



**APLIKASI SISTEM PAKAR UNTUK MENDETEKSI
ERUSAKAN PADA PRINTER PRINTONIX**

Laporan Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:

WAHYU MUNANDAR
4150411-065

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2009**

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

NIM : 4150411-065
Nama : WAHYU MUNANDAR
Judul Skripsi : APLIKASI SISTEM PAKAR UNTUK MENDETEKSI
KERUSAKAN PADA PRINTER PRINTONIX

Menyatakan bahwa skripsi tersebut diatas adalah hasil karya saya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan skripsi saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, May 2009

Wahyu Munandar

LEMBAR PERSETUJUAN

NIM : 4150411-065
Nama : WAHYU MUNANDAR
Judul Skripsi : APLIKASI SISTEM PAKAR UNTUK MENDETEKSI
KERUSAKAN PADA PRINTER PRINTONIX

SKRIPSI INI TELAH DIPERIKSA DAN DISETUJUI

JAKARTA, MAY 2009

Dr. Rusdianto Rustam
Pembimbing

Devi Fitrihanah, S.Kom., MTI
MT
Koord. Tugas Akhir Teknik Informatika
Informatika

Abdusy Syarif, ST.,
KaProdi Teknik

KATA PENGANTAR

BISMILLAHIRROHMANIRROHIIM.

Assalamu'alaikum wr, wb.

Dengan memanjatkan puji dan syukur kepada Allah SWT atas rahmat dan hidayah-NYA, Tugas Akhir dengan judul “APLIKASI SISTEM PAKAR UNTUK MENDETEKSI KERUSAKAN PADA PRINTER PRINTONIX” ini dapat diselesaikan. Tugas Akhir ini diajukan guna memenuhi salah satu syarat kelulusan program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Mercu Buana Jakarta.

Tanpa dukungan berbagai pihak baik materil maupun non materil sangat mustahil Tugas Akhir ini dapat diselesaikan, untuk itu penulis ingin mengucapkan rasa terima kasih yang setinggi-tingginya kepada :

1. Allah SWT yang telah mewujudkan segala usaha yang telah saya lakukan.
2. Kedua orang tua tercinta berserta keluarga yang telah memberikan do'a, motivasi dan dukungan yang tiada henti.
3. Wira Oktaviar, Istri saya yang selalu memberikan do'a, motivasi, dukungan dan segalanya untuk terselesaikanya tugas ini.
4. Bapak Dr.Rusdianto Rustam selaku Dosen Pembimbing Tugas Akhir ini sehingga atas bimbingan dan dukungannya Tugas Akhir ini dapat diselesaikan.
5. Bapak Abdusy Syarif, ST.MT. selaku Ketua Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Mercu Buana Jakarta.

6. Bapak dan Ibu Dosen yang telah memberikan bimbingannya kepada penulis selama studi di Fakultas Ilmu Komputer Universitas Mercu Buana Jakarta.
7. Rekan-rekan para Ahli printer dan tehnisi printer yang tidak dapat disebutkan satu persatu atas dukungannya baik berupa informasi maupun pengalamannya yang sangat menunjang dalam penulisan Tugas Akhir ini.

Semoga semua dukungan dan bantuan yang telah diberikan kepada penulis akan mendapatkan berkah dan pahala yang berlipat ganda dari Allah SWT.

Akhir kata semoga Tugas Akhir ini dapat berguna dan bermanfaat bagi masyarakat yang membutuhkan juga kalangan ahli yang dapat memberikan saran serta perbaikan karena penulis merasa banyak kekurangan dalam hal pembuatan Tugas Akhir ini. Kritik dan saran yang membangun sangat diperlukan untuk memperbaiki Tugas Akhir ini.

Wassalamu'alaikum Wr, Wb.

Jakarta, May 2009

Penulis,

ABSTRAK

Salah satu komponen penunjang operasional dalam bidang teknologi yang sering kita temukan dalam kehidupan sehari-hari adalah printer, printer merupakan objek atau barang yang banyak dibutuhkan dalam menunjang aktivitas keseharian, baik hanya sebagai objek penunjang atau maupun sebagai ujung tombak bagi beberapa bidang usaha diantaranya percetakan, desain dan lain-lain.

Kemajuan ilmu pengetahuan dan teknologi dalam bidang komputer semakin pesat diantaranya perkembangan sistem pakar yang merupakan bagian dari kecerdasan buatan. Pengetahuan yang dimiliki seorang pakar dapat digabungkan dengan kemampuan komputer sehingga dapat menghasilkan sebuah sistem pakar. Sistem pakar ini akan bekerja dengan mengakses basis pengetahuan yang menampung pengetahuan mengenai berbagai kerusakan pada printer serta perbaikannya. Basis pengetahuan menggunakan sistem basis data relasional. Mesin inferensi menggunakan teknik pencarian melebar pertama yaitu sistem akan memeriksa mulai dari simpul akar yang ada pada setiap tingkat sebelum pindah ke tingkat selanjutnya. Representasi pengetahuan yang digunakan adalah metode kaidah produksi yang memiliki keuntungan yaitu mudah dalam pemeliharannya.

Penelitian ini bertujuan membangun sebuah sistem pakar yang dapat mendeteksi kerusakan pada printer. Diharapkan sistem pakar ini dapat berguna bagi masyarakat yang membutuhkan khususnya para teknisi printer.

Kata kunci : Mesin inferensi, basis pengetahuan

ABSTRACT

Printer is one of important component operational support in technology which we always meet every day. Printer is object with a lot of function that we needed in our activity every day, not only support but in the many industry like design industry; printer is very imported than else.

Nowadays the improvement of computer science and technology growth rapidly such as the expert system development which is a part of artificial intelligent. The expert knowledge can be combined with computer ability so can produce the expert system. This expert system will work by accessing the database which contain all the knowledge about various of printer printonix diseases and it's solutions. The database use the relational database management system. The inference engine use breadth first searching technique which will check from the root of every level then move to the next level. The knowledge representation used is production method that has advantage in maintenance friendly.

This research aimed to develop an expert system to detect printer printonix disturb. Hopely this expert system give an advantage for the people especially who involve in printer technicians.

Keywords : Inference Engine, Knowledge Base

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	i
LEMBAR PERNYATAAN	ii
KATA PENGANTAR	iii
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Tujuan Pembahasan	2
1.3 Batasan Masalah	3
1.4 Metodologi Perangkat Lunak	3
1.5 Sistematika Penulisan	4
BAB II LANDASAN TEORI	
2.1 Pengertian Sistem Pakar	6
2.2 Konsep Dasar Sistem Pakar	9
2.2.1 Keahlian	9
2.2.2 Ahli	10
2.2.3 Pengalihan Keahlian	11
2.2.4 Inferensi	11
2.2.5 Aturan	12
2.2.6 Kemampuan Menjelaskan	12
2.3 Struktur Sistem Pakar	13
2.3.1 Antarmuka Pengguna	14
2.3.2 Basis Pengetahuan	14

2.3.3 Akuisisi Pengetahuan	15
2.3.4 Mesin Inferensi	16
2.3.5 Pengembangan Sistem Pakar	20
2.4 Representasi Pengetahuan Kaidah Produksi	22
2.5 Diagram Use Case	23
2.6 Manajemen Sistem Basis Data	25
2.6.1 Diagram Alir Data	26
2.6.2 Model Basis Data	28
2.6.3 Teknik Normalisasi	31
2.7 Pengembangan sistem berbasis/berorientasi objek	33
2.7.1 Karakteristik Berorientasi Objek	33
2.7.2 Metodologi Berorientasi Objek	36
BAB III ANALISIS DAN PERANCANGAN	
3.1 Analisis Masalah	38
3.1.1 Inferensi	41
3.1.1 Kaidah Produksi	42
3.2 Perancangan sistem	44
3.2.1 Perancangan Aliran Data	44
3.2.1.1 Flow Chart	49
3.2.1.2 Implementasi Diagram Use Case	55
3.2.1.3 Kaidah Produksi	61
3.2.2 Perancangan Basis Data	62
3.2.3 Rancangan Layar Tampilan	70
BAB IV IMPLEMENTASI DAN PENGUJIAN	
4.1 Skenario Pengujian Sistem	78
4.2 Pengujian	81
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan	85
5.2 Saran	85
DAFTAR PUSTAKA	87

DAFTAR GAMBAR

Gambar 2.1	Arsitektur sistem pakar	14
Gambar 2.2	Penalaran Mundur	17
Gambar 2.3	Penalaran Maju	17
Gambar 2.4	Pencarian Mendalam Pertama	18
Gambar 2.5	Pencarian Melebar Pertama	19
Gambar 2.6	Pencarian Terbaik Pertama	20
Gambar 2.7	Tahapan Pengembangan Sistem Informasi	22
Gambar 2.8	Diagram Use Case	25
Gambar 2.9	Notasi Dalam Diagram Alir Data	27
Gambar 2.10	DAD Konteks Sistem Penanganan Royalti Buku	28
Gambar 2.11	Model Hirarki	29
Gambar 2.12	Model Jaringan	30
Gambar 2.13	Model Relasional	31
Gambar 3.1	Pencarian Mendalam Pertama	42
Gambar 3.2	Diagram Tree Identifikasi	43
Gambar 3.3	Diagram Konteks	45
Gambar 3.4	DAD Rinci Proses Konsultasi	45
Gambar 3.5	DAD Rinci Proses Manipulasi Data	46
Gambar 3.6	DAD Rinci Kamus Istilah	47
Gambar 3.7	DAD Rinci Pengelolaan Pengguna	47
Gambar 3.8	Diagram Hubungan Antar Entitas	48
Gambar 3.9	Menu Utama	49
Gambar 3.10	Menu File	50
Gambar 3.11	Submenu Konsultasi	51
Gambar 3.12	Submenu Manipulasi Data	52

Gambar 3.13	Menu Help	53
Gambar 3.14	Diagram Kamus Istilah	54
Gambar 3.15	Pengelolaan Pengguna	55
Gambar 3.16	Diagram Use Case	56
Gambar 3.17	Layar Login	71
Gambar 3.18	Layar Menu Utama	72
Gambar 3.19	Tampilan Submenu Konsultasi	73
Gambar 3.20	Manipulasi Data Tipe Kerusakan	74
Gambar 3.21	Manipulasi Data Gejala Umum	75
Gambar 3.22	Manipulasi Data Gejala Khusus	76
Gambar 3.23	Tampilan Kamus Istilah	77
Gambar 3.24	Tampilan Pengelolaan Pengguna	77
Gambar 4.1	Tampilan Login	82
Gambar 4.2	Tampilan Pesan pada saat Login	82
Gambar 4.3	Tampilan Menu File	83
Gambar 4.4	Tampilan Konsultasi Kerusakan	84

DAFTAR TABEL

Tabel 2.1	Table Bentuk Tidak Normal	31
Tabel 2.2	Tabel Bentuk Normal Pertama	32
Tabel 2.3	Tabel Relasi SISWA	32
Tabel 2.4	Tabel Relasi AMBILKELAS	32
Tabel 3.1	Tabel Pengelompokan Tipe Kerusakan	62
Tabel 3.2	Tabel Pengelompokan Gejala Umum	63
Tabel 3.3	Tabel Pengelompokan Gejala Khusus	64
Tabel 3.4	Tabel Tipe Kerusakan	68
Tabel 3.5	Tabel Gejala Umum	68
Tabel 3.6	Tabel Gejala Khusus	69
Tabel 3.7	Tabel Kamus Istilah	74
Tabel 3.8	Tabel Pengguna	74
Tabel 4.1	Tabel Hasil Pengujian	84

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era globalisasi sekarang ini, peran teknologi informasi sangat di butuhkan dan telah menjadi bagian dari kehidupan kita sehari-hari, kebutuhan informasi yang terus berkembang secara dinamis serta ditambah dengan persaingan dalam dunia usaha yang semakin kompetitif mendorong lahirnya ide-ide kreatif untuk menciptakan inovasi-inovasi baru yang dapat digunakan secara cepat dan tepat untuk memenuhi kebutuhan-kebutuhan tersebut.

Salah satu komponen penunjang operasional dalam bidang teknologi yang sering kita temukan dalam kehidupan sehari-hari adalah printer, printer merupakan objek atau barang yang banyak dibutuhkan dalam menunjang aktifitas keseharian, baik hanya sebagai objek penunjang atau maupun sebagai ujung tombak bagi beberapa bidang usaha diantaranya percetakan, desain dan lain-lain. Dan sebagai bagian dari barang elektronik yang bergantung pada komponen lainnya, serta sudah pasti memiliki masalah kerusakan atau gangguan dalam pengoperasiannya,. Terbatasnya tenaga ahli atau teknisi ditambah dengan mahalnya biaya sewa teknisi dan keterbatasan dana yang mungkin dialami oleh perusahaan dalam perawatan printer tersebut maka menjadi salah satu faktor penyebab terhambatnya kegiatan operasional. Untuk itu diperlukan sarana untuk penyaluran informasi tentang permasalahan pada printer yaitu suatu sumber

pengetahuan yang dapat menggantikan peran para ahli atau minimal membantu dalam proses pendeteksian permasalahan secara dini .

Salah satu keistimewaan sistem pakar yaitu dapat digunakan oleh pemakai yang bukan pakar tetapi dengan kecanggihan pemikiran seorang pakar. Permasalahan yang sering terjadi yaitu bahwa seseorang sering kebingungan tentang satu hal tetapi untuk menanyakan kepada pakarnya sering kali sulit ditemui, salah satu contohnya dalam hal bidang usaha yang membutuhkan printer untuk mencetak yaitu yang berkaitan dengan keterbatasan pengetahuan seseorang dalam mengetahui tentang kerusakan pada printer.

Atas dasar itulah penulis mencoba memberikan suatu pilihan dengan penggunaan sistem pakar untuk mendeteksi kerusakan pada printer. Diharapkan dengan adanya sistem pakar ini para pengguna printer menjadi lebih tahu dan mengerti sehingga pada akhirnya proses operasional yang menggunakan printer dapat berjalan dengan baik dan lebih cepat serta efisien.

1.2 Tujuan Pembahasan

Tujuan pembahasan dari tugas akhir ini adalah:

1. Dapat memberikan alternatif solusi sumber pengetahuan bagi para teknisi atau IT Support dalam melaksanakan tugasnya.
2. Memberikan pengetahuan bagi pengguna komputer yang memiliki printer, khususnya perusahaan-perusahaan yang sangat bergantung pada printer dalam menunjang operasional sehari-hari dalam mengakses informasi mengenai kerusakan printer. Dengan adanya sistem pakar ini diharapkan para pengguna

printer dapat mengatasi permasalahannya sendiri, mengingat terbatasnya jumlah ahli yang ada saat ini serta mahalnya biaya perawatan printer.

1.3 Batasan Masalah

Basis pengetahuan sistem pakar untuk mendeteksi kerusakan pada printer menggunakan model basis data relasional. Teknik pencarian menggunakan teknik pencarian melebar pertama dan representasi pengetahuan menggunakan kaidah produksi.

1.4 Metodologi Perangkat lunak

Dalam penulisan tugas akhir ini digunakan metode rekayasa perangkat lunak berupa metode sekuensial linier atau lebih dikenal dengan nama metode air terjun. Metode ini mengusulkan sebuah pendekatan yang sistematis mulai pada tingkat analisis, desain, penulisan kode program dan pengujian.

1. Tahap analisis

Pada tahap ini ditentukan spesifikasi guna membangun sebuah basis pengetahuan kerusakan pada printer yang akan diimplementasikan dalam sistem pakar untuk mendeteksi kerusakan pada printer.

2. Tahap desain sistem

Pada tahap ini dilakukan perancangan basis pengetahuan dan desain sistem berdasarkan hasil pada tahapan analisis.

3. Tahap penulisan kode program

Hasil perancangan dan desain sistem akan diimplementasikan kedalam bahasa pemrograman menjadi aplikasi sistem pakar untuk mendeteksi kerusakan pada printer.

4. Tahap pengujian

Proses pengujian dilakukan untuk menemukan kesalahan-kesalahan dan memastikan hasil yang diperoleh sesuai dengan kebutuhan.

1.5 Sistematika Penulisan

Penulisan Tugas Akhir ini dibagi menjadi beberapa bab yang saling berkaitan antara lain:

Bab I : PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang, tujuan pembahasan, batasan masalah, metodologi perangkat lunak dan sistematika penulisan.

Bab II : LANDASAN TEORI

Bab ini menjelaskan landasan teori yang dipakai dalam penulisan tugas akhir ini, antara lain teori mengenai sistem pakar, *unified modelling language* dan manajemen sistem basis data.

Bab III : ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai analisis masalah, perancangan basis pengetahuan yang akan diimplementasikan dalam sistem pakar dan langkah-langkah perancangan perangkat lunak lainnya.

Bab IV : IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi program yang telah dibuat serta pengujiannya.

Bab V : KESIMPULAN DAN SARAN

Bab terakhir ini akan membahas mengenai kesimpulan dari sistem pakar yang telah dibuat dan saran-saran yang diperlukan guna pengembangan sistem pakar yang lebih baik.

BAB II

LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai berbagai teori yang dipakai dalam perancangan sistem pakar untuk mendeteksi kerusakan pada printer printonix serta penjelasan mengenai printer printonix..

Printer Printonix adalah printer dengan tipe line matrix yang biasa digunakan untuk mencetak dokumen dalam jumlah lembar yyang banyak, dengan menggunakan kertas yang berkelanjutan (*continous form*). Salah satu contoh yaitu untuk mencetak data tagihan telpon dari satu district.Terdapat type dari jenis printer ini dengan keunggulanya masing-masing.

Beberapa teori yang digunakan diantaranya mengenai sistem pakar, Pengembangan aplikasi berbasis/berorientasi objek.

2.1 Pengertian Sistem Pakar

Sistem pakar adalah salah satu sub bidang komputer yang di tujukan untuk membuat software dan hardware yang sepenuhnya bisa menirukan beberapa fungsi otak manusia. Membuat software expert systems/program sebagai penasihat atau konsultan pakar. Dapat mengumpulkan dan penyimpan pengetahuan seseorang atau beberapa orang pakar kedalam komputer untuk digunakan oleh semua orang yang membutuhkan. Tidak untuk menggantikan kedudukan seorang pakar tetapi untuk memasarakatkan pengetahuan dan pengalaman pakar tersebut.

Sistem pakar adalah suatu program komputer yang cerdas yang menggunakan pengetahuan dan prosedur inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang ahli untuk menyelesaikannya (Feigenbaum,1982).

Sistem pakar merupakan suatu bidang teknik kecerdasan yang paling populer sekarang ini. Sistem pakar dapat diartikan sebagai sebuah perangkat lunak komputer yang memiliki basis pengetahuan untuk domain tertentu dan menggunakan penalaran inferensi menyerupai seorang pakar dalam memecahkan suatu masalah.

Tiap-tiap orang mempunyai keahlian masing-masing yang mungkin satu dengan yang lainnya mempunyai keahlian yang berbeda, tergantung dari keahliannya masing-masing. Ada ahli kimia, dokter dan sebagainya. Komputer dapat diprogram untuk berbuat seperti orang yang ahli dalam bidang tertentu. Komputer yang demikian dapat dijadikan seperti konsultan atau tenaga ahli di bidang tertentu yang dapat menjawab pertanyaan dan memberikan nasihat-nasihat yang dibutuhkan. Sistem demikian di sebut dengan *expert sistem* atau sistem pakar, (Jogianto H.M, 1988).

Sistem pakar adalah suatu program komputer yang dirancang untuk mengambil keputusan seperti keputusan yang diambil oleh seorang atau beberapa orang pakar. Menurut Marimin (1992), sistem pakar adalah sistem perangkat lunak komputer yang menggunakan ilmu, fakta, dan teknik berpikir dalam pengambilan keputusan untuk menyelesaikan masalah-masalah yang biasanya hanya dapat diselesaikan oleh tenaga ahli dalam bidang yang bersangkutan

Sistem pakar merupakan salah satu bidang teknik kecerdasan buatan yang cukup diminati karena penerapannya diberbagai bidang baik bidang ilmu pengetahuan maupun bisnis yang terbukti sangat membantu dalam mengambil keputusan dan sangat luas penerapannya. Kemampuan sistem pakar didukung oleh adanya basis pengetahuan yang berupa pengetahuan nonformal yang sebagian diperoleh dari pengalaman, bukan dari pengetahuan formal.

Beberapa manfaat yang diperoleh dengan mengembangkan sistem pakar antara lain:

- 1) Sistem pakar dapat digunakan setiap hari menyerupai sebuah mesin sedangkan seorang pakar tidak mungkin bekerja terus menerus setiap hari tanpa beristirahat.
- 2) Sistem pakar merupakan suatu perangkat lunak yang dapat diperbanyak dan kemudian dibagikan ke berbagai lokasi maupun tempat yang berbeda-beda untuk digunakan, sedangkan seorang pakar hanya bekerja pada satu tempat dan pada saat yang bersamaan.
- 3) Suatu sistem pakar dapat diberi pengamanan untuk menentukan siapa saja yang memiliki akses untuk menggunakannya dan jawaban yang diberikan oleh sistem terbebas dari proses intimidasi atau ancaman, sedangkan seorang pakar bisa saja mendapat ancaman atau tekanan pada saat menyelesaikan permasalahan.
- 4) Pengetahuan yang disimpan pada sistem pakar tidak akan bisa hilang atau lupa dengan syarat perawatan yang baik, sedangkan pengetahuan pakar

manusia lambat laun akan hilang karena meninggal atau usia yang semakin tua.

- 5) Biaya menggaji pakar manusia lebih mahal jika dibandingkan dengan penggunaan program sistem pakar yaitu dengan asumsi sistem pakar itu sudah ada.
- 6) Menjadikan pengetahuan dan nasihat lebih mudah didapat, menyimpan keahlian seorang pakar dan memberikan tanggapan yang cepat.

2.2 Konsep Dasar Sistem Pakar

Konsep dasar dari suatu sistem pakar menurut Turban (1993) adalah: keahlian, ahli, pengalihan keahlian, inferensi, aturan, dan kemampuan memberikan penjelasan.

2.2.1 Keahlian

Keahlian merupakan suatu penguasaan pengetahuan di bidang tertentu yang didapatkan dari pelatihan, membaca atau dari pengalaman. Beberapa pengetahuan yang merupakan keahlian antara lain:

- 1) Fakta-fakta pada lingkup suatu permasalahan tertentu.
- 2) Teori-teori pada lingkup suatu permasalahan tertentu.
- 3) Prosedur-prosedur dan aturan-aturan berkenaan dengan lingkup suatu permasalahan tertentu.
- 4) Strategi-strategi untuk menyelesaikan suatu permasalahan tertentu.
- 5) Pengetahuan tentang pengetahuan.

Bentuk tersebut memungkinkan para ahli untuk dapat mengambil keputusan para ahli untuk dapat mengambil keputusan lebih cepat dan lebih baik dari seorang yang bukan ahli.

2.2.2 Ahli

Pakar adalah seseorang yang telah mempelajari fakta- fakta, buku teks, dan pengetahuan bidangnya, serta mengembangkan pengetahuannya yang telah terdokumentasi dengan mempelajari kasus-kasus yang pernah ditemukannya. Seorang pakar memakai strategi yang mencakup mengajukan pertanyaan tertentu, melakukan pengujian kasar, mengubah taktik, mengulangi bukti-bukti dan sebagainya.

Tetapi kalau dilihat dari derajat atau level keahlian, sulit untuk mendefinisikan seorang pakar. Pertanyaannya adalah seberapa banyak keahlian yang harusnya dimiliki seseorang sebelum memenuhi syarat sebagai seorang ahli.

Secara khusus keahlian manusia termasuk kumpulan perilaku yang terlibat dalam aktivitas dibawah ini :

- Pengenalan dan formulasi masalah
- Memecahkan masalah secara cepat dan benar
- Menjelaskan solusi
- Belajar dari pengalaman
- Restruktur pengetahuan
- Pemecahan rule-rule
- Menentukan relevansi

- Kesadaran akan limit

Pakar dapat mengambil masalah yang dijanjikan dalam beberapa cara yang berubah-ubah dan mengubahnya ke bentuk yang memberikan solusi yang cepat dan efektif. Pakar harus mampu menjelaskan hasilnya, belajar hal-hal baru tentang domain, restrukturisasi pengetahuan kapanpun dibutuhkan, pemecahan rule-rule, dan menentukan kapan keahlian mereka relevan.

2.2.3 Pengalihan keahlian

Tujuan utama sistem pakar adalah pengalihan keahlian dari para ahli untuk kemudian digunakan lagi oleh orang lain yang bukan ahli. Proses ini membutuhkan empat tahapan yaitu tambahan pengetahuan dari para ahli atau sumber-sumber lainnya, representasi pengetahuan ke komputer, inferensi pengetahuan dan pengalihan pengetahuan ke pengguna yang disimpan ke dalam komputer yang dinamakan basis pengetahuan. Ada dua tipe pengetahuan yaitu fakta dan prosedur.

2.2.4 Inferensi

Kemampuan untuk menalar harus dimiliki oleh sistem pakar. Jika keahlian-keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basis data, maka komputer harus dapat di program untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk mesin inferensi (inference engine).

2.2.5 Aturan

Pengetahuan dari suatu sistem pakar mungkin dapat direpresentasikan dalam sejumlah aturan. Salah satu aturan yang paling umum untuk merepresentasikan pengetahuan adalah dalam bentuk tipe aturan IF-THEN (jika-maka).

2.2.6 Kemampuan memberikan penjelasan

Kemampuan melacak pertanggungjawaban untuk konklusi ke sumber adalah krusial dalam transfer keahlian dan dalam pemecahan masalah. Subsistem penjelasan dapat melacak tanggungjawab dan penjelasan perilaku sistem pakar dengan secara interaktif menjawab pertanyaan

Untuk lebih meningkatkan kemampuan program sistem pakar diperlukan komponen-komponen tambahan yaitu fasilitas penjelasan, kemudahan memodifikasi, kompatibilitas dan kemampuan belajar.

1) Fasilitas Penjelasan.

Sistem pakar harus mampu menjelaskan pemikiran yang dilakukan untuk mencapai kesimpulan. Penjelasan ini terdiri dari identifikasi, langkah-langkah proses pemikiran dan pernyataan kebenaran setiap langkah serta penjelasan operasi yang sedang dilakukan.

2) Kemudahan memodifikasi.

Kemudahan memodifikasi merupakan satu hal penting, karena ilmu pengetahuan berkembang terus dan kemampuan seorang pakar terus bertambah. Oleh karena itu suatu program sistem pakar harus mudah

untuk dimodifikasi terutama dalam hal basis pengetahuan dari sistem pakar tersebut.

3) Kompatibilitas

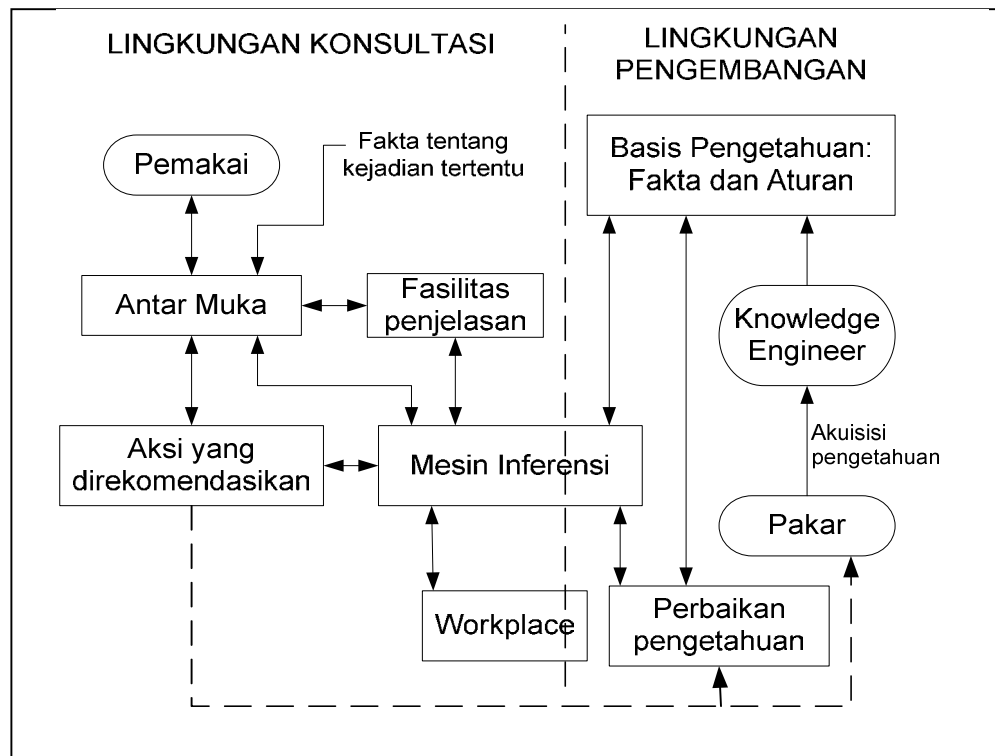
Kompatibilitas merupakan kemampuan dari program sistem pakar untuk dijalankan. Program sistem pakar dibuat untuk dijalankan pada suatu konfigurasi komputer tertentu. Hal ini kadang menyulitkan. Kemampuan suatu program sistem pakar untuk dapat dipakai dalam berbagai konfigurasi komputer merupakan salah satu nilai lebih karena akan memperluas pemakai dari sistem pakar yang dibuat.

4) Kemampuan belajar

Kemampuan suatu sistem pakar untuk menambah sendiri pengetahuannya selama berinteraksi dengan pemakai adalah kemampuan belajar dari sebuah sistem pakar. Beberapa sistem pakar pada saat ini telah memiliki kemampuan tersebut. Kemampuan belajar ini merupakan syarat utama bagi program sistem pakar di masa mendatang.

2.3 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan dan konsultasi (Turban,1995). Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat pada gambar 2.1.



Gambar.2.1 Arsitektur sistem pakar (sumber: Turban, 1995)

2.3.1 Antarmuka Pengguna (*user interface*)

Sistem pakar berisi bahasa prosesor untuk komunikasi yang *familiar*, berorientasi pada masalah antara *user* dan komputer. Komunikasi ini dibawa oleh *natural language interface*, dan dalam beberapa kasus *user interface* ditambahkan dengan menu-menu dan grafik. *Natural Language Interface* memiliki tujuan mendesain dan membangun suatu sistem komputer yang akan mampu menganalisa dan mengerti akan bahasa manusia.

2.3.2 Basis Pengetahuan (*Knowledge Base*)

Keahlian berisi pengetahuan penting untuk pengertian, formulasi, dan pemecahan masalah.

Keahlian memiliki dua elemen dasar, yaitu :

- *Facts*, seperti situasi masalah dan teori dari area masalah.
- *Heuristic* khusus atau rule-rule yang menghubungkan penggunaan pengetahuan untuk pemecahan masalah spesifik dalam sebuah domain khusus.

2.3.3 Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian pemecahan masalah dari beberapa sumber pengetahuan ke program komputer untuk konstruksi atau perluasan basis pengetahuan. Sumber-sumber pengetahuan potensial termasuk *human expert*, teksbook, database, laporan penelitian khusus dan gambar-gambar.

Pengakuisian pengetahuan dari pakar adalah tugas kompleks yang rumit dalam konstruksi sistem pakar. Dalam hal ini, dibutuhkan *knowledge engineer* untuk berinteraksi dengan satu atau lebih pakar manusia dalam membangun basis pengetahuan. Secara khusus, *knowledge engineer* membantu pakar menstruktur area masalah dengan interpretasi dan integrasi jawaban ke pertanyaan manusia, analogi penggambaran, *posing counterexamples*, dan membawa untuk menerangkan kesulitan konseptual.

Metode akuisisi dapat dilakukan secara :

1. Manual, yaitu dengan cara melakukan interview, observasi, tracking terhadap proses pengambilan/pencapaian kesimpulan/tujuan, analisa kasus, analisa kejadian kritis, komentar dan diagram konsep. Dalam metode ini, peran

dari *knowledge engineer* (orang yang berperan dalam mengakuisisi pengetahuan *expert* menjadi basis pengetahuan dalam komputer) dan *expert* sangat diperlukan.

2. Semiotomatis, yaitu dengan menggunakan alat bantu pembangun sistem basis pengetahuan. Dalam metode ini, peran *knowledge engineer* kecil sedangkan *expert* memegang peran dalam pembangunan basis pengetahuan.

3. Otomatis, yaitu pengetahuan digenerate dari kejadian yang ada. Dalam metode ini, peran pakar dan *knowledge engineer* hampir tidak diperlukan. Salah satu metode yang digunakan adalah induksi.

2.3.4 Mesin Inferensi

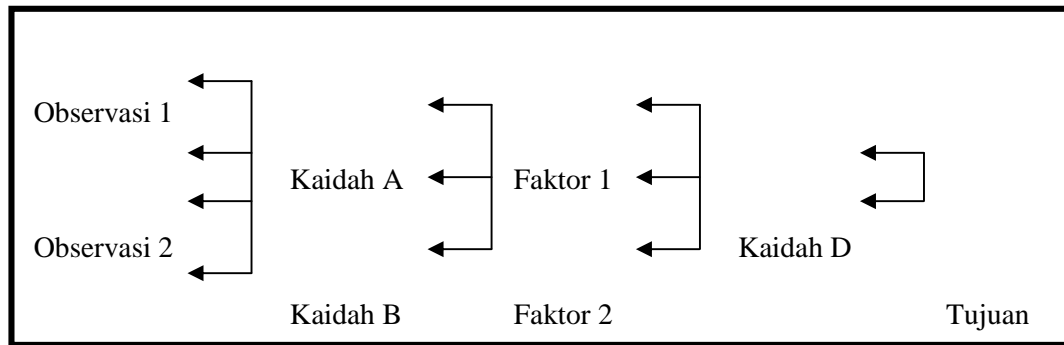
Inference engine adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam workplace, dan untuk memformulasikan kesimpulan (Turban, 1995).

Secara deduktif mesin inferensi memilih mesin yang relevan dalam rangka mencapai kesimpulan. Dengan demikian sistem ini dapat menjawab pertanyaan pemakai meskipun jawaban tersebut tidak tersimpan secara eksplisit didalam basis pengetahuan. Mesin inferensi memulai pelacakannya dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data. Secara umum ada dua tipe teknik inferensi yang ada yaitu:

1) Penalaran Mundur (*backward chaining*).

Algoritma ini dimulai dengan mencari kaidah yang sesuai dengan fakta yang 'diketahui'. Kaidah yang dievaluasi akan menghasilkan fakta baru yang selanjutnya akan menyebabkan kaidah lainnya untuk dievaluasi.

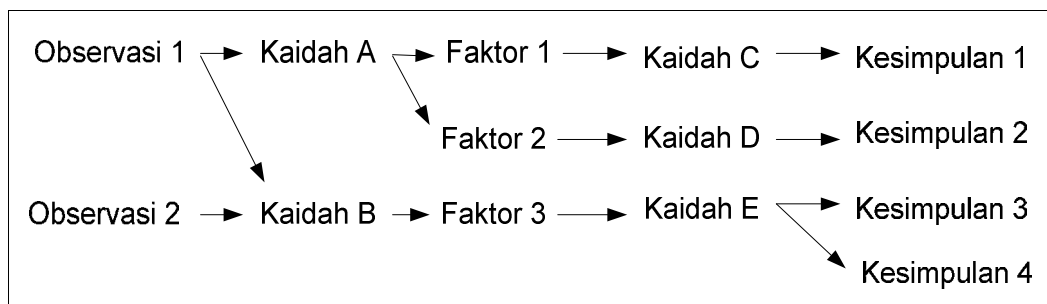
Algoritma ini akan berhenti jika tidak ada lagi fakta yang memenuhi kaidah dalam basis kaidah. Penalaran mundur dapat dilihat pada gambar 2.2.



Gambar 2.2 Penalaran Mundur

2) Penalaran Maju (*forward chaining*).

Algoritma ini merupakan sistem pembuktian teorema (theorem proving). cara kerja algoritma ini dimulai dengan fakta yang ingin dibuktikan kebenarannya, lalu kaidah-kaidah yang 'dicurigai' akan membuktikan fakta tersebut akan dievaluasi. algoritma ini akan berhenti jika fakta yang menjadi *goal* berhasil dibuktikan atau tidak berhasil dibuktikan. Penalaran maju dapat dilihat pada gambar 2.3.

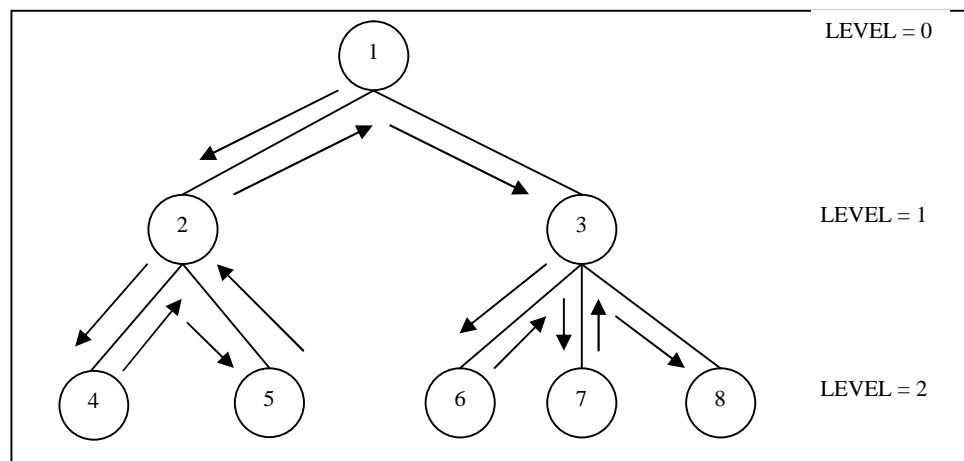


Gambar 2.3 Penalaran Maju

Kedua metode inferensi tersebut, dipengaruhi oleh tiga macam teknik penelusuran atau pencarian, yaitu:

1) Pencarian mendalam pertama (*Depth-first search*).

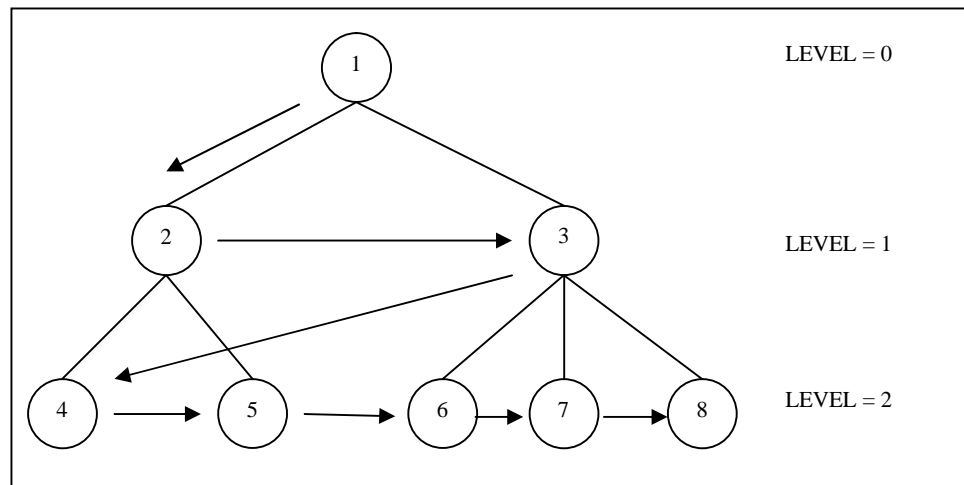
Pencarian mendalam pertama atau *Depth-first search* merupakan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan. Pada *Depth First Search*, proses pencarian akan dilakukan pada semua anaknya sebelum dilakukan pencarian ke node-node yang selevel. Pencarian dimulai dari node akar ke level yang lebih tinggi. Proses ini diulangi terus hingga ditemukannya solusi. Pencarian ini dapat dilihat pada gambar 2.4.



Gambar. 2.4 Pencarian Mendalam Pertama

2) Pencarian melebar pertama (*Breadth-first search*).

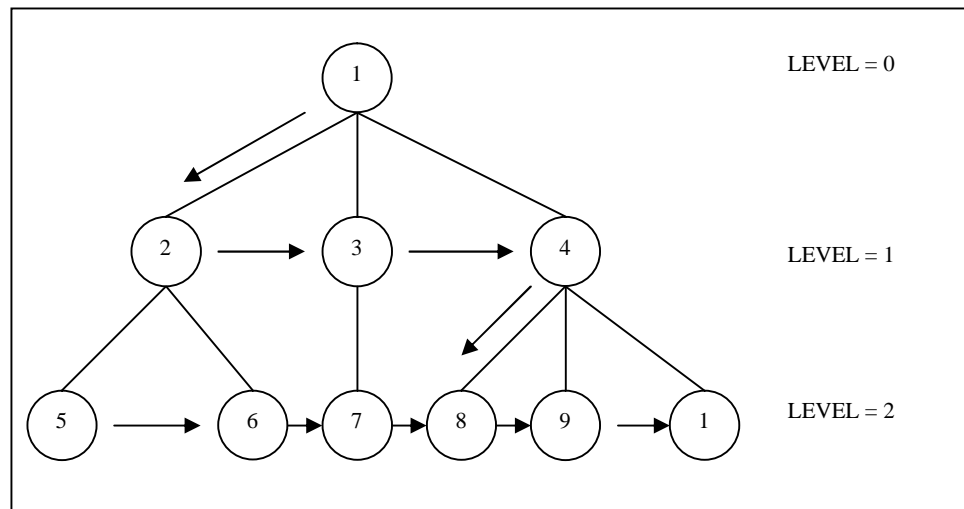
Pencarian Melebar Pertama atau *Breadth-first search* bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat selanjutnya. Metode ini memulai pencarian dari akar dan dilanjutkan ke level selanjutnya. Pencarian dilakukan dengan melacak semua node yang memiliki level yang sama sampai menemukan solusi. Pencarian ini dapat dilihat pada gambar 2.5.



Gambar. 2.5 Pencarian Melebar Pertama

3) Pencarian terbaik pertama (*Best-first search*).

Pencarian terbaik pertama atau *Best-first search* bekerja berdasarkan kombinasi kedua metode sebelumnya. Pada pencarian ini diperbolehkan mengunjungi node yang ada di level yang lebih rendah, jika ternyata pada node yang lebih tinggi memiliki nilai yang lebih buruk. Keuntungan metode ini adalah mengurangi beban komputasi karena hanya solusi yang memberikan harapan saja yang diuji dan akan berhenti apabila solusi sudah mendekati yang terbaik. Ini merupakan model yang hampir menyerupai cara manusia mengambil solusi. Metode ini dapat dilihat pada gambar 2.6.



Gambar 2.6 Pencarian Terbaik Pertama

Dalam memilih apakah akan menggunakan penalaran maju atau mundur, semuanya tergantung pada masalah yang akan dibuat sistem pakarnya, dan sebelum dibuktikan mana yang lebih baik diantara kedua metode inferensi ini.

2.3.5 Pengembangan Sistem Pakar

Proses pembuatan dan pengembangan sistem pakar melibatkan beberapa unsur yang saling berinteraksi yaitu sistem pakar itu sendiri, seorang pakar atau lebih, *knowledge engineer*, alat pengembang sistem pakar dan pemakai.

Terdapat 6 tahapan dalam pengembangan sistem :

1) Tahap Perencanaan

Tahap ini membahas permasalahan yang akan diketengahkan dalam membangun sistem pakar dan *knowledge engineer* harus berdiskusi dan bekerja sama dalam menentukan batasan yang dikaji. Pada tahap ini mulai ditentukan komponen-komponen kunci dalam sistem yang sedang dibangun.

2) Tahap Analisa

Pada tahap ini *knowledge engineer* dan pakar menentukan konsep yang akan dikembangkan menjadi sistem pakar.

3) Tahap Perancangan

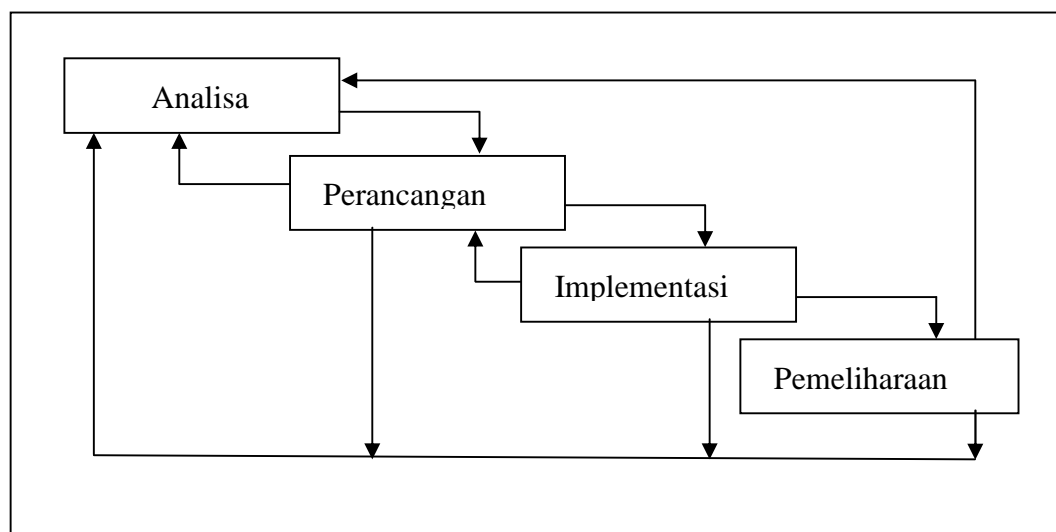
Tahap ini meliputi penataan konsep-konsep kunci, sub masalah dan bentuk aliran informasi yang telah ditentukan pada tahap sebelumnya ke dalam representasi formal yang paling sesuai dengan masalah yang ada. Hubungan antar unsur digambarkan dalam bentuk format yang biasa digunakan oleh sistem pakar. Struktur data dan teknik inferensi juga ditentukan, maka *knowledge engineer* menentukan alat pembangun sistem pakar yang digunakan.

4) Tahap Implementasi

Pada tahap ini *knowledge engineer* menerjemahkan bentuk hubungan antara unsur ke dalam bahasa pemrograman yang telah dibuat sebelumnya agar program yang diinginkan terlaksana.

5) Pemeliharaan sistem.

Pemeliharaan sistem diperlukan sehingga sistem yang dibangun tidak menjadi usang dan investasi tidak menjadi sia-sia. Pemeliharaan sistem yang paling berguna adalah proses dokumentasi sistem dimana di dalamnya tersimpan semua hal penting yang dapat menjadi tolak ukur pengembangan sistem di masa yang akan datang. Tahapan dalam pengembangan sistem dapat dilihat pada gambar 2.7



Gambar 2.7 Tahapan pengembangan sistem informasi

2.4 Representasi Pengetahuan Kaidah Produksi

Kaidah Sistem produksi menjadi acuan yang sangat sering digunakan oleh metode inferensi sistem pakar dan dalam kasus penyelesaian masalah tingkah laku manusia, atau dalam produksi sederhana. Pengetahuan dalam sistem produksi dapat direpresentasikan oleh himpunan kaidah dalam bentuk **:IF** [kondisi] **THEN** [aksi]. Contoh produksi dari adanya seseorang terkena penyakit demam berdarah yaitu :

**IF [demam tinggi AND pendarahan dihidung AND pendarahan digusi
AND mual sampai muntah darah AND bintik merah pada kulit AND
tekanan darah menurun] THEN [demam berdarah]**

Beberapa keuntungan yang dapat diperoleh dari sistem produksi yaitu :

1) Expressiveness dan intuitiveness

Karena berkaitan dengan pengalaman pekerjaan manusia dan diorganisasikan dalam bentuk saran atau nasehat.

2) Sederhana

Sederhana karena menggunakan kaidah produksi berupa IF..THEN sehingga tepat mengarah kepada pendokumentasian.

3) Kemampuan modularity dan kemampuan memodifikasi

Informasi dapat ditambahkan dan dihapus dari sistem secara esensial dan tidak mengakibatkan efek samping serta dapat menunjukkan kenaikan produksi dengan tidak mengalami pengurangan kinerja (Ciri modular).

4) Pengetahuan intensif

Pengetahuan intensif yang diberikan berupa pengetahuan murni karena setiap kaidah produksi ekuivalen untuk suatu ringkasan dan kejelasan.

2.5 Diagram Use Case

Diagram Use-Case merupakan salah satu diagram untuk menggambarkan sekelompok *use cases* dan aktor yang disertai dengan hubungan diantaranya. Diagram *use cases* ini menjelaskan dan menerangkan kebutuhan / requirement yang diinginkan / dikehendaki *user*/pengguna, serta sangat berguna dalam

menentukan struktur organisasi dan model dari pada sebuah sistem. Diagram-diagram Use-Case merupakan pusat pemodelan perilaku sistem, subsistem dan kelas. Elemen diagram Use-Case antara lain:

1. Aktor

Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah sesuatu atau seseorang yang berinteraksi dengan sistem, yaitu siapa atau apa yang menggunakan sistem. Aktor merepresentasikan peran bukan pemakai individu dari sistem. Aktor memiliki nama yang dipilih seharusnya menyatakan peran aktor. Aktor berkomunikasi dengan sistem lewat pengiriman dan penerimaan pesan. Use-Case selalu diawali oleh aktor yang mengirim pesan, disebut dengan stimulus. Ketika satu use-case dilakukan, use-case dapat mengirim pesan ke satu aktor atau lebih.

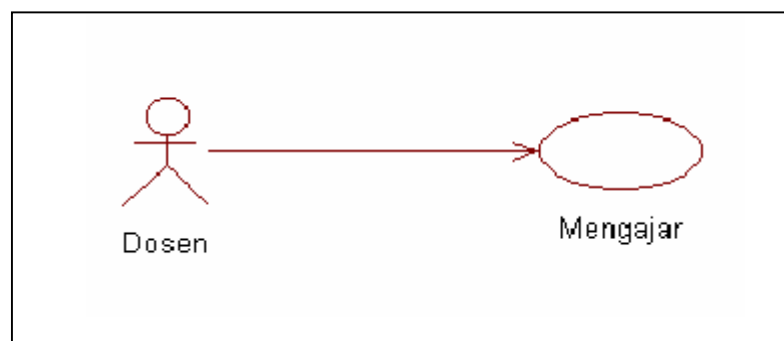
2. Use-Case

Sebuah *use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase *requirements*, model *use case* menggambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari input user dan respon-respon sistem. Setiap use case menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya. Selama pembuatan model use case secara paralel juga harus ditetapkan obyek-obyek yang terlibat dalam setiap use case.

3. Hubungan ketergantungan, generalisasi dan asosiasi.

Hubungan antar use-case dengan use-case yang lain berupa generalisasi antara lain: Include yaitu perilaku use-case merupakan bagian dari use-case yang lain dan Extend yaitu perilaku use-case memperluas perilaku use-case yang lain. Untuk menjabarkan use case dalam sistem, sangat baik bila dimulai dengan memperhatikan aktor dan actions/aksi yang mereka lakukan dalam sistem. Setiap use case menggambarkan suatu urutan interaksi antara aktor dengan sistem. Sebuah use case harus memberikan sejumlah nilai pada satu.

Diagram Use-Case digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Diagram Use-Case menyediakan cara mendeskripsikan pandangan eksternal terhadap sistem dan interaksi-interaksinya dengan dunia luar. Dengan cara ini, diagram Use-Case menggantikan diagram konteks pada pemodelan konvensional. Contoh diagram Use-Case dapat dilihat pada gambar 2.8.



Gambar 2.8 Diagram Use Case

2.6 Manajemen Sistem Basis Data

Basis Data adalah sekelompok data atau informasi yang diatur sedemikian rupa agar dapat berhubungan dengan pengolahan data dan penyediaan informasi.

Dalam basis data terdapat beberapa field. Field adalah tempat informasi ditampung. Informasi yang ada dalam suatu database dinamakan record. Record yaitu kumpulan informasi yang berhubungan dengan informasi pada field.

Suatu Database Management System (DBMS) berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi DBMS terdiri dari Database dan set program pengelola untuk menambah data, menghapus data, mengambil dan membaca data. Database adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam lingkup perusahaan atau instansi. Dalam satu file terdapat record-record yang sejenis, sama besar, sama bentuk merupakan satu kumpulan entitas yang seragam. Satu record terdiri dari field-field yang saling berhubungan untuk menunjukkan bahwa field tersebut dalam satu pengertian yang lengkap dan direkam dalam satu record. (Ir. Harianto Kristanto, *Konsep dan Perancangan Database*, Penerbit Andi Yogyakarta, 1994, halaman 1).

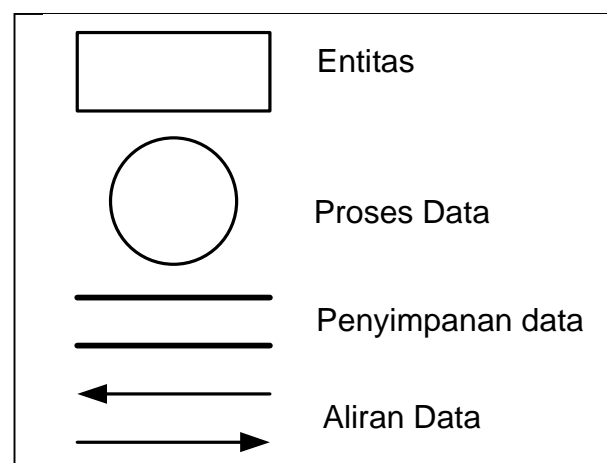
Dalam perancangan sebuah basis data, terlebih dahulu harus ditentukan ruang lingkup sistem yang akan dibuat. Hal tersebut dapat dijabarkan dengan Diagram Alir Data. Kemudian penentuan hubungan antar entitas dan pengujian basis data yang telah dibuat dengan teknik normalisasi guna menghindari kemubaziran data.

2.6.1 Diagram Alir Data (*Data Flow Diagram*)

Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD) merupakan alat yang biasa dipakai untuk mendokumentasikan proses dalam sistem. DAD

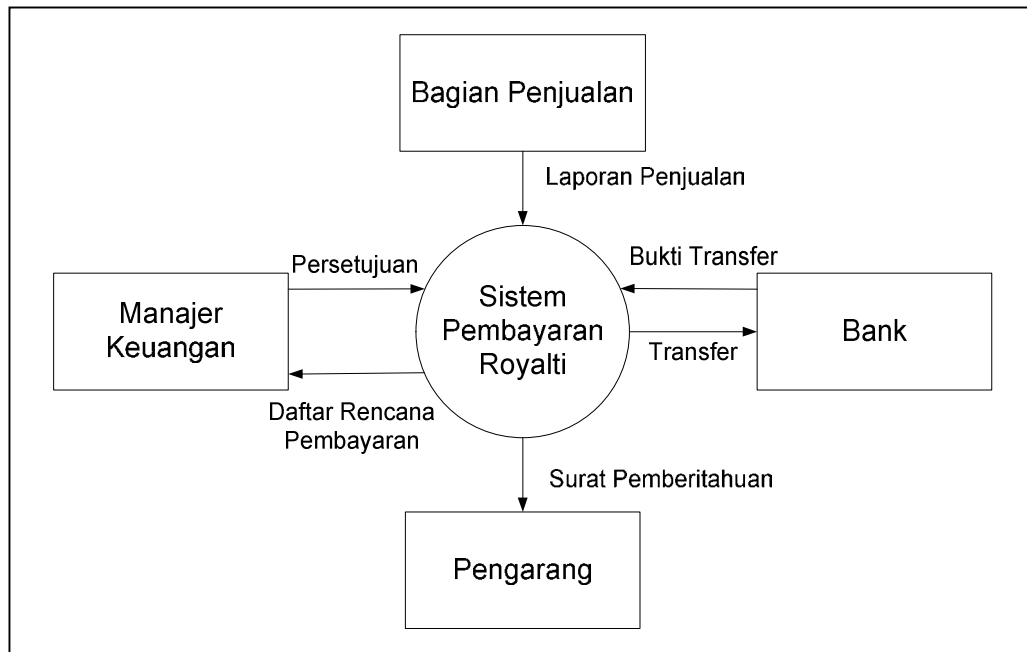
menekankan pada fungsi-fungsi di dalam sistem, cara menggunakan informasi yang tersimpan dan pemindahan informasi antar fungsi di dalam sistem. DAD konteks adalah DAD yang memperlihatkan sistem sebagai sebuah proses. Tujuannya adalah memberikan pandangan umum sistem. DAD konteks memperlihatkan sebuah proses yang berinteraksi dengan lingkungannya. Ada pihak luar atau lingkungan yang memberi masukan dan ada pihak luar yang menerima keluaran sistem. Dalam hal ini pihak luar (sering disebut *terminator*) dapat berupa sistem lain, suatu perangkat keras, orang atau organisasi. (*Abdul Kadir, Konsep dan Tuntunan Praktis Basis Data, Penerbit Andi Yogyakarta, 1994, halaman 40--41*).

Simbol-simbol yang dipakai pada DAD mempunyai nama sebagaimana diperlihatkan pada gambar 2.9.



.Gambar 2.9 Notasi Dalam Diagram Alir Data

Gambar 2.10 memperlihatkan contoh DAD konteks, yang menggambarkan secara umum sistem yang dipakai untuk menangani pembayaran royalti buku.



Gambar 2.10 DAD Konteks Sistem Penanganan Royalti Buku (sumber: Abdul Kadir,1999)

Diagram konteks di atas memberikan gambaran bahwa sistem berinteraksi dengan empat terminator: Bagian Penjualan, Manajer Keuangan, Bank, dan Pengarang. Tanda panah menyatakan masukan dan keluaran sistem. Sebagai contoh, sistem menerima masukan berupa laporan penjualan dari bagian penjualan.

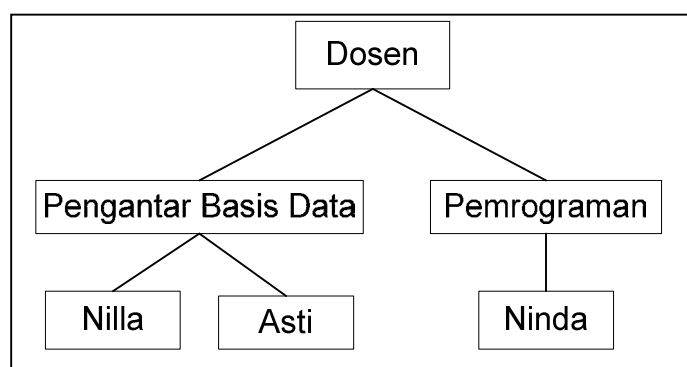
Pada tahapan selanjutnya, tahapan analisis sistem, DAD konteks akan dijabarkan ke pandangan yang lebih detail. DAD yang lebih detail daripada DAD konteks disebut DAD analisis dan DAD model.

2.6.2 Model Basis Data

Model Basis Data menyatakan hubungan antar rekaman yang tersimpan dalam Basis Data. Model dasar basis data yang paling umum ada tiga macam yaitu:

1. Model Hirarki

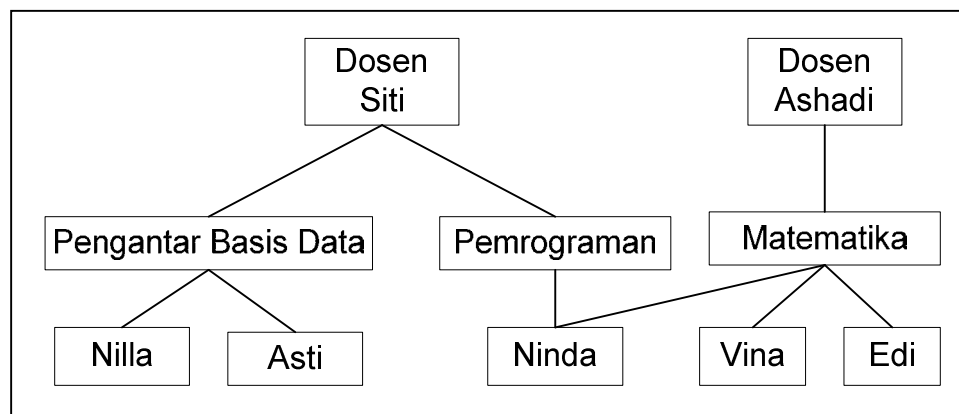
Model hirarkis biasa disebut model pohon, karena menyerupai pohon yang dibalik. Model ini menggunakan pola hubungan orang tua-anak. Setiap simpul (biasa dinyatakan dengan lingkaran atau kotak) menyatakan sekumpulan medan. Simpul yang terhubung ke simpul pada level di bawahnya disebut orang tua. Setiap orang tua bisa memiliki satu (hubungan 1:1) atau beberapa anak (hubungan 1:M), tetapi setiap anak hanya memiliki satu orang tua. Simpul – simpul yang dibawah oleh simpul orang tua disebut anak. Simpul orang tua yang tidak memiliki orang tua disebut akar. Simpul yang tidak mempunyai anak disebut daun. Adapun hubungan antara anak dan orang tua disebut cabang. Contoh dapat dilihat pada gambar. 2.11.



Gambar 2.11 Model Hirarki

2. Model Jaringan

Model ini menyerupai model hirarkis, dengan perbedaan suatu simpul anak bisa memiliki lebih dari satu orang tua. Oleh karena sifatnya demikian, model ini bias menyatakan hubungan 1:1 (satu orang tua punya satu anak), 1:M (satu orang tua punya banyak anak), maupun N:M (beberapa anak bisa mempunyai beberapa orangtua). Pada model jaringan, orang tua disebut pemilik dan anak disebut anggota.. Contoh dari model jaringan ini dapat dilihat pada gambar. 2.12.

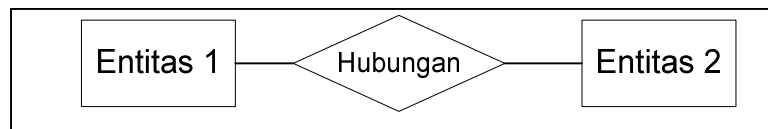


Gambar 2.12 Model Jaringan

3. Model Relasional

Model relasional adalah model data yang paling banyak digunakan saat ini.

Pembahasan pokok pada model ini adalah relasi, yang dimisalkan sebagai himpunan dari record. Deskripsi data dalam istilah model data disebut skema. Pada model relasional, skema untuk relasi ditentukan oleh nama, nama dari tiap field (atau atribut atau kolom), dan tipe dari tiap field. Contoh dapat dilihat pada gambar 2.13.



Gambar 2.13 Model Relasional

2.6.3 Teknik Normalisasi

Proses normalisasi merupakan proses pengelompokan data elemen menjadi tabel yang menunjukkan entity dan relasinya. Pada proses normalisasi selalu diuji pada beberapa kondisi. Apakah ada kesulitan pada saat menambah/insert, menghapus/delete, mengubah/update, membaca/retrieve pada satu database. Bila ada kesulitan pada pengujian tersebut maka relasi tersebut dipecahkan pada beberapa tabel lagi atau dengan kata lain perancangan belumlah mendapat database yang optimal. (*Ir. Harianto Kristanto, Konsep dan Perancangan Database, Penerbit Andi Yogyakarta, 1994, halaman 18*). Beberapa level yang biasa digunakan dan umum dipakai dalam normalisasi adalah:

1. Bentuk Normal Pertama (1NF / *First Normal Form*)

Bentuk normal pertama biasa dikenakan pada tabel yang belum ternormalisasi yaitu tabel yang memiliki atribut yang berulang. Suatu relasi dikatakan dalam bentuk normal pertama, jika dan hanya jika setiap atribut bernilai tunggal untuk setiap baris. Contoh tabel tidak normal (*unnormalized form*) dapat dilihat pada tabel 2.1.

Tabel 2.1 Table Bentuk Tidak Normal

No_siswa	Nama	Wali_Studi	Kelas1	Kelas2	Kelas3
22890100	Tanzania	Zaman	1234	1543	
22890101	Nilla	Riki	1234	1775	1543

Bentuk normal pertama dari tabel diatas dapat dilihat pada tabel 2.2.

Tabel 2.2 Tabel Bentuk Normal Pertama

No_siswa	Nama	Wali_studi	Kode_kelas
22890100	Tanzania	Zaman	1234
22890100	Tanzania	Zaman	1543
22890101	Nilla	Riki	1234
22890101	Nilla	Riki	1775
22890101	Nilla	Riki	1543

2. Bentuk Normal kedua (2NF / *Second Normal Form*)

Bentuk normal kedua memiliki syarat bentuk data telah memenuhi kriteria bentuk normal pertama. Atribut bukan kunci haruslah bergantung secara fungsi kepada kunci utama. Untuk membentuk table yang memenuhi bentuk normal kedua haruslah sudah ditentukan kunci untuk setiap field. Kunci ini harus unik dan mewakili atribut yang menjadi anggotanya. Dari contoh table normal pertama yang kemudian dirubah menjadi tabel yang memenuhi bentuk normal kedua, dapat dilihat pada gambar dan tabel 2.3 dan tabel 2.3.

Tabel 2.3 Tabel Relasi SISWA

No_siswa	Nama	Wali_studi
22890100	Tanzania	Zaman
22890101	Nilla	Riki

Tabel 2.4 Tabel Relasi AMBILKELAS

No_siswa	Kode_kelas
22890100	1234
22890100	1543
22890101	1234
22890101	1775
22890101	1543

3. Bentuk Normal Ketiga (3 NF / *Third Normal Form*)

Tabel yang memenuhi bentuk normal ketiga sebelumnya harus memenuhi bentuk normal kedua. Suatu table dapat dikatakan memenuhi bentuk normal ketiga yaitu setiap atribut bukan kunci haruslah bergantung hanya pada kunci primer secara menyeluruh. Contoh bentuk normal ketiga dapat dilihat pada contoh bentuk normal kedua diatas karena seluruh atribut yang ada bergantung penuh pada kunci primernya.

2.7 Pengembangan Sistem berbasis/berorientasi objek

Pemrograman berorientasi Objek atau *Object Oriented Programming* (disingkat OOP) adalah salah satu pendekatan pemrograman atau paradigma untuk pengembangan/development suatu perangkat lunak komputer dimana dalam struktur perangkat lunak tersebut didasarkan kepada interaksi objek dalam penyelesaian suatu proses / tugas. Dalam konsep *object oriented* akan kita temukan kata objek dan class, class merupakan pola / template yang menggambarkan kumpulan objek yang mempunyai sifat yang sama, perilaku, atau disebut dengan himpunan objek sejenis. Sementara objek adalah implementasi dari class.

2.7.1. Karakteristik *Object Oriented*

Konsep *object oriented* memiliki karakteristik utama yaitu :

1) Class

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh Class kerusakan printer adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang

menunjuk pada berbagai macam detil/keterangan dari kerusakan printer. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2) Objek

Abstraksi dari sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat kejadian, struktur, status atau hal-hal lain yang bersifat abstrak. Objek juga merupakan suatu entitas yang mampu menyimpan informasi dan mempunyai operasi yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Contoh dari objek pada sistem pakar ini adalah objek kerusakan pada printer.

3) Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari “pelaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan

berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4) Enkapsulasi

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

○ *Polimorphism*

Melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan “gerak cepat”, dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda

di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

○ *Inheritance*

Mengatur polimorfisme dan enkapsulasi dengan mengizinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada - objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas.)

2.7.2. Metodologi *Object Oriented*

Metodologi *object oriented* adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemograman berorientasi objek dan pengujian berorientasi objek. Metodologi *object oriented* memiliki beberapa keuntungan diantaranya :

- Meningkatkan produktivitas kerja

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut.

- Kecepatan pengembangan

Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

- Kemudahan pemeliharaan

Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dari pola-pola yang mungkin sering berubah-ubah.

- Adanya konsistensi

Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan, maupun pengkodean.

- Meningkatkan kualitas perangkat lunak

Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai pembuatan basis pengetahuan yang akan diimplementasikan dalam rancangan sistem pakar untuk mendeteksi kerusakan pada printer.

3.1 Analisa Masalah

Dalam membangun sebuah sistem pakar yang digunakan mendeteksi kerusakan pada printer Printronix, diperlukan pemilihan model basis data yang sesuai untuk menyimpan sebuah basis pengetahuan. Basis pengetahuan ini mempresentasikan data mengenai gejala kerusakan printer serta perbaikannya.

Pengetahuan kerusakan pada printer didasarkan pada hasil studi pustaka yaitu referensi dari buku-buku pengetahuan mengenai berbagai kerusakan pada printer. Hasil dari studi pustaka tersebut kemudian dicocokkan dengan pengetahuan yang dimiliki oleh pakar printer. Pakar yang dimaksud adalah teknisi yang ahli dalam bidang perawatan dan perbaikan printer. Pengetahuan yang didapat dari hasil studi pustaka kemudian dicocokkan dengan hasil wawancara dengan para ahli tersebut. Pengetahuan ini kemudian akan dipergunakan dalam basis pengetahuan sistem pakar. Identifikasi masalah ini merupakan tahapan awal dalam proses pengembangan sebuah sistem pakar.

Untuk memudahkan sistem dalam menganalisa permasalahan uraian kelompok kerusakan utama maka perlu dibuat suatu aturan untuk menguraikan kelompok tersebut dalam beberapa kelompok yang lebih banyak serta beberapa

tingkatan uraian yang dapat menjelaskan secara detil mengenai gejala-gejala yang didapat dari kelompok kerusakan pada tingkatan sebelumnya. Dan dari hasil penelusuran masalah utama serta gejala-gejala kerusakan pada printer printronik ini maka dibuat menjadi tiga tingkatan penelusuran yang terdiri dari :

- 1) Tingkatan pertama, berisikan kelompok kerusakan utama yang terdiri dari empat kelompok kerusakan utama yang merupakan hasil dari identifikasi masalah utama pada printer printronik kelompok pertama ini disebut juga sebagai kelompok tipe kerusakan.
- 2) Tingkatan kedua yaitu uraian permasalahan dari kelompok kerusakan utama dimana uraian-uraian ini merupakan penjelasan umum atau gejala-gejala umum yang dapat diuraikan lagi menjadi penjelasan yang lebih detil. Satu kelompok pada tingkatan pertama dapat memiliki lebih dari satu penjelasan/gejala umum. kelompok ini disebut juga sebagai kelompok kerusakan umum.
- 3) Tingkatan ketiga yaitu tingkatan terakhir yang menjelaskan secara detil tentang penjelasan umum sehingga pada tingkatan ini sistem bisa memberikan kesimpulan jawaban dari uraian-uraian permasalahan diatas, Satu tingkatan umum dapat memiliki lebih dari satu penjelasan/gejala detil. kelompok ini dapat disimpulkan sebagai kelompok kerusakan khusus.

Masing-masing kelompok tingkatan tersebut memiliki hubungan dengan tingkatan lainnya yang ada di bawahnya yang berfungsi sebagai identitas penghubung yang

dapat mengidentifikasi dari setiap kelompok masalah serta penjelasannya hingga kesimpulan akhir.

Pada tingkatan pertama, setelah mekanisme data yang dibutuhkan untuk membangun sistem pakar selesai diidentifikasi maka didapatkan gambaran global pengetahuan untuk mendeteksi kerusakan printer yang dikelompokkan menjadi:

- 1) Kerusakan pada spare part printer yang terdiri atas bagian-bagian dari perangkat keras printer itu sendiri.
- 2) Kerusakan pada jaringan/network yang terdiri atas jaringan berupa hardware penghubung antara printer dan computer.
- 3) Kerusakan pada perangkat lunak printer yang terdiri atas software yang ada di dalam memory printer.
- 4) Kerusakan pada computer.

Keempat kelompok diatas diuraikan menjadi beberapa gejala dan identifikasi kerusakan beserta usulan solusi..

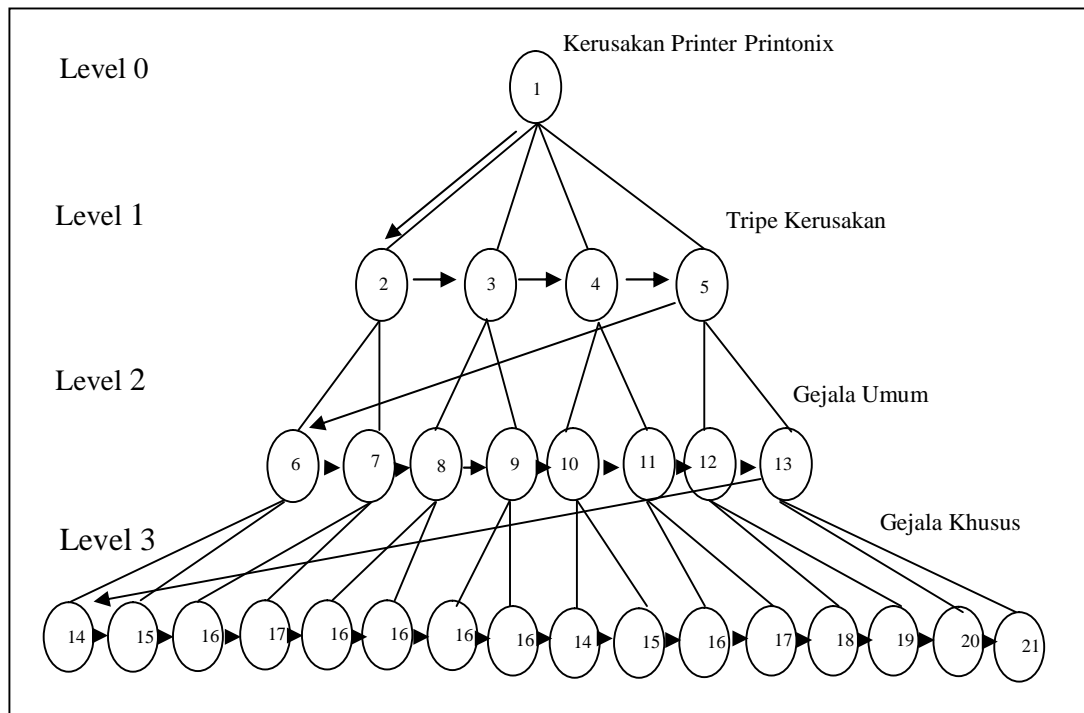
Sistem pakar ini melibatkan orang awam sebagai user yang menggunakan sistem ini untuk mencari solusi dari permasalahan yang dialami dan ahli sebagai pakar yang berperan dalam memasukan data master kerusakan printer yang merupakan data-data yang akan dipakai oleh user untuk mengkonsultasikan permasalahannya. Maka itu sistem yang di buat nanti harus dapat membedakan antara user biasa dan user pakar dimana memiliki akses lebih banyak dari pada user biasa. Untuk membantu user dalam proses konsultasi perbaikan printer maka perlu ditambahkan menu-menu tambahan yang dapat menyajikan informasi

mengenai istilah-istilah kata yang tidak umum yang digunakan pada sistem ini, serta informasi nama-nama distributor yang menyrdiakan spare part printer ini beserta tempat service printer

3.1.1 Inferensi

Untuk membangun sistem ini di butuhkan suatu metode yang dapat memudahkan dalam proses penalaran. Berdasarkan uraian dari indetifikasi permasalahan dimana usulan solusi yang merupakan hipotesa didapat dari uraian gejala hasil identifikasi yang merupakan fakta. Sehingga hipotesa didapat dari fakta-fakata yang ada, maka metode inferensi yang akan dipakai adalah metode penalaran maju yang memiliki cara membentuk penalaran dari fakta menuju hipotesa atau kesimpulan bukan penalaran mundur dimana penalaran tersebut dimulai dari hipotesa dalam hal ini solusi yang digunakan untuk mendapatkan sekumpulan fakta.

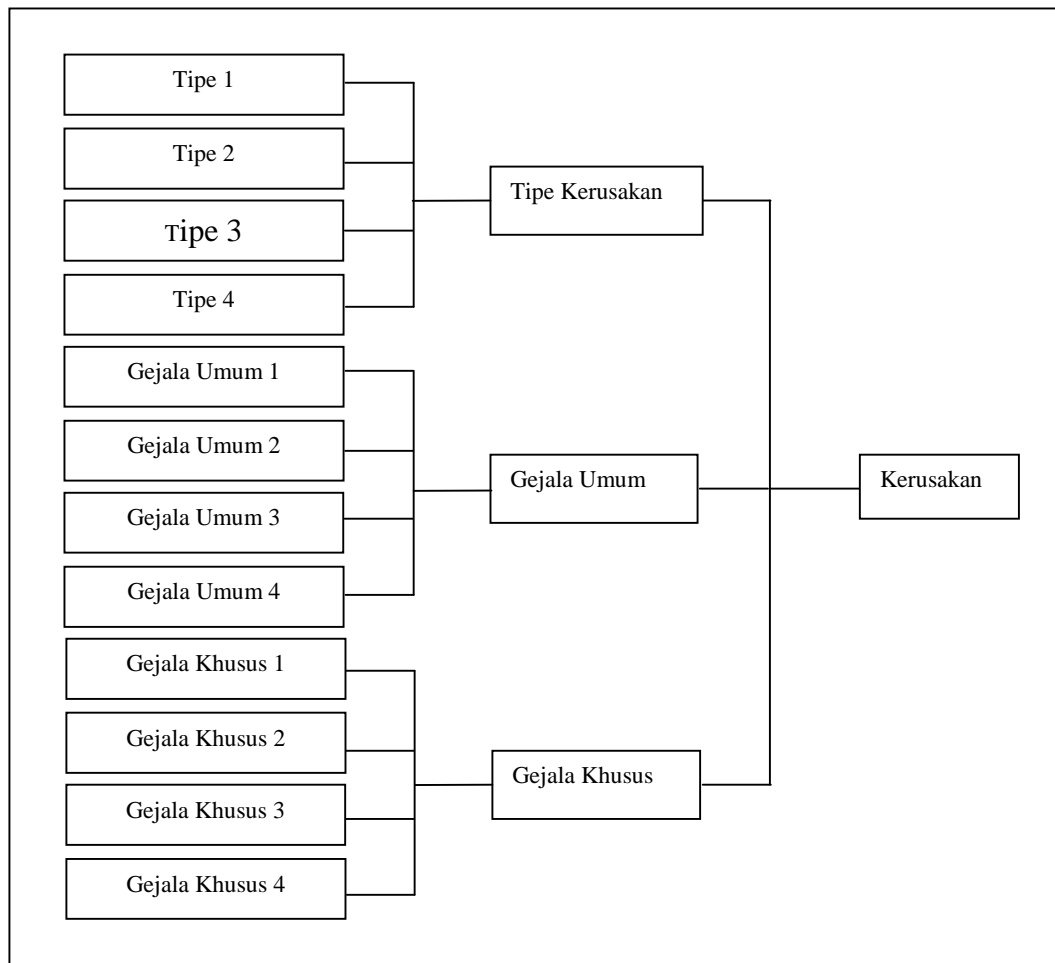
Teknik pencarian yang paling tepat untuk sistem pakar ini adalah teknik pencarian melebar pertama yaitu pencarian yang bergerak dari simpul akar yang paling atas dan menguji setiap tingkat atau level sebelum pindah ke level selanjutnya. Pencarian ini dapat dilihat pada gambar 3.1.



Gambar. 3.1 Pencarian Mendalam Pertama

3.1.2 Kaidah produksi

Metode representasi pengetahuan yang akan dipakai pada sistem pakar ini adalah metode kaidah produksi. Kaidah produksi dikatakan sebagai implikasi dua bagian, yaitu bagian premise atau kondisi dan bagian konklusi atau maka. Apabila bagian premise dipenuhi maka bagian konklusi akan bernilai benar. Bila bagian premise tidak dipenuhi maka akan melompat ke bagian premise lainnya berikutnya. Suatu kaidah produksi dapat terdiri dari beberapa premise dan lebih dari satu konklusi.



Gambar. 3.2 Diagram Tree Identifikasi Kerusakan Printer Printonix

Kaidah produksi ini dipilih karena mudah dimengerti dan penjelasan yang diperoleh mudah diturunkan dan dimodifikasi. Contoh kaidah produksi dari sistem pakar ini adalah sebagai berikut :

Rule 1 :

IF Tipe Kerusakan 1 dan Gejala Umum1 dan Gejala Khusus 1

THEN

Kerusakan 1

Rule 2 :

IF Tipe Kerusakan 2 dan Gejala Umum 2 dan Gejala Khusus 2

THEN

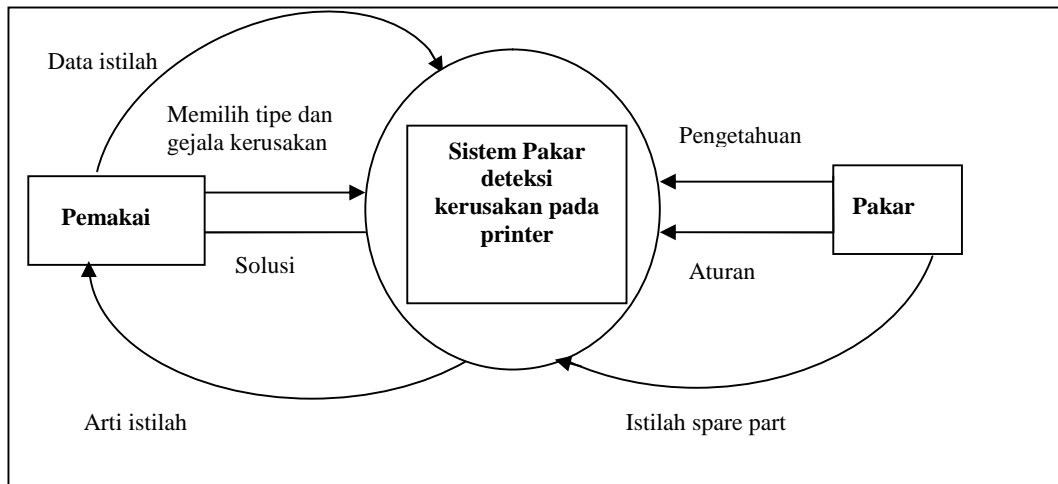
Kerusakan 2

3.2 Perancangan Sistem

Setelah tahapan analisis permasalahan, langkah selanjutnya adalah melakukan perancangan sistem yang terdiri dari perancangan aliran data dan perancangan struktur basis data.

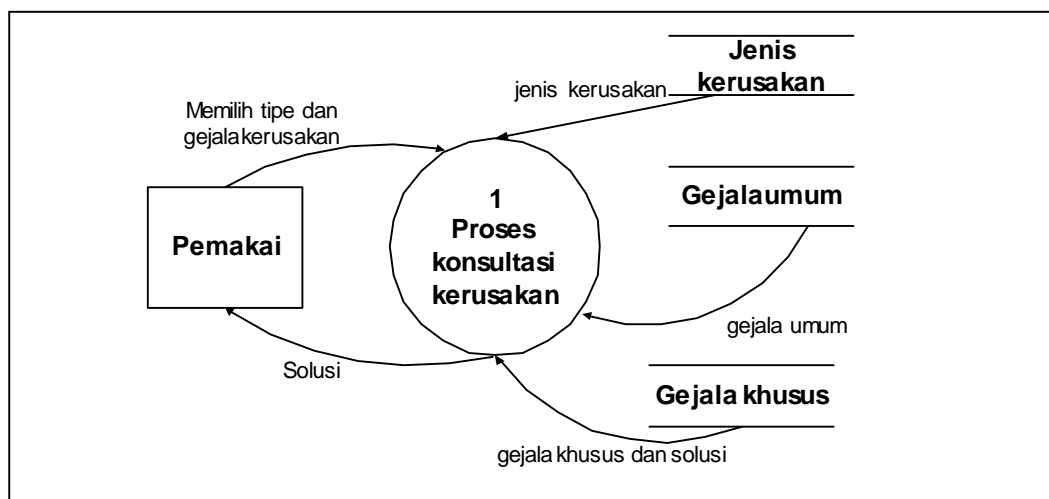
3.2.1 Perancangan Aliran Data

Untuk mendapatkan tabel yang sesuai dengan kebutuhan dalam perancangan sistem pakar ini, terlebih dahulu akan dijelaskan mengenai aliran data dalam sistem pakar untuk mendeteksi kerusakan pada printer. Gambar 3.3 menjelaskan aliran data yang masuk dan keluar dari sistem pakar secara umum. Terdapat dua eksternal entitas yaitu pemakai dan pakar. Aliran data yang masuk dari pemakai adalah pemakai memilih tipe kerusakan, gejala umum dan gejala khusus. Pemakai juga dapat memilih arti istilah yang kurang dimengerti. Keluaran atau output dari sistem yang mengalir ke pemakai adalah solusi dan arti istilah. Aliran data yang masuk dari entitas pakar adalah masukan mengenai pengetahuan, aturan dan istilah Spare part.



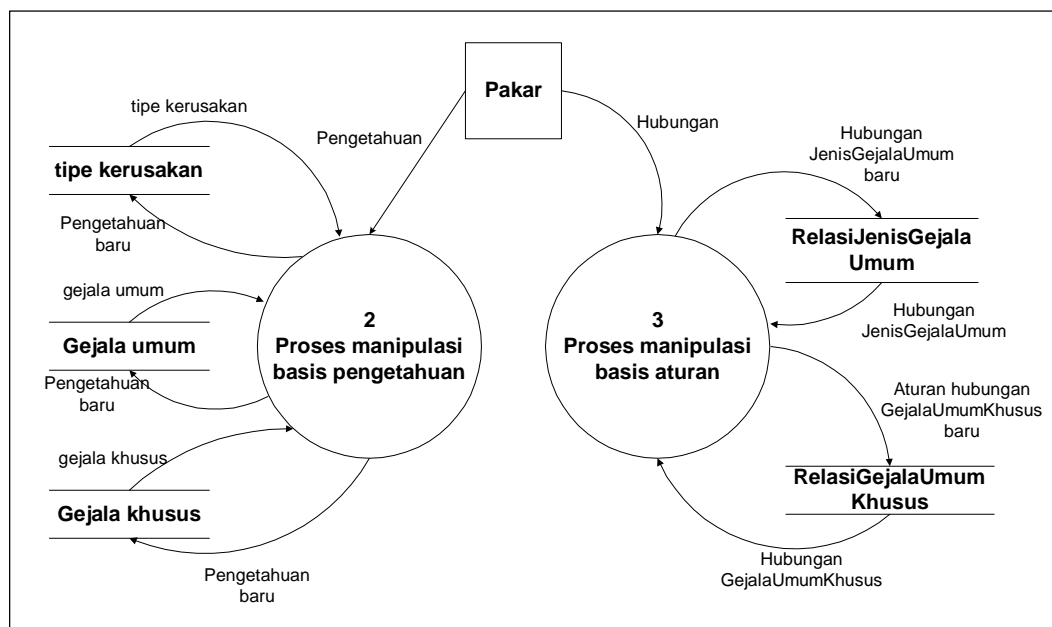
Gambar 3.3 Diagram Konteks

Aliran data lebih rinci pada proses konsultasi kerusakan dapat dilihat pada gambar 3.4. proses ini menggambarkan alir data di mulai pada saat pemakai memilih tipe kerusakan, gejala umum dan gejala khusus. Data-data tipe kerusakan, gejala umum dan gejala khusus diperoleh dari simpanan data tipe kerusakan, gejala umum dan gejala khusus. Output yang diperoleh oleh pemakai adalah solusi.



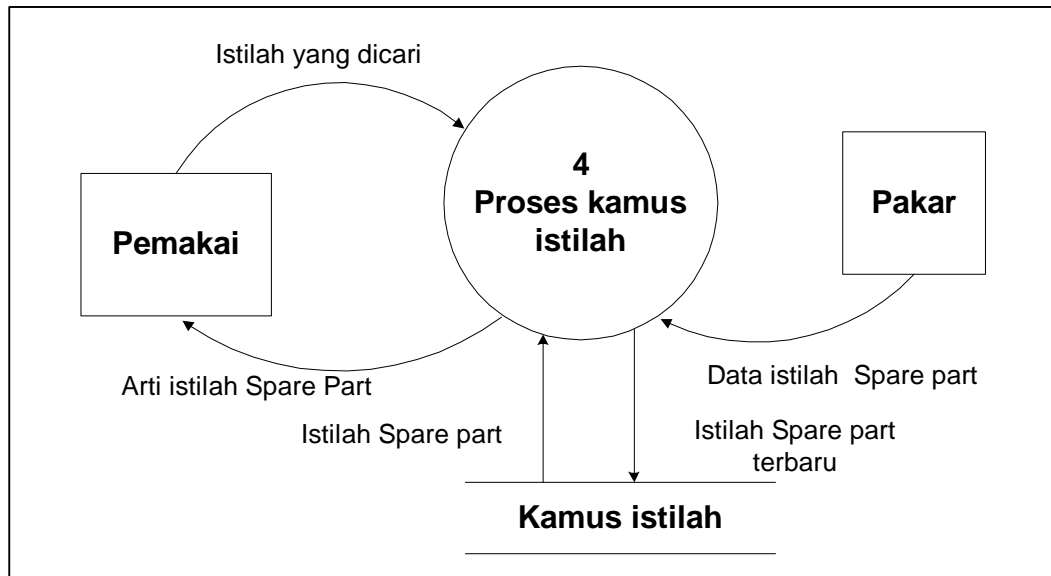
Gambar 3.4 DAD Rinci Proses Konsultasi

Pada proses manipulasi data terdapat proses manipulasi basis pengetahuan dan basis aturan. Proses manipulasi basis pengetahuan dimulai dari data masukan berupa pengetahuan mengenai tipe kerusakan, gejala umum, gejala khusus dan solusi. Pengetahuan ini disimpan dalam simpanan data berupa simpanan tipe kerusakan, gejala umum dan gejala khusus. Pada proses manipulasi basis aturan, aliran data yang mengalir adalah aturan yang berasal dari pakar yang disimpan pada simpanan data tipe kerusakan, gejala umum dan gejala khusus.



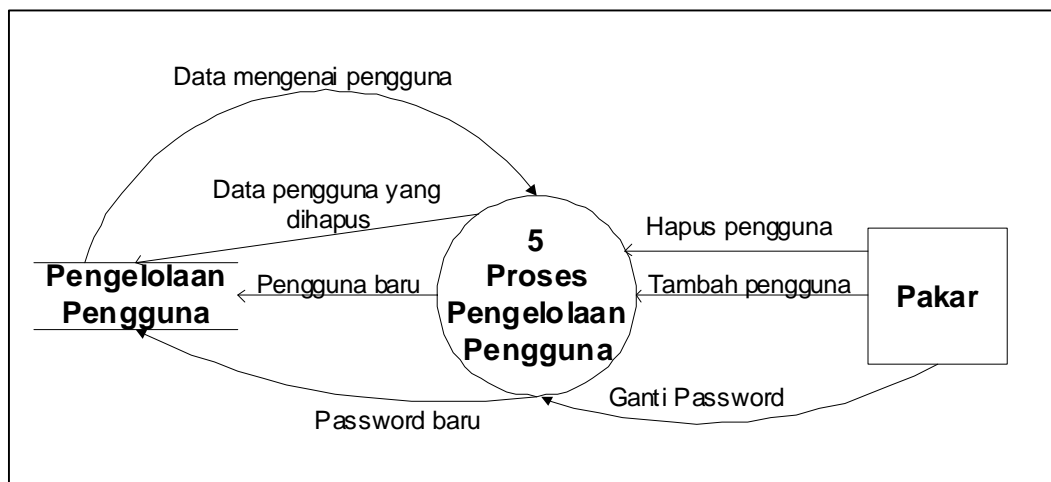
Gambar 3.5 DAD Rinci Proses Manipulasi Data

Aliran data proses kamus istilah dimulai pada saat pemakai memasukan data istilah yang tidak dimengerti. Hasil keluaran berupa informasi istilah Spare part. Data pengetahuan mengenai istilah mengalir dari pakar yang disimpan dalam simpanan data kamus istilah.



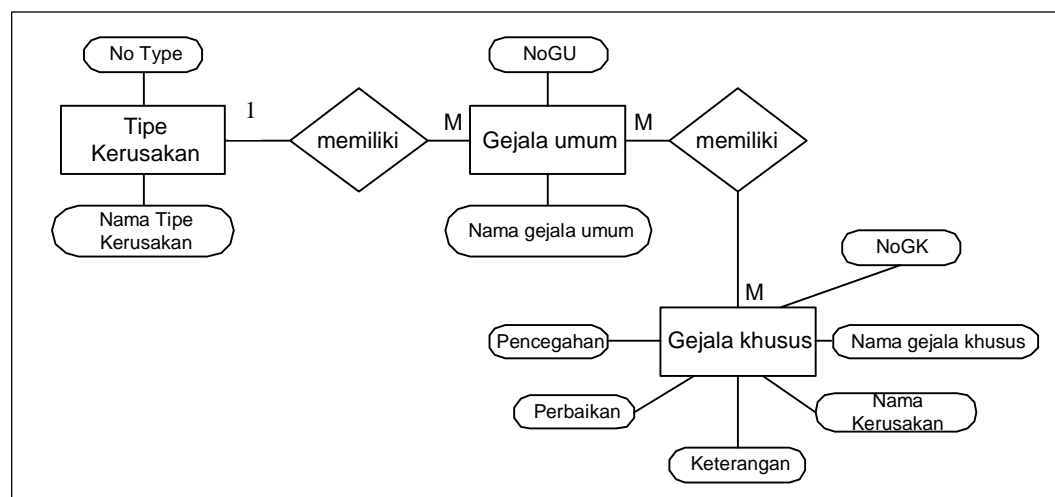
Gambar 3.6 DAD Rinci Kamus Istilah

Aliran data masukan pada proses pengelolaan pengguna berupa password baru, data pengguna baru dan data pengguna yang dihapus yang mengalir dari pakar. Simpanan data pengelolaan pengguna memberikan data mengenai pengguna kepada sistem pengelolaan pengguna.



Gambar 3.7 DAD Rinci Pengelolaan Pengguna

Hubungan simpanan data antara tipe kerusakan, gejala umum dan gejala khusus pada proses diatas dapat digambarkan dalam diagram hubungan antar entitas. Hubungan tipe kerusakan dan gejala umum adalah satu ke banyak dimana satu tipe kerusakan dapat memiliki banyak gejala umum. Hubungan antara gejala umum dan gejala khusus adalah satu ke banyak dimana satu gejala umum dapat memiliki banyak gejala khusus. Simpanan data gejala khusus merupakan data yang unik dimana satu gejala khusus hanya dimiliki oleh satu nama kerusakan yang akan menentukan solusi dari kerusakan tersebut. Hubungan antar entitas tersebut dapat dilihat pada gambar 3.8.



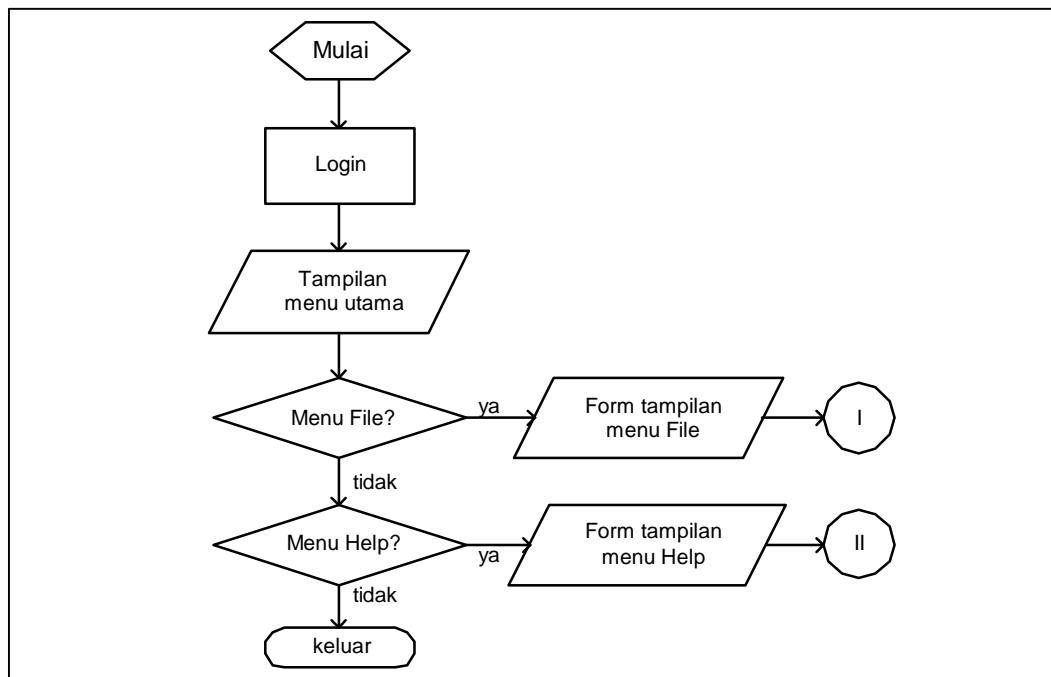
Gambar 3.8 Diagram Hubungan Antar Entitas

Dari hubungan antar simpanan data tipe kerusakan, gejala umum dan gejala khusus, dapat dibuat menjadi tiga buah tabel yang akan digunakan dalam proses konsultasi dan proses manipulasi data. Tabel kamus istilah digunakan untuk proses kamus istilah Spare part. Tabel tambahan berupa tabel pengguna yang akan menyimpan data mengenai kata kunci untuk pakar atau admin. Aliran data yang mengalir pada setiap proses dapat dilihat pada kamus data sebagai berikut:

- a. Data Tipe kerusakan = { Kode Tipe + keterangan }
 - b. Data Gejala umum = { Kode + Gejala }
 - c. Data Gejala khusus = { Kode + Gejala + Kerusakan + Pencegahan + Perbaikan }
 - d. Kamus Istilah = { ID + Kata + Definisi }
- Pengguna = { Pakar + Password }

3.2.1.1 Flow Chart Sistem

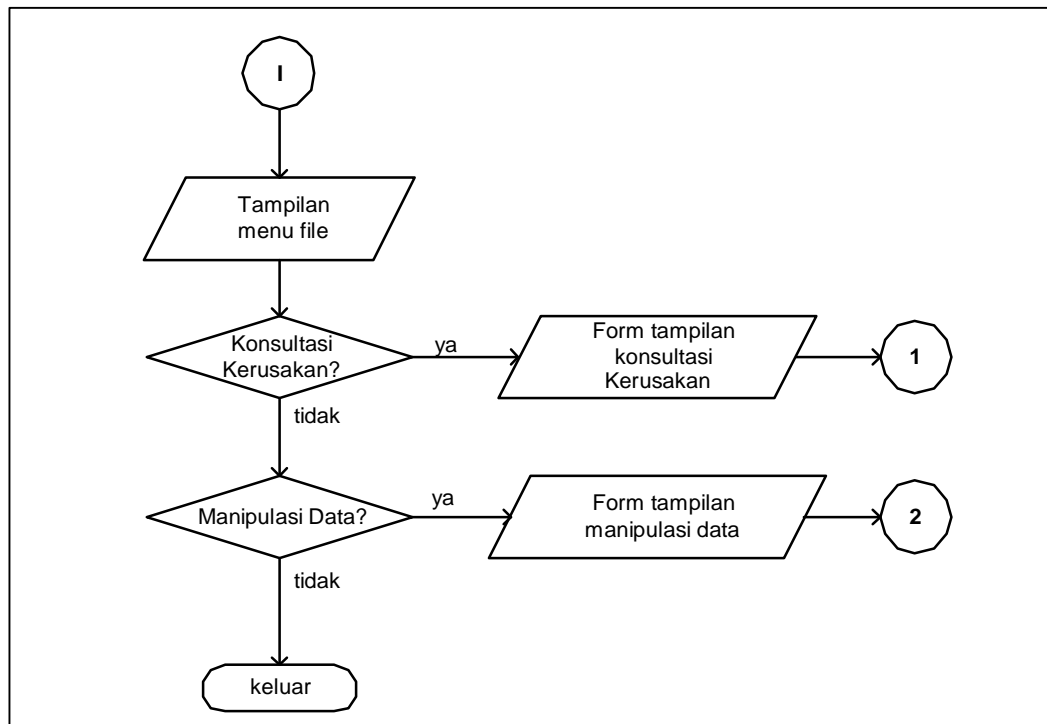
Dalam menggambarkan proses yang terjadi dalam program sistem pakar diagnosa kerusakan pada printer digunakan beberapa flow chart. Sistem akan menampilkan tampilan menu utama yang berisi menu file dan help. Gambar flow chart untuk proses pada menu utama dapat dilihat pada gambar 3.9.



Gambar 3.9 Menu Utama

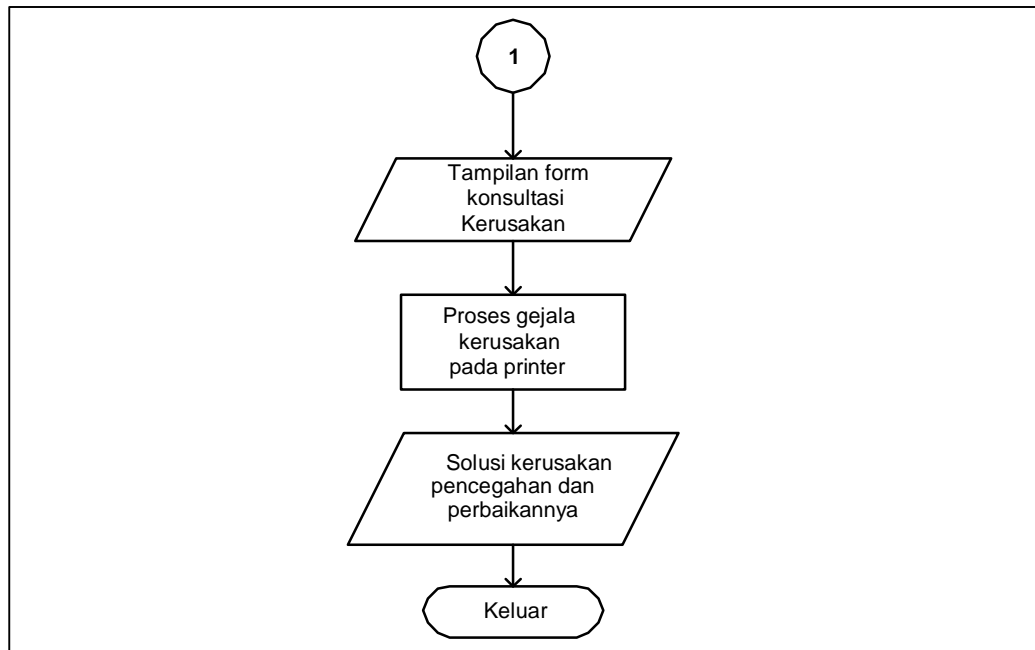
1. Flow chart untuk menu file

Flow chart pada menu file menggambarkan proses setelah pemakai memilih menu file. Menu file terdiri dari submenu konsultasi dan manipulasi pengetahuan. Gambar flow chart menu file dapat dilihat pada gambar 3.10.



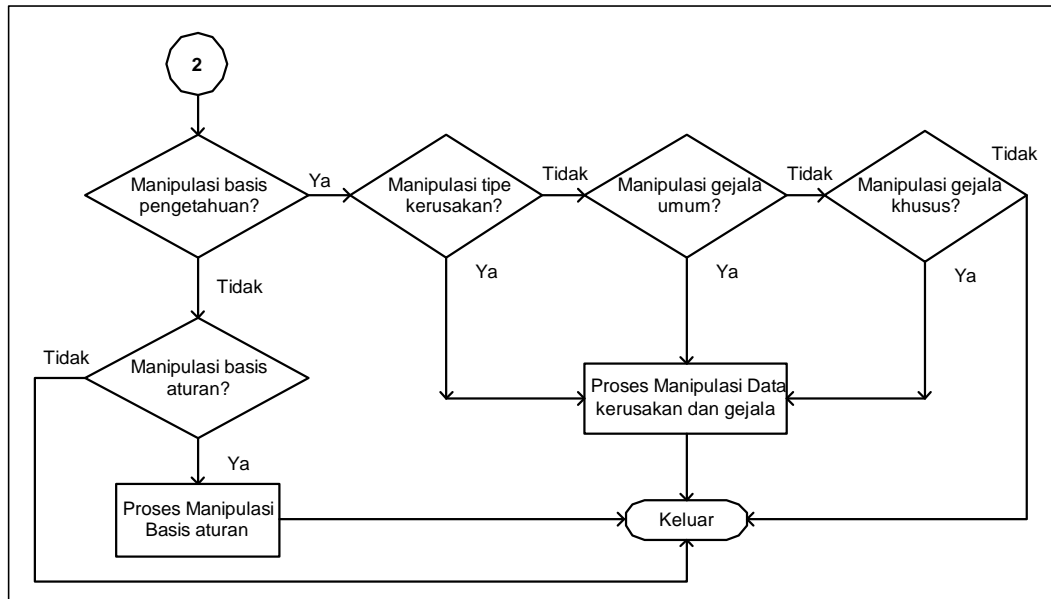
Gambar 3.10 Menu File

Pada saat pemakai memilih submenu konsultasi, sistem akan menampilkan tampilan menu konsultasi. Pemakai akan memilih tipe dan gejala kerusakan sampai solusi yang sesuai ditemukan. Alur submenu konsultasi dapat dilihat pada gambar 3.9.



Gambar 3.11 Submenu Konsultasi

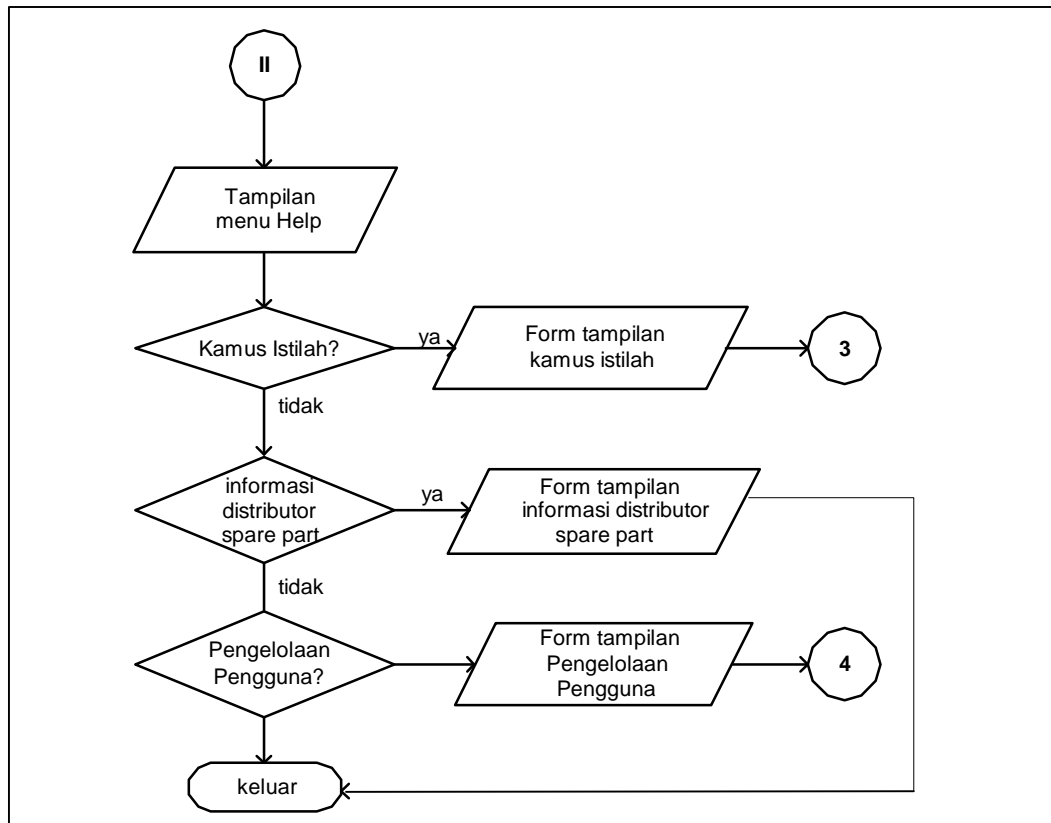
Submenu manipulasi data hanya dapat diakses oleh pakar atau admin setelah memasukkan password atau kata kunci. Alur proses ini dimulai ketika pakar memilih submenu manipulasi data. Pakar dapat memilih manipulasi basis pengetahuan atau basis aturan. Pada manipulasi basis pengetahuan, pakar dapat memilih manipulasi tipe kerusakan, gejala umum atau gejala khusus. Pada basis aturan, pakar dapat memanipulasi hubungan antara tipe kerusakan, gejala umum atau gejala khusus yang akan memberikan solusi. Gambar alur proses submenu manipulasi data dapat dilihat pada gambar 3.12.



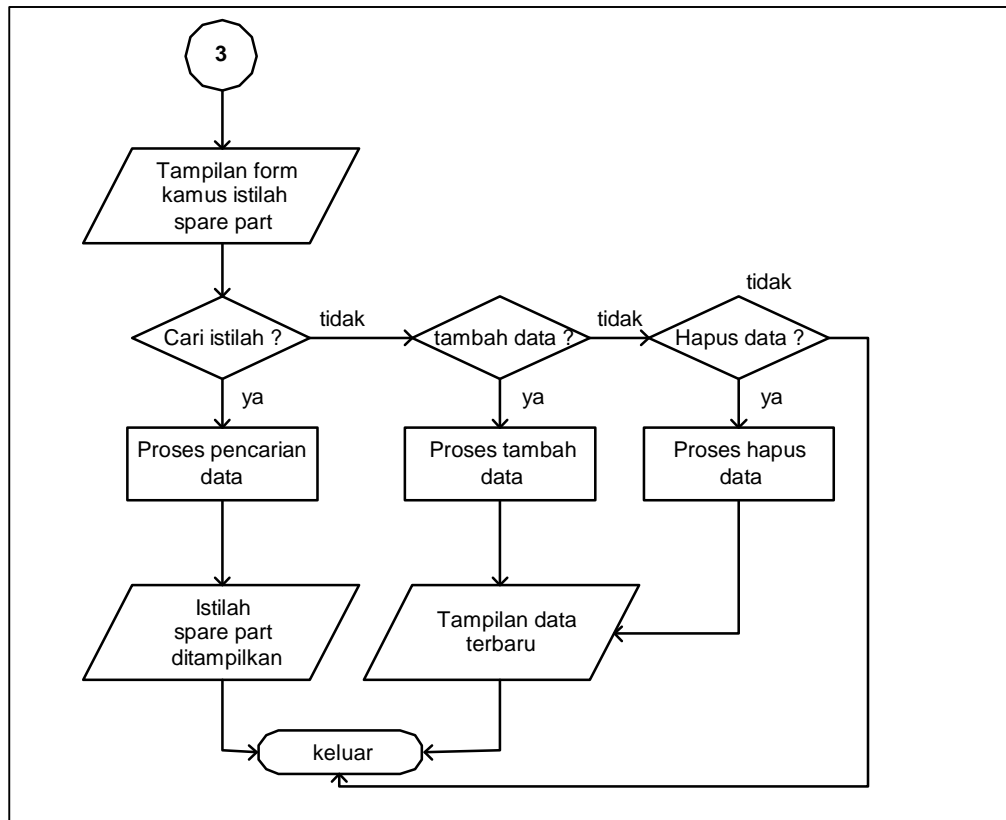
Gambar 3.12 Submenu Manipulasi Data

2. Flow Chart untuk menu help

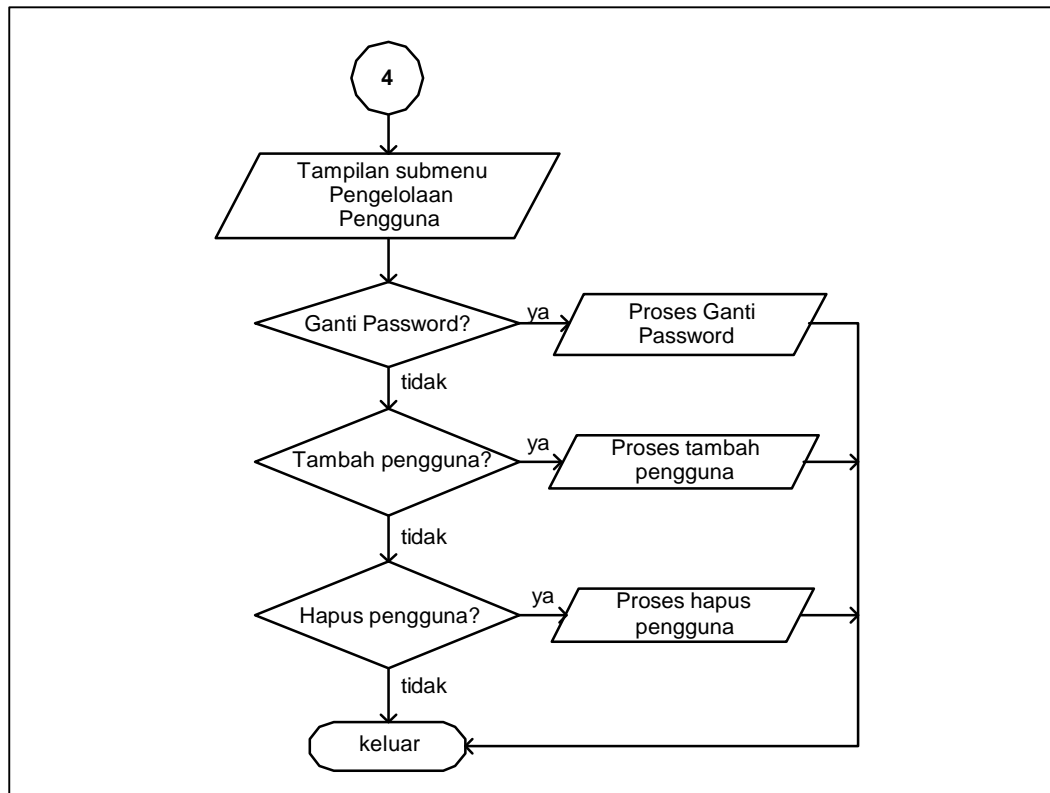
Pada menu help terdapat beberapa submenu seperti kamus istilah dan daftar informasi distributor Spare part untuk kerusakan pada printer serta pengelolaan pengguna. Alur proses menu help dapat dilihat pada gambar 3.13. Pada submenu kamus istilah, dapat dilihat alur proses setelah pemakai memilih submenu kamus istilah. Pemakai dapat mencari kata yang kurang dimengerti. Khusus untuk pakar dapat memanipulasi data kamus istilah. Submenu kamus istilah dapat dilihat pada gambar 3.14. Pakar / admin juga dapat menggunakan submenu Pengelolaan Pengguna untuk mengganti password, menambah dan menghapus pengguna yang dapat menggunakan sistem pakar diagnosa kerusakan pada printer. Flowchart pengelolaan pengguna dapat dilihat pada gambar 3.15.



Gambar 3.13 Menu Help



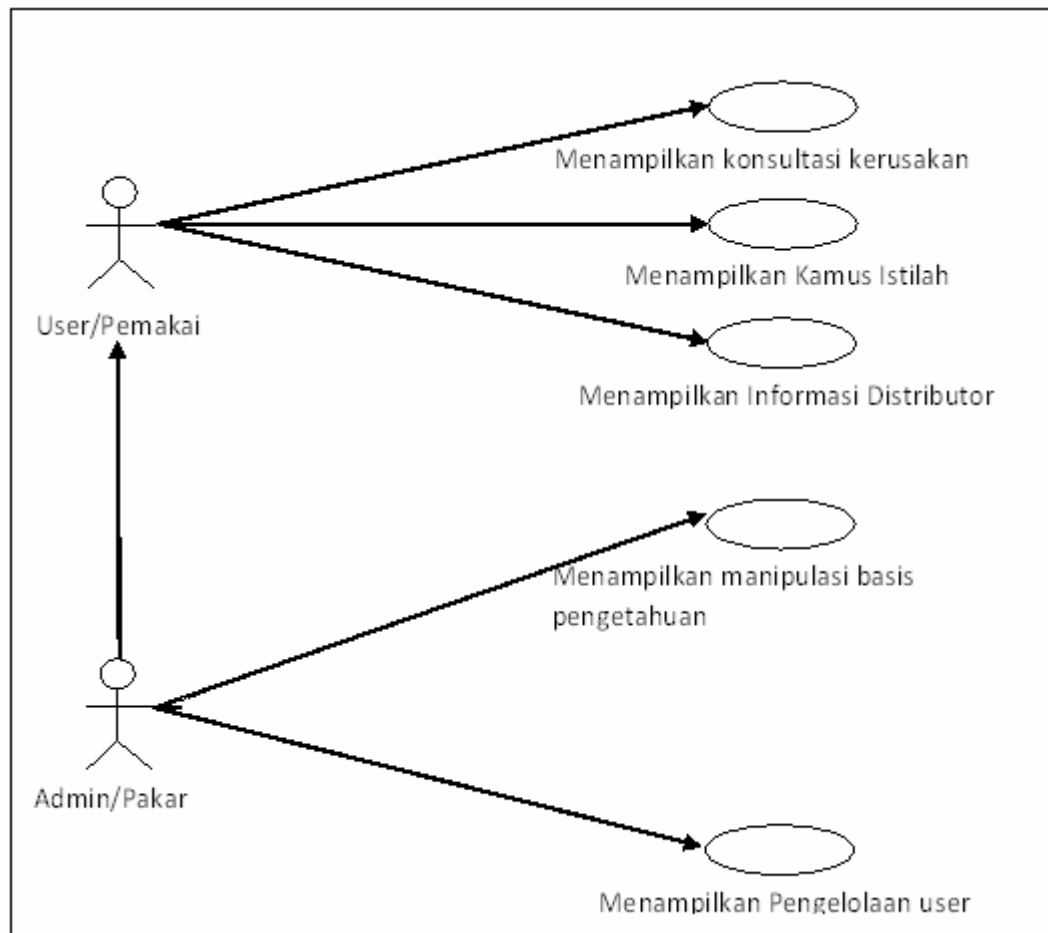
Gambar 3.14 Kamus Istilah



Gambar 3.15 Pengelolaan Pengguna

3.2.1.2 Implementasi Diagram Use Case

Use Case untuk sistem pakar diagnosa kerusakan pada printer ini dapat dilihat pada gambar 3.16 dimana pada sistem pakar ini administrator atau pakar dapat berperan sebagai pemakai atau user. Diagram use case menggambarkan interaksi antara pemakai dengan sistem.



Gambar 3.16 Diagram Use Case

Diagram Use Case pada gambar 3.14 harus dideskripsikan dalam dokumen yang disebut dengan flow of event sebagai berikut:

1. Use case untuk form konsultasi kerusakan.

Nama use case : Konsultasi kerusakan.

Deskripsi singkat : Use Case memungkinkan admin dan pemakai untuk mengakses form konsultasi kerusakan yang berisi gejala-gejala kerusakan, nama kerusakan, perbaikan dan pencegahannya.

Aktor : Pakar/Admin dan pemakai.

Precondition : Pakar/Admin atau pengguna mengakses menu File kemudian submenu konsultasi kerusakan. Sistem menampilkan form konsultasi kerusakan pada printer.

Main flow : Use Case ini dimulai ketika pakar/admin atau pengguna memilih menu file dan submenu konsultasi kerusakan. Sistem akan menampilkan tipe kerusakan pada printer. Admin/pakar atau pengguna akan memilih tipe kerusakan pada printer dilanjutkan dengan gejala umum dan gejala khusus. Setelah ditemukan nama kerusakan yang cocok, sistem akan menampilkan informasi nama kerusakan dan cara pencegahan serta cara perbaikannya.

Alternative flow : Admin/pakar atau pengguna dapat memilih tombol selesai untuk kembali ke menu utama.

Postcondition : Jika use case sukses dijalankan maka sistem akan menampilkan informasi kerusakan, cara perbaikan dan pencegahannya.

2. Use case untuk form kamus istilah.

Nama use case : Kamus istilah.

Deskripsi singkat : Pengguna dapat mencari istilah spare part printer yang kurang dimengerti. Use case ini juga memungkinkan pakar/admin dapat memanipulasi istilah-istilah spare part printer.

Actor : pakar/admin dan pengguna.

Precondition : pakar/admin atau pengguna memilih menu help dan submenu kamus istilah. Sistem akan menampilkan form kamus istilah.

Main flow : pemakai hanya dapat mengakses informasi kamus istilah.

Pemakai memilih tombol cari data, sistem akan mencari data atau istilah Spare part.

Pakar/admin dapat memanipulasi istilah Spare part.

Alternative flow : Jika tidak ditemukan data istilah Spare part maka sistem akan menampilkan pesan bahwa data yang dicari tidak tersedia.

Postcondition : Jika use case form kamus istilah sukses dijalankan maka sistem akan menampilkan informasi istilah Spare part.

3. Diagram use case untuk form tampilan informasi distributor spare part printer.

Nama use case : Informasi Distributor Spare part.

Deskripsi singkat : Pemakai dapat melihat tampilan Alamat Distributor Spare part

Actor : Pemakai dan Pakar / admin.

Precondition : Pemakai atau pakar/admin memilih menu Help dan submenu Alamat Distributor Spare part.

Main flow : Pemakai atau Pakar/admin memilih menu Help yang dilanjutkan memilih submenu Alamat Distributor Spare part.

Alternative flow : Pemakai atau Pakar/admin memilih tombol keluar untuk ke menu utama.

Postcondition : Sistem akan menampilkan form tampilan Alamat Distributor Spare part

4. Diagram use case untuk form tampilan Manipulasi Data Basis Pengetahuan.

Nama use case : Manipulasi Data Basis Pengetahuan.

Deskripsi singkat : Pemakai dengan level Pakar/admin dapat memanipulasi data mengenai tipe kerusakan, gejala umum, gejala khusus dan solusi.

Actor : Pakar/admin.

Precondition : Pakar/admin memilih menu File dan sub menu manipulasi Data Basis Pengetahuan.

Main flow : Pakar / admin masuk ke menu File dan submenu Manipulasi Data Basis Pengetahuan. Pakar dapat memanipulasi data tipe kerusakan, data gejala umum, data gejala khusus dan solusi, pencegahan serta perbaikan kerusakan.

Alternative flow : Pakar memilih tombol keluar dan sistem akan menampilkan tampilan menu utama.

Poscondition : Jika use case sukses dijalankan sistem akan menampilkan tampilan Manipulasi Data Basis Pengetahuan.

5. Diagram use case untuk form tampilan Pengelolaan Pengguna.

Nama use case : Pengelolaan Pengguna.

Deskripsi singkat : Pemakai dengan level Pakar/admin dapat memanipulasi data mengenai pengguna yang dapat menggunakan sistem pakar untuk mendeteksi kerusakan pada printer.

Actor : Pakar/admin.

Precondition : Pakar/admin memilih menu Help dan submenu Pengelolaan Pengguna.

Main flow : Pakar / admin masuk ke menu Help dan submenu Pengelolaan Pengguna. Pakar dapat memanipulasi data mengenai pengguna yang dapat menggunakan sistem pakar.

Alternative flow : Pakar memilih tombol keluar dan sistem akan menampilkan tampilan menu utama.

Poscondition : Jika use case sukses dijalankan sistem akan menampilkan tampilan Pengelolaan Pengguna.

6. Diagram use case untuk form tampilan Login.

Nama use case : Login.

Deskripsi singkat : Pemakai dengan level Pakar/admin dapat memanipulasi data mengenai pengguna yang dapat menggunakan sistem pakar untuk mendeteksi kerusakan pada printer, sehingga untuk bias melakukan itu harus memasukan password terlebih dahulu, sedangkan user biasa bias langsung masuk kedalam aplikasi dan hanya bias mengakses tampilan konsultasi kerusakan.

Actor : Pakar/admin dan pemakai.

Precondition : Pakar/Admin dan pemakai membuka aplikasi sitem pakar.

Main flow : Pakar/admin memilih daftar pilihan menu dan memasukan password untuk bias masuk kedalam aplikasi sementara unser biasa hanya menekan tombol masuk untuk bisa masuk kedalam aplikasi.

Alternative flow : Pakar memilih tombol keluar dan sistem akan keluar dari aplikasi.

Poscondition : Jika use case sukses dijalankan sistem akan menampilkan tampilan menu utama.

3.2.1.3 Kaidah Produksi

Tipe dan gejala kerusakan yang telah dikelompokkan diatas kemudian dibuat menjadi basis pengetahuan. Dari berbagai metode representasi pengetahuan yang biasa digunakan, penulis menggunakan metode representasi pengetahuan kaidah produksi. Metode ini memiliki keuntungan yaitu mudah dimengerti, penjelasan mudah diperoleh dan diturunkan, modifikasi dan perawatan yang lebih mudah. Metode kaidah produksi biasanya dituliskan dalam bentuk jika – maka (if - then). Kaidah produksi dapat dikatakan sebagai hubungan dua bagian yaitu bagian jika (premise) dan bagian maka (konklusi). Apabila bagian premise dipenuhi maka bagian konklusi juga bernilai benar. Suatu kaidah juga dapat terdiri dari beberapa premise dan lebih dari satu konklusi. Antara premise dan konklusi dapat dihubungkan dengan ”atau” atau ”dan”.

Dalam menentukan tipe kerusakan pada printer maka dibutuhkan beberapa kaidah produksi yang harus dipenuhi dalam bentuk if then antara lain:

Rule 1a IF Tipe kerusakan 01 AND kerusakan umum 01 AND kerusakan khusus 01 THEN kerusakan 01

Rule 1b IF Tipe kerusakan 02 AND kerusakan umum 01 AND kerusakan khusus 01 THEN kerusakan 02

3.2.2 Perancangan Basis Data

Data-data gejala kerusakan yang telah diperoleh diatas kemudian dikelompokkan menjadi tiga tingkatan kategori. Yang pertama adalah pengelompokan data berdasarkan tipe kerusakan, yang kedua adalah gejala umum dan yang ketiga adalah gejala khusus. Setiap gejala-gejala tersebut memiliki hubungan satu dengan yang lainnya. Tiga tingkatan tersebut diaplikasikan kedalam tiga tabel yang berbeda yang didalamnya terdapat field yang menghubungkan satu tingkatan dengan tingkatan lainnya.

1. Pengelompokan Data Tipe Kerusakan

Dari hasil identifikasi masalah, terdapat empat tipe kerusakan. Penomoran tipe kerusakan menggunakan huruf T untuk merepresentasikan tipe dan satu digit angka berikutnya adalah merupakan sequence atau urutan nomor yang terdiri dari nomor satu sampai dengan empat. Tabel 3.1 digunakan untuk merepresentasikan pengelompokan tipe kerusakan pada printer.

Tabel 3.1 Tabel Pengelompokan Tipe Kerusakan

Kode	Nama Tipe Kerusakan
T01	Tipe kerusakan 1
T02	Tipe kerusakan 2
T03	Tipe kerusakan 3
T04	Tipe kerusakan 4

2. Pengelompokan Data Gejala Umum

Gejala umum adalah gejala-gejala yang ditemukan dari beberapa kerusakan. Penomoran menggunakan huruf U untuk merepresentasikan gejala umum dan dua digit angka merupakan sequence atau urutan nomor. Sequence ini terdiri dari dua puluh data dengan maksimal indentifikasi data sejumlah sembilan

puluh sembilan data. Setiap tipe kerusakan memiliki lebih dari satu gejala umum, sebagai contoh satu tipe kerusakan memiliki lima data gejala kerusakan umum. Tabel gejala umum dapat dilihat pada tabel. 3.2.

Tabel 3.2 Tabel Pengelompokan Gejala Umum

Kode	Kode_Umum	Nama Gejala Umum
T01	U01	Gejala Kerusakan Umum 01 dari tipe 01
T01	U02	Gejala Kerusakan Umum 02 dari tipe 01
T01	U03	Gejala Kerusakan Umum 03 dari tipe 01
T01	U04	Gejala Kerusakan Umum 04 dari tipe 01
T01	U05	Gejala Kerusakan Umum 05 dari tipe 01
T02	U01	Gejala Kerusakan Umum 01 dari tipe 02
T02	U02	Gejala Kerusakan Umum 02 dari tipe 02
T02	U03	Gejala Kerusakan Umum 03 dari tipe 02
T02	U04	Gejala Kerusakan Umum 04 dari tipe 02
T02	U05	Gejala Kerusakan Umum 05 dari tipe 02
T03	U01	Gejala Kerusakan Umum 01 dari tipe 03
T03	U02	Gejala Kerusakan Umum 02 dari tipe 03
T03	U03	Gejala Kerusakan Umum 03 dari tipe 03
T03	U04	Gejala Kerusakan Umum 04 dari tipe 03
T03	U05	Gejala Kerusakan Umum 05 dari tipe 03
T04	U01	Gejala Kerusakan Umum 01 dari tipe 04
T04	U02	Gejala Kerusakan Umum 02 dari tipe 04
T04	U03	Gejala Kerusakan Umum 03 dari tipe 04
T04	U04	Gejala Kerusakan Umum 04 dari tipe 04
T04	U05	Gejala Kerusakan Umum 05 dari tipe 04

3. Pengelompokan Gejala Khusus

Penomoran dalam pengelompokan gejala khusus menggunakan huruf K untuk merepresentasikan gejala khusus dan dua digit angka merupakan sequence atau urutan nomor. Sequence ini terdiri dari enam puluh data dengan maksimal indentifikasi data sejumlah sembilan puluh sembilan data. Setiap kerusakan umum dapat memiliki lebih dari satu gejala khusus ,sebagai contoh satu kerusakan umum memiliki tiga data gejala kerusakan umum, sedangkan untuk tabel yang berisi data lengkapnya dapat dilihat pada halaman lampiran. Satu gejala khusus merepresentasikan gejala-gejala kerusakan yang khusus dimiliki oleh satu tipe kerusakan umum dan satu tipe kerusakan terdiri dari beberapa gejala kerusakan umum. Pengelompokan gejala khusus dapat dilihat pada tabel 3.3.

Tabel 3.3 Tabel Pengelompokan Gejala Khusus

Kode	Kode_Umum	Kode_Khusus	Nama Gejala Khusus
T01	U01	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 01 dengan tipe 01
T01	U01	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 01 dengan tipe 01
T01	U01	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 01 dengan tipe 01
T01	U02	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 02 dengan tipe 01
T01	U02	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 02 dengan tipe 01
T01	U02	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 02 dengan tipe 01
T01	U03	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 03 dengan tipe 01
T01	U03	K02	Gejala Kerusakan Khusus 02

			dari Kerusakan umum 03 dengan tipe 01
T01	U03	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 03 dengan tipe 01
T01	U04	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 04 dengan tipe 01
T01	U04	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 04 dengan tipe 01
T01	U04	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 04 dengan tipe 01
T01	U05	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 05 dengan tipe 01
T01	U05	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 05 dengan tipe 01
T01	U05	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 05 dengan tipe 01
T02	U01	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 01 dengan tipe 02
T02	U01	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 01 dengan tipe 02
T02	U01	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 01 dengan tipe 02
T02	U02	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 02 dengan tipe 02
T02	U02	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 02 dengan tipe 02
T02	U02	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 02 dengan tipe 02
T02	U03	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 03 dengan tipe 02
T02	U03	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 03

			dengan tipe 02
T02	U03	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 03 dengan tipe 02
T02	U04	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 04 dengan tipe 02
T02	U04	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 04 dengan tipe 02
T02	U04	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 04 dengan tipe 02
T02	U05	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 05 dengan tipe 02
T02	U05	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 05 dengan tipe 02
T02	U05	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 05 dengan tipe 02
T03	U01	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 01 dengan tipe 03
T03	U01	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 01 dengan tipe 03
T03	U01	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 01 dengan tipe 03
T03	U02	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 02 dengan tipe 03
T03	U02	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 02 dengan tipe 03
T03	U02	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 02 dengan tipe 03
T03	U03	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 03 dengan tipe 03
T03	U03	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 03 dengan tipe 03

T03	U03	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 03 dengan tipe 03
T03	U04	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 04 dengan tipe 03
T03	U04	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 04 dengan tipe 03
T03	U04	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 04 dengan tipe 03
T03	U05	K01	Gejala Kerusakan Khusus 01 dari Kerusakan umum 05 dengan tipe 03
T03	U05	K02	Gejala Kerusakan Khusus 02 dari Kerusakan umum 05 dengan tipe 03
T03	U05	K03	Gejala Kerusakan Khusus 03 dari Kerusakan umum 05 dengan tipe 03

Dalam perancangan sistem pakar ini digunakan tujuh buah tabel antara lain tabel tipe kerusakan, tabel gejala umum, tabel gejala khusus, tabel kamus istilah dan tabel users.

1. Tabel TipeKerusakan

Tabel tipe kerusakan memiliki dua buah field yaitu Kode dan Nama_tipe. Field Kode memiliki tipe data text tanpa duplikat karena Kode adalah unik dan ukuran 3 untuk memuat tiga digit kode. Field Nama_tipe memiliki tipe data text dan ukuran 50 untuk menyimpan nama kerusakan. Tabel tipe kerusakan dapat dilihat pada tabel 3.4 berikut:

Tabel 3.4 Tabel Tipe Kerusakan

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Kode_tipe	Text	3	Berisi nomor tipe kerusakan

2	Nama_tipe	Text	50	Berisi nama-nama tipe kerusakan
---	-----------	------	----	---------------------------------

2. Tabel GejalaUmum

Tabel gejala umum terdiri dari dua buah field yaitu Kode dan Gejala. Field Kode memiliki tipe data text tanpa duplikat karena unik dan ukuran tiga digit untuk menyimpan nomor gejala umum. Field Gejala memiliki tipe data text dan ukuran 50 untuk menyimpan nama gejala umum. Dapat dilihat pada tabel 3.5.

Tabel 3.5 Tabel Gejala Umum

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Kode_tipe	Text	3	Berisi nomor tipe gejala
2	Kode_umum	Text	3	Berisi nomor gejala umum
3	Gejala	Text	50	Berisi nama-nama gejala umum

3. Tabel GejalaKhusus

Tabel Gejala Khusus terdiri dari enam buah field yaitu Kode, Gejala, Kerusakan, Keterangan, Pencegahan dan Perbaikan. Field Kode memiliki tipe data text tanpa duplikat karena unik dan empat digit untuk menyimpan kode. Field Gejala memiliki tipe data text dan ukuran 50 untuk menyimpan nama gejala khusus. Field Kerusakan memiliki tipe data text dan ukuran 50 untuk menyimpan nama kerusakan. Field Keterangan memiliki tipe data memo untuk menyimpan keterangan mengenai kerusakan. Field Pencegahan memiliki tipe data memo untuk menyimpan keterangan mengenai pencegahan kerusakan. Field Perbaikan memiliki tipe data memo untuk menyimpan keterangan mengenai perbaikan kerusakan. Dapat dilihat pada tabel 3.6.

Tabel 3.6 Tabel Gejala Khusus

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Kode_tipe	Text	3	Berisi nomor tipe gejala
2	Kode_umum	Text	3	Berisi nomor gejala umum
3	Kode_khuus	Text	3	Berisi nomor gejala khusus
4	Gejala	Text	50	Berisi nama gejala khusus
5	Kerusakan	Text	50	Berisi nama kerusakan
6	Keterangan	Memo		Berisi keterangan mengenai kerusakan
7	Pencegahan	Memo		Berisi mengenai pencegahan
8	Perbaikan	Memo		Berisi mengenai perbaikan

4. Tabel Kamus Istilah

Tabel ini memiliki tiga buah field yaitu ID, istilah dan Definisi. Field ID berisi nomor istilah Spare part memiliki tipe data auto number dengan ukuran long integer dan tanpa duplikat karena unik.. Field kata memiliki tipe data text dan ukuran 50 untuk menyimpan istilah Spare part. Field Definisi memiliki tipe data memo untuk menyimpan arti dari istilah Spare part. Dapat dilihat pada tabel 3.7.

Tabel 3.7 Tabel Kamus Istilah

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	ID	Auto Number	Integer	Berisi nomor istilah Spare part
2	Istilah	Text	50	Berisi istilah-istilah dalam Spare part
3	Definisi	memo		Berisi arti dari istilah-istilah

5. Tabel Pengguna

Tabel pengguna memiliki dua buah field yaitu Nama dan Password. Field Nama memiliki tipe data text dan ukuran 10 digit untuk menyimpan nama pemakai. Field Password memiliki tipe data text dan ukuran 12 untuk menyimpan kata kunci. Dapat dilihat pada tabel 3.8.

Tabel 3.8 Tabel Pengguna

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Nama	Text	10	Berisi nama pemakai
2	Password	Text	12	Berisi kata kunci

3.2.3 Rancangan Layar Tampilan

Rancangan layar tampilan yang akan digunakan dalam sistem pakar ini adalah sebagai berikut:

1. Tampilan layar login

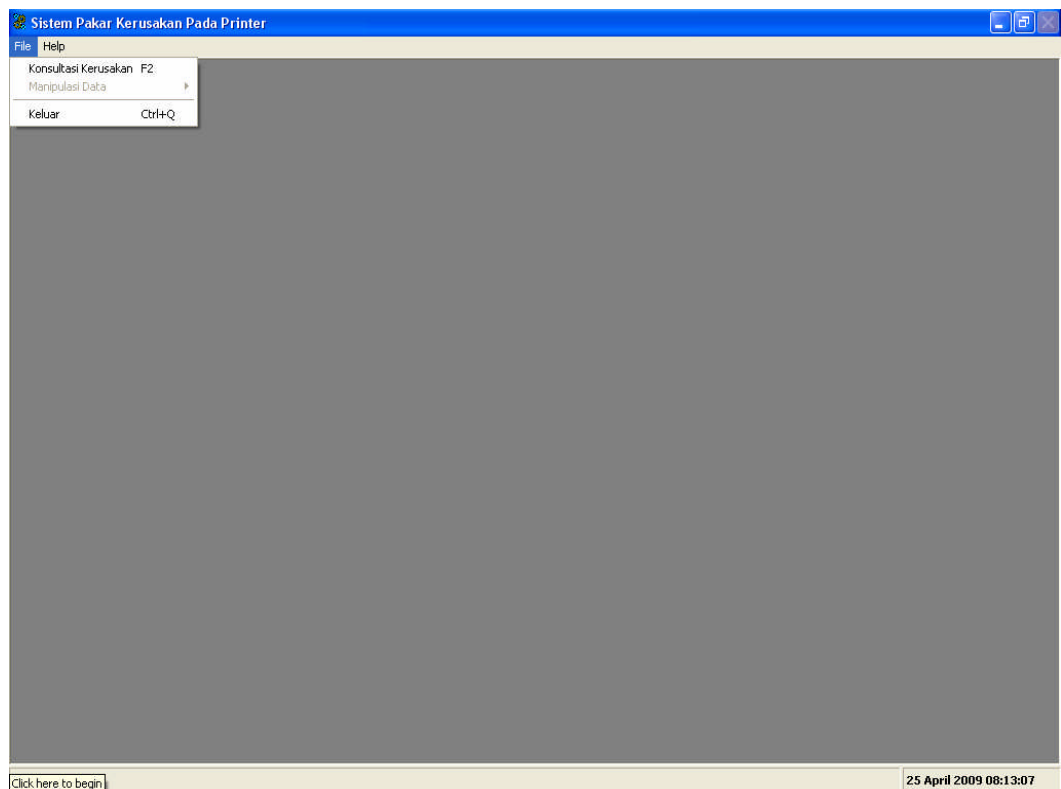
Tampilan ini terdiri dari dua buah textbox yaitu untuk memasukan nama dan password. Dua buah command button digunakan untuk perintah OK dan Cancel. Sebuah checkbox digunakan khusus untuk pakar. Sebuah label digunakan untuk mengganti password. Rancangan tampilan login dapat dilihat pada gambar 3.17 berikut:



Gambar 3.17 Layar Login

2. Tampilan layar menu utama

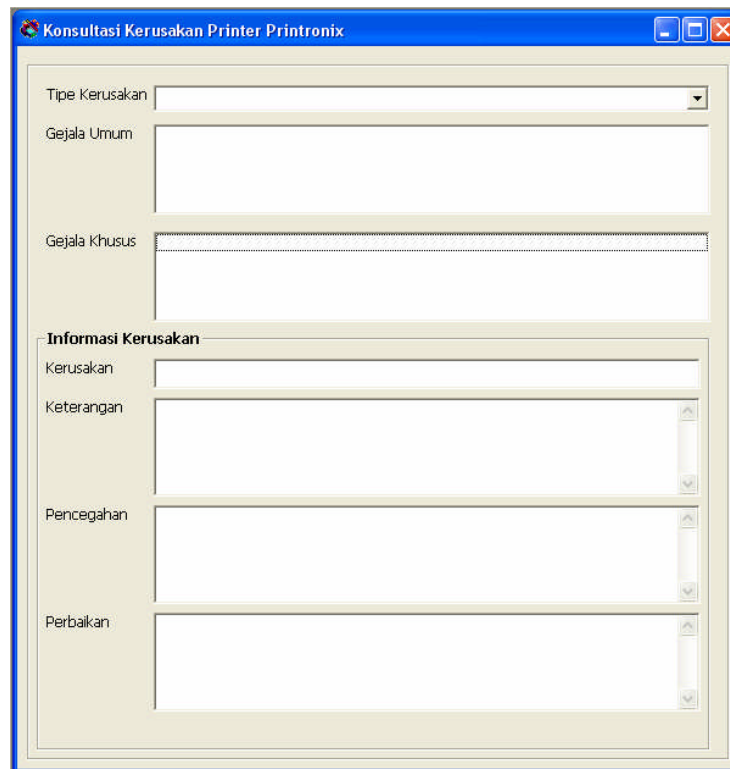
Pada layar menu utama terdapat menu File dan Help. Pemakai dapat memilih salah satu dari menu tersebut. Menu file terdiri dari submenu Konsultasi, Manipulasi Data, basis aturan dan Keluar. Khusus untuk menu tampilan data dan basis aturan hanya dapat diakses oleh pakar atau admin. Submenu Manipulasi data terdiri dari basis pengetahuan, basis aturan dan keluar. Submenu Help terdiri dari Kamus istilah, petunjuk membedah bangkai, alamat distributor Spare part dan keluar. Tampilan layar menu utama dapat dilihat pada gambar 3.18.



Gambar 3.18 Layar Menu Utama

3. Tampilan menu File

Rancangan tampilan menu file submenu konsultasi terdiri dari tiga buah listbox yang memuat daftar tipe kerusakan, gejala umum dan gejala khusus. Sebuah label untuk menampilkan nama kerusakan dan tiga buah textbox untuk menampilkan keterangan kerusakan, pencegahan dan perbaikan. Dapat dilihat pada gambar 3.19.



Gambar 3.19 Tampilan Submenu Konsultasi

Tampilan submenu manipulasi data terdiri dari manipulasi data tipe kerusakan, data gejala umum dan data gejala khusus. Rancangan tampilan untuk manipulasi basis pengetahuan tipe kerusakan terdiri dari sebuah DataGridView dua buah text box untuk memasukkan kode dan tipe kerusakan serta empat buah command button untuk memasukkan refresh, tambah rubah dan hapus. Manipulasi gejala kerusakan dapat dilihat pada gambar 3.25.

Kode: T01

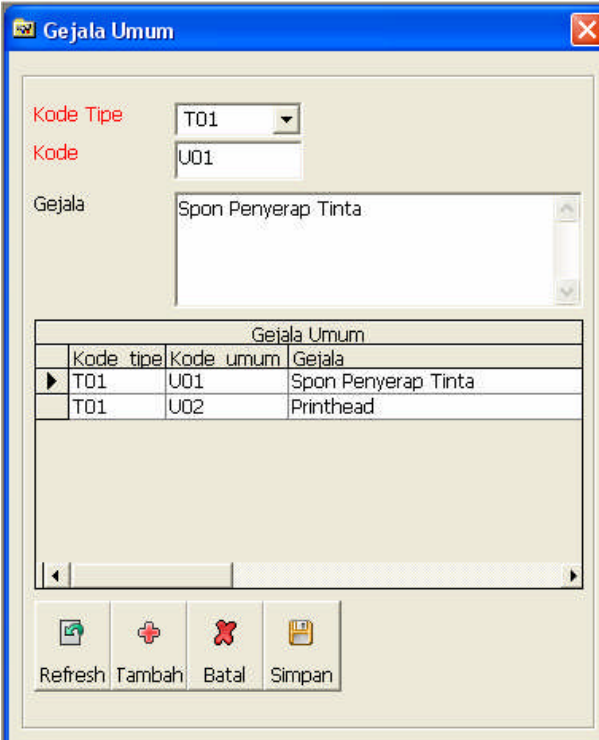
Tipe Kerusakan: Kerusakan pada spare part printer

Tipe Kerusakan	
Kode	Keterangan
T01	Kerusakan pada spare part printer
T02	Kerusakan pada jaringan/network
T03	Kerusakan pada perangkat lunak printer
T04	Kerusakan pada komputer

Refresh Tambah Batal Simpan

Gambar 3.20 Manipulasi Data Tipe Kerusakan

Tampilan untuk manipulasi data Gejala Umum terdiri dari dua buah textbox untuk memuat kode dan gejala, sebuah datagrid dan empat buah command button untuk memuat refresh, tambah, ubah dan hapus. Dapat dilihat pada gambar. 3.21.



Gejala Umum

Kode Tipe: T01

Kode: U01

Gejala: Spon Penyerap Tinta

Gejala Umum		
Kode tipe	Kode umum	Gejala
T01	U01	Spon Penyerap Tinta
T01	U02	Printhead

Refresh Tambah Batal Simpan

Gambar 3.21 Manipulasi Data Gejala Umum

Tampilan untuk manipulasi gejala khusus terdiri dari enam buah textbox untuk memuat kode, gejala, kerusakan, keterangan, pencegahan dan perbaikan. Sebuah datagrid dan empat buah command button untuk memuat perintah refresh, tambah, ubah dan hapus. Dapat dilihat pada gambar. 3.22.

Gejala Khusus

Kode Tipe: T01
 Kode Gejala Umum: U01
 Kode: K01

Gejala: Blingking tapi tidak bisa direset
 Pencegahan: Spon perlu bersihkan secara rutin
 Kerusakan: Penuhnya spon penyerap tinta
 Perbaikan: Spon perlu dicuci, kemudian lakukan reset seperti biasa
 Keterangan: -

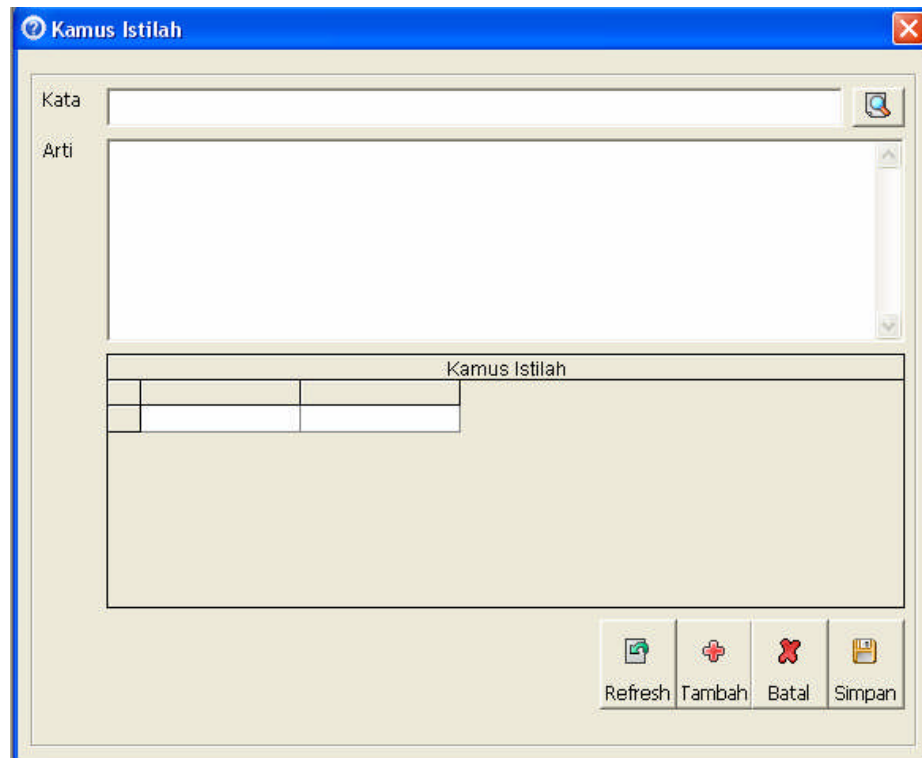
Gejala Khusus					
Kode tipe	Kode umum	Kode khusus	Gejala	Kerusakan	Keterangan
T01	U01	K01	Blingking tapi tidak bisa direset	Penuhnya spon penyerap tinta	-
T01	U01	K02	Waktu isi ulang tinta	Spon sudah kering dan keras	-
T01	U02	K01	Printer jalan normal tapi	Printhead buntu	-

Refresh Tambah Batal Simpan

Gambar 3.22 Manipulasi Data Gejala Khusus

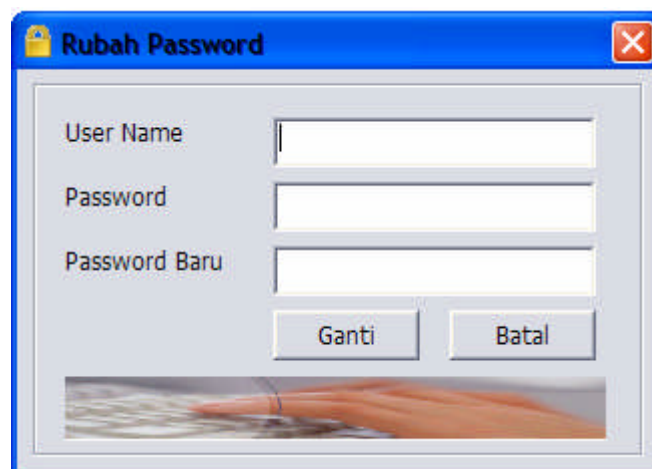
4. Menu Help

Menu help terdiri dari kamus istilah dan daftar alamat distributor Spare part serta pengelolaan pengguna. Pada rancangan tampilan kamus istilah digunakan dua buah textbox untuk memuat kata dan arti, sebuah datagrid dan empat buah tombol command button untuk memuat refresh, tambah, ubah dan hapus. Dapat dilihat pada gambar 3.23.



Gambar 3.23 Tampilan Kamus Istilah

Pengelolaan pengguna menggunakan 3 buah textbox untuk memuat nama pengguna, password dan ulang password. Sebuah listbox untuk daftar pengguna dan 3 buah command button untuk memuat perintah mengganti, menambah dan menghapus pengguna. Lihat pada gambar 3.24.



Gambar 3.24 Tampilan Pengelolaan Pengguna

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Bab ini akan diawali dengan skenario yang akan menjelaskan bagaimana sistem diterapkan khususnya dalam mendapatkan respon dari sistem terhadap masukan yang diberikan, setelah itu dilanjutkan dengan pengujian dengan cara menerapkan skenario diatas dan melihat respon secara nyata dari sistem, sehingga dengan cara ini dapat mewakili pengujian sistem bahwa sistem berjalan dengan baik.

4.1 Skenario Pengujian Sistem

Tema dari skenario yang dibuat adalah user yang memiliki masalah pada printer akan mendapatkan jawaban berupa saran atau solusi dari sistem untuk memperbaiki kerusakan printer printonix. Langkah –langkah yang dilakukan adalah :

- User akan memasukan nama user dan password kedalam kotak isian lalu memilih tipe user apakah user biasa atau administrator.
- Bila nama user tidak ada dalam database maka sistem akan memberikan pesan “ User salah atau belum terdaftar !” dan bila salah memasukan password maka akan tampil pesan “Password Salah, silahkan coba lagi!”. Kedua pesan tersebut digunakan untuk menjelaskan kepada user bahwa tidak semua orang diberikan hak untuk menggunakan sistem ini, hanya orang-orang yang namanya

telah didaftarkan oleh administrator yang dapat mengakses sistem ini.

- Setelah user memasukan inputan nama dan password dengan benar maka user akan masuk kedalam tampilan menu utama dimana user akan milih menu konsultasi kerusakan.
- Pada menu manipulasi data terdapat sub menu tipe kerusakan, hanya pakar atau admin dapat mengakses sub menu tersebut.
- Pada sub menu tipe kerusakan, user pakar atau admin dapat menambahkan data dengan menekan tombol tambah, lalu memasukan data tipe kerusakan dan menekan tombol simpan bila ingin menyimpan atau menekan tombol batal bila ingin membatalkan proses penambahan data. Bila pakar atau admin ingin merubah data yang sudah ada maka harus memilih data yang akan di rubah, lalu menekan tombol ubah, data yang akan diubah akan tampil pada kotak isian nama tipe yang dapat di rubah, setelah itu user dapat menyimpan data dengan menekan tombol simpan atau batal bila tidak menginginkan perubahan data.
- Pada sub menu kerusakan umum, user pakar atau admin dapat menambahkan data dengan menekan tombol tambah, lalu memilih data tipe kerusakan, mengisi data kerusakan umum dan menekan tombol simpan bila ingin menyimpan atau menekan tombol batal bila ingin membatalkan proses penambahan data. Bila pakar atau admin ingin merubah data yang sudah ada maka harus memilih

data yang akan di rubah, lalu menekan tombol ubah, data yang akan diubah akan tampil pada kotak isian kode tipe, kode gejala umum dan nama gejala umum yang dapat di rubah, setelah itu user dapat menyimpan data dengan menekan tombol simpan atau batal bila tidak menginginkan perubahan data

- Pada sub menu kerusakan khusus, user pakar atau admin dapat menambahkan data dengan menekan tombol tambah, lalu memilih data tipe kerusakan, memilih kode gejala umum, mengisi data kerusakan khusus dan menekan tombol simpan bila ingin menyimpan atau menekan tombol batal bila ingin membatalkan proses penambahan data. Bila pakar atau admin ingin merubah data yang sudah ada maka harus memilih data yang akan di rubah, lalu menekan tombol ubah, data yang akan diubah akan tampil pada kotak isian kode tipe, kode gejala umum, kode gejala khusus dan nama gejala khusus yang dapat di rubah, setelah itu user dapat menyimpan data dengan menekan tombol simpan atau batal bila tidak menginginkan perubahan data..
- Pada menu konsultasi kerusakan user akan memilih tipe kerusakan setelah itu user akan memilih gejala umum dan gejala khusus dari kerusakan yang didapat setelah itu sistem akan menampilkan nama atau keterangan kerusakan serta solusi kepada user berupa langkah-langkah pencegahan agar tidak terjadi lagi dan juga saran perbaikan yang harus dilakukan.

- Setelah selesai user akan membaca solusi dan saran yang didapat, lalu bila ada kata-kata atau istilah printer yang tidak dimengerti maka user akan memilih menu kamus istilah lalu pada kamus istilah user akan memasukan istilah kata yang akan dicari setelah itu sistem akan menampilkan penjelasan dari kata yang akan dicari.

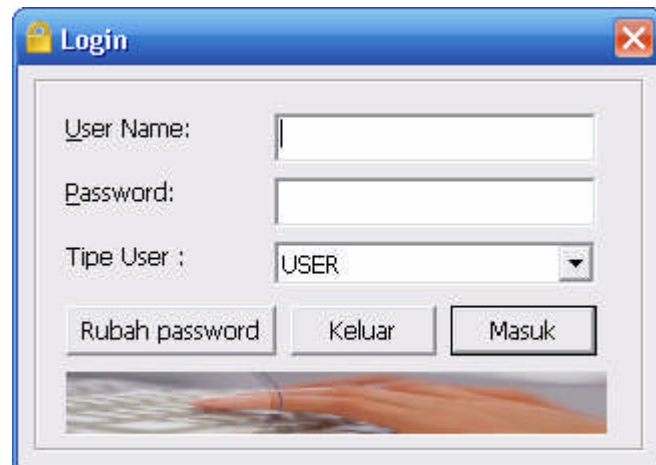
4.2 Pengujian

Pengujian yang dilakukan terhadap program aplikasi sistem pakar ini meliputi pengujian modul-modul dengan tujuan untuk mengetahui hasil yang diharapkan sesuai dengan tujuan yang telah ditentukan. Pengujian meliputi pengujian modul Konsultasi Kerusakan, Manipulasi Data, Basis Aturan, Kamus Istilah, Informasi Distributor Spare part dan Pengelolaan Pengguna. Parameter yang digunakan untuk pengujian diantaranya koneksi dengan basis data, keterkaitan antar form, manipulasi data yaitu penambahan data baru, edit dan hapus data, kebenaran sintak, pemanggilan modul, kebenaran seperti tujuan yang diharapkan.

a. Tampilan Login

Pada saat login pemakai dapat memasukan kata kunci atau *password*. Login berfungsi untuk menentukan pemakai sebagai pakar atau pemakai biasa. Pemakai biasa hanya bisa menggunakan submenu konsultasi, cari istilah dan alamat distributor spare part. Tampilan login dapat dilihat pada gambar 4.1. jika record nama dan password ditemukan maka pemakai akan login sebagai pakar dan sistem akan menampilkan menu utama yang dapat dilihat pada gambar 4.3., jika salah, sistem akan menampilkan kotak pesan berisi

“Password Salah, silahkan coba lagi” yang dapat dilihat pada gambar 4.2. , selain itu pemakai akan login sebagai pemakai biasa.



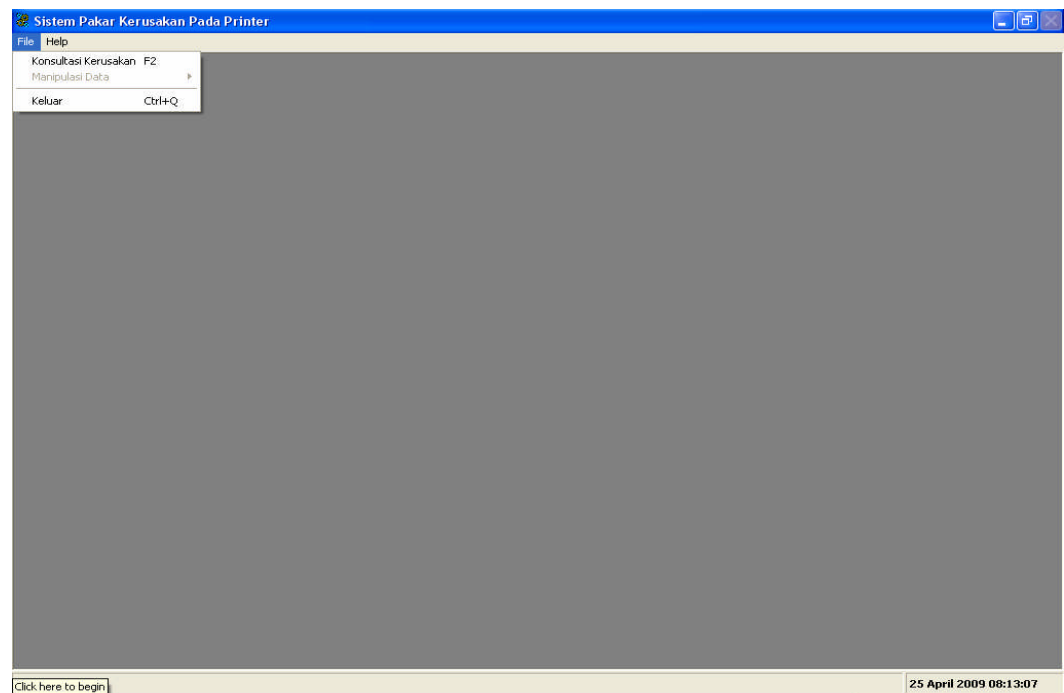
Gambar 4.1 Tampilan Login



Gambar 4.2 Tampilan Pesan pada saat Login

b. Tampilan Menu File

Saat pemakai memilih menu File akan tampil submenu Konsultasi Kerusakan, Manipulasi Data, Basis aturan dan keluar. Tampilan menu File dapat dilihat pada gambar 4.3.



Gambar 4.3 Tampilan Menu File

c. Tampilan submenu Konsultasi Kerusakan

Pada submenu konsultasi kerusakan, pemakai dapat memilih tipe kerusakan kemudian gejala umum dan gejala khusus. Tipe kerusakan, gejala umum dan gejala khusus yang telah dipilih kemudian akan diproses sehingga dapat diketahui nama kerusakan, pencegahan serta spare partnya.

Gambar 4.4 Tampilan Konsultasi Kerusakan

Untuk hasil pengujian sistem berdasarkan simulasi dapat dilihat pada tabel 4.1 berikut:

Tabel 4.1 Tabel Hasil Pengujian

Modul	Hasil Pengujian	Keterangan
Login	Tidak ada kesalahan	Sesuai tujuan
Tipe Kerusakan	Tidak ada kesalahan	Sesuai tujuan
Gejala Umum	Tidak ada kesalahan	Sesuai tujuan
Gejala Khusus	Tidak ada kesalahan	Sesuai tujuan
Konsultasi Kerusakan	Tidak ada kesalahan	Sesuai tujuan
Kamus Istilah	Tidak ada kesalahan	Sesuai tujuan

BAB V

KESIMPULAN DAN SARAN

Bab ini akan membahas tentang kesimpulan dan saran dari proses pembuatan sistem pakar.

5.1 Kesimpulan

Kesimpulan yang didapat adalah sebagai berikut :

- Dari hasil pengujian dapat disimpulkan bahwa sistem berjalan dengan baik. Terlihat dari gejala-gejala yang dimasukan bisa beroperasi sehingga bisa menghasilkan solusi atau saran kurang realistis.
- Penggunaan bahasa visual basic bisa dengan baik diimplementasikan atau mewujudkan sistem pakar ini.
- Pemanfaatan dari sistem ini masih kurang flexibel karena tidak bisa diakses tanpa menginstal terlebih dahulu.
- Penggunaan database Ms.Access tidak bisa memberikan respon yang cepat terutama bila diakses oleh multi user.

5.2 Saran

Saran yang dapat diberikan adalah sebagai berikut :

- Sebaiknya sistem ini dapat dibuat dengan bahasa pemograman berbasis web yang dapat di akses dengan bebas secara umum dimana saja oleh user tanpa harus menginstal pogram aplikasi

disetiap komputer sehingga sistem ini akan menjadi sistem yang flexibel yang dapat membantu para user dimana saja.

- Sebaiknya untuk pengembangan lebih lanjut, sistem ini dapat dibuat dengan bahasa pemograman berbasis web Asp.net karena dengan asp.net yang memiliki object yang bervariasi seperti object msgridline yang dapat menampilkan data sekaligus object combo didalamnya ataupun check box selain row data biasa sehingga pada tampilan pun dapat menjadi lebih menarik yang dapat menggantikan datagrid pada vb.6.
- Selain itu dengan Asp.net proses pembuatan program dapat di kombinasikan dengan bahasa pemograman lainya selain vb diantaranya java atau vb script karena pada asp.net memiliki multi platform yang dapat menampung beberapa bahasa dan mampu menggabungkanya.

DAFTAR PUSTAKA

- Arhami, Muhammad. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Penerbit Andi.
- Aziz, M. Farid. 1994. *Belajar Sendiri Pemrograman Sistem Pakar*. Jakarta: PT. Elek Media Komputindo.
- Jogiyanto, H.M. 1999. *Analisis & Disain Sistem Informasi: pendekatan terstruktur teori dan praktek aplikasi bisnis* Yogyakarta: Penerbit Andi.
- Kadir, Abdul. 1999. *Konsep dan Tuntunan Praktis Basis Data*. Yogyakarta: Penerbit Andi.
- Kristanto, Harianto. 1999. *Konsep Dan Perancangan Database*. Yogyakarta: Penerbit Andi.
- Suryadi H.S. 1994. *Pengantar Sistem Pakar*. Jakarta: Penerbit Gunadarma.

KONEKSI KE BASIS DATA

Option Explicit

Private conDB As Connection
Private bolAdm As Boolean

Public Function
OpenDatabase(DatabaseName As String) As
Boolean

On Error GoTo OpenError

Set conDB = New Connection
conDB.CursorLocation = adUseClient
conDB.Open
"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=" & App.Path & "\" &
DatabaseName & ";Persist Security
Info=False"
OpenDatabase = True
Exit Function

OpenError:
MsgBox "Error " &
conDB.Errors.Item(0).Number, vbCritical,
"ERROR!!!"
Set conDB = Nothing
End Function

Public Function CloseDatabase() As
Boolean
Set conDB = Nothing
End Function

Public Function Execute(SQL As String) As
Boolean
On Error GoTo ExecuteError

conDB.Execute SQL
Execute = True
Exit Function

ExecuteError:
MsgBox
conDB.Errors.Item(0).Description
conDB.Errors.Clear
End Function

Public Function OpenRecordSet(SQL As
String, Optional CursorType As
CursorTypeEnum = adOpenStatic) As
Recordset

On Error GoTo ReadError

Set OpenRecordSet = New Recordset

OpenRecordSet.CursorLocation =
adUseClient
OpenRecordSet.Open SQL, conDB,
CursorType, adLockOptimistic
Exit Function

ReadError:
Set OpenRecordSet = Nothing
MsgBox
conDB.Errors.Item(0).Description
conDB.Errors.Clear
End Function

Public Function IsAdmin() As Boolean
IsAdmin = bolAdm
End Function

Public Sub SetAdmin(State As Boolean)
bolAdm = State
End Sub

LOGIN

Option Explicit

'menampilkan box_ dialog ganti password
Private Sub cmd_rubah_Click()
Me.Caption = "Rubah Password"
Box_1.Visible = False
Box_2.Visible = True
txtNama.SetFocus
End Sub

'menampilkan box_ dialog Masuk
Private Sub cmdBatal_Click()
Me.Caption = "Login"
Box_1.Visible = True
Box_2.Visible = False
txtUser.SetFocus
End Sub

'tutup window jika Cancel ditekan
Private Sub cmdOut_Click()
Unload Me
End Sub

'tombol ganti password
Private Sub cmdGanti_Click()
Dim rsUser As Recordset
Dim userName As String, password As
String, newPassword

'Batalkan operasi jika nama/password
kosong
If txtNama.Text = vbNullString Or
txtPasswd.Text = vbNullString Or

```

txtNewPasswd.Text = vbNullString Then
Exit Sub

'ganti karakter '(apostrop) dengan "
userName = Replace(txtNama.Text, "'",
"")
password = Replace(txtPasswd.Text, "'",
"")
newPassword =
Replace(txtNewPasswd.Text, "'", "''")

'buka table user dengan nama & password
tersebut
Set rsUser = OpenRecordSet("Select
Nama,Passwd From Pengguna Where
Nama='" & userName & "' And Passwd='"
& password & "'")

'jika table terbuka (sukses)
If Not (rsUser Is Nothing) Then
If rsUser.State = adStateOpen Then
'apakah record tersebut ditemukan?
If rsUser.RecordCount > 0 Then
Set rsUser = Nothing
'rubah password lama dalam table
users dengan password baru
If Execute("Update Pengguna Set
Passwd='" & newPassword & "' Where
Nama='" & userName & "' And Passwd='"
& password & "'") Then
'pesan perubahan berhasil
MsgBox "Password user " &
userName & " telah berhasil dirubah.",
vbInformation
Else
'pesan perubahan gagal
MsgBox "Gagal merubah
password user " & userName, vbCritical
End If
Else
'tdk ada record dengan nama &
password tsb
MsgBox "User tidak ditemukan!",
vbCritical, "Ganti Password"
txtNama.SetFocus
SendKeys "{Home}+{End}"
End If
End If
End If

'hancurkan object rsUser
Set rsUser = Nothing
'panggil sub Batal_Click() untuk
menampilkan kembali dialog login
Call cmdBatal_Click

```

```

End Sub

'tombol Masuk
Private Sub cmdIn_Click()
Dim userName As String, password As
String
Dim rsUser As Recordset

'jika checkbox pakar tidak diceklis,
langsung masuk ke window utama
If cbo_TipeUser.ListIndex = 0 Then
SPMDIForm.Show
Me.Hide
Else

'buka table users
Set rsUser = OpenRecordSet("Select
Nama,Passwd From Pengguna Where
Nama='" & txtUser & "' And Passwd='" &
txtPassword & "'")

'jika table berhasil dibuka...
If Not (rsUser Is Nothing) Then
If rsUser.State = adStateOpen Then
'jika record dengan nama &
password ditemukan...
If rsUser.RecordCount > 0 Then
Set rsUser = Nothing
SetAdmin True
SPMDIForm.Show
Me.Hide
Else
MsgBox "Password Salah,
silahkan coba lagi!", , "Login"
txtPassword.SetFocus
SendKeys "{Home}+{End}"
End If
End If
End If
End If

Set rsUser = Nothing
End Sub

'event aktivasi window
Private Sub Form_Activate()
SetAdmin (False)
If Me.WindowState = vbMinimized Then
Me.WindowState = vbNormal
End If
txtUser.Text = ""
txtPassword.Text = ""
txtNama.Text = ""
txtPasswd.Text = ""
txtNewPasswd.Text = ""

```

```

txtUser.SetFocus

cbo_TipeUser.Clear
cbo_TipeUser.AddItem "USER"
cbo_TipeUser.AddItem "PAKAR"
cbo_TipeUser.ListIndex = 0
End Sub

'event window load
Private Sub Form_Load()
    'buka database
    OpenDatabase ("Printer.mdb")
End Sub

'event window close
Private Sub Form_Unload(Cancel As Integer)
    'tutup database
    CloseDatabase
End Sub

MENAMPILKAN MENU UTAMA
Option Explicit

'event window load
Private Sub MDIForm_Load()
    Dim bgImage As String
    'file background
    bgImage = App.Path & "\BG.jpg"
    'jika file background ada...tampilkan sbg
    latar belakang
    If Dir(bgImage) <> vbNullString Then
        'Set Me.Picture =
        LoadPicture(bgImage)
    End If
    'menu berikut hanya aktif jika user=pakar
    menuFileBasisAturan.Enabled = IsAdmin
    menuFilePengetahuan.Enabled = IsAdmin
    menuFileManageUsers.Enabled =
    IsAdmin
    If IsAdmin = True Then
        StatusBar1.Panels(1) = "Pakar" Else
        StatusBar1.Panels(1) = frmLogin.txtUser

        StatusBar1.Panels(2).Text =
        Format(Now(), "DD MMMM YYYY
        hh:mm:ss")
        Show
    End Sub

'event window close
Private Sub MDIForm_Unload(Cancel As Integer)
    'load frmLogin
    frmLogin.Show

```

```

End Sub

Private Sub menuCascade_Click()
    Me.Arrange vbCascade
End Sub

Private Sub menuFileAnalisa_Click()
    Analisa.Show
End Sub

Private Sub menuFileBasisAturan_Click()
    BasisAturan.Show
End Sub

Private Sub menuFileExit_Click()
    Unload Me
End Sub

Private Sub menuFileGK_Click()
    MasterGK.Show
End Sub

Private Sub menuFileGU_Click()
    MasterGU.Show
End Sub

Private Sub
menuFileTypeKerusakan_Click()
    MasterTipeKerusakan.Show
End Sub

Private Sub menuFileJenisPenyakit_Click()
    MasterTipeKerusakan.Show
End Sub

Private Sub menuFileManageUsers_Click()
    Users.Show
End Sub

Private Sub menuHelpAbout_Click()
    frmSplash.Show vbModal, Me
End Sub

Private Sub menuHelpBedah_Click()
    BedahBangkai.Show
End Sub

Private Sub menuHelpKamus_Click()
    KamusIstilah.Show
End Sub

Private Sub menuHelpProdusen_Click()
    Produsen.Show
End Sub

```

```

Private Sub menuTileH_Click()
    Me.Arrange vbTileHorizontal
End Sub

Private Sub menuTileV_Click()
    Me.Arrange vbTileVertical
End Sub

Private Sub Timer1_Timer()
    StatusBar1.Panels(2).Text =
Format(Now(), "DD MMMM YYYY
hh:mm:ss")
End Sub

```

PROSES KONSULTASI KERUSAKAN

Option Explicit

```

'table objects
Private rsJenis As Recordset
Private rsUmum As Recordset
Private rsKhusus As Recordset

'event window load
Private Sub Form_Load()
    'buka table TipeKerusakan,GejalaUmum
    dan GejalaKhusus
    Set rsJenis = OpenRecordSet("Select *
From TipeKerusakan")
    Set rsUmum = OpenRecordSet("Select
distinct j.KodeJenis,u.Kode,u.Gejala From
GejalaUmum u Inner Join
relasiJenisGejalaUmum j On
j.KodeGejalaUmum=u.Kode")
    Set rsKhusus = OpenRecordSet("Select
u.KodeGejalaUmum,k.Kode,k.Gejala,k.Ker
usakan,k.Keterangan,k.Pencegahan,k.Perbai
kan From GejalaKhusus k Inner Join
relasiGejalaUmumKhusus u On
u.KodeGejalaKhusus=k.Kode")

    'tambahkan record dari TipeKerusakan ke
    combo box TipeKerusakan
    Do While Not rsJenis.EOF
        DoEvents
        TipeKerusakan.AddItem
        rsJenis.Fields(0).Value & " - " &
        rsJenis.Fields(1).Value
        rsJenis.MoveNext
    Loop
End Sub

Private Sub Form_Resize()
    If Me.WindowState <> vbMinimized
Then

```

```

        TipeKerusakan.Width = Me.Width -
1680
        GejalaUmum.Width = Me.Width -
1680
        GejalaKhusus.Width = Me.Width -
1680
        NamaKerusakan.Width = Me.Width -
1680
        Keterangan.Width = Me.Width - 1680
        Pencegahan.Width = Me.Width - 1680
        Perbaikan.Width = Me.Width - 1680
    End If
End Sub

```

```

'event window close
Private Sub Form_Unload(Cancel As
Integer)
    'hancurkan object
    Set rsJenis = Nothing
    Set rsUmum = Nothing
    Set rsKhusus = Nothing
End Sub

```

```

'salah satu gejala umum dipilih
Private Sub GejalaUmum_Click()
    'filter table GejalaKhusus dengan gejala
    umum = 3 karakter pertama dari list yg
    terpilih
    rsKhusus.Filter = "KodeGejalaUmum=" &
    & Left(GejalaUmum.Text, 3) & ""
    GejalaKhusus.Clear
    NamaKerusakan.Caption = vbNullString
    Keterangan.Text = vbNullString
    Pencegahan.Text = vbNullString
    Perbaikan.Text = vbNullString
    Do While Not rsKhusus.EOF
        DoEvents
        GejalaKhusus.AddItem
        rsKhusus.Fields(1).Value & " - " &
        rsKhusus.Fields(2).Value
        rsKhusus.MoveNext
    Loop
End Sub

```

```

'salah satu gejala khusus dipilih
Private Sub GejalaKhusus_Click()
    Dim Kode As String

    'variable (kode) Kerusakan adalah 4
    karakter pertama dari list yg terpilih
    Kode = Left(GejalaKhusus.Text, 4)
    'kosongkan dulu label NamaKerusakan
    NamaKerusakan.Caption = vbNullString

```

```

'filter GejalaKhusus, dgn field
kode=variable(kode)
rsKhusus.MoveFirst
rsKhusus.Find ("Kode=" & Kode & "")

'jika ada record
If Not rsKhusus.EOF Then
    'label NamaKerusakan=field ke 3
    (Nama Kerusakan)
    If Not
IsNull(rsKhusus.Fields(3).Value) Then
        NamaKerusakan.Caption =
rsKhusus.Fields(3).Value
    Else
        NamaKerusakan.Caption =
vbNullString
    End If

    'jika field ke 4 ada isinya...
    'text ->textbox Keterangan=field ke 4
    (Keterangan Kerusakan)
    If Not
IsNull(rsKhusus.Fields(4).Value) Then
        Keterangan.Text =
rsKhusus.Fields(4).Value
    Else
        Keterangan.Text = vbNullString
    End If

    'jika field ke 5 ada isinya...
    'text ->textbox Pencegahan=field ke 5
    (Pencegahan Kerusakan)
    If Not
IsNull(rsKhusus.Fields(5).Value) Then
        Pencegahan.Text =
rsKhusus.Fields(5).Value
    Else
        Pencegahan.Text = vbNullString
    End If

    'jika field ke 6 ada isinya...
    'text ->textbox Perbaikan=field ke 6
    (Perbaikan Kerusakan)
    If Not
IsNull(rsKhusus.Fields(6).Value) Then
        Perbaikan.Text =
rsKhusus.Fields(6).Value
    Else
        Perbaikan.Text = vbNullString
    End If
End If

'kosongkan variable(kode)
Kode = vbNullString
End Sub

```

```

'user memilih salah satu jenis Kerusakan
Private Sub TipeKerusakan_Click()
    'filter table GejalaUmum dengan
    kodejenis=3 karakter dari list yg terpilih
    rsUmum.Filter = "KodeJenis=" &
Left(TipeKerusakan.Text, 3) & ""
    'kosongkan listbox GejalaUmum
    GejalaUmum.Clear
    'Isikan data dari table GejalaUmum ke
    listbox GejalaUmum
    Do While Not rsUmum.EOF
        DoEvents
        GejalaUmum.AddItem
rsUmum.Fields(1).Value & " - " &
rsUmum.Fields(2).Value
        rsUmum.MoveNext
    Loop
    GejalaKhusus.Clear
    NamaKerusakan.Caption = vbNullString
    Keterangan.Text = vbNullString
    Pencegahan.Text = vbNullString
    Perbaikan.Text = vbNullString
End Sub

```

MANIPULASI DATA TIPE KERUSAKAN

Option Explicit

```

Private Const CONFIRM_DELETE As
String = "Anda akan menghapus data jenis
Kerusakan berikut. Apakah anda yakin?"

Private rsTipe As Recordset

Private Sub cmdBatal_Click()
    If rsTipe.EditMode <> adEditNone Then
        rsTipe.CancelBatch
    End If
    cmdRefresh_Click
    SetButtons True
End Sub

Private Sub cmdHapus_Click()
    If MsgBox(CONFIRM_DELETE,
vbQuestion Or vbYesNo) = vbYes Then
        rsTipe.Delete
    End If
End Sub

Private Sub cmdRefresh_Click()
    If Not (rsTipe Is Nothing) Then
        rsTipe.Requery
        grdJenis.Columns(1).Width = 3500
    End If

```



```

End Sub

Private Sub cmdRubah_Click()
    SetButtons False
End Sub

Private Sub cmdSave_Click()
    rsTipe.Update
    SetButtons True
End Sub

Private Sub cmdTambah_Click()
    rsTipe.AddNew
    SetButtons False
End Sub

Private Sub Form_Load()
    Set rsTipe = OpenRecordSet("Select *
From TipeKerusakan", adOpenStatic)
    Set Kode.DataSource = rsTipe
    Kode.DataField = rsTipe.Fields(0).Name
    Set Keterangan.DataSource = rsTipe
    Keterangan.DataField =
rsTipe.Fields(1).Name
    Set grdJenis.DataSource = rsTipe
    grdJenis.Columns(1).Width = 3500
End Sub

```

```

Private Sub Form_Unload(Cancel As
Integer)
    Set rsTipe = Nothing
End Sub

```

```

Private Sub SetButtons(Optional bState As
Boolean = True)
    cmdRefresh.Visible = bState
    cmdTambah.Visible = bState
    cmdRubah.Visible = bState
    cmdHapus.Visible = bState
    cmdSave.Visible = Not bState
    cmdBatal.Visible = Not bState
End Sub

```

MANIPULASI DATA GEJALA UMUM

Option Explicit

```

Private Const CONFIRM_DELETE As
String = "Anda akan menghapus data gejala
umum berikut. Apakah anda yakin?"

```

```

Private rsGU As Recordset

```

```

Private Sub cmdBatal_Click()
    If rsGU.EditMode <> adEditNone Then
        rsGU.CancelBatch
    End If
End Sub

```

```

End If
cmdRefresh_Click
SetButtons True
End Sub

```

```

Private Sub cmdHapus_Click()
    If MsgBox(CONFIRM_DELETE,
vbQuestion Or vbYesNo) = vbYes Then
        rsGU.Delete
    End If
End Sub

```

```

Private Sub cmdRefresh_Click()
    If Not (rsGU Is Nothing) Then
        rsGU.Requery
        grdJenis.Columns(1).Width = 3500
    End If
End Sub

```

```

Private Sub cmdRubah_Click()
    SetButtons False
End Sub

```

```

Private Sub cmdSave_Click()
    rsGU.Update
    SetButtons True
End Sub

```

```

Private Sub cmdTambah_Click()
    rsGU.AddNew
    SetButtons False
End Sub

```

```

Private Sub Form_Load()
    Set rsGU = OpenRecordSet("Select *
From GejalaUmum Order By Kode",
adOpenStatic)
    Set Kode.DataSource = rsGU
    Kode.DataField = rsGU.Fields(0).Name
    Set Keterangan.DataSource = rsGU
    Keterangan.DataField =
rsGU.Fields(1).Name
    Set grdJenis.DataSource = rsGU
    grdJenis.Columns(1).Width = 3500
End Sub

```

```

Private Sub Form_Unload(Cancel As
Integer)
    Set rsGU = Nothing
End Sub

```

```

Private Sub SetButtons(Optional bState As
Boolean = True)
    cmdRefresh.Visible = bState
    cmdTambah.Visible = bState

```

```

cmdRubah.Visible = bState
cmdHapus.Visible = bState
cmdSave.Visible = Not bState
cmdBatal.Visible = Not bState
End Sub

```

MANIPULASI DATA GEJALA KHUSUS

Option Explicit

```

Private Const CONFIRM_DELETE As
String = "Anda akan menghapus data gejala
khusus berikut. Apakah anda yakin?"

```

```

Private rsGK As Recordset

```

```

Private Sub cmdBatal_Click()
    If rsGK.EditMode <> adEditNone Then
        rsGK.CancelBatch
    End If
    cmdRefresh_Click
    SetButtons True
End Sub

```

```

Private Sub cmdHapus_Click()
    If MsgBox(CONFIRM_DELETE,
vbQuestion Or vbYesNo) = vbYes Then
        rsGK.Delete
    End If
End Sub

```

```

Private Sub cmdRefresh_Click()
    If Not (rsGK Is Nothing) Then
        rsGK.Requery
        grdGK.Columns(1).Width = 3500
        grdGK.Columns(2).Width = 3500
        grdGK.Columns(3).Width = 3500
        grdGK.Columns(4).Width = 3500
        grdGK.Columns(5).Width = 3500
    End If
End Sub

```

```

Private Sub cmdRubah_Click()
    SetButtons False
End Sub

```

```

Private Sub cmdSave_Click()
    rsGK.Update
    SetButtons True
End Sub

```

```

Private Sub cmdTambah_Click()
    rsGK.AddNew
    SetButtons False

```

```

End Sub

```

```

Private Sub Form_Load()
    Set rsGK = OpenRecordSet("Select *
From GejalaKhusus Order By Kode",
adOpenStatic)
    Set Kode.DataSource = rsGK
    Kode.DataField = rsGK.Fields(0).Name
    Set Gejala.DataSource = rsGK
    Gejala.DataField = rsGK.Fields(1).Name
    Set Kerusakan.DataSource = rsGK
    Kerusakan.DataField =
rsGK.Fields(2).Name
    Set Keterangan.DataSource = rsGK
    Keterangan.DataField =
rsGK.Fields(3).Name
    Set Pencegahan.DataSource = rsGK
    Pencegahan.DataField =
rsGK.Fields(4).Name
    Set Perbaikan.DataSource = rsGK
    Perbaikan.DataField =
rsGK.Fields(5).Name
    Set grdGK.DataSource = rsGK
    grdGK.Columns(1).Width = 3500
    grdGK.Columns(2).Width = 3500
    grdGK.Columns(3).Width = 3500
    grdGK.Columns(4).Width = 3500
    grdGK.Columns(5).Width = 3500
End Sub

```

```

Private Sub Form_Unload(Cancel As
Integer)
    Set rsGK = Nothing
End Sub

```

```

Private Sub SetButtons(Optional bState As
Boolean = True)
    cmdRefresh.Visible = bState
    cmdTambah.Visible = bState
    cmdRubah.Visible = bState
    cmdHapus.Visible = bState
    cmdSave.Visible = Not bState
    cmdBatal.Visible = Not bState
End Sub

```

BASIS ATURAN

Option Explicit

```

Private Const CONFIRM_DELETE As
String = "Anda akan menghapus aturan
berikut. Apakah anda yakin?"

```

```

Private rsTipe As Recordset
Private rsGU As Recordset
Private rsGK As Recordset

```

```

Private bMode As Byte

Private Sub cboJenis_Click()
    Dim kj As String

    kj = Left(cboJenis.Text, 3)
    listGU.Clear
    rsGU.MoveFirst
    Do While Not rsGU.EOF
        If rsGU.Fields(0).Value = kj Then
            listGU.AddItem
rsGU.Fields(1).Value & " " &
rsGU.Fields(2).Value
            listGU.Selected(listGU.NewIndex) =
True
        Else
            listGU.AddItem
rsGU.Fields(1).Value & " " &
rsGU.Fields(2).Value
            listGU.Selected(listGU.NewIndex) =
False
        End If
        rsGU.MoveNext
    Loop

    listGU.ListIndex = 0
End Sub

Private Sub cmdAddJU_Click()
    Dim KodeJenis As String, sQuery As
String
    Dim i As Integer
    Dim rsJU As Recordset

    'Hapus relasi jenis
    KodeJenis = Left(cboJenis.Text, 3)
    sQuery = "Delete From
RelasiJenisGejalaUmum Where
KodeJenis=" & KodeJenis & ""
    If Execute(sQuery) Then
        Set rsJU =
OpenRecordSet("RelasiJenisGejalaUmum")
        For i = 0 To listGU.ListCount - 1
            If listGU.Selected(i) Then
                rsJU.AddNew
                rsJU.Fields(0).Value = KodeJenis
                rsJU.Fields(1).Value =
Left(listGU.List(i), 3)
                rsJU.Update
            End If
        Next i
    End If

    Set rsJU = Nothing

```

```

End Sub

Private Sub cmdAddUK_Click()
    Dim KodeGU As String, sQuery As
String
    Dim i As Integer
    Dim rsUK As Recordset

    'Hapus relasi jenis
    KodeGU = Left(listGU.Text, 3)
    sQuery = "Delete From
RelasiGejalaUmumKhusus Where
KodeGejalaUmum=" & KodeGU & ""
    If Execute(sQuery) Then
        Set rsUK =
OpenRecordSet("RelasiGejalaUmumKhusus
")
        For i = 0 To listGK.ListCount - 1
            If listGK.Selected(i) Then
                rsUK.AddNew
                rsUK.Fields(0).Value = KodeGU
                rsUK.Fields(1).Value =
Left(listGK.List(i), 4)
                rsUK.Update
            End If
        Next i
    End If

    Set rsUK = Nothing
End Sub

Private Sub Form_Load()
    Dim sQuery As String

    Set rsTipe = OpenRecordSet("Select *
From TipeKerusakan Order By Kode")

    sQuery = "Select
ju.KodeJenis,gu.Kode,gu.Gejala From
RelasiJenisGejalaUmum ju " & _
"Right Outer Join GejalaUmum gu On
ju.KodeGejalaUmum=gu.Kode "
    Set rsGU = OpenRecordSet(sQuery)

    sQuery = "Select
uk.KodeGejalaUmum,gk.Kode,gk.Gejala
From RelasiGejalaUmumKhusus uk " & _
"Right Outer Join GejalaKhusus gk On
uk.KodeGejalaKhusus=gk.Kode " & _
"Order By gk.Kode"
    Set rsGK = OpenRecordSet(sQuery)

    cboJenis.Clear
    Do While Not rsTipe.EOF

```

```

        cboJenis.AddItem
rsTipe.Fields(0).Value & " " &
rsTipe.Fields(1).Value
        rsTipe.MoveNext
    Loop
End Sub

Private Sub Form_Resize()
    If Me.WindowState <> vbMinimized
Then
        listGU.Width = Me.Width - 1680
        listGK.Width = Me.Width - 1680
        cmdAddJU.Left = Me.Width - 4160
        cmdAddUK.Left = Me.Width - 4160
    End If
End Sub

Private Sub Form_Unload(Cancel As
Integer)
    Set rsTipe = Nothing
    Set rsGU = Nothing
    Set rsGK = Nothing
End Sub

Private Sub listGU_Click()
    Dim ku As String

    ku = Left(listGU.Text, 3)
    listGK.Clear
    rsGK.MoveFirst
    Do While Not rsGK.EOF
        If rsGK.Fields(0).Value = ku Then
            listGK.AddItem
rsGK.Fields(1).Value & " " &
rsGK.Fields(2).Value
            listGK.Selected(listGK.NewIndex) =
True
        Else
            listGK.AddItem
rsGK.Fields(1).Value & " " &
rsGK.Fields(2).Value
            listGK.Selected(listGK.NewIndex) =
False
        End If

        rsGK.MoveNext
    Loop

    listGK.ListIndex = 0
End Sub

```

KAMUS ISTILAH

Option Explicit

```

Private Const CONFIRM_DELETE As
String = "Anda akan menghapus istilah
berikut dari kamus istilah. Apakah anda
yakin?"
Private Const ERR_NOT_FOUND As
String = "ISTILAH YANG ANDA CARI
TIDAK DAPAT DITEMUKAN."

'table object
Private rsKamus As Recordset

'tombol cari di-click
Private Sub Cari_Click()
    Dim sKata As String
    'kata yg dicari
    sKata = Replace(Kata.Text, "", "")
    'pindahkan pointer table kamus ke record
pertama
    rsKamus.MoveFirst
    'cari kata tersebut
    rsKamus.Find "Kata Like "*" & sKata &
"*"
    'jika pointer berada pada ujung file (End
Of File-EOF)
    'berarti data tidak ketemu
    If rsKamus.EOF Then
        Definisi.Text = ERR_NOT_FOUND
    Else
        'data ketemu
        Kata.Text = rsKamus.Fields(1).Value
        Definisi.Text =
rsKamus.Fields(2).Value
    End If
End Sub

Private Sub cmdBatal_Click()
    If rsKamus.EditMode <> adEditNone
Then
        rsKamus.CancelBatch
    End If
    cmdRefresh_Click
    SetButtons True
End Sub

Private Sub cmdHapus_Click()
    If MsgBox(CONFIRM_DELETE,
vbQuestion Or vbYesNo) = vbYes Then
        rsKamus.Delete
    End If
End Sub

Private Sub cmdRefresh_Click()
    If Not (rsKamus Is Nothing) Then
        rsKamus.Requery
    End If

```

```

End Sub

Private Sub cmdRubah_Click()
    SetButtons False
End Sub

Private Sub cmdSave_Click()
    rsKamus.Update
    cmdRefresh_Click
    SetButtons True
End Sub

Private Sub cmdTambah_Click()
    rsKamus.AddNew
    SetButtons False
End Sub

'event window load
Private Sub Form_Load()
    'buka table KamusIstilah
    Set rsKamus = OpenRecordSet("Select *
From KamusIstilah Order By Kata")

    'tampilkan DataGrid jika
    user=Admin(pakar), untuk mengelola data
    kamus
    If IsAdmin Then
        If Not (rsKamus Is Nothing) Then
            If rsKamus.State = adStateOpen
Then
                'Bind control with recordset
                Set Kata.DataSource = rsKamus
                Kata.DataField =
rsKamus.Fields(1).Name
                Set Definisi.DataSource =
rsKamus
                Definisi.DataField =
rsKamus.Fields(2).Name
                'hubungkan table rsKamus dengan
                control DataGrid
                Set GridKamus.DataSource =
rsKamus
            End If
        End If

    Else
        Me.Height = 3015
    End If
    Cari.Enabled = Not IsAdmin
End Sub

'event window close
Private Sub Form_Unload(Cancel As
Integer)

```

```

'putuskan hubungan table rsKamus
dengan control DataGrid
Set Kata.DataSource = Nothing
Set Definisi.DataSource = Nothing
Set GridKamus.DataSource = Nothing
Set rsKamus = Nothing
End Sub

```

```

'tekan Enter = click tombol Cari
Private Sub Kata_KeyUp(KeyCode As
Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        If Not (IsAdmin) Then
            Call Cari_Click
        End If
    End If
End Sub

```

```

Private Sub SetButtons(Optional bState As
Boolean = True)
    cmdRefresh.Visible = bState
    cmdTambah.Visible = bState
    cmdRubah.Visible = bState
    cmdHapus.Visible = bState
    cmdSave.Visible = Not bState
    cmdBatal.Visible = Not bState
End Sub

```

MENAMPILKAN FORM ALAMAT DISTRIBUTOR SPARE PART

Option Explicit

```

Private Sub Form_Load()
    Dim fileName As String
    fileName = App.Path & "\Distributor.rtf"
    If Dir(fileName) <> vbNullString Then
        RTFWindow.LoadFile fileName
    End If
End Sub

```

PENGELOLAAN PENGGUNA

Option Explicit

```

Dim rsUsers As Recordset

Private Sub ShowUsers()
    Set rsUsers = OpenRecordSet("Select *
From Pengguna Order By Nama")
    If Not (rsUsers Is Nothing) Then
        If rsUsers.State = adStateOpen Then
            UserList.Clear
            Do While Not rsUsers.EOF

```

```

        UserList.AddItem
rsUsers.Fields(0).Value
        rsUsers.MoveNext
    Loop
    End If
End If

AddUser.Visible = True
DelUser.Visible = True
DelUser.Enabled = (UserList.ListCount >
0)
ChgPassword.Visible = True
cmdSavePassword.Visible = False
Batal.Visible = False
Txt1.Text = vbNullString
Txt2.Text = vbNullString
Txt3.Text = vbNullString
Txt4.Text = vbNullString

Param3.Caption = "Ulang Password"
Param4.Visible = False
Txt4.Visible = False
End Sub

Private Sub AddUser_Click()
    If Txt1.Text = vbNullString Or Txt2.Text
= vbNullString Or Txt3.Text = vbNullString
Then Exit Sub
    'jika password dan konfirmasi password
tdk sama... keluar
    If Txt2.Text <> Txt3.Text Then
        MsgBox "Password dan konfirmasi
password harus sama.", vbExclamation
    Else
        If Execute("Insert Into Pengguna
Values('" & Txt1.Text & "','" & Txt2.Text &
"')") Then
            'pesan berhasil
            MsgBox "Pengguna baru dengan
nama " & Txt1.Text & vbCr & "Berhasil
ditambahkan", vbInformation
            ShowUsers
        Else
            'pesan gagal
            MsgBox "Penambahan pengguna
baru gagal.", vbCritical, "Error!!!"
        End If
    End If
End Sub

user
Private Sub Batal_Click()
    ShowUsers
End Sub

```

```

Private Sub ChgPassword_Click()
    If UserList.ListCount = 0 Then Exit Sub
    If UserList.ListIndex = -1 Then Exit Sub
    AddUser.Visible = False
    DelUser.Visible = False
    ChgPassword.Visible = False
    cmdSavePassword.Visible = True
    Batal.Visible = True

    Txt1.Text = UserList.Text
    Txt2.Text = vbNullString
    Txt3.Text = vbNullString
    Txt4.Text = vbNullString

    Param3.Caption = "Password Baru"
    Param4.Caption = "Ulang Password"
    Param4.Visible = True
    Txt4.Visible = True
    Txt2.SetFocus
End Sub

baru
Private Sub cmdSavePassword_Click()
    Dim rsUser As Recordset
    Dim userName As String, password As
String, newPassword
    If Txt1.Text = vbNullString Or Txt2.Text
= vbNullString Or Txt3.Text = vbNullString
Then Exit Sub
    userName = Replace(Txt1.Text, "", "")
    password = Replace(Txt2.Text, "", "")
    newPassword = Replace(Txt3.Text, "",
"")

    If Txt3.Text <> Txt4.Text Then
        MsgBox "Password baru dan
konfirmasi password baru harus sama.",
vbExclamation
    Else
        Set rsUser = OpenRecordSet("Select
Nama,Passwd From Pengguna Where
Nama='" & userName & "' And Passwd='"
& password & "'")
        If Not (rsUser Is Nothing) Then
            If rsUser.State = adStateOpen Then
                If rsUser.RecordCount > 0 Then
                    Set rsUser = Nothing
                    If Execute("Update Pengguna
Set Passwd='" & newPassword & "' Where
Nama='" & userName & "' And Passwd='"
& password & "'") Then
                        'pesan berhasil
                        MsgBox "Password user " &
userName & " telah berhasil dirubah.",
vbInformation

```

```

        Else
            'pesan gagal
            MsgBox "Gagal merubah
password user " & userName, vbCritical
        End If
    Else
        MsgBox "User tidak
ditemukan!", vbCritical, "Ganti Password"
        Txt1.SetFocus
        SendKeys "{Home}+{End}"
    End If
End If
End If
End If
Set rsUser = Nothing
ShowUsers
End Sub

Private Sub DelUser_Click()
    If UserList.ListCount = 0 Then Exit Sub
    If Execute("Delete From Pengguna Where
Nama=" & UserList.Text & """) Then
        MsgBox "User " & UserList.Text & "
berhasil dihapus.", vbInformation
    Else
        MsgBox "Gagal menghapus user " &
UserList.Text, vbCritical, "Error"
    End If
    ShowUsers
End Sub

Private Sub Form_Load()
    ShowUsers
End Sub

Private Sub Form_Unload(Cancel As
Integer)
    Set rsUsers = Nothing
End Sub

```