman logika, yakni dengan klausa logika kita dapat melakukan perhitungan.

2.5.4 Ciri Bahasa Prolog

Ciri yang menonjol dalam Prolog adalah disamping mencari jawaban secara logika terhadap pernyataan yang kita ajukan, juga dapat memberi jawaban lain atau memberi semua kemungkinan jawaban. Selain hanya berjalan dari awal sampai akhir program, Prolog dapat berjalan mundur dan mencari lebih dari satu cara penyelesaian tiap bagian masalah.

Secara sistematis, ciri lain prolog yang menonjol adalah:

a. Predikat

Prolog membutuhkan fakta-fakta yang terungkap dalam relasi dan sifat untuk mencari jawaban. Sebagai contoh:

Irham suka permen
Permen manis

Ini menggambarkan objek Irham dan permen dengan relasi suka, dan sifat manis pada permen. Seringkali, relasi bernilai benar jika memenuhi kondisi tertentu. Hubungan ini disebut aturan (*rule*).

Contoh aturan adalah:

Irham suka permen jika rasa permen asam

Dalam prolog, aturan dan fakta dinyatakan dalam kalimat yang disebut klausa (*clause*) seperti;

suka (Irham,permen).

Sintaks untuk menyatakan fakta seperti diatas disebut logika predikat (*predicate logic*). Untuk aturan diatas, kita tuliskan dalam Prolog:

Suka(Irham, permen) if rasa (permen, asam)

atau dapat pula berbentuk:

suka (Irham, permen) if asam (permen)

Selanjutnya, logika predikat ini kita sebut saja predikat, yang merupakan nama simbolik suatu relasi atau sifat, sedangkan objek yang suka dikaitkan disebut *argument*. Contoh predikat diatas adalah *suka* dengan argumen *Irham* dan *permen*.

b. Basis Data

Sebagai bahasa yang berorientasi pada fakta dan untuk menunjang basis pengetahuan, Prolog menyediakan fasilitas basis data.

c. Deduksi

Prolog dapat melakukan deduksi (penarikan kesimpulan) seperti:

Diberikan fakta:

Dede suka Iik
Cece suka Rayan

dan aturan:

Celung suka Rayan

d. Pemadanan dan Unifikasi

Dalam mencari jawaban atau menarik sesuatu kesimpulan (*reasoning*), Prolog melakukan pemadanan. Argumen yang pertama dipadankan dengan argumen pertama predikat lain yang sama, dan seterusnya. Tujuan melakukan pemadanan adalah agar dapat melakukan unifikasi, yakni penyatuan *argument* dalam suatu predikat dengan predikat lain yang sudah identik. Sebagai contoh, kita semua tahu bahwa bujur sangkar adalah suatu bangun segi empat, berisi sama dan bersudut 90 derajat. Andaikan kita diminta untuk memeriksa apakah suatu bangun berbentuk bujur sangkar. Pertama kita menduga bahwa suatu bangun berbentuk bujur sangkar. Jika kemudian ketiga syarat diatas terpenuhi, maka dugaan kita benar bahwa bangun yang kita periksa adalah bujur sangkar. Dalam Prolog, dugaan bahwa suatu bangun berbentuk bujur sangkar adalah *unifikasi variabel*, katakanlah X, dengan nilai bujur sangkar. Ini terjadi karena ada pemadanan antara bangun (X) dengan bangun (bujur sangkar)

yang sudah identik, kemudian dilakukan unifikasi antara X dengan bujur sangkar.

Dalam bahasa lain, pemberian nilai terhadap suatu variabel dilakukan dengan statemen pemberian (*assignment statement*) seperti $y = 2$. Ini bersifat permanen selama tidak ada statemen lain yang mengubahnya. Unifikasi bersifat sementara selama belum ada lacak balik dan hanya berlaku satu klausa, sebagaimana hanya berlaku dalam satu fungsi seperti dalam C.

e. Pembuktian Mundur

Dari contoh pembuktian bahwa suatu bangun berbentuk bujur sangkar, prolog menganggap suatu bangun berbentuk bujur sangkar terlebih dahulu, baru kemudian dilakukan pembuktian. Cara ini disebut pembuktian mundur (*backward chaining*) atau disebut juga metoda *top-down* karena dari kenyataan pokok (bujur sangkar) kemasalah rinci (sudut dan sisi).

f. Lacak Balik (*backtracking*)

Dalam mencari jawaban, Prolog melacak suatu fakta atau aturan dengan melakukan pepadanan (*matching*) berturut-turut. Jika sepadan, akan terjadi unifikasi dan dilanjutkan pembuktian yang selanjutnya dilakukan pepadanan serupa.

Katakanlah dalam pembuktian suatu bentuk seperti diatas, kita menemui kegagalan, misalnya, karena sudut-sudutnya tidak sama dengan 90 derajat.

Yang pertama kita lakukan adalah mengubah anggapan bahwa suatu bangun berbentuk bujur sangkar. Langkah berikutnya mencari anggapan baru dengan definisi baru, misalnya bentuk *trapezium*. Dalam prolog, kejadian seperti ini disebut sebagai lacak balik (*backtracking*). Jadi, jika kemudian menemui kegagalan, artinya hipotesa pertama tidak dapat digunakan harus dibatalkan. Kemudian Prolog melakukan pelacakan mundur (lacak balik) untuk mencari hipotesa baru. Lacak balik akan berhenti jika ditemui jawaban atau kehabisa fakta untuk menyimpulkan jawaban.

g. Rekursi

Prolog merupakan program yang paling banyak memanfaatkan rekursi, baik dalam aturan maupun data. Rekursi berarti cara memecahkan masalah dengan menguraikan masalah tadi menjadi lebih kecil tetapi memiliki struktur atau model yang sama. Dalam bahasa procedural, kita kenal rekursi sebagai suatu prosedur yang memanggil dirinya sendiri dengan memperkecil permasalahan. Akan tetapi tidak semua bahasa mampu melakukan rekursi, misalnya: Fortran dan Cobol.

Contoh bentuk rekursi yang sederhana adalah faktorial. Untuk mencari faktorial 4 misalnya, dicari faktorial 3 untuk dikalikan dengan 4. Faktorial 3 diperoleh dari faktorial 2 kali 3 dan seterusnya sehingga sampai pada faktorial yang diketahui nilainya. Jelas bahwa disini ada usaha memperkecil permasalahan, sehingga masalah yang lebih dasar dapat dipecahkan melalui masalah yang lebih kecil dan mudah dengan bentuk yang serupa. Contoh data rekursif adalah list, yakni sekumpulan data

sejenis dalam satu besaran data. List terdiri dari kepala (*unsure* pertama) dan list ekor yang lebih kecil, dan seterusnya.

h. Pengolahan Simbol

Prolog ditekankan juga pada pengolahan informasi simbolik seperti pada contoh diatas, juga untuk pengolahan struktur data dinamis dan membuat struktur data baru. Selain itu prolog juga dapat memanipulasi simbol seperti dalam pemecahan persamaan matematik, diferensial dan integral, dan lain sebagainya.

i. Meta Programming

Dalam bahasa lain, apabila program dijalankan maka algoritma pemrograman tidak dapat diubah lagi, baik dalam *interpreter* apalagi melalui *compiler*. Pengubahan algoritma hanya dapat dilakukan pada program sumber melalui suatu editor. Dengan prolog, kita dapat membuat program yang dapat diubah prosedurnya atau dalam hal ini logikanya pada saat program dijalankan.

j. Paralel

Meskipun kebanyakan implementasi Prolog sampai sekarang ini belum bersifat paralel, Prolog merupakan bahasa yang berstruktur paralel dari sifatnya. Ini karena beberapa klausa (yang sama) dapat dieksekusi secara bersamaan (*concurrent*). Paralelisme demikian disebut *Or-Paralellism*. Dalam klausa sendiri, beberapa sub-klausa dapat dieksekusi secara paralel, dan disebut *And-Parallelism*. Hal ini membuat Prolog sebagai calon kuat bahasa komputer masa depan dengan arsitektur paralel.

2.5.5 Bahasa Deklaratif

Pokok perbedaan prolog dari bahasa lain adalah karena bersifat deskriptif atau deklaratif, sedangkan bahasa lain bersifat *procedural* atau imperatif. Artinya, Prolog hanya membutuhkan deklarasi atau uraian masalah sedangkan yang lain memerlukan perintah. Dalam memecahkan masalah, prolog melakukan deduksi (dari sekumpulan fakta ditarik kesimpulan). Disini kewajiban kita adalah member perincian masalah secara benar, bukannya membuat sekumpulan perintah untuk dikerjakan komputer.

Perbedaan deklaratif dan procedural dapat diterangkan dengan perumpamaan seperti kita naik taksi. Secara deklaratif, kita hanya meminta supir taksi untuk membawa kita ke suatu tempat tujuan, tidak peduli jalan mana saja yang harus ditempuh. Penentuan jalan yang ditempuh diserahkan kepada supir. Dalam pengertian procedural, kita yang menentukan jalan mana saja yang harus ditempuh dan boleh saja supir taksi tidak tahu kemana tujuan kita sebenarnya.

Sebagai bukti bahwa Prolog merupakan bahasa deklaratif adalah:

1. Jika kita ingin menerangkan bahwa A adalah orang tua B, dalam Prolog cukup ditulis:

Orangtua(A,B).

2. Jika kita menerangkan bahwa A adalah leluhur Z, maka harus didefinisikan terlebih dahulu logikanya yang dalam bahasa Indonesia berbunyi sebagai berikut:

A adalah leluhur Z jika A orang tua Z atau
A adalah leluhur Z jika orang tua B dan B leluhur Z.

Aturan seperti ini dalam prolog ditulis sebagai aturan rekursif seperti berikut:

leluhur(A,Z) if orangtua(A,Z).
leluhur(A,Z) if orangtua(A,B) and leluhur(B,Z).

Hanya dengan menuliskan aturan seperti ini, kita dapat menentukan siapa leluhur Z, siapa keturunan A atau membuktikan kebenaran bahwa A leluhur Z dengan memberikan data hubungan (fakta) orang tua secukupnya seperti diatas tanpa harus merinci cara mencari jawaban atau cara membuktikannya.

Aturan seperti diatas diturunkan berdasarkan logika, sehingga pemrograman prolog, seperti namanya, disebut pemrograman logika (*logic programming*).

Dari contoh diatas jelas bahwa Prolog adalah bahasa berjenis Apa yakni Apa yang harus dipecahkan atau disebut juga *Goal Oriented*. Sedangkan yang lain berjenis Bagaimana , yakni bagaimana memecahkan masalah. Dalam hal terakhir ini komputer hanya sebagai pembantu untuk mempercepat penyelesaian masalah. Dengan adanya Prolog, diharapkan bahwa komputerlah yang berfungsi sebagai pemecah masalah (*Intelligence*) atau komputer yang harus *memahami* masalah yang harus dipecahkan.

Dengan demikian, pengenalan prolog sangat penting bagi mahasiswa ilmu komputer (*Computer Science*) karena tidak ada jalan lain kecuali mengetahui bagaimana gagasan pemrograman berjenis apa tersebut. Demikian menurut P.H. Winston, seorang pakar Lisp dari tempat kelahiran Lisp (MIT) ^[9] .

2.5.6 Prolog dan Proyek Komputer Generasi Kelima

Prolog lahir di Eropa, tapi yang tidak juga kecil peranannya dalam membesarkan prolog adalah Proyek Komputer Generasi Kelima yang dicanangkan Jepang. Memang sejauh ini perkembangan teknologi komputer dari satu generasi ke generasi berikutnya hanyalah berkisar pada kemajuan peralatannya dari tabung hampa ke transistor, ke IC dan sekarang ke LSI. Gejala ini mengatakan bahwa

dalam filosofi desain dan sasaran utama pemamfaatan komputer, tidak ada perubahan besar.

Dengan komputer generasi nkelima, diharapkan bahwa perubahan generasi tidak saja melibatkan perubahan perangkat keras seperti penggunaan VLSI, tapi juga dalam perubahan filosofi desain dan bisang penerapannya. Demikian pernyataan Lembaga Komputer Generasi Baru Jepang (ICOT).

Lebih jauh ditegaskan bahwa perubahan tersebut melibatkan perubahan dari komputer numerik ke komputer yang dapat menafsirkan arti informasi dan dapat memecahkan masalah ^[7].

Dari cita-cita besar ini, memang tidak salah untuk mengambil Prolog sebagai bahasa inti untuk komputer generasi kelima tersebut, selain dari sifat paralelnya.

2.5.7 Turbo Prolog dan Prolog Tradisional

Di awal kelahirannya, Prolog masih merupakan bahasa murni yang didasarkan pada keindahan logika. Prolog masih berkembang lambat dan hanya digunakan di kalangan peneliti di perguruan tinggi terkenal karena mereka menyadari arti pentingnya Prolog. Akhirnya mereka menemukan adanya kebutuhan untuk penggunaan praktis dan keampuhan pada prolog untuk bidang penelitian yang luas.

Diawal tahun 1980-an, dunia industri menyadari kemajuan prolog terhadap bahasa lain. Disini muncul kebutuhan untuk memiliki prolog yang cepat dari bahasa lain, memori lebih kecil dan karakteristik praktis lainnya.

Dengan campur tangan dunia industri, muncul dua kubu pemrogram prolog dengan filosofi yang berbeda. Yang pertama tetap yakin bahwa Prolog adalah

bahasa riset dan dipihak lain yakin bahwa Prolog akan menjadi bahasa yang ampuh untuk pengembangan program terapan.

Turbo Prolog merupakan hasil perkembangan ini. Turbo Prolog menggabungkan kemampuan Prolog tradisional dengan kecepatan dan kemampuan bahasa lainnya.

Perbedaan utama antara Turbo Prolog dengan Prolog tradisional adalah:

- a. Turbo Prolog merupakan kompiler Prolog bertipe yakni semua objek dan relasi/relasi harus dideklarasikan dalam program.
- b. Dalam Turbo Prolog, metaprogramming bukan merupakan fasilitas jadi yang siap pakai tapi dapat dimodelkan.

Menurut pembuatnya, alasan penting yang melatar belakangi hal tersebut adalah:

- a. Peningkatan kecepatan yang sangat besar.
- b. Daya baca program lebih besar.
- c. Waktu untuk pengembangan program lebih pendek.
- d. Deteksi galat (*error*) lebih besar.
- e. Pemeliharaan program lebih murah.

2.5.8 Penerapan Prolog

Prolog sangat membantu dalam program untuk menirukan cara berfikir manusia dengan mesin (komputer) karena Prolog memiliki sifat atau cirri yang sesuai dengan metoda pemrograman kecerdasan buatan. Dengan kata lain Prolog sangat besar manfaatnya dalam kecerdasan buatan yang meliputi berbagai bidang:

- a. *Sistem pakar*: Secara sederhana, sistem pakar adalah suatu program yang ditulis untuk dapat menirukan keahlian seorang pakar (dalam bidang keahlian yang sangat khusus) dalam menjawab persoalan,

seperti mendiagnosa infeksi bakteri (MYCIN), mengidentifikasi suatu jenis binatang, untuk konsultasi, analisa, prediksi dan sebagainya. Sebagai bahasa yang berdasarkan pada pemrograman logika, Prolog lebih menguntungkan, baik segi representasi pengetahuan maupun cara pencarian/penarikan kesimpulan.

- b. *Permainan*: Sekarang banyak beredar program untuk memecahkan masalah permainan seperti untuk bermain poker, *bridge*, catur dan sebagainya. Sebagai contoh, program catur Mephisto III yang pernah bermain seri dengan juara dunia Anatoly Karpov. Dalam catur misalnya, pemain harus mencari segala kemungkinan langkah terbaik untuk memenangkan permainan. Namun jika kita menjajagi semua kemungkinan langkah dan akibatnya, tentu akan lama. Ivan Bratko telah banyak mengupas tentang Advice Languages yang dapat memberi saran untuk langkah berikutnya secara deklaratif. Saran-saran ini dinyatakan dalam sekumpulan goal dan cara mencapai goal tersebut.
- c. *Searching atau pencarian jawaban*: Program untuk menyederhanakan rumus matematik, mencari jejak, atau membuktikan suatu kebenaran. Dalam hal ini masalah yang dihadapi biasanya adalah tidak diketahuinya algoritma pemecahannya. Sebagai bahasa deklaratif, Prolog tampil sebagai calon terbaik.
- d. *Pengolahan bahasa alami*: Ini mencakup penterjemahan antar-bahasa, humor, percakapan dengan mesin, analisa kalimat dan sebagainya. Kelebihan Prolog dalam hal ini adalah bahwa program untuk pengolahan bahasa akan lebih mudah diwujudkan dan lebih jelas secara

logika sehingga mudah dalam pemeliharaan dan pengembangan programnya.

- e. *Robot*: Ini mencakup bidang yang lebih luas seperti pengendalian gerak, percakapan robot (SHRDLU), sistem pakar robotik (ARGOS II). Prolog dipicu dengan suatu sasaran (*goal-driven*) sedangkan pemrograman dalam robot pada umumnya lebih mudah jika diwujudkan secara goal-driven, seperti untuk mengambil kubus dengan bagian goal: *gerakkan manipulator kearah kubus*.
- f. *Belajar* atau *Machine Learning*: Program yang dibuat agar komputer dapat mengambil pelajaran (pengalaman), misalnya *Hacker*. Program yang demikian, umumnya selalu memperbaiki atau menambah data untuk peningkatan performansi dalam hal kecerdikan memberi jawaban. Prolog didukung dengan fasilitas untuk dapat mengubah fakta dan aturan pada saat program berjalan, yang sesuai dengan cara kita belajar.
- g. *Pengenalan Citra (Pattern Recognition)*: Ini mengusahakan agar komputer dapat mengenal bentuk yang agak berbeda seperti membaca tulisan tangan. Ini bermanfaat misalnya dalam komputer palmtop yang tidak memerlukan keyboard. Dalam pengolahan visual, nampaknya lebih menguntungkan jika memanfaatkan paralelisme, deklaratif dan pepadanan Prolog.

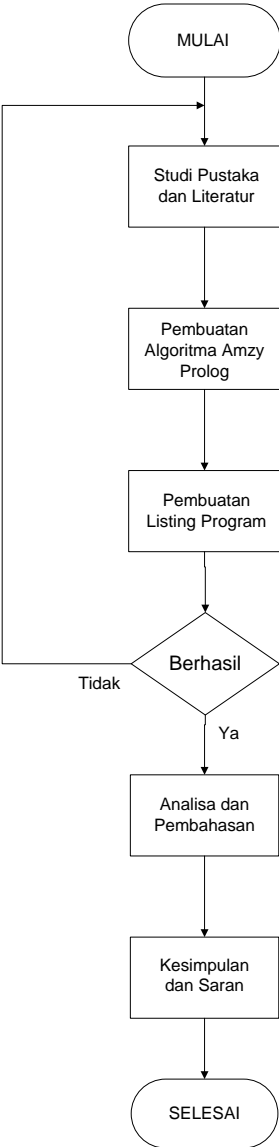
Meskipun Prolog merupakan bahasa deklaratif, dapat pula ditafsirkan dari segi *procedural*, sehingga dapat mencakup bidang penerapan yang lebih luas. Selain

penerapan seperti diatas, Prolog khususnya Turbo Prolog, dapat pula digunakan untuk:

- Pengaturan dan pemantauan proses dalam industri.
- Mewujudkan program basis data yang canggih.
- Membuat program aplikasi untuk bisnis.
- Menulis (membuat) kompuler.
- Merencanakan dan penjadwalan projek, kegiatan, kurikulum dan sebagainya.
- Pendidikan seperti sejarah, hukum ilmu komputer, logika matematik, dan sebagainya.
- Pemodelan dan simulasi

BAB III METODE PENELITIAN

3.1 Bagan Alir Penelitian



Gambar 3.1 Alur Penelitian

3.3 Waktu dan Tempat

Penelitian ini dilakukan dari Bulan April 2015 di Laboratorium Terpadu Teknik Sistem Komunikasi dan Informatika Jurusan Teknik Elektro, Universitas Lampung.

3.4 Alat dan Bahan

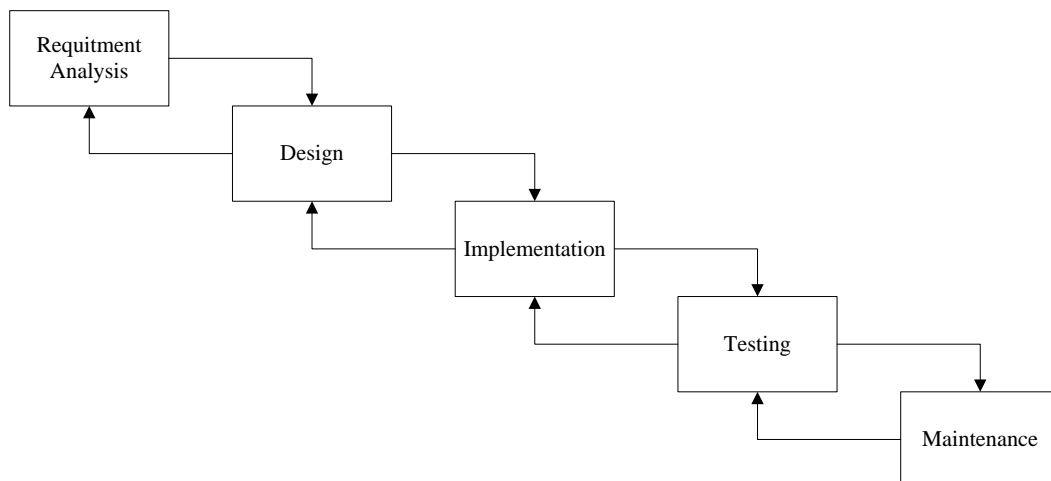
Adapun alat dan bahan yang akan digunakan pada penelitian ini yaitu *Personal Computer* (PC) Intel (R) Core (TM) i7 3930 CPU 3,20 GHz dengan RAM 8 GB yang ter-*install* perangkat lunak *Amzi Prolog* sebagai program untuk mensimulasikan penelitian yang dilaksanakan.

3.5 Studi Literatur

Metode yang digunakan dalam penelitian ini yaitu metode *Modified Waterfall*. Metode ini sangat cocok digunakan untuk kebutuhan pelanggan yang sudah sangat dipahami dan kemungkinan terjadinya perubahan kebutuhan selama pengembangan perangkat lunak kecil. Hal positif dari *Modified Waterfall* adalah struktur tahap pengembangan sistem jelas dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan.

Metode ini digunakan karena merupakan suatu metode yang praktis dan cukup menghemat biaya karena semua parameter-parameter yang dibutuhkan serta hasil yang diinginkan dapat langsung dimodelkan dan disimulasikan dengan menggunakan suatu program komputer (*Personal Computer*) dalam bentuk perangkat lunak berbasis sistem pakar.

Modified Waterfall menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai analisis, desain, implementasi, pengujian, dan perawatan. Berikut adalah gambar model air terjun (*waterfall*) :



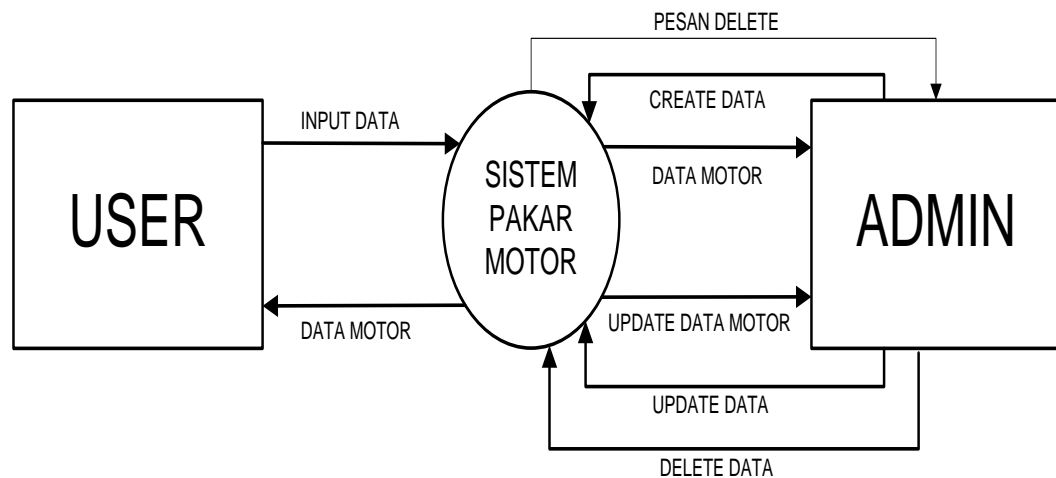
Gambar 3.1.*Modified Waterfall Model*

- Analisis kebutuhan perangkat lunak (*Requierment Analysis*)
Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan *user*. Kebutuhan yang telah didapat yaitu berdasarkan hasil wawancara ke dengan ahli sepeda motor. Dalam penelitian ini ahli sepeda motor yang dimaksud adalah Mekanik. Data tersebut diubah ke dalam bentuk kode program.
- Perancangan (*Design*)
Perancangan yang dilakukan yaitu perancangan perangkat lunak. Pada tahap ini dilakukannya mengumpulkan kebutuhan yang dibutuhkan pada sistem dan memulai perancangan konseptual yang meliputi *Context Diagram* setelah itu *Data Flow Diagram* dan *Entity Relationship Diagram* serta Desain *Database* kerusakan yang diperlukan sistem nantinya.

1. Context Diagram (CD)

Langkah awal dalam perancangan sistem ini adalah pembuatan sistem *Context Diagram* yang ditunjukkan pada gambar 3.2. yang berfungsi untuk menggambarkan hubungan antar entitas pada sistem informasi. *Context Diagram* ini merupakan gambaran awal dari sistem pakar diagnosa kerusakan sepeda motor secara umum, yang menggambarkan sistem beserta hubungannya dengan lingkungan luar dan bagaimana sistem ini berinteraksi. Entitas yang akan ditampilkan pada sistem ini informasi yaitu pengelola dan *user*.

Berikut adalah CD level 0 dari studi kasus Perangkat Lunak Diagnosa Kerusakan Sepeda Motor Honda Berbasis Sistem Pakar.



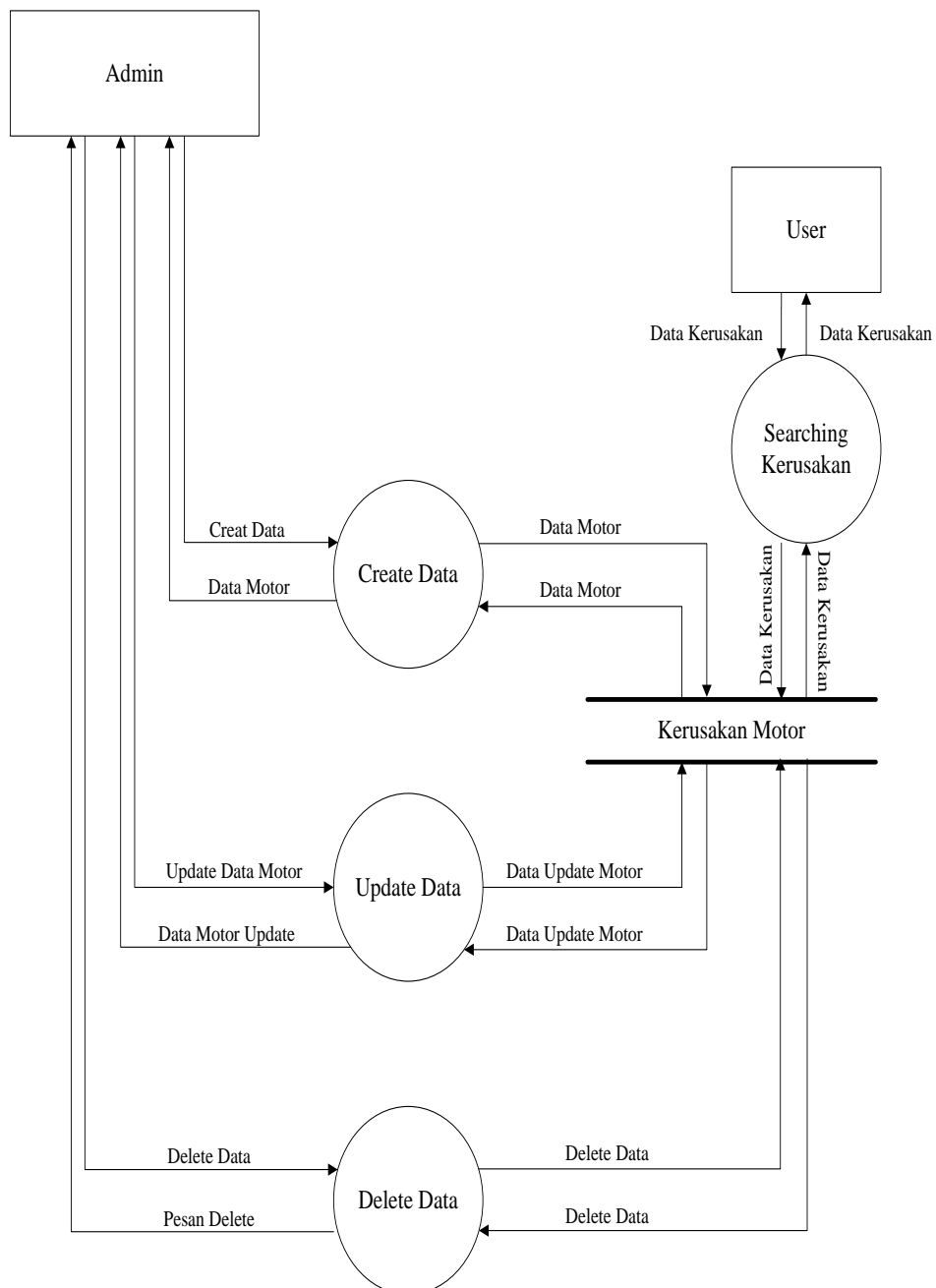
Gambar 3.2 *Context Diagram*

2. Data Flow Diagram (DFD)

DFD adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). DFD dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level

abstraksi. DFD dapat dibagi menjadi beberapa *level* yang lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail.

Berikutnya adalah DFD level 1 dari studi kasus Perangkat Lunak Diagnosa Kerusakan Sepeda Motor Honda Berbasis Sistem Pakar.



Gambar 3.3 DFD Level 1

3. Pada tahap selanjutnya akan dilakukannya pembuatan rancangan arsitektur yang terdiri atas desain *database* ^[4].

Desain Database

1. Desain Database Konseptual (*Conceptual Database Design*)

Perancangan *database* konseptual ini dapat dikatakan sebagai kelanjutan tahap analisis pada metode *modified waterfall* yang mencari segala sesuatu yang berhubungan dengan sistem yang dibuat setelah mengumpulkan dan mencari data berupa data gejala kerusakan serta perbaikan untuk kerusakan tersebut. Dalam tahap ini, setidaknya dapat dijawab pertanyaan seperti yang di bawah ini:

- a. Informasi (*output*) apa yang diinginkan dari *database*?

Informasi (*output*) yang diinginkan dari *database* adalah nama kerusakan dan perbaikan untuk kerusakan tersebut.

- b. Pengembangan sistem di masa mendatang.

Pada pengembangan sistem di masa mendatang diharapkan sistem ini dapat digunakan oleh seluruh masyarakat Indonesia.

- c. Siapa saja yang terlibat dalam sistem yang dibangun nantinya?

Yang terlibat dalam sistem yang dibangun nantinya adalah Admin, Mekanik, dan *User*.

- d. Apa saja *input* yang diperlukan?

Input yang diperlukan adalah pernyataan Ya / Tidak dari pengguna tentang gejala kerusakan yang ditanyakan oleh sistem kepada pengguna.

2. *Desain Database Logic (Logical Database Design)*

Tahap perancangan ini disebut juga pemetaan model data. Berikut langkah-langkah dalam merancang *database logic*:

- a. Mendefinisikan *Entity* yang dibutuhkan.

Entitas pada sistem yang dibangun ialah: Kerusakan

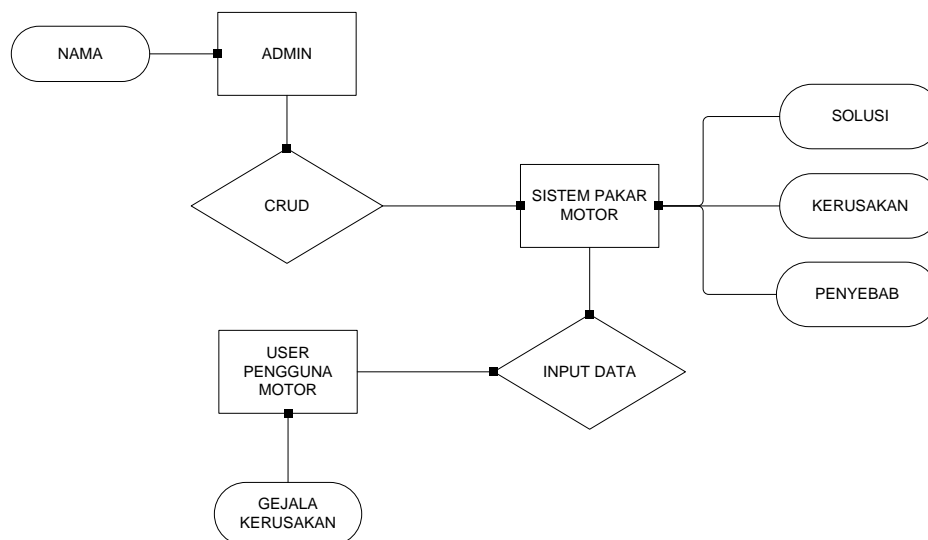
- b. Menentukan *Attribute* setiap *Entity* beserta kuncinya.

Tabel 3.3. Daftar *attribute* beserta kuncinya.

<i>Entity</i>	<i>Attribute</i>
Kerusakan	Gejala 1
	Gejala 2
	Gejala 3
	Gejala 4
	Gejala ...
Perbaikan	Kerusakan

3. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) terdiri atas table-table yang memiliki relasi dengan table lainnya. *Entity Relationship Diagram (ERD)* digunakan untuk pemodelan basis data relasional. Dari tabel yang telah dirancang sebelumnya dapat dibuat tabel data dan atribut Perangkat Lunak Diagnosa Kerusakan Sepeda Motor Berbasis Sistem Pakar sebagai berikut.



Gambar 3.5 Diagram ER

Dari gambar diatas dapat dilihat hubungan antara kedua database bahwa jenis kerusakan akan diketahui ketika gejala kerusakan terpenuhi dan table perbaikan akan aktif ketika kerusakan sudah diketahui.

- Implementasi (*Implementation*)

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain. Pada tahapan ini setelah mendapatkan semua kebutuhan *Requierment Analysis* yang dilakukan dengan cara wawancara kerusakan sepeda motor pada ahli mesin atau mekanik maka hasil dari kebutuhan tersebut dibuat tabel data kerusakan dan selanjutnya diubah ke dalam bahasa pemrograman prolog. Pada prolog terdapat dua file yang digunakan, yaitu kode program yang berfungsi sebagai *inference engine* dan informasi tentang basis data kerusakan atau disebut juga *knowledge base*.

- Pengujian (*Testing*)

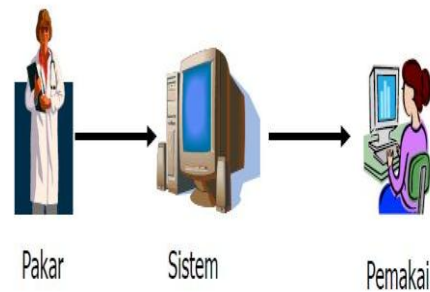
Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Setelah tahapan implementasi berhasil dilakukan maka tahapan selanjutnya yaitu pengujian sitem pakar diagnosa kerusakan. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Pengujian dilakukan pada dua bengkel sepeda motor yang berbeda dan mekanik yang berbeda pula.

- Pemeliharaan (*maintenance*)

Tahap terakhir dari model *modified waterfall* yaitu pemeliharaan. Hal ini tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan yang baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru. Dari *maintenece* ini terdapat masukan dari *user* bahwa Perangkat Diagnosa Kerusakan Sepeda Motor Berbasis Sistem Pakar ini perlu adanya peningkatan bahasa pemrograman dari bahasa pemrograman berbasis teks ke dalam bentuk bahasa pemrograman berbasis GUI (*Graphical User Interface*) agar lebih mudah dan menarik dalam menggunakan program tersebut.

3.6 Gambaran Umum Penelitian

Gambaran umum mengenai sistem ini dapat digambarkan sebagai berikut:



Gambar 3.2 Gambaran Umum Sistem

Berdasarkan gambar 3.2 dapat dilihat bahwa PC (*Personal Computer*) yang digunakan untuk menjalankan sistem ini dapat diakses dari manapun juga, termasuk *netbook* yang bersifat *mobile* dan praktis untuk digunakan setiap orang. PC yang digunakan memiliki perangkat lunak Amzi Prolog agar dapat menjalankan simulasi program sebelum dibuat menjadi perangkat lunak pendiagnosa kerusakan kendaraan pada sepeda motor berbasis sistem pakar.

V SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil yang didapat serta analisa yang dilakukan untuk diagnosa kerusakan sepeda motor, maka dapat diambil kesimpulan sebagai berikut :

1. Sistem pakar ini mampu melakukan diagnosa kerusakan serta menghasilkan kesimpulan dan memudahkan pengguna sepeda motor untuk mengetahui penyebab, akibat dan gejala-gejala yang ditimbulkan dari sepeda motor.
2. Memudahkan para pengguna sepeda motor untuk mencari solusi kerusakan sepeda motor.
3. Sistem pakar ini dapat melakukan proses *backward chaining*.
4. Memudahkan para mekanik memperbaiki sepeda motor, apabila lupa dengan gejala kerusakan sepeda motor yang dialami.
5. Sistem pakar ini telah diuji di dua bengkel yang berbeda dan mekanik yang berbeda pula dan menghasilkan persentase keberhasilan sebesar 75%.

5.2 Saran

Pada penelitian selanjutnya diagnosa kerusakan sepeda motor berbasis sistem pakar ini dapat dikembangkan lagi jenis kerusakannya pada motor mesin 4 langkah menggunakan injeksi, dan pengembangan bahasa program berbasis teks menjadi program berbasis GUI dan berbasis Android agar dapat mudah digunakan oleh pengguna.

DAFTAR PUSTAKA

- [1] Astra Honda, <http://www.astra-honda.com/index.php/sejarah-astra-honda-motor/>, terakhir diakses pada bulan Mei 2015.
- [2] Andri Kristanto, 2004, *Kecerdasan Buatan*, Graha Ilmu Indonesia, Yogyakarta.
- [3] M. Salahudin, Rosa A.S, 2004, *Rekayasa Perangkat Lunak, Terstruktur dan Berorientasi Objek*, Informatika Indonesia, Bandung.
- [4] Nita Merliana, Rahmat Hidayat, 2012, *Perancangan Sistem Pakar*, Ghalia Indonesia, Bogor.
- [5] Uung Ungkawa, 1992, *Bahasa Pemrograman Logika Turbo Prolog*, Andi Offset Yogyakarta.
- [6] Amble, T., 1987, *Logic Programing and Knowledge Engineering*, Adison-Wesley Publishing Co.
- [7] Ennals, R., 1985, *Artificial intellingence, Aplication to Logical Reasoning and Historical Research*, Ellis Horwood Ltd.
- [8] Simon, G.L., 1984, *Introducing Artificial Intelligence*, NCC Publication.
- [9] Bratko, I., 1986, *Prolog Programing for Artificial Intelligence*, Addison-Wesley Publishing Co.
- [10] Adam Hussein, 2015, *Perangkat Lunak Diagnosa Penyakit Pada Manusia Berbasis Sistem Pakar*, Teknik Elektro Universitas Lampung.